Allocation des Ressources pour la Gestion Dynamique du Spectre dans les Réseaux Ad hoc Clustérisés

Jérôme Gaveau

université
PARIS-SACLAY

NNT: 2018SACLC060

Thèse de doctorat

CentraleSupélec

Allocation des Ressources pour la Gestion Dynamique de Spectre dans les Réseaux Ad hoc Clustérisés

Thèse de doctorat de l'Université Paris-Saclay
préparée à CentraleSupélec

École doctorale n° 580
Spécialité de doctorat: réseaux, information et communications

Thèse présentée et soutenue à Gif-sur-Yvette, le 11 Juillet 2018, par

Jérôme Gaveau

Composition du jury:

M. Philippe Ciblat Professeur, Télécom ParisTech	Président
M. Xavier Lagrange Professeur, IMT Atlantique	Rapporteur
M. Stefano Secci Professeur, Sorbonne Université	Rapporteur
M. Tijani Chahed Professeur, Telecom SudParis	Examinateur
M. Thierry Defaix Ingénieur, DGA MI	Examinateur
Mme. Berna Seyrac Ingénieure, Orange Labs	Examinateur
M. Mohamad Assaad Professeur, CentraleSupélec	Directeur de Thèse
M. Christophe Le Martret Ingénieur, Thales Communication & Security	Directeur de Thèse

Remerciements

Ce travail de thèse est une aventure aussi bien professionnelle que personnelle. J'ai parcouru les différentes épreuves de ce voyage grâce à un entourage qui m'a soutenu constamment. Pour cela, je tiens chaleureusement à vous en remercier.

Tout d'abord, je remercie **Mohamad Assaad** et **Christophe Le Martret**, mes directeurs de thèse, de m'avoir appris, transmis leur savoir, écouté, encadré patiemment durant ces trois années. Je les remercie également d'avoir placé leur confiance en moi pour mener à bien ce projet.

Je tiens à remercier les membres de mon jury, **Philippe Ciblat**, **Xavier Lagrange**, **Stefano Secci**, **Tijani Chahed**, **Thierry Defaix** et **Berna Sayrac** d'avoir accepté de relire mon travail, pour leurs commentaires et les discussions enrichissantes qui ont suivis.

Je remercie Thales Communication & Security d'avoir permis la possibilité de ce travail et plus particulièrement **Jean-Luc Peron**, responsable du service WFD, de m'avoir fait une place dans l'équipe. Merci à mes amis et collègues de Thales pour leur accueil, les bons moments en pause, tous les midis, les discussions riches et techniques, **Raphaël**, **Christophe**, **Luxmiram**, **Olivier**, **Laurent**, **Arnaud**, **Helene**, **Antonio**, **Dorine**, **Benoît**, **Simon**, **Serdar**, **Elie**, **Sylvain**, **Valentin**, **Adrien**, **Philippe**, **Xavier** et bien sûr **Anaël**, merci pour tout.

Merci également à mes amis et collègues de CentraleSupélec, **Evgeny**, **Salah Eddine**, **Kenza**, **Matha**, **Gil**, **Ejder**, **Rita**, **Hafiz**, **Julien**, **Romain** et **Richard**.

Merci à mes amis et à votre soutien permanent depuis des années, **Julien**, **Jean**, **Robin**, **Mathilde**, **Marion**, **Héloïse**, **Cécile**, **Pauline P.**, **Pauline T.**, **Emmanuelle**, **Sina**, **Timothée**, **Massimo**, **Vincent**, **Addo**, **Agnès**, **Otto**, **Jules**, **Clément**, **Romain**, **Sylvain**, **Julia**, **Greg**, **Guillaume**, **Francis**, **Etienne**, **Baptiste**, **Benjamin**, **Matthieu**, **Gautier**, **Adrien**, **Simon**, **Jonathan**, **Alexandre**... et tous ceux que j'aurai oubliés.

Merci infiniment à ma famille, ma belle famille, mes parents, mon frère, mes beaux parents, belles sœurs et beaux frères pour leurs encouragements, les bons moments et l'amour qu'ils me donnent. Enfin, je ne pourrai pas terminer cette liste sans remercier **Tiffany** qui éclaire ma vie, et m'a soutenu sans faille durant ces trois années.

Merci à tous.

Contents

List of Acronyms

AWGN	Additive White and Gaussian Noise
BER	Bit Error Rate
CH	Cluster Head
ccdf	complementary cumulative density function
cdf	Cumulative Density Function
EFHT	Expected First Hitting Time
EESM	Effective Exponential Signal to Noise Ratio (SNR) Mapping
FSC	Finite State Controller
GoF	Goodness of Fit
GT	Game theory
iid	independent and identically distributed
INR	Interference to Noise Ratio
KS	Kolmogorov-Smirnov
lhs	left hand side
LUT	Look Up Table
MC	Markov Chain
MCS	Modulation and Coding Scheme
MNE	Mixed Nash Equilibrium
NB	Narrow Band
NE	Nash Equilibrium
ODL	Optimal Distributed Learning
OFDM	Orthogonal Frequency Division Multiplexing
pdf	probability density function
PER	Packet Error Rate
PNE	Pure Nash Equilibrium
QoS	Quality of Service
RC	Recurrence Classes
RL	Reinforcement Learning
RODL	Robust Optimal Distributed Learning

RTEL	Robust Trial and Error Learning
RRC	Reduced Recurrence Classes
RRS	Reduced Recurrent State
rv	random variable
SINR	Signal to Noise plus Interference Ratio
SNR	Signal to Noise Ratio
SSS	Stochastic Stable State
TDMA	Time Division Multiple Access
TE	Trial and Error
TEL	Trial and Error Learning
WB	Wideband

Résumé de la thèse en français

0.1 Introduction

Les réseaux ad hoc ont fait l'objet de nombreuses études grâce à leur capacité de déploiement sans infrastructure (*e.g.* [1] et les références associées). Ces réseaux subissent de nombreux changement dynamiques dus au fait que les nœuds sont mobiles et que les connexions entre ces derniers sont sujettes aux phénomènes aléatoires inhérents aux canaux de communications sans fil. C'est pourquoi la gestion des interférences dans ces réseaux constitue un défi dont la difficulté se trouve augmentée par l'absence d'un contrôleur central pouvant le réaliser. Ce dernier obstacle constitue une des raisons de la faible mise en œuvre de ces réseaux en pratique.

Parmi les alternatives de gestion des ressources dans les réseaux ad hoc, la solution clustérisée est celle qui assure de bonnes performances de communications tout en tenant compte de la mobilité et du passage à l'échelle. La clustérisation garde les avantages des réseaux ad hoc (pas d'infrastructure) et réduit de manière significative la signalisation ce qui permet la mise en œuvre de ces réseaux en pratique [2, 3, 4]. L'idée est de rassembler localement (et dynamiquement) les nœuds en groupes appelés "cluster" dans le but d'optimiser la performance du réseau et de gérer localement les ressources du cluster. Cette tâche est souvent effectuée par un chef de cluster (CH en anglais) qui est élu parmi les nœuds du cluster. Le rôle du CH est de gérer l'allocation des ressources et des schémas de transmissions au sein de son cluster. Pour ce faire, les nœuds de chaque cluster échangent de la signalisation uniquement avec leur CH, ce qui réduit significativement la signalisation par rapport à un réseau ad hoc traditionnel. Les réseaux ad hoc organisés en clusters sont utilisés dans les contextes des réseaux de véhicules, des réseaux cognitifs ainsi que dans les réseaux militaires où les nœuds proches géographiquement appartiennent au même cluster [5].

Un des problèmes principaux dans les réseaux ad hoc clustérisés concerne l'allocation de spectre afin de permettre les communications sans fil dans les clusters. Une solution idéale serait d'effectuer une allocation orthogonale des bandes de fréquences à l'initialisation de la procédure mais ce n'est pas possible pour les deux raisons suivantes. Tout d'abord, à cause de la nature dynamique du réseau, le nombre de clusters peut varier et donc aussi le nombre de bandes de fréquences nécessaires à son bon fonctionnement. Ensuite, pour passer à l'échelle, le réseau a besoin de plus en plus de bandes de fréquences à mesure que le nombre de clusters augmente ce qui n'est pas réalisable à cause de la pénurie de spectre. Pour pallier à ces difficultés, on limite la quantité de bandes de fréquences et une solution pratique consiste à regarder les allocations dynamiques permettant une réutilisation spatiale des canaux. En effet, les liens radios peuvent continuer à fonctionner lorsque l'interférence est maîtrisée. Par exemple, on peut fixer un SINR minimum en dessous duquel le lien radio est considéré non fonctionnel. Par la suite, on peut utiliser la décroissance naturelle de la puissance des signaux radios avec la distance afin de réutiliser des canaux fréquentiels par différent clusters.

Dans cette thèse, on considère un réseau ad hoc clustérisé dans lequel les nœuds de chaque cluster communiquent entre eux et échangent de la signalisation avec le CH. Nous supposons qu'il n'y a pas d'échange d'information entre les clusters. Afin de réduire la quantité d'interférence dans le réseau, et du fait qu'il n'existe pas de contrôleur central pour réaliser la gestion au fil de l'eau des interférences, les clusters peuvent utiliser différent canaux fréquentiels. Cependant, les bandes de fréquences sont des ressources rares ce qui oblige des clusters à utiliser la même bande. Une allocation des fréquences doit être réalisée dans le but de maximiser les performances du réseau. Malheureusement, cette allocation ne peut être planifiée pour les raisons suivantes. La charge au niveau de chaque cluster varie dans le temps (car le trafic varie dans le temps) ce qui signifie que l'interférence des clusters entre eux varie également. Les clusters sont également mobiles, et deux clusters voisins à un instant donné peuvent se retrouver éloignés plus tard. Ainsi l'allocation doit être dynamique pour faire face à la mobilité et aux variations de trafic. L'absence de contrôleur central impose également que l'allocation de fréquence soit effectuée de manière distribuée. Dans chaque cluster, le CH décide du choix du canal fréquentiel utilisé par les nœuds de son cluster et cela sans échanger de signalisation avec les autres clusters. De plus, cette allocation doit être robuste aux diverses perturbations aléatoires affectant les liens radios. Les CHs doivent prendre des décisions basées sur des mesures qui sont potentiellement perturbées par la large gamme de variations dynamiques inhérentes aux réseaux ad hoc sans fil. Dans un contexte où les informations et les décisions sont distribuées, il est naturel pour les CHs de considérer des changements de configuration du réseau lorsque les mesures varient. Dans un environnement perturbé par des phénomènes dynamique ce n'est plus le cas, ce qui peut mener le réseau à avoir un comportement incontrôlable et détériorer sérieusement ses performances.

0.2 État de l'art

Cette section présente plus formellement le contexte et les axes de l'étude, et décrit l'état de l'art concernant l'allocation distribuée des fréquences dans les réseaux ad hoc clustérisés.

Dans le contexte de l'allocation distribuée des ressources, il est usuel d'employer la théorie des jeux (GT en anglais) comme cadre pour la modélisation du problème. Les concepts de bases de la GT utiles dans cette thèse sont développés dans le Chapitre 2. Le problème est donc modélisé par un jeu dans lequel les clusters sont les joueurs et leurs stratégies sont les choix de fréquences. Les décisions des clusters sont guidées par une métrique appelée utilité. Le cadre du problème d'allocation distribuée des fréquences peut être résumé par les propriétés suivantes :

(i) le choix de fréquence est effectué indépendamment par chaque cluster,

(ii) la stratégie de chaque cluster dépend seulement de ses propres actions et mesures d'utilité courante et passées (l'information est locale),

(iii) l'utilité n'a pas de forme exacte dans le sens où c'est une fonction qui dépend du choix des autres clusters et elle prend des valeurs discrètes,

(iv) l'utilité peut être perturbée par des processus stochastiques (*e.g.* évanouissements de Rayleigh),

(v) le contexte de l'étude est dynamique dans le sens où les clusters peuvent bouger, entrer et quitter le réseau, et la charge de trafic peut varier,

(vi) l'ensemble des stratégies est discret (sélection des canaux fréquentiels).

Les propriétés *(i)* et *(ii)* définissent un schéma "complètement découplé" [6] que l'on nomme *complètement distribué* dans cette thèse [7, 8]. Il est important de noter que dans le cadre des propriétés *(i)* et *(ii)*, et dans une large classe de jeux de tailles finies, il n'existe pas de procédure adaptative qui converge presque sûrement vers un équilibre de Nash (NE en anglais) [6, 9, 10]. Cependant, comme nous allons le voir dans cette thèse, il est possible d'étudier une forme de convergence plus faible (voir les états stochastiques stables (SSS en anglais) dans la Section 2.2 pour les détails).

Vis à vis de la propriété *(iii)*, les méthodes d'apprentissages sans modèle sont intéressantes pour aborder notre problème [11], car dans ce paradigme, les joueurs n'essaient ni de modéliser leur environnement, ni d'obtenir un modèle explicite de l'utilité. Ils considèrent l'environnement comme une boîte noire et apprennent en interagissant avec lui (*e.g.* trials and errors).

Ce cadre de travail, bien que très restrictif, peut être rencontré dans une grande variété d'exemples. Par exemple, dans un champ d'éoliennes, chacune des éoliennes contrôle la puissance qu'elle extrait du vent [12]. Il est très difficile, voire impossible, de modéliser l'influence d'une turbine sur les autres. De plus, l'absence de communications entre les différentes éoliennes rend impossible toute coopération. Un autre exemple concerne les personnes se rendant au travail dans une ville. Ces derniers désirent éviter les bouchons, cependant il ne connaissent pas les choix stratégiques des autres, ni l'influence de leur propre stratégie sur leur propre préférence et celles des autres [13].

La propriété *(iv)* représente le fait que dans un contexte de communications sans fil, le canal est perturbé par des processus stochastiques qui à leur tour affectent l'utilité. Ces perturbations de l'utilité se répercutent sur les décisions des joueurs ce qui peut faire perdre les bonnes propriétés de convergence des algorithmes d'allocation des ressources, et diminuer la performance du réseau.

Nous rencontrons des algorithmes d'allocation des fréquences complètement distribués dans beaucoup de contextes sans fil car les nœuds/joueurs ne peuvent pas forcément échanger de la signalisation entre eux au risque de diminuer le trafic. De plus, les fonctions d'utilité réalistes n'ont régulièrement pas de forme exacte comme mentionné par la propriété *(iii)* (*e.g.* Qualité d'expérience, nombre de paquets correctement décodés, etc). Néanmoins, dans ce contexte, des algorithmes d'allocation de ressources basés sur des apprentissages sans modèle ont été développés. Parmi eux, nous avons identifié les méthodes basées sur le "Q-learning", le "learning automata", le "combined payoff and strategy learning", le "no-regret learning" et les méthodes "trial and error" (TE en anglais) comme des solutions potentielles. Pour chacune des procédures d'apprentissage précédentes, nous présentons le principe de fonctionnement général sans rentrer dans les détails.

Dans [14, 15] les auteurs utilisent le Q-learning pour sélectionner les canaux fréquentiels parmi les femtocells. Dans [16], le Q-learning est employé afin d'allouer de manière complètement distribuée les canaux fréquentiels et la puissance dans un réseau. Le principe général de la méthode est de résoudre les équations d'optimalité de Bellman [17]. La solution de ces équations maximise la somme des utilités cumulées au cours du temps. Cependant, il y a quelques contraintes à utiliser cette méthode dans un contexte complètement distribué. Tout d'abord, pour prendre leur décision, les joueurs ont besoin de connaître l'état du système ce qu'ils n'ont pas (ils ne connaissent pas les choix d'actions des autres joueurs). Ensuite, l'optimalité de la solution peut être atteinte dans un environnement stationnaire ce qui n'est pas le cas dans les jeux à plusieurs agents. La décision d'un joueur influence celle des autres et réciproquement. Par conséquent, l'environnement n'est pas stationnaire et la convergence de la procédure n'est plus assurée [11]. C'est pourquoi, lors de l'implémentation du Q-learning dans un environnement multi-agents, chaque joueur considère peu ou pas d'interaction avec les autres joueurs. De plus, dans les cas où la convergence peut être atteinte (*i.e.* peu de couplage entre les joueurs), cela peut prendre beaucoup de temps [11].

Avec le Q-learning, chaque joueur essaye d'apprendre les valeurs d'utilités cumulées et choisit la stratégie dans le but de maximiser cette valeur. Il existe des algorithmes qui ont un fonctionnement opposé dans le sens où la valeur d'utilité reçue modifie directement le choix des stratégies. Ces méthodes sont connues sous le nom anglais de learning automata [18, 19]. Dans [20, 21, 22], les auteurs proposent un algorithme basé sur le principe du learning automata présenté dans [23]. La preuve de convergence utilise les équations différentielles ordinaires et les approximations stochastiques [24]. Chaque stratégie est jouée avec une probabilité qui est mise à jour à chaque itération en utilisant une approximation stochastique. Pour une action donnée, la probabilité de la jouer lors de la prochaine itération est d'autant plus importante que l'utilité reçue est grande. Les probabilités des autres stratégies sont misent à jour en utilisant la conservation des probabilités. Il est prouvé dans [23] que les états stationnaires de l'algorithme sont les équilibres de Nash purs (PNE en anglais) (voir la Définition 2.2 à la page 12 pour de plus amples détails). Un des principaux avantages de ce type d'algorithmes, en plus d'être complètement distribués, est leur capacité à évoluer dans un environnement où l'utilité est perturbée. Cela est dû à l'utilisation d'approximations stochastiques. Cependant, il existe quelques inconvénients lors de l'usage de ces algorithmes. Le premier est que bien qu'un PNE puisse être atteint, il n'y a pas d'information supplémentaire sur sa performance. Par exemple, il est montré dans [25] que l'algorithme converge vers un état sous optimal. De plus, dans le cadre restrictif de cette thèse, l'existence d'un PNE n'est pas assurée, ce qui peut remettre en cause la convergence de ces algorithmes.

Les algorithmes précédents apprennent l'utilité moyenne ou la stratégie, mais il existe des algorithmes qui utilisent les deux procédures simultanément. Dans [25, 26] les auteurs implémentent l'approche combined payoff and strategy learning afin de partager les ressources entre les femtocells et les bornes Wifi respectivement. Le principe de fonctionnement de ces algorithmes peut être décrit en deux étapes. Durant la première étape, l'utilité moyenne correspondant à l'action courante est mise à jour en utilisant l'utilité reçue et une méthode d'approximation stochastique. Dans un second temps, la distribution de probabilité des stratégies est mise à jour de manière similaire au cas des learning automata à la différence près qu'une fonction de correspondance avec des propriétés particulière est utilisée (voir [26] pour plus de détails). Grâce à cette fonction et au caractère fini du jeu, l'algorithme possède un point fixe vers lequel il peut converger. Ce qui est intéressant, c'est qu'en réglant un paramètre de température, on peut faire en sorte que ce point fixe soit une action pure ou une probabilité uniforme de jouer n'importe quelle action. Il est également intéressant de noter que, comme dans le learning automata, cet algorithme est robuste aux perturbations de l'utilité grâce à l'utilisation d'une approximation stochastique. Par rapport au learning automata, l'apprentissage combiné améliore les performances de fonctionnement [25]. Cependant, les auteurs de [25] mentionnent que bien que le point fixe peut correspondre à un PNE, la convergence vers ce point peut ne pas être sûre si la fonction d'utilité est trop égoïste (*e.g.* pas de communication entre les joueurs). De plus, il n'y pas de garantie de performance de l'équilibre atteint. Il est intéressant de noter que ces algorithmes font partie d'un concept plus général nommé en anglais combined fully distributed payoff and strategy reinforcement learning (CODIPAS-RL) décrit dans [7].

La procédure no-regret est employée dans [27, 28] afin d'allouer de manière complètement distribuée la puissance sur certaines bandes de fréquences et des canaux fréquentiels respectivement. Par rapport aux méthodes précédentes, l'apprentissage no-regret possède une philosophie différente. Les joueurs n'essaient pas directement d'améliorer leur utilité mais essaient plutôt de diminuer leur regret. Le regret de jouer une action est la différence de l'utilité moyenne qu'il reçoit en jouant l'action courante et ce qu'il aurait pu obtenir en jouant une action différente. Le regret peut ensuite être utilisé pour mettre à jour la distribution de probabilité des stratégies comme dans la méthode learning automata (*e.g.* [29]). Par exemple, on peut faire en sorte d'augmenter la probabilité de jouer une stratégie rapportant un faible regret. Les procédures no-regret comme celle présentée dans [29] convergent vers l'ensemble des équilibres corrélés connus pour contenir l'ensemble des NEs [30]. Dans ce sens, n'importe quel jeu fini possède au moins un équilibre corrélé. Bien que l'ensemble des équilibres corrélés soit plus général que celui des NEs il n'y a pas de garanties théoriques d'obtenir de meilleurs performances. Cependant, il a été observé que rechercher des équilibres corrélés peut permettre d'obtenir de meilleurs performance sociales que juste chercher les PNEs [11]. Il existe quelques inconvénients à l'usage de ces procédures. Tout d'abord, initialement cette procédure [29] n'est pas complètement distribuée dans le sens où pour calculer le regret il faut connaitre les actions des autres joueurs. Dans [27, 28], les auteurs modifient l'algorithme pour le rendre complètement distribué. De plus, l'algorithme [29] n'est pas robuste aux perturbations de l'utilité c'est pourquoi [28] propose des modifications afin de palier à cette difficulté. Enfin, pour calculer le regret, le joueur doit pouvoir connaitre l'utilité reçue s'il avait joué une action différente ce qui dans notre cas est impossible et doit être appris d'une autre manière.

La dernière classe d'algorithmes que nous présentons ici est basée sur le trial and error et est noté TE. La plupart des travaux basés sur le TE se concentrent sur des fonctions d'utilité déterministes dans lesquelles l'utilité n'est pas perturbée par des phénomènes stochastiques. Les auteurs de [31, 32] utilisent deux méthodes TE nommées ODL [12] et TEL [10] afin d'allouer de manière complètement distribuée les bandes de fréquences ainsi que les puissances d'utilisations de ces canaux dans un réseau ad hoc clustérisé. Une version modifiée d'ODL a été employée dans [33] afin d'allouer des bandes de fréquences de manière complètement distribuée dans un contexte de réseaux ad hoc et d'évanouissements lents. La description détaillée de ces méthodes TE est reléguée au Chapitre 3. De manière simple, chaque joueurs implémente un contrôleur à états fini dans lequel deux modes de fonctionnement se distinguent. Dans le premier mode, le joueur expérimente une nouvelle stratégie avec une probabilité faible ce qui correspond à une version standard de phase exploration/recherche. Dans un second mode, le joueur expérimente de manière aléatoire n'importe quelle stratégie à chaque itération de l'algorithme, ce qui est considéré comme une phase de recherche bruité. Dans chacun des modes, la probabilité d'accepter une nouvelle stratégie comme stratégie future s'accroît avec la valeur de l'utilité reçue lors de l'expérimentation. La machine d'état définit quel mode utiliser en fonction de la valeur de l'utilité reçue. L'algorithme ODL converge (au sens des SSSs) vers un état qui maximise la somme des utilités de tous les joueurs, aussi connu sous le nom de fonction de bien-être social (voir la Définition 2.3 à la page 12). L'algorithme TEL quant à lui possède le résultat intéressant suivant : s'il y a un PNE, l'algorithme converge (au sens SSS) vers un état qui maximise la fonction de bien-être social parmi ces PNE, sinon, l'algorithme converge vers un état qui maximise un compromis entre la fonction de bien-être social et une fonction de stabilité. La difficulté est que les propriétés de convergence des algorithmes mentionnés ne sont plus valides dans un contexte où l'utilité est perturbée par des processus stochastiques. Cela pose un problème dans notre cadre car ces phénomènes sont omniprésents dans les canaux de propagations sans fil (évanouissement, ombrage). Dans le domaine du contrôle, un travail récent [34] est le premier à proposer des modifications de ODL pour faire face aux perturbations stochastiques. Dans ces modifications, la déviation maximum du bruit qui affecte l'utilité est supposée constante et connue en avance.

Parmi les procédures d'allocation complètement distribuées présentés, le principe TE semble être pour le moment, la solution qui possède la convergence la plus générale et la mieux définie, et qui remplit l'ensemble des propriétés mentionnées en début de cette section sauf la *(iv)*. La méthode Q-learning essaie directement de maximiser la somme cumulée de l'utilité. Cependant, elle est originellement construite pour être utilisée dans des contextes avec un seul joueur ce qui rend la convergence dans notre cadre peu sûre ou très longue. Les procédures learning automata convergent vers les PNEs dont l'existence n'est pas assurée dans ce travail. De plus, la performance de ces équilibres n'est pas détaillée. Combined learning of payoff and strategy peut converger vers un point fixe dont l'efficacité n'est pas spécifiée non plus. Ensuite, cette convergence n'est pas certifiée dans notre contexte. Cependant, les performances de fonctionnement semblent meilleur que celles de Learning automata [26]. Les procédures no-regret convergent vers un type d'équilibres différents (*i.e.* équilibres corrélés) qui contiennent les NEs. Cette classe d'algorithmes n'est pas initialement construite pour fonctionner dans un contexte complètement distribué. Leur adaptation nécessite des modifications qui rendent la convergence non sûre ou très lente.

En comparaison aux autres procédures, les approches TE possèdent d'intéressantes propriétés de convergence globales bien que le problème soit non coopératif. Elles sont également simples à implémenter. La possibilité de caractériser aussi précisément leur convergence provient de la notion de convergence utilisée, les états stochastiques stables (SSS en anglais) sont moins contraignant que la convergence presque sûre. Avec ce concept, plutôt que de converger avec une probabilité 1 vers un état, le réseau est amené à passer une grande proportion du temps dans un état optimal et peut le quitter à tout moment. Il est prouvé dans [6] qu'il n'existe pas de procédure complètement distribuée convergeant presque sûrement ou avec probabilité 1 vers un NE dans la plupart des jeux, ce qui fait que les résultats du TEL sont les meilleurs que l'on peut atteindre dans ce contexte. Néanmoins, il a été observé dans les travaux [35], que ces procédures deviennent instables quand l'utilité est sujette à des perturbations stochastiques. Plus précisément, on démontre dans cette thèse la perte des propriétés de convergence lorsque l'utilité est perturbée (voir Chapitre 4). La cause de cette sensibilité aux perturbations est que chaque joueur interprète les variations d'utilités comme des changements d'état du système et réagit localement ce qui perturbe de manière globale le réseau. Une des contributions majeure de ce travail et de définir des algorithmes TE robustes aux perturbations de l'utilité.

De plus, au-delà des notions de convergence et du fait que les algorithmes atteignent un état satisfaisant dans notre problème, l'étude des caractéristiques de convergence reste une question ouverte [10, 11]. La principale difficulté provient de la complexité des chaînes de Markov (MC en anglais) induites par l'utilisation des algorithmes. Le jeu dans lequel les joueurs utilisent les algorithmes TE peuvent être représentés par des MCs discrètes avec énormément d'états. C'est pourquoi, obtenir la matrice de transition de ces MCs est difficile ce qui rend l'analyse du temps de convergence également très difficile (même numériquement). La résolution de ce problème dans un cas spécifique est une des contributions majeure de ce travail.

0.3 Structure de la thèse

Cette thèse est composée de six chapitres en incluant celui-ci : Les aspects théoriques (Chapitre 2), Allocation complètement distribuée des fréquences sans perturbations (Chapitre 3), Allocation complètement distribuée des fréquences avec perturbations (Chapitre 4), Les statistiques de l'utilité perturbée (Chapitre 5) et la conclusion (Chapitre 6). Dans cette section, nous décrivons un bref résumé de chaque chapitre.

Le Chapitre 2 présente les notations théoriques et les outils nécessaires à la compréhension de la modélisation du problème d'allocation de ressources, et des principaux résultats théoriques. Plus spécifiquement, la Section 2.1 introduit les concepts de théorie des jeux que nous utilisons dans cette thèse. La Section 2.2 introduit la représentation sous forme de MC et le concept de MC régulièrement perturbée dont la définition est clé pour comprendre les démonstrations théoriques de convergence des algorithmes employés.

Le chapitre 3 rappelle le problème d'allocation des fréquences de manière complètement distribuée dans un contexte déterministe [35]. Le problème est résolu de manière efficace avec les algorithmes TE dont le fonctionnement et les résultats de convergence sont présentés en détails. La seconde partie de ce chapitre concerne l'étude de leur performance qui reste une question ouverte à cause de la grande dimension des MCs sous-jacentes. Le principe de l'analyse est de réduire la dimension de la MC pour pouvoir réaliser les calculs numériques. La réduction se fait en se basant sur des symétries de la MC liées à la particularité du modèle de l'utilité utilisé. Grâce à ces simplification, nous sommes capables de comparer les algorithmes TE présentés dans des régions difficilement atteignables par simulations de Monte Carlo.

Dans le Chapitre 4, nous considérons toujours le problème d'allocation des fréquences sauf que nous ajoutons des perturbations stochastiques de l'utilité. Nous introduisons tout d'abord un modèle général de perturbations dont le but est de représenter l'influence générale des perturbations possibles sur la mesure d'utilité. Dans ce cadre, le comportement des algorithmes TE est analysé et décrypté. Nous sommes capables de démontrer l'incapacité des algorithmes TE standards à conserver leurs propriétés de convergences. C'est pourquoi dans la suite du chapitre, nous proposons des modifications pour rendre ces algorithmes robustes, et nous prouvons la convergence de ces derniers. Ensuite, nous étendons ces solutions à un cadre plus général dans lequel des hypothèses faites sur les perturbations sont relâchées. Finalement, nous étudions plus en détail les solutions proposées afin de montrer l'amélioration des performances en présence de perturbations vis à vis des algorithmes TE initiaux.

Le Chapitre 5 est dévoué à l'analyse de l'impact des évanouissements de Rayleigh sur les statistiques de l'utilité. L'objectif est de comprendre comment ces évanouissements peuvent influencer l'utilité et le processus d'allocation de ressources dans le Chapitre 4. Le calcul de ces statistiques requièrent les statistiques du SINR car c'est ce qui est utilisé afin de calculer les valeurs de l'utilité. Pour cela, on considère une transmission OFDM, une des technologies les plus utilisées de nos jours. Tout d'abord on considère les évanouissements plats que l'on note NB, et par la suite, nous considérons les évanouissements sélectifs en fréquence que l'on note WB. Dans le cas WB, l'abstraction de la couche physique utilise une métrique de SNR équivalente nommée EESM qui associe aux différentes valeurs de SNR corrélées sur les sous porteuses une seule valeur de SNR équivalent. La corrélation des variables aléatoires dans la métrique EESM nous empêche de trouver une forme exacte pour ses statistiques. C'est pourquoi, nous supposons que les évanouissements corrélés du canal peuvent être représentés par une succession d'évanouissements constant sur une bande de cohérence et indépendant et identiquement distribués d'une bande à l'autre. Cela permet de développer des calculs numériques afin de trouver les statistiques d'EESM. Pour compléter la procédure, on met en place une méthode permettant de calculer la valeur de la bande de cohérence afin que les statistiques calculées avec la méthode proposée correspondent aux statistiques d'EESM obtenues à travers le canal initial. Les résultats nous permettent de comprendre clairement le comportement de l'utilité en présence d'évanouissements de Rayleigh. Cela nous permet également de prédire les performances d'un système OFDM qui utilise la métrique EESM ce qui pourrait donner des pistes de travail sur le design des futurs systèmes sans fil utilisant OFDM. Enfin, le modèle de canal par blocs utilisé peut servir d'abstraction de couche physique avec une complexité peu élevée dans les simulations du Chapitre 4.

Le Chapitre 6 conclue cette thèse et fournit des perspectives de travail.

Chapter 1

General Introduction

1.1 Introduction

Ad hoc wireless networks have been attracted intensive studies in the past due to their ability of being deployed without infrastructure (*e.g.* [1] and the references therein). These networks experience a wide variety of dynamic changes due to the fact that nodes are mobile and their connectivity is subject to time varying wireless channel conditions. The interference management in such networks is therefore challenging especially with the lack of a central entity to perform it. This explains the limited implementation of these networks in practice.

Among the alternatives to manage resources in ad hoc networks, the clustered one is the method that ensures good performance when dealing with scale effects and mobility. Clustering keeps advantage of ad hoc networks (no infrastructure) while reducing the signaling overhead and allowing the implementation of such networks in practice [2, 3, 4]. The idea consists in gathering locally (and dynamically) nodes in "clusters" in order to optimize network performance and to manage locally cluster's resources. This task is often devoted to a Cluster Head (CH) that is elected among the cluster nodes. The role of the CH is to control the resource allocation and transmission schemes inside the cluster. The nodes in each cluster exchange signaling only with the CH, which reduces the signaling overhead as compared to traditional ad hoc networks. Clustered ad hoc networks are used in the context of vehicular networks, cognitive radio networks and military networks where nodes close to each other belong to the same cluster [5].

One of the major problem in clustered ad hoc networks deals with spectrum band allocated to clusters in order to make them able to operate. An ideal solution is to allocate orthogonally the frequency bands at the starting step of the algorithm but it is not operationally possible for the two following reasons. Firstly, because of the inherent dynamic nature of the network, the number of clusters may vary and so does the number of required frequency channels. Secondly, to cope with the scaling effect, in which the network is able to deal with an increasing number of clusters, the network would require more and more channels as the network grows up, which is impossible due to spectrum scarcity. To overcome these difficulties, a practical solution considers a finite spectrum resource and tries to look for a possible dynamic re-use of frequency bands. Indeed, the radio links are able to work when the interference is limited, for example, one may set a minimum Signal to Noise plus Interference Ratio (SINR). In this way, we can take advantage of the electromagnetic waves natural power decrease with respect to the distance (or obstacles) in order to re-use the same frequency band by different clusters.

In this thesis, we consider a clustered network where the nodes in each cluster communicate with each other and exchange signaling information with the CH. No information is exchanged between different clusters. In order to reduce the interference in the network, especially that no central entity exists in the network to perform instantaneous interference management, the clusters may use different frequency bands. Bandwidth is however a scarce resource which implies that several clusters may use the same frequency. A frequency allocation must then be conducted in a such a way to maximize the network performance. This allocation cannot unfortunately be handled off-line for the following reasons. The load of each cluster is time varying (since the traffic is usually time varying) which means that the interference from one cluster on another is time varying. The clusters are also mobile which means that two neighbor clusters could become distant after some time. The frequency allocation must then be dynamic to cope with mobility and dynamic clusters' load. The lack of central controller in the network implies that such a frequency allocation must be done in a distributed way. In each cluster, the CH decides at a given time the frequency band to be used in its cluster without exchanging any signaling information with other clusters. In addition, the allocation have to be robust to disturbances. The CH realizes the frequency allocation based on measurements that are possibly disturbed due to wide variety of dynamic changes that face ad hoc networks. In a distributed information/decision perspective it is natural to expect a change in the state of the network if the measurements have changed. In a disturbed environment this can lead the network to uncontrolled and unexpected decisions and deteriorate performance heavily.

1.2 State of the art

This section establishes more formally the context and the focus of the work and presents the state of the art of distributed resource allocation in clustered ad hoc networks.

In the context of distributed resource allocation, it is usual to employ Game theory (GT) as a modeling framework. The concepts of GT useful in this thesis are described in Chapter 2. The problem is thus modeled by a game in which we consider the clusters as players and their actions as frequency channel choices. Then, each cluster takes decision based on a metric named utility (or payoff). The framework of the resource allocation problem considered can be summarized by the following properties:

(i) the decision is made separately by each cluster,

(ii) the cluster' strategy depends only on its current and past utilities and actions (local information),

(iii) the utility does not have a closed form expression as it is a function of other clusters' actions and takes discrete values,

(iv) the utility can be disturbed by stochastic process (*e.g.* fast fading),

(v) the context is dynamic in the sense that clusters can move, leave or enter the network, and the load can vary,

(vi) the action set is discrete (selection of frequency channels).

Properties *(i)* and *(ii)* define a completely uncoupled scheme [6] that we name *fully distributed* in the thesis [7, 8]. It is important to specify that in the context of properties *(i)* and *(ii)*, and in a large class of games with finite memory, there exists no adaptive algorithm that converges almost surely to a Nash Equilibrium (NE) [6, 9, 10]. However, as one will see, it is possible to consider a weaker type of convergence (see Stochastic Stable State (SSS) and Section 2.2 for details).

With property *(iii)*, it appears that the model-free strategy learning algorithms are very appealing approaches [11] since in this paradigm, players neither try to model the environment nor try to have a specific/explicit utility form. They simply consider the environment as a black box and learn by interactions (*e.g.* trials and errors). This context, though very restrictive, can be encountered in a wide variety of examples. For instance, in a wind farm, each turbine controls the power that it extracts from the wind [12]. It is very difficult, if not intractable, to model the impact of a turbine on other turbines. In addition, the lack of communications between them makes impossible any cooperation. Another example is the case of commuters in city that want to avoid traffic jams but, they neither know the strategies of other commuters nor the impact of their strategy on the achieved rewards [13].

Property *(iv)* represents the fact that in the context of wireless telecommunications, the medium is disturbed by stochastic processes which, by consequence, affect the utility. These utility perturbations impact players decisions which can lead to the loss of convergence properties and thus to performance degradations.

Fully distributed resource allocation can be encountered in many wireless contexts since the nodes/players may not be able to exchange information between each other in order not to increase the overhead in the network. Also realistic utility functions of the users may not have closed form expression as denoted by property *(iii)* (*e.g.* Quality of Experience, number of correctly decoded packets, etc). Nonetheless, in this context some approaches to allocate resources with model-free strategic algorithms have been developed. Among them we have identified Q-learning, learning automata, combined payoff and strategy learning, no-regret learning and Trial and Error (TE) learning as potential solutions. For each of the previous of learning procedures, we present the basic principle without going into details of the analysis.

In [14, 15] authors use Q-learning algorithm to select frequency channels among femtocells. In [16] Q-learning is employed to allocate frequency channel and power in a fully distributed network. The basic principle of this learning procedure is to solve the Bellman's optimality equation [17]. The solution to these equations maximizes the sum of cumulative utilities over time. There are some drawbacks related to the use of this procedure in a fully distributed scheme. First of all, players require the state of the system which they do not have (they do not know the action of other players). Then, the optimality can be reached in a stationary environment which is not the case in multi player games. The decision of a player has an impact on future other players decisions and vice versa. Consequently, the environment is not stationary and the convergence of the previous algorithm is not ensured [11]. Usually, when implementing Q-learning in multi players environment, each player considers low or no interaction with others. In addition, in some cases where convergence can be obtained (*e.g.* low coupling between players), the convergence takes a large amount of time [11].

With Q-learning, each player tries to learn the expected cumulative reward to find a policy. In the opposite, there exists algorithms that directly learn the policy based on the received reward. They are known as learning automata [18, 19]. In [20, 21, 22] authors proposed an algorithm based on learning automata scheme whose principle is presented in [23]. The convergence proof is based on ordinary differential equation and stochastic approximation [24]. Each action is played with a given probability which is updated at each iteration using a stochastic approximation. For a given action selected, the probability to play it in the next iteration increases especially as the received utility is great. Other probabilities are computed using conservation probability. It is proved in [23] that the stationary points of the algorithm are Pure Nash Equilibrium (PNE) (see Definition 2.2 in page 12 for details). The main advantage of these type of algorithms, in addition to be fully distributed, is that they are designed to evolve in an environment where the utility can be disturbed. This is a consequence of the stochastic approximation procedure. However, there are some issues arising when using the above methods. The first one is that nothing more can be said about the efficiency of the PNE reached. For instance, it is shown in [25] that the algorithm can converge to a suboptimal state. In addition, in the restrictive environment considered in this thesis the existence of PNE is difficult to predict which makes the convergence of the above procedure not ensured.

Previous algorithms learn the expected reward or the policy but there exists schemes that use both information at the same time. In [25, 26] authors use combined payoff and strategy learning approaches to respectively share the resources among femtocells or wifi access points. The basic principle of these algorithms can be described as follows. It is composed of two steps. During one step, the average utility value corresponding to the use of that action is updated using the received utility and a stochastic approximation method. In a second step, the probability distribution over actions is updated similarly to the learning automata scheme at the difference that a mapping function with some specific properties is involved (see [26] for more details). Thanks to this function and the finiteness of the game, the algorithm possesses a unique fixed point to which the algorithm may converge to. One can force the fixed point to be a uniform distribution over the action space or a pure action. It is interesting to note that as in the learning automata procedure, the algorithm is naturally robust to utility disturbances thanks to the stochastic approximation. In comparison to learning automata, combined learning improves the convergence performance as observed in [25]. However, the authors in [25] also note that while the fixed point equilibrium can correspond to a PNE, the convergence to this equilibrium may not be ensured if the utility is too selfishness (no communication between players). In addition to that, there are no guaranties on the performance of the equilibrium reached. It is interesting to note that these types of algorithms are part of a more general concept named combined fully distributed payoff and strategy reinforcement learning (CODIPAS-RL) described in [7].

The no-regret procedure is employed in [27, 28] in order to allocate power to channel and frequency channel respectively, in a fully distributed way. In comparison to algorithms presented previously, the no-regret learning has a totally different philosophy. Players do not try to directly increase their utility but rather try to minimize their regret. The regret of playing an action is the difference between what a player get in average by playing this action and what he would have received by playing a different one. The regret can then be used to update the probability among actions just as a learning automata would have done (*e.g.* [29]). For instance, one can set this update such that the probability of playing actions with low regret is high. No-regret procedures as the adaptive process in [29] converge to the set of correlated equilibrium which contains the set of NE [30]. In this sense any finite game has always at least one correlated equilibrium. Even though a correlated equilibrium is more general than a PNE, there are no theoretical guaranties of better performance. However, looking for a correlated equilibrium can highlight better social performance than just looking for a PNE [11]. The no-regret learning scheme [29] is not fully distributed in the sense that players need to observe actions of others to compute regret. In [27, 28] authors modified the algorithm to make it fully distributed. Algorithm [29] is not robust to disturbances that is why [28] proposes a modification to face disturbance using stochastic approximation of the regret. Finally, to compute the regret, the player needs to know the utility it would have received if it has played a different action, which in our case is not available at each step of the algorithm and needs to be learnt.

The last class of algorithms presented here are the TE based learning procedures. Most of the existing work based on TE schemes focuses on deterministic payoff functions in which the utility is not disturbed by a stochastic process. Authors in [31, 32] use two TE based algorithms, namely the ODL [12] and the TEL [10] to allocate in a fully distributed way frequency and power to a clustered ad hoc network. A modified version of ODL has been used in [33] to allocate frequency channels in an ad hoc network and in the context of very slow shadowing variations, in which the algorithm can be run and converge before the variation of the shadowing. The description of both TE algorithms is relegated to Chapter 3. Basically, each player implements a state machine in which two different behaviors can be distinguished. In the first mode, a player experiments with low probability a different action which is a standard search/exploration phase. In the second mode, a player experiments randomly any action which is a "noisy" search phase. In both modes, the experimented action has a probability to be accepted that increases with the utility received. The finite state machine defines which mode to use depending on the received utility. Algorithm ODL achieves a state that maximizes the social welfare value (sum of all clusters' utilities, see Definition 2.3 in page 12) regardless the existence of PNE. Algorithm TEL has the following particular interest: if a PNE exists, the system spends a high amount of time in a state that maximizes the social welfare function whereas, if a PNE does not exist, the algorithm allows achieving a state that represents a trade-off between stochastic stability and maximizing the social welfare. The aforementioned TE based algorithms loose their convergence property when the payoff is affected by a stochastic process, which is a common assumption in radio channel propagation (fading, shadowing). In the domain of control, recent work [34] is the first to propose a modification of ODL to face disturbances. In this modification, the maximum disturbance deviation that affects the utility is supposed to be constant and known in advance.

Among the model-free learning approaches discussed, TE schemes seems to be, for the time being, the solution whose convergence is well specified and that fulfills almost all the conditions required except property *(iv)*. Q-learning directly tries to maximize a cumulative sum of rewards. However, it has originally been designed for a single player problem making the convergence in multi players not ensured or possibly very long. Learning automata converges to the set of PNE whose existence is not ensured in our cases. In addition the performance of the NE reached is not guaranteed to be efficient. Combined learning of payoff and strategy can converge to a fixed point that corresponds in some cases to a PNE but the efficiency of the fixed point is not known. In addition, in a fully distributed scheme the procedure is not guaranteed to converge. However it seems to outperform learning automata [26]. Regret learning converges to a different type of equilibrium (*i.e.* correlated equilibrium) that embeds the NE equilibrium. To that end it converges in a larger class of games and can reach more efficient states than NE. This class of learning procedures are however not initially designed to work in a fully distributed scheme. They require some modifications to be adapted that make the convergence much slower and not ensured.

The TE approaches possess very attractive global convergence behavior despite the non-cooperativeness of the problem, while remaining quite simple to implement. The possibility to characterize so accurately the theoretical performance of TE schemes is due to the convergence notion of SSS that is less restrictive than almost sure convergence. In this convergence concept, rather than converging with probability one, the network spends a high proportion of time in the optimal state and can leave it. However, it is proved in [6] that there exists no fully distributed procedures that converges almost surely to a NE and thus, the convergence result of the TEL algorithm is the strongest that one would achieve in our thesis. Nonetheless, as it is noted in [35], these procedures exhibit instability when the utility is subject to disturbances. More specifically, the aforementioned TE algorithms loose their convergence property when the utility is disturbed by a stochastic process (it is proved in Chapter 4). The reason is that they are very sensitive to utility changes in the network. In fact, for the same network action, the utility takes several values due to stochastic variations. It follows that clusters interpret these disturbances as network action changes and, they react to these stimulus perturbing the overall network. One of the main contribution of this thesis consists in analyzing and proposing a solution to the problem of disturbed utility in TE schemes (TEL and ODL) under specific settings.

Beyond the convergence notion and the fact that the above two algorithms converge to a desired state, the study of these specific convergence properties (*i.e.* SSS) remains an open question [10, 11]. The main reason comes from the computation complexity of the inherent Markov Chain (MC) generated by these two algorithms. In fact, the game in which players employ these learning schemes can be represented by discrete MCs with huge number of states. Obtaining the transitions matrix of these MCs is therefore not tractable which makes the analysis of the convergence rate not possible (even numerically). This problem is one of the important contributions studied in this work.

1.3 Outlines

This thesis is composed of six chapters including this one: Theoretical aspects (Chapter 2), Fully distributed channel allocation without disturbances (Chapter 3), Fully distributed channel allocation with disturbances (Chapter 4), Statistics of the disturbed utility (Chapter 5) and a conclusion (Chapter 6). We provide hereafter a brief summary of each chapter.

Chapter 2 presents the theoretical notations and background that are necessary to understand the resource allocation modeling and the main results of the thesis. More specifically, Section 2.1 introduces the game theory and associated concepts related to our work. Section 2.2 introduces the MC representation and the extended concept of regularly perturbed MC that is of main interest in the convergence study of algorithms employed.

Chapter 3 recalls the problem of frequency resource allocation without disturbances [35]. The problem can be solved efficiently with TE based algorithms whose operations as well as their convergence results are presented in details. However, their convergence rates remains an open question due to the huge dimension of the inherent MC generated by these algorithms. Thus, the second part of this chapter is dedicated to their performance analysis under a simplified utility model. The principle of this analysis is to propose an approximation of the MC to reduce the number of states such that it enables numerical computation of performance. This complexity reduction is based on finding symmetries in the MC and on approximations of algorithms transitions. With these approximations, we are able to realize a comparison of two TE based algorithms in regions that are unreachable with Monte Carlo simulations.

Chapter 4 still considers the problem of frequency resource allocation except that we include stochastic disturbances in the utility function. We first introduce a general disturbed utility model whose aim is to model the impact of stochastic perturbation on the utility. In this disturbed context, the behavior of standard TE algorithms presented in Chapter 3 is analyzed and decrypted. It is shown, theoretically, that in a disturbed environment those algorithms fail to keep good convergence properties. Therefore, we propose modifications of these algorithms and, we provide theoretical proofs of convergence. These solutions are then extended to a more general framework in which some assumptions about the disturbance are released. Then, we study with more details the impact of the proposed solutions on the global convergence of the algorithms.

Chapter 5 is devoted to the analysis of the physical layer abstraction used in Chapter 4 in the context of Rayleigh fading. The specific goal of this chapter is to study the statistics of the utility to understand the impact of Rayleigh fading on resource allocation algorithms in Chapter 4. These statistics computations require the statistics of the SINR since it is the input of the utility function. We consider an Orthogonal Frequency Division Multiplexing (OFDM) transmission scheme as it is one of the most widely used technology nowadays. First, we deal with flat fading referred to as Narrow Band (NB) fading and then, we consider the frequency selective fading referred to as Wideband (WB) fading. In the later case, the abstraction of the physical layer uses the effective SNR metric named Effective Exponential SNR Mapping (EESM) that maps the correlated SNRs over each subcarrier into one effective SNR. The EESM does not possess an appropriate form for mathematical manipulation which prevents us from providing any theoretical analysis of its statistics and hence of the utility statistics. We thus extend this abstraction assuming that the channel exhibits a coherence bandwidth which enables us to consider the fading flat inside successive bandwidths. To complete this procedure, we provide a method based on statistical tests to set the coherence bandwidth parameter such that the new abstraction represents faithfully the effective SNR computed using a real channel. The proposed abstraction allows numerical computation of the utility statistics which enables us to clearly understand its behavior. It also enables us to predict the performance of a WB OFDM system, to provide insights on the design of future wireless system based on OFDM modulation, and to give a simplified abstraction to model WB fading in numerical simulations.

Chapter 6 concludes this thesis and provides outlooks for future works.

1.4 Contributions

This work has led to a publication and a patent.

Conference

- J. Gaveau, C. J. Le Martret, M. Assaad, " Grouping of subcarriers and effective SNR statistics in wideband OFDM systems using EESM", in **2017 IEEE 13th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)**, Oct 2017, pp. 1-7.

Patent

- J. Gaveau, C. J. Le Martret, M. Assaad, "Procédé d'allocation de ressources radio dans un réseau sans fil, par apprentissage", **INPI**, 2017.

Chapter 2

Theoretical aspects

This chapter presents the main theoretical tools of interest used in the thesis. The problem considered deals with the fully distributed allocation of frequency channels among clusters. Since game theory studies behavior of interacting agents, it appears as a natural tools to tackle this problem. In this context, we analyze algorithms whose theoretical proofs of convergence rely on perturbed Markov Chain (MC) theory. This chapter is organized as follows. First, we present basic concepts of game theory. In the second section, preliminary definitions on MC and the specific case of regularly perturbed MC are highlighted.

2.1 Game theory

Game theory (GT) is a mathematical framework to study and analyze the outcome resulting from the interaction between decision-takers named players. Each player chooses an action from a set and receives an outcome named utility (or payoff). A game is the association of a set of players, a set of strategies and utilities. Classically, GT is separated in cooperative and non-cooperative games. In cooperative GT, players can form coalitions to maximize their utility. Whereas, in non cooperative GT, players act independently without cooperating. In both cases, GT studies the equilibrium reached through players' interactions. In this thesis, we assume that players that represent clusters cannot communicate and thus do not cooperate. Thus, non-cooperative GT is our main focus. We model the network using the normal form as follows.

Definition 2.1 (Normal form game). *A normal form game with K players is defined by the triplet $\mathcal{G} = (\mathcal{K}, (\mathcal{N}_k)_{k\in\mathcal{K}}, (u_k)_{k\in\mathcal{K}})$ where $\mathcal{K}$ is the set of players, $\mathcal{N}_k$ is the set of player k strategies and, $\forall k \in \mathcal{K}$, $u_k : \mathcal{N}_1 \times \cdots \times \mathcal{N}_K \to \mathbb{R}$ is the utility function of player k.*

The vector of all strategies or actions, named action profile, is noted $\mathbf{a} = (a_1, \dots, a_K) \in \mathcal{N} = \mathcal{N}_1 \times \dots \times \mathcal{N}_K$. The utility received by each player depends on other players' actions so it is a function if this action profile. It is convenient to highlight the action taken by a given player in the action profile as follows $\mathbf{a} = (a_k, \mathbf{a}_{-k})$ where a_k is the action of cluster k and $\mathbf{a}_{-k}$ is the action of all players except k. One of the most studied solution concept in non-cooperative GT is the Pure Nash Equilibrium (PNE) defined as follows.

Definition 2.2 (Pure Nash Equilibrium)**.** *A pure Nash equilibrium is an action profile $\mathbf{a}^*$ in which no player has an incentive to deviate from its action. Then $\forall k \in \mathcal{K}$ and $\forall a_k \in \mathcal{N}_k$,*

$$u_k(a_k, \mathbf{a}^*_{-k}) \leq u_k(\mathbf{a}^*). \tag{2.1}$$

The PNE defines actions that are played with probability one. In this sense it is a specific case of Mixed Nash Equilibrium (MNE). In this last case, each player can select actions according to a probability distribution which is not restricted to play one action with probability one as in the PNE. The MNE is thus a generalization of PNE in which no player can increase its expected utility by unilaterally changing its probability distribution over actions. In any finite game, the existence of a PNE is not always ensured contrary to the MNE [36].

The overall performance of a game can be measured in terms of the sum of all utilities named the social welfare.

Definition 2.3 (Social welfare)**.** *For any action profile $\mathbf{a}$, the social welfare is defined as*

$$W(\mathbf{a}) := \sum_{k \in \mathcal{K}} u_k(\mathbf{a}). \tag{2.2}$$

and the average social welfare with respect to the number of player is

$$W_m(\mathbf{a}) := \frac{W(\mathbf{a})}{K}. \tag{2.3}$$

Going further with GT necessitates some assumptions in the structure of the game. For instance, games can be zero sum, potential, congestion, etc or a combination of these properties. There are many types of games and giving an exhaustive list would be irrelevant for the comprehension of this work (for more details see [37]). However, it is important to understand that each property gives insights on the interactions between players through their utility function and provides information on the existence of specific equilibriums such as the PNE. In this work we consider a very general assumption supposing that our game is *interdependent*.

Definition 2.4 (Interdependence). *The game $\mathcal{G}$ is interdependent if, $\forall \mathbf{a} \in \mathcal{N}$, and all proper subset of cluster $\emptyset \subset J \subset \mathcal{K}$, $\exists i \notin J$, $\exists \mathbf{a}'_J \in \prod_{k \in J} \mathcal{N}_k$ such that $u_i((\mathbf{a}'_J, \mathbf{a}_{-J})) \neq u_i(\mathbf{a})$.*

Literally, any subset of player(s) can modify the utility of a player not in the set by an appropriate action change. It is a less restrictive hypothesis than genericity (*i.e.* games in which the utility of each player is modified by any action change) which implies interdependence [10]. The meaningful example of a traffic game given in [10] gives one pertinent view of this phenomenon. In this game, agents can select the route of their choice. There are payoff ties because a local change of route by one player does not change the payoffs of players on different routes. However, it satisfies the interdependence condition because a given player, or set of players, can switch to a route that is being used by another player and hence change his payoff.

The interdependent property, regardless of the utility function form, is the only assumption made about the structure of the game in this thesis. It means that we consider a very general framework. In that context, we study algorithms whose convergence proofs rely on perturbed MC presented in the next section.

2.2 Markov chains and regular perturbations

This section presents notions relative to MC and perturbed MC. The convergence proofs of the proposed algorithms that are given in Chapter 4 rely on perturbed MC properties. This requires the introduction of MC notions and their transition matrix representation.

Let $n \in \mathbb{Z}$ and $(X_n)_n$ be a sequence on a countable space Ξ. This process can be represented by a matrix if it is a homogeneous MC.

Definition 2.5 (Markov property). *The process $(X_n)_n$ is a MC if and only if for all n and all $(x_0, \ldots, x_n) \in \Xi^{n+1}$,*

$$\Pr\{X_n = x_n | X_{n-1} = x_{n-1}, \ldots, X_0 = x_0\} = \Pr\{X_n = x_n | X_{n-1} = x_{n-1}\},$$

where $\Pr\{X|Y\}$ is the probability for event X to happen knowing that Y happened. In other word, a process is a MC if the probability that an event happens depends only on the previous event.

Definition 2.6 (Homogeneous Markov chain)**.** *The process $(X_n)_n$ is a homogeneous MC if and only if for all n and all $(x_0, x_1) \in \Xi^2$,*

$$\Pr\{X_n = x_1 | X_{n-1} = x_0\} = \Pr\{X_1 = x_1 | X_0 = x_0\}.$$

With these properties, one can represent the process $(X_n)_n$ with one transition matrix. This representation is defined as follows.

Definition 2.7 (Transition matrix)**.** *The transition matrix of the homogeneous MC $(X_n)_n$ is $(\mathbf{P}_{x,x'})_{(x,x')\in\Xi^2}$ where*

$$P_{x,x'} = \Pr\{X_1 = x' | X_0 = x\}.$$

This representation appears to be very useful to compute probabilities in MC. For instance, the probability to get from x to x' in p steps is given by $\Pr\left\{X_p = x' | X_1 = x\right\} = (\mathbf{P}^p)_{x,x'}$.

The study of perturbed MC relies on trees manipulations which can be seen as specific type of graph. With the previous matrix representation, one can see the MC as a graph where some states are *accessible* from one another.

Definition 2.8 (Accessibility)**.** *Let $(x, y) \in \Xi^2$, a state x is accessible from y, and is noted $x \leftarrow y$, if there exists a finite path with positive probability from x to y.*

In other words, there exists a finite $n > 0$ such that $(\mathbf{P}^n)_{y,x} > 0$. One can see the process induced by $\mathbf{P}$ as a graph where the state in Ξ are the vertices and edges are any two states with a positive transition in $\mathbf{P}$ (*i.e.* $(x, y) \in \Xi^2$ is an edge if $P_{xy} > 0$). In addition, if $x \leftarrow y$ and $x \rightarrow y$, x and y are said to *communicate* which is noted $x \leftrightarrow y$. All states that communicate form a *communication class*. All states of a class are either *transient* or *recurrent*.

Definition 2.9 (Transience and recurrence)**.** *A state in Ξ is recurrent if there is a probability 1 to return to it. In the contrary, a state in Ξ is transient if there is a positive probability to never return to it.*

Note that, a class with recurrent (transient) states is a recurrent (transient) class. Intuitively, there is a positive probability to leave and never return a class with transient states whereas, there is a null probability to leave a class with recurrent states once entered.

Proposition 2.1. *All communication classes of a given MC form a partition of the set Ξ. Let note $C_1, C_2 \ldots$ all communication classes, that are disjoints, then*

$$\Xi = \bigcup_{i=1} C_i.$$

Definition 2.10 (Irreducibility)**.** *A MC composed of one communication class is irreducible.*

In addition to that, any MC that is finite and irreducible is recurrent. The study of perturbed MC (defined in the sequel) is based on the study of the unique stationary distribution of that MC. In few words, the stationary distribution of a MC is the probability distribution over states after a sufficient long time. The proportion of time spends in states converges to the stationary distribution if in addition to be irreducible, the finite MC is *aperiodic*.

Definition 2.11 (Aperiodicity)**.** *A state $x \in \Xi$ is aperiodic if and only if there exists $n > 0$ such that $(\mathbf{P}^n)_{x,x} > 0$. A MC is aperiodic if and only if x is aperiodic for all $x \in \Xi$.*

Contrary to the case in which we consider a MC *periodic*, that is to say when the probability of reaching a given state is positive only periodically, when the MC is *aperiodic* (and irreducible) the process converges to one stationary distribution.

We now have all the notions to describe a regular perturbed Markov chain. In what follows, the exponent in $\mathbf{P}^v$ defines the value of a perturbation v (not the number of times the matrix is multiplied by itself).

Definition 2.12 (Regular Perturbed Markov Process [38])**.** *A Markov process $\mathbf{P}^\epsilon$ is a regular perturbation of $\mathbf{P}^0$ if the next conditions hold for all $x, y \in \Xi$*

(i) $\mathbf{P}^\epsilon$ *is aperiodic and irreducible for all* $\epsilon > 0$.

(ii) $\lim_{\epsilon \to 0} \mathbf{P}^\epsilon_{x,y} = \mathbf{P}^0_{x,y}$.

(iii) *If* $\mathbf{P}^\epsilon_{x,y} > 0$, $\exists r \geq 0$, *s.t.* $0 < \lim_{\epsilon \to 0} \epsilon^{-r} P^\epsilon_{x,y} < \infty$.

We define r_{xy} the resistance of the transition (x, y) as $r_{xy} := r$.

From property *(i)* and previous definitions, the matrix $\mathbf{P}^\epsilon$ has a unique stationary distribution noted π^ϵ. In general, $\mathbf{P}^0$ is composed of several communication classes (see Proposition 2.1) and ends necessarily in one of them (see Definition 2.9 and the remark below). Therefore, the process $\mathbf{P}^0$ has several stationary distributions (one for each recurrent class) and converges to one of them. When the perturbation ϵ is strictly positive, the perturbed process $\mathbf{P}^\epsilon$ is able to jump from one recurrent class of $\mathbf{P}^0$ to others. Consequently, $\mathbf{P}^\epsilon$ has a unique communication class and a unique stationary distribution. When the perturbation decreases to 0, condition *(ii)* specifies that $\mathbf{P}^\epsilon$ becomes progressively $\mathbf{P}^0$. This means that the process selects progressively one of the recurrent class of $\mathbf{P}^0$ and that the states of this class have a non vanishing stationary probability. This introduces the concept of SSS.

Definition 2.13 (Stochastic stable state)**.** *A state $x \in \Xi$ is a SSS relative to the process* $\mathbf{P}^\epsilon$ *if* $\lim_{\epsilon \to 0} \pi^\epsilon > 0$.

When the perturbation goes to 0 the probability to observe a state that is not a SSS decreases to 0. This type of convergence is less restrictive than almost sure convergence or convergence with probability 1.

The convergence of the algorithms studied in Chapters 3 and 4 relies on the specific property that possesses the SSS of the regularly pertubed MC. This property is linked to the studies of the spanning trees over states in Ξ (*i.e.* the trees whose vertex are all states in Ξ). The process described by $\mathbf{P}^\epsilon$ can be seen as a graph with states in Ξ as vertices. Each edge links a couple of states that communicate in one step of the process (that has a positive probability in $\mathbf{P}^\epsilon$). In perturbed MC theory, the edges of the trees are weighted by the resistance defined in condition *(iii)* of Definition 2.12. In addition, one needs to define the resistance of a tree.

Definition 2.14 (Resistance of a tree)**.** *Let τ_x be a spanning tree in Ξ that is rooted in $x \in \Xi$. The resistance of τ_x is given by the sum of the resistance of all its edges,*

$$r(\tau_x) = \sum_{(i,j) \in \tau_x} r_{ij} \tag{2.4}$$

The set of trees rooted in x is noted $\mathcal{T}_x$. The minimum resistance among all spanning trees rooted at a given state is the *stochastic potential* of this vertex.

Definition 2.15 (Stochastic potential)**.** *The stochastic potential of a state $x \in \Xi$ is the least resistance tree among all spanning trees rooted at x*

$$\rho(x) = \min_{\tau \in \mathcal{T}_x} r(\tau). \tag{2.5}$$

With the previous definitions, it is possible to state the major result related to regularly perturbed MCs that is proved in [38].

Theorem 2.1. *Let $\mathbf{P}^\epsilon$ be a regularly perturbed MC. The SSSs are the states contained in the recurrent communication classes of $\mathbf{P}^0$ with the minimum stochastic potential.*

The details of the proof are out of the scope of this chapter but their meaning is useful to understand the convergence proofs of the algorithms studied in this thesis. From this theorem, it follows that when ϵ decreases to 0, the states with the minimum stochastic potential have a non vanishing stationary probability in comparison to other states. Thus, the process remains a longer time in those states than in others. Minimizing the stochastic potential is equivalent to solve an optimization problem. The particularity of the algorithms studied in this thesis is that despite the non-cooperative characteristic of the considered games, they find a global cooperative optimum in the long run. They converge to a state that minimizes the stochastic potential which has the particularity to take into account the utility of all players in the game. To prove such a result, one needs to compute the stochastic potential of the MC process over Ξ which can be really cumbersome for two reasons. The first reason is the dimension of the state space Ξ and the second reason is the complexity of MC's structures induced by the algorithms. The first difficulty is lowered in the same seminal paper [38]. The author shows that the study of the trees induced by the process $\mathbf{P}^{\epsilon}$ (*i.e.* over all states in Ξ) can be reduced to the analysis of graphs considering only the recurrence classes of $\mathbf{P}^{0}$ as vertices. The number of edges in the new graph is hence significantly reduced. The second difficulty is solved by appropriately designing the algorithms such that, the study of the trees becomes feasible and, the stochastic potential corresponds to the desirable global behavior of the system [10, 12].

Chapter 3

Distributed channel allocation without disturbances

This chapter describes solutions to the decentralized resource allocation scheme based on two Trial and Error (TE) algorithms namely, Trial and Error Learning (TEL) and Optimal Distributed Learning (ODL). They are initially designed to evolve in a deterministic environment (no random varying utility). They show the particularity to exhibit interesting cooperative convergence properties in a broad class of games even though the game is non cooperative. In fact, ODL has the property to spend a high proportion of time in an optimum state that maximizes the sum of the utilities of all players. The TEL has the property to spend a high proportion of time in an optimum state that maximizes the sum of the utilities of all players if there exists a Pure Nash Equilibrium (PNE), otherwise, it spends a high proportion of time in an optimum state that maximizes a trade-off between the sum of the utilities of all players and a predefined stability function. The author in [35] proposed an adaptation of these algorithms to the deterministic telecommunication context. It is thus of main importance, in Section 3.1, to present a brief summary of this adaptation. Section 3.2 discusses the existence of PNE in the game considered. Section 3.3 describes the TE algorithms' mechanisms. Section 3.4 presents the associated theoretical results and highlights preliminary numerical results. Section 3.5 tackles the problem of analyzing the performance of these algorithms. Even though they converge to a desired state, the convergence rate remains an open question [10, 11]. For instance, estimating the mean fraction of time spent in the optimum state as well as the mean time duration to reach it is challenging due to the high complexity and dimension of the inherent Markov Chain (MC). Section 3.6 concludes this chapter.

3.1 Model

This section presents the deterministic model of clustered ad hoc wireless networks. Deterministic means that no stochastic variations are considered in the model (static nodes, no time varying channel). Nonetheless, stochastic variations will be taken into account in Chapter 4 and the model will be modified accordingly. First, we present the wireless telecommunication model and its mathematical representation. Then, the metric available to take decision are detailed and a game model is formulated.

We consider a wireless network composed of K clusters with possibly different sizes. The set of clusters is denoted by $\mathcal{K} = \{1, \ldots, K\}$. The allocation of resources in each cluster $k \in \mathcal{K}$ is handled by one node called the Cluster Head (CH). The CH chooses the frequency channel to use in its own cluster without communicating with other CHs. The set of N possible resources is denoted by $\mathcal{N} = \{r_1, \ldots, r_N\}$. For any cluster $k \in \mathcal{K}$, the action of selecting a given resource allocation is denoted by the discrete variable $a_k \in \mathcal{N}$. Vector $\mathbf{a} = (a_1, \ldots, a_K) \in \mathcal{N}^K$ represents the actions of all clusters. In a cluster, a link i is a couple composed of a transmitter i_t and a receiver i_r and is denoted by $i = (i_t, i_r)$. A link i is said to be a working link if its SINR at the receiver side i_r is above a predefined threshold $\Gamma_0 \geq 0$. Interference is caused by the clusters using the same resource. The instantaneous SINR expression of link i in cluster k, considering only Additive White and Gaussian Noise (AWGN) channels is (time index is removed to simplify the notation)

$$\mathrm{SINR}^k_{i_t,i_r} = \frac{\mathrm{SNR}^k_{i_t,i_r}}{1 + \sum_{n \in \mathcal{I}_k} \sum_{j \in \mathcal{M}_n} \mathrm{INR}^{n,k}_{j,i_r}}, \tag{3.1}$$

where $\mathcal{M}_n$ is the set of transmitting nodes at time t in the cluster $n \in \mathcal{K}$, $\mathcal{I}_k$ is set of clusters interfering (operating on the same resource) with the cluster k, $\mathrm{SNR}^k_{j,i} = \frac{g^k_{j,i} P^k_j}{N_0 W_{nb}}$ and $\mathrm{INR}^{n,k}_{j,i} = \frac{g^{n,k}_{k,i} P^n_j}{N_0 W_{nb}}$, with $g^{n,k}_{j,i}$ the path loss from user j in cluster n to user i in cluster k (if the two nodes j and i are in the same cluster k then we write any value $x^{k,k}_{j,i} = x^k_{j,i}$, *e.g.* $g^{k,k}_{j,i} = g^k_{j,i}$), with P^k_j is the transmit power by node j in cluster k, N_0 is the thermal noise density such that $N_0 = k_b T$ with k_b the Boltzmann constant, T the temperature and finally, W_{nb} is the signal bandwidth. Note that the noise power in this band is $P_b = N_0 W_{nb}$. The channel gain is computed with a three slopes model, $g^{n,k}_{j,i} = \frac{(d^{n,k}_{j,i})^{-2}}{4\pi} \left(1 + \frac{d^{n,k}_{j,i}}{100}\right)^{-1} \left(1 + \frac{d^{n,k}_{j,i}}{1000}\right)^{-1}$, where $d^{n,k}_{j,i}$ is the distance between node j in cluster n and node i in cluster k. In the sequel, for each cluster k, the couple transmitter-receiver $i = (i_t, i_r)$ refers to link i of cluster k and, when there is no ambiguity, it is simply denoted by i for ease of notation (*e.g.* $\mathrm{SINR}^k_{i_t,i_r}$ becomes

SINR_i^k). It is worth mentioning that the set of interfering clusters $\mathcal{I}_k$ depends on the discrete action vector $\mathbf{a}$. At each iteration, a link $i = (i_t, i_r)$ in cluster k is said to be working if the received SINR is greater than a given threshold Γ_0. Receiver i_r computes for transmitter i_t in its cluster the working link metric of i

$$\ell_i^k(\mathbf{a}) := \mathbb{1}_{\{SINR_i^k(\mathbf{a})>\Gamma_0\}}, \tag{3.2}$$

where $\mathbb{1}_{\{x\}}$ is the indicator function equal to 1 if the condition into bracket is verified and 0 otherwise, $\mathrm{SINR}_i(\mathbf{a})$ is the SINR measured at the current iteration with (3.1) while the action profile is $\mathbf{a}$. Note that this metric is already used to design a utility function in [39] where the utility of each cluster is represented by one link (*e.g.* without power considerations $u_k(\mathbf{a}) = \ell_i^k(\mathbf{a})$).

A more detailed model has been proposed in [31] in which the utility is composed of the working metric of all links inside the cluster as follows

$$u_k(\mathbf{a}) := \frac{1}{|\mathcal{L}_k|} \sum_{i \in \mathcal{L}_k} \ell_i^k(\mathbf{a}), \tag{3.3}$$

where $\mathcal{L}_k$ is the set of links in cluster k and, $|\mathcal{L}_k|$ is the number of links in cluster k. Then, $u_k \in \mathcal{U} = [0, 1]$ and is discrete. Note that it is assumed that the CH can communicate with all nodes even if their respective communication links are not working. For instance, the nodes could feedback the link metric on a control channel with low throughput that is more robust than the intended communication link. This model of utility is used in Section 3.4.2 and will be adapted to the stochastic context in Chapter 4.

The resource allocation problem in the clustered ad hoc network can be represented by a non-cooperative game in normal form $\mathcal{G} = (\mathcal{K}, (\mathcal{N})_{j\in\mathcal{K}}, (u_j(\mathbf{a}))_{j\in\mathcal{K},\mathbf{a}\in\mathcal{N}^K})$. The allocation is then realized in this way. At each time, the receiver of link i in cluster k feedbacks the bit ℓ_i^k to the CH. The CH of cluster k knows hence the value of the working metric for all link i inside its cluster. Each CH knows a numerical realization of the instantaneous utility of its own cluster only (no signaling between clusters). Based on this knowledge, the CH decides a resource allocation action a_k in such a way to maximize its utility. It is worth mentioning that due to the lack of signaling between the clusters, the CH cannot know the interference.

3.2 Discussion on PNE existence

Naturally, the question of the existence of a PNE in game $\mathcal{G}$ arises. It is worth mentioning that from Nash theorem in [36] game $\mathcal{G}$ possesses at least one NE (at least a mixed NE), which follows from the finiteness of the game. First of all, when the number of resources is greater or equal than the number of players (*i.e.* $N \geq K$) the existence of a PNE is ensured. For instance, when all players use a different resource then, their utilities is maximum. However, when $N < K$ the existence of a PNE turns out to be not ensured. Authors in [40] propose a meaningful example to justify the possible non existence of PNE in clustered ad hoc networks. Briefly, the lack of an equilibrium is due to the asymmetry of interference between clusters.

We illustrate this phenomenon by considering the network illustrated in Figure 3.1 and the model described in Section 3.1. There are three clusters with two nodes each and $N = 2$. In this model, the received utility is represented in the cube in Figure 3.2. Each corner represents a network resource configuration $\mathbf{a}$ (the top vector) and, the associated utility vector $\mathbf{u}$ (the bottom vector). Due to the asymmetry of interference, when two clusters interfere, one of them has always an incentive to deviate. This later, when changing its action is still interfered but less than before. The reason of this asymmetry is well illustrated in Figure 3.1. For instance, let consider that clusters 1 and 2 employ the same resource. Then, both nodes of cluster 2 are strongly interfered whereas only one node is strongly interfered in cluster 1. Consequently, one link in cluster 1 works whereas no link in cluster 2 works and the utilities are $u_1 = 0.5$ and $u_2 = 0$. The same reasoning applies successively to clusters 2 and 3 and, clusters 3 and 1. In Figure 3.2, the best response cycle represented by the red arrows shows that there is no PNE in this case.

Consequently, this example shows that depending on the number of resources N and, on nodes' positions, the existence of a PNE is not ensured. That is why we focus in this work on two fully distributed algorithms that exhibit good convergence properties even if a PNE is not available. These algorithms are presented in the next section. It is worth mentioning that one of the two algorithms still highlights specific convergence properties when a PNE exists.

3.3 Fully distributed trial and error algorithms

This section focuses on the description of two fully distributed algorithms that can be adapted to tackle the resource allocation problem. These algorithms, TEL and ODL, exhibit good convergence properties as it will be presented in Section 3.4.1. Both share common characteristics. Each player $k \in \mathcal{K}$ implements a Finite State Controller (FSC) composed of states called moods and noted m_k and,

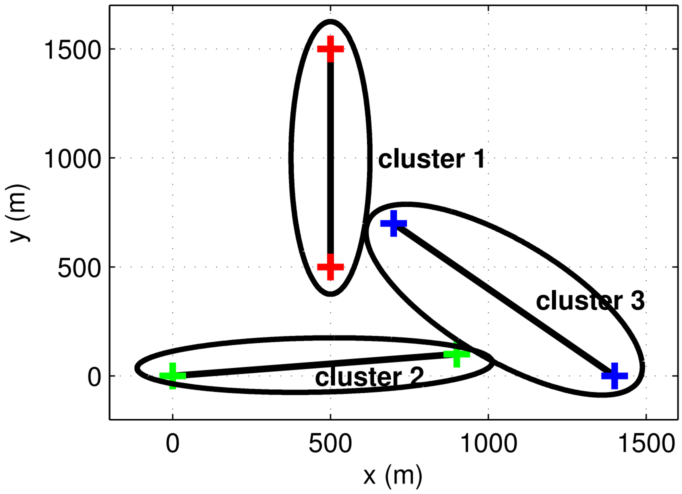


Figure 3.1: Wireless ad hoc clustered network with 3 clusters composed of 2 nodes each. Crosses represent mobile terminals, and ellipses the affiliations of terminals to cluster.

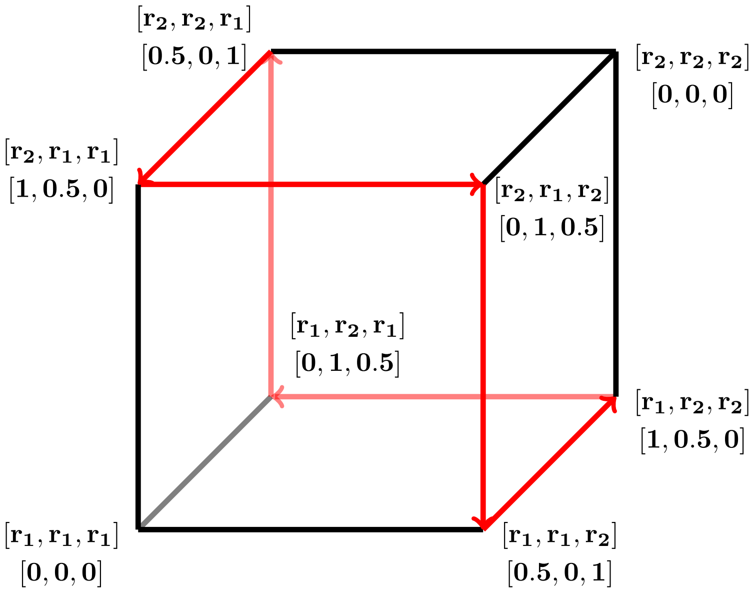

$[\mathbf{r_2}, \mathbf{r_2}, \mathbf{r_1}]$
$[0.5, 0, 1]$
$[\mathbf{r_2}, \mathbf{r_2}, \mathbf{r_2}]$
$[0, 0, 0]$
$[\mathbf{r_2}, \mathbf{r_1}, \mathbf{r_1}]$
$[1, 0.5, 0]$
$[\mathbf{r_2}, \mathbf{r_1}, \mathbf{r_2}]$
$[0, 1, 0.5]$
$[\mathbf{r_1}, \mathbf{r_2}, \mathbf{r_1}]$
$[0, 1, 0.5]$
$[\mathbf{r_1}, \mathbf{r_2}, \mathbf{r_2}]$
$[1, 0.5, 0]$
$[\mathbf{r_1}, \mathbf{r_1}, \mathbf{r_1}]$
$[0, 0, 0]$
$[\mathbf{r_1}, \mathbf{r_1}, \mathbf{r_2}]$
$[0.5, 0, 1]$

Figure 3.2: Cube of utility of players. On each corner, the top vector represents the network action **a** and, the bottom vector is the utility computed, **u**, by the CH. The red arrows represent the best response cycle. There are two resources.

$\mathbf{m} = (m_1, \ldots, m_K)$ is the mood vector of the network. In the TEL, there are four moods called Content (C), Watchful (W), Hopeful (H) and Discontent (D), whereas the ODL controller is solely composed of the two moods C and D. Furthermore, each player has a benchmark action and a benchmark utility denoted respectively by $\bar{a}_k \in \mathcal{N}$ and $\bar{u}_k \in \mathcal{U}$ that are kept in memory. The benchmarks of the network are then denoted by $\bar{\mathbf{a}} = (\bar{a}_1, \ldots, \bar{a}_K) \in \mathcal{N}^K$ and $\bar{\mathbf{u}} = (\bar{u}_1, \ldots, \bar{u}_K) \in \mathcal{U}^K$.

Definition 3.1 (State of the network)**.** *The different states taken by the network that uses any of the two algorithms are defined by* $\mathbf{z} = (\mathbf{m}, \mathbf{a}, \bar{\mathbf{a}}, \mathbf{u}, \bar{\mathbf{u}})$ *where* $\mathbf{m} = (m_1, \ldots, m_K)$ *is the mood vector of the network,* $\mathbf{a} = (a_1, \ldots, a_K)$ *is the network vector of played actions,* $\bar{\mathbf{a}} = (\bar{a}_1, \ldots, \bar{a}_K) \in \mathcal{N}^K$ *is the benchmark action vector of all players,* $\mathbf{u} = (u_1, \ldots, u_K)$ *is the network vector of received utilities, and* $\bar{\mathbf{u}} = (\bar{u}_1, \ldots, \bar{u}_K) \in \mathcal{U}^K$ *is the benchmark utility vector.*

At each iteration, every player either selects to use the benchmark action (*i.e.* $a_k = \bar{a}_k$) or decides to try a new one $a_k \neq \bar{a}_k$. Then, the player observes the obtained utility u_k and compares it to its benchmark utility $\bar{u}_k$. Depending on the result of this comparison, the player, with some probability, updates its benchmark and change its mood or not. The benchmark update consists in replacing the current benchmark (*i.e.* $(\bar{a}_k, \bar{u}_k)$) by the current action and received utility (*i.e.* (a_k, u_k)). Detailed descriptions of both algorithms, including the rules used to define/update the benchmark actions and utilities, are provided in the next sections.

3.3.1 TEL

This section describes the rules applied in the TEL controller presented in [10] of any $k \in \mathcal{K}$ and for any $\epsilon \in]0, 1]$:

- $m_k = C$, there are two cases to consider :

  1) with probability $1-\epsilon$, the player keeps playing its benchmark (*i.e.* $a_k = \bar{a}_k$). The next state changes to H if $u_k > \bar{u}_k$ or, it changes to W if $u_k < \bar{u}_k$ or, it remains C if $u_k = \bar{u}_k$.

  2) with probability ϵ, the player experiments a new action, *i.e.* $a_k \in \mathcal{N} \backslash \{\bar{a}_k\}$. The action experimented is selected randomly among $\mathcal{N} \backslash \{\bar{a}_k\}$ (*i.e.* $\Pr\{a_k = r_i\} = \frac{1}{N-1}, \forall r_i \neq \bar{a}_k$) and, the next state remains $m_k = C$. When $u_k > \bar{u}_k$, player k updates its benchmark with probability $\epsilon^{G(u_k - \bar{u}_k)}$, where $G(x) = -\nu_1 x + \nu_2$, with $\nu_1 > 0$ and ν_2 such that $0 < G(u_k - \bar{u}_k) < 1/2$. An update consists in changing the benchmark by the played action and the received utility in the next iteration as follows, $\bar{u}_k \leftarrow u_k$ and $\bar{a}_k \leftarrow a_k$.

- $m_k = H$: $a_k = \bar{a}_k$ and the next state changes to C with a utility benchmark update (*i.e.* $\bar{u}_k \leftarrow u_k$) if $u_k > \bar{u}_k$ or, it changes to W if $u_k < \bar{u}_k$ or, it changes to C if $u_k = \bar{u}_k$.

- $m_k = W$: $a_k = \bar{a}_k$ and the next state changes to H if $u_k > \bar{u}_k$ or, it changes to D if $u_k < \bar{u}_k$ or, it changes to C if $u_k = \bar{u}_k$.

- $m_k = D$: an action a_k is randomly selected among $\mathcal{N}$ (*i.e.* $\Pr\{a_k = r_i\} = \frac{1}{N}, \forall r_i \in \mathcal{N}$) with probability 1. The next state m_k changes to C with probability $\epsilon^{F(u_k)}$, where $F(u) = -\phi_1 u + \phi_2$ with, $\phi_1 > 0$ and ϕ_2 such that $0 < F(u) < 1/2K$, with a benchmark update (*i.e.* $\bar{u}_k \leftarrow u_k$ and $\bar{a}_k \leftarrow a_k$), otherwise, with probability $1 - \epsilon^{F(u_k)}$, $m_k = D$.

3.3.2 ODL

This section described the rules applied in the ODL controller given in [12] of any player $k \in \mathcal{K}$ and for any $\epsilon \in]0, 1]$:

- $m_k = C$, there are two cases to consider :

  1) with probability $1 - \epsilon^c$, where $c > K$ is a real constant, $a_k = \bar{a}_k$. If $u_k \neq \bar{u}_k$ then the state m_k changes to D with probability $1 - \epsilon^{1-u_k}$. Otherwise, with probability ϵ^{1-u_k}, the cluster updates its benchmark (*i.e.* $\bar{u}_k \leftarrow u_k$) and remains C.

  2) with probability $\epsilon^c > 0$, a new action is experimented, $a_k \in \mathcal{N} \backslash \{\bar{a}_k\}$. The new action is selected randomly in the set $\mathcal{N} \backslash \{\bar{a}_k\}$. If $u_k \neq \bar{u}_k$, the state m_k changes to D with probability $1 - \epsilon^{1-u_k}$. Otherwise, with probability ϵ^{1-u_k}, the cluster updates its benchmark (*i.e.* $\bar{u}_k \leftarrow u_k$ and $\bar{a}_k \leftarrow a_k$) and remains in C.

- $m_k = D$: an action a_k is randomly chosen among $\mathcal{N}$. The cluster switches to C with probability ϵ^{1-u_k} and updates its benchmark (*i.e.* $\bar{u}_k \leftarrow u_k$ and $\bar{a}_k \leftarrow a_k$), otherwise with probability $1 - \epsilon^{1-u_k}$, it remains D.

3.4 Results

In a deterministic context (static nodes, no time varying channel), the resource allocation problem can be efficiently handled by TE based methods. Assuming interdependency (see Definition 2.4), these methods have an interesting property of spending a high amount of time in stable states that maximizes a specific function related to the network performance. Although the interdependence property may appear restrictive, it is actually met in a broad class of games and makes the TEL and the ODL algorithms very appealing. This section presents the theoretical and numerical results obtained by using the TE algorithms presented in Section 3.3.

3.4.1 Theoretical results

The next results require notions that we present briefly for clarity. The set C^0 represent a set in which all players are content (*i.e.* $\forall k \in \mathcal{K}$, $m_k = C$) and aligned. A cluster is said to be aligned if $u = \bar{u}$ and $a = \bar{a}$. In addition, we use the definition of SSS (see Definition 2.13 in Chapter 2). The stationary probability of states that are not SSSs vanishes when the perturbation ϵ goes to zero. Therefore, over the long run, the SSSs are observed more frequently than other states provided that ϵ is small enough. The theoretical result obtained with the TEL algorithm is presenting in the next theorem.

Theorem 3.1 ([10]). *If the K persons game $\mathcal{G}$ is interdependent with deterministic utilities and all players use the TEL,*

(i) if the set of PNE is not empty, every PNE that maximizes the social welfare is an SSS,

(ii) otherwise, every state $\mathbf{z}^* \in C^0$ that maximizes a trade-off between the social welfare and the instability as follows is an SSS

$$\phi_1 W(\mathbf{z}^*) - \nu_1 S_0(\mathbf{z}^*). \tag{3.4}$$

where the instability $S_x(\mathbf{z}) := \max_{k\in\mathcal{K}} \max_{a'_k \in C\backslash a_k}\{0, u_k(a'_k, \mathbf{a_{-k}}) - \bar{u}_k \geq x\}$ represents the tendency of the system to leave $\mathbf{z}$. In this state, at least one player has the possibility to increase its utility benchmark. The greater this increase, the greater the probability to accept the new action. If a PNE exists, and the network is in this state, no one can improve its utility benchmark by an action change because $S_0(\mathbf{z}) = 0$. The network spends a high amount of time in this state. If PNE does not exist, then $S_0(\mathbf{z}) > 0$ and the algorithm spends a high amount of time in a stable state that maximizes a trade-off between the social welfare and an instability function. The theoretical result of the ODL algorithm is described by the following theorem.

Theorem 3.2 ([12]). *If the K persons game $\mathcal{G}$ is interdependent with deterministic utilities, every state that maximizes the social welfare is a SSS.*

No matter if a PNE exists or not, the ODL algorithm selects a state that maximizes the social welfare, which is a general result.

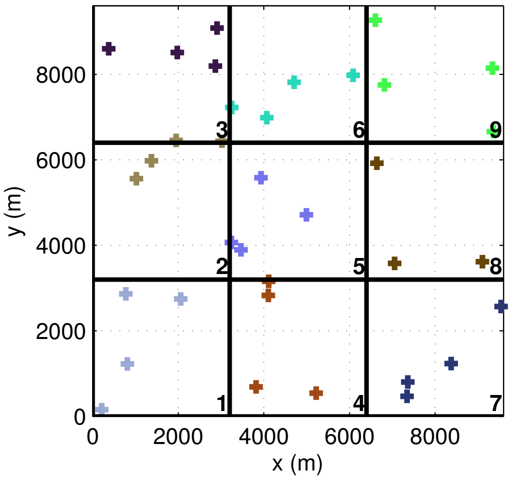


Figure 3.3: Wireless ad hoc clustered network with 9 clusters composed of 4 nodes each. Crosses represent mobile terminals, and colors the affiliations of terminals to cluster identified by the number in the bottom right of each square.

Table 3.1: Simulation parameters values

N	9	k_b	$1.38\ 10^{-23}\ WK^{-1}s$	Γ_0	4.3 dB
P_{Tx}	1 W	T	238 K	n_{it}	15000
B	250 KHz	P_b	$7.9\ 10^{-15}$ W		

3.4.2 Numerical results

In this section, numerical results are presented using the wireless model described in Section 3.1. The goal is twofold. First it highlights the behavior of both algorithms in a deterministic environment and the link with the analytical performance predicted in Section 3.4.1. Secondly, it provides a reference of their behavior in an allocation problem that is deterministic. These results will serve as a comparison in Chapter 4 where stochastic perturbation of the utility is considered. The ad hoc clustered network used in simulation is given in Figure 3.3. Parameters are listed in Table 3.1. The parameter Γ_0 is set such that the Bit Error Rate (BER) is 10^{-2}. Unless specified, the experimentation probability in TEL is set to $\epsilon = 10^{-3}$ whereas for ODL, one need a lower value to get similar performance $\epsilon^{\nu} = 10^{-4}$ (*i.e.* $\epsilon = 0.36$).

Figures 3.4 present the average social welfare of TEL and ODL in the deterministic context described in Section 3.1. The black curve is the average social welfare (see Definition 2.3 in page 12) noted W_m. It is the normalized sum of all clusters' utilities. The red curve is the average benchmark social welfare noted $\overline{W}_m$ and defined afterwards.

Definition 3.2 (Benchmark social welfare)**.** *The benchmark social welfare is the sum of all players' benchmark and it is given, for any* $\mathbf{z} \in \Xi$*, by*

$$\overline{W}(\mathbf{z}) = \sum_{i \in \mathcal{K}} \bar{u}_i. \tag{3.5}$$

The average benchmark social welfare is

$$\overline{W}_m(\mathbf{z}) = \frac{\overline{W}(\mathbf{z})}{K}. \tag{3.6}$$

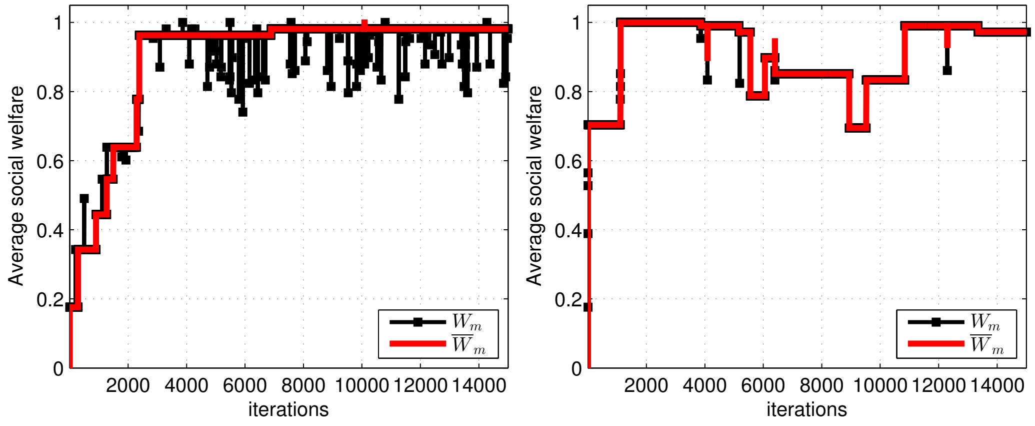
The average benchmark social welfare is hence the normalized sum of all clusters' benchmark utilities.

First, one can see that the network tends to maximizes the average social welfare for both algorithms and to keep it at a high value. The fact that the network does not remain in a maximum social welfare state in Figure 3.4b is expected as the Theorems 3.2 and 3.1 in Section 3.4.1 specify that this state is an SSS. So the network spends a high amount of time in this state. As one will see in Section 3.5, this amount of time is related to ϵ. Note that as long as $\epsilon > 0$, the network has a positive probability to leave each state.

The red curve variations in both figures are expected because the perturbation ϵ is kept constant along the simulation. The clusters keep experimenting and consequently, the state of the system changes regularly. Notice that the black curve variations in Figure 3.4a happen more often than in Figure 3.4b. This is also an expected results since, when the current state is content, the probability to experiment a new action in TEL is ten times higher than the one in ODL.

When the red curve is constant, it means that no one in the network changes its benchmark and the network is relatively stable. Even if the black curve is subject to variations due to experimentations, the red curve tends to remain stable. The clusters keep their benchmarks constant during ephemeral network modifications. This behavior is important and it is made possible thanks to the FSC of each algorithm presented in Section 3.3. They are constructed such that they absorb these short network changes due to explorations of other clusters. Without this property, the network would adapt to any perceived change which would cause a lot of instability and hence interference.

As an other observation, the ODL reaches faster a quite good social welfare than the TEL but, it oscillates more often around this value than the TEL. This highlights the stability capacity of the TEL whereas its probability to experiment ($\epsilon = 10^{-3}$) is much more greater than the one in ODL ($\epsilon^c = 10^{-4}$).



(a) TEL (b) ODL

Figure 3.4: Resource allocation in a deterministic environment with TEL and ODL algorithms.

More generally, as one will see in Section 3.5, when ϵ is large (*e.g.* $\epsilon > 0.1$) the network goes fast to good social welfare but is less stable. Conversely, if ϵ is small (*e.g.* $\epsilon < 10^{-4}$), the network is very stable and converges slowly to high social welfare values. To tackle the intricate convergence speed/stability problem, a solution consists in decreasing ϵ. In [41] authors prove that the convergence is guaranteed when the annealing of ϵ is of order $1/\sqrt{t}$ which is very long. The slow convergence speed is a general result when dealing with algorithms that look for global optimum states such as simulated annealing algorithms. It is worth mentioning that [31] proposes an enhancement of TEL in which ϵ is decreased very fast if the utility is high enough. In this case however, the convergence of algorithms cannot be guaranteed. This approach is discussed in Chapter 4. For this moment let us analyze more precisely the performance of both algorithms.

3.5 Performance analysis

This section presents one of the main contribution of this thesis [42]. We study the performance of both TEL and ODL algorithms for a specific interference model. We have seen in the previous sections the capacity of these algorithms to exhibit good cooperative performance for a broad class of games. Even though the above two algorithms converge to a desired state, their convergence rate remains an open question [10, 11]. The main reason comes from the computation complexity of the inherent MC generated by these two algorithms. In fact, the game in which players employ these learning schemes can be represented by discrete MCs based on states $\mathbf{z}$ (see Section 3.3) whose number is huge. Obtaining the transitions matrix of these MCs is therefore not tractable which makes the analysis of the convergence rate not possible (even numerically).

The main contributions of this section are fourfold. We are interested in computing the mean time these algorithms spend in a desired state as well as the mean time required to achieve that state under a given model. Due to the huge dimension of the MCs, only approximations can be employed to compute a close approximation of the aforementioned convergence metrics. The first contribution is to provide an approximation of the MC associated to the TEL algorithm. The second contribution is to also provide such an approximation for the ODL algorithm. In addition, we explain the methodology to obtain them. To the best of our knowledge, a first attempt to analyze the convergence rate of TEL in a practical context was addressed in [31]. However, the analysis proceeded in this thesis provides a better approximation (as one will see in the sequel). In addition, no attempt has been made to analyze the convergence properties of ODL. Third, with the numerical results, we study the convergence properties of each algorithm. Last, this allows us to provide a comparison between these two algorithms. To the best of our knowledge, this comparison has not been addressed under a practical system model before.

This section is organized as follows. Section 3.5.1 presents the system model and assumptions. Section 3.5.2 presents the MC representation of both algorithms and the figure of merit to be computed. Section 3.5.3 summarizes the main results of this work. The detailed analysis of the convergence (*i.e.* mean convergence time to a desired state and mean time spent in that state), including the reduction of the MCs, is provided in Sections 3.5.4, 3.5.5 and 3.5.6. Section 3.5.7 highlights the complexity reduction induced by our method. Section 3.5.8 presents the algorithm necessary to compute the transition matrix of both reduced MCs. Numerical results are provided in Section 3.5.9.

3.5.1 Simplified model

We consider as in Section 3.1, a network/set of K players $\mathcal{K}$, that interact among each other. The context is quickly recalled for clarity. The players share a set of resources $\mathcal{N}$. Each player k chooses an action a_k, which consists in selecting without exchanging any information with the other players a resource inside the set $\mathcal{N}$. When two players choose the same resource they interfere with each other. This problem can be modeled as the normal form game $\mathcal{G}$ with appropriate utility function. Since we consider a general game model, we make in the following some assumptions in order to ensure the existence of a PNE. We suppose that the number of available resources N is greater or equal to the number of players K (see Section 3.2). We assume that the utility can take binary values (*i.e.* $\mathbf{u} \in \{0,1\}^K$). Note that, such a hard threshold utility model is commonly encountered in the literature [33, 39, 43]. Furthermore, we assume that if two players interfere with each other (*i.e.* choose the same resource) then their utilities are equal to 0. The utility of a player is then equal to 1 when no other player chooses the same resource. These simplified assumptions can be first justified by the fact that our objective in this section is to study the performance of TEL and ODL algorithms and not to study the existence of PNE for some game models. Then, it is worth mentioning that even under the above assumptions the problem is still challenging. Firstly, players cannot communicate with each other and then cannot be aware of the others' actions and they can only observe the result of their own actions (*e.g.* a player cannot know how many players have chosen the same resource). Secondly, the resulting Markov chain, as one will see in the sequel, remains very complex to analyze under this model.

3.5.2 Markov Chain representation and performance metrics

The different states taken by the network are defined by $\mathbf{z} = (\mathbf{m}, \mathbf{a}, \bar{\mathbf{a}}, \mathbf{u}, \bar{\mathbf{u}})$, where $\mathbf{m} = (m_1, \ldots, m_K)$, $\mathbf{a} = (a_1, \ldots, a_K)$, $\bar{\mathbf{a}} = (\bar{a}_1, \ldots, \bar{a}_K)$, $\mathbf{u} = (u_1, \ldots, u_K)$, and $\bar{\mathbf{u}} = (\bar{u}_1, \ldots, \bar{u}_K)$ are $(1 \times K)$ vectors representing the moods, the actions, the benchmark actions, the utilities, and the benchmark utilities respectively (see Sections 3.1 and 3.3). These states represent a MC noted Ξ_{TEL} if the TEL is used by all players or Ξ_{ODL} if it is ODL. Unless there is an ambiguity, we drop the indices and call the MC Ξ in the sequel.

The convergence performance is evaluated along two features: *i*) the mean time duration to reach the SSS maximizing the social welfare, starting from a specific initialization point, also known as Expected First Hitting Time (EFHT) denoted by T_{EFHT}, *ii*) the mean fraction of time duration spent on that state denoted by α.

It is of interest to note that these algorithms are known to converge under the *interdependence* property (see Definition 2.4). In few words, the *interdependence* is the property that for any set of players, there exists an action that changes the utility of a player not in the set. This condition is a sufficient condition as the analysis in [10, 12] was done for more general game model than the one considered in this thesis. In our case, the above condition is not needed. In fact, thanks to the presence of the probability ϵ in TEL and ODL (see Section 3.3), all states of the MC Ξ communicate and form a unique communication class. The MC Ξ is then ergodic (*i.e.* it is a finite, irreducible and aperiodic MC) and possesses a unique invariant distribution. This property ensures a non null transition probability between all states for a sufficient number of transitions and a non null probability of the corresponding state. In the next section, we present the main results of this work and the figure of merit tendency that can be deduced from numerical results presented in Section 3.5.9.

3.5.3 Performance observations

The main result of this work is to provide an efficient approximation of the MC for TEL and ODL algorithms that allows an accurate numerical convergence analysis. The approximated MC is denoted by $\widetilde{\Xi}$. In next sections, we describe the procedure to approximate and reduce the MC dimensionality so as to realize the convergence analysis. It is worth mentioning that the number of states in the original MC is huge, which makes very hard the computation (even numerically) of the performance metrics T_{EFHT} and α for both algorithms.

Using the proposed efficient approximation, we were able to find interesting results (that are presented in Section 3.5.9). Based on the obtained results, the following observations can be highlighted. We use the Landau notation $\mathcal{O}(.)$ to specify the rate of convergence when K becomes large or when ϵ is close to 0 and strictly positive. In this notation, K and ϵ are dropped for clarity.

Observation 1. *For the TEL, the EFHT* $T_{EFHT} = \mathcal{O}(\frac{1}{\epsilon^{a_1}})$ *and* $T_{EFHT} = \mathcal{O}(K^{a_2})$ *where* $a_1, a_2 > 0$ *and,* $1 - \alpha = \mathcal{O}(\epsilon^{a_3})$ *and* $1 - \alpha = \mathcal{O}(K^{a_4})$ *where* $a_3, a_4 > 0$.

Observation 2. *For the ODL,* $T_{EFHT} = \mathcal{O}(\frac{1}{\epsilon^{cb_1}})$ *and* $T_{EFHT} = \mathcal{O}(b_2^K)$ *where* $b_1 > 0$ *,*$b_2 > 1$ *and, the stability is* $1 - \alpha = \mathcal{O}(\epsilon^{b_3})$ *and* $\alpha = \mathcal{O}(b_4^K)$*, where* $b_3 > 0$, $1 > b_4 > 0$.

From these observations some interesting comparisons can be done. Both algorithms have a convergence time inversely proportional to ϵ and a stability that decreases polynomially with ϵ. However, ODL has a convergence time which is exponential with respect to K contrarily to TEL which is polynomial. At low K, the convergence time of ODL is relatively similar to the TEL one, but at higher K, TEL converges faster than ODL. In addition, for ODL, the stability decreases exponentially with respect to K whereas, for the TEL, the stability decreases polynomially. It follows that the TEL is much more stable than the ODL. At low number of players, the convergence time of both algorithms are similar but the stability of TEL is better. At higher number of players, the TEL performs better than ODL for both convergence metrics. These observations result from the analysis of numerical figures of merit computed using the formulas presented in the next section.

3.5.4 Metrics computation

In this section, we present how to compute the figures of merit of both algorithms using the transition matrix $\mathbf{P}_0$ of Ξ. The method is based on the generalized fundamental matrix $\mathbf{F}$ for ergodic MC developed in [44]. The matrix $\mathbf{F}$, which is an extension of the fundamental matrix introduced in [45], is defined by

$$\mathbf{F} := (\mathbf{I} - \mathbf{P}_0 + \mathbf{1}\mathbf{b}^t)^{-1}, \tag{3.7}$$

where $\mathbf{I}$ is the identity matrix, $\mathbf{1}$ is a column vector filled with 1, and $\mathbf{b}$ is any arbitrary column vector such that $\mathbf{b}^t\mathbf{1} \neq 0$. In the simulations, we use $\mathbf{b} = \mathbf{1}$.

The first feature deals with the EFHT to a given state j from a state i, $T_{EFHT}(i, j) = \mathbb{E}_i\left[T_j\right]$, and is given by ([44] equation (30))

$$\mathbb{E}_i\left[T_j\right] = \frac{\mathbf{F}_{jj} - \mathbf{F}_{ij}}{\pi_j}, \tag{3.8}$$

where $\mathbf{F}_{ij}$ is the term in line i and column j of matrix $\mathbf{F}$, and π_j is the stationary probability of state j. The stationary distribution is given by ([44] equation (28))

$$\mathbf{b}^t\mathbf{F} = \pi. \tag{3.9}$$

The second feature that describes the performance of stochastic stable algorithms is the mean fraction of time spent in the state that maximizes the social welfare. In an ergodic MC, the proportion of time α_j spent in a state j is equal to its stationary probability $\alpha_j = \pi_j$ ([45] Theorem 4.2.1) that can be computed using (3.9).

The convergence analysis realization requires the manipulation of transitions matrices. The huge number of states $\mathcal{S}$ grows exponentially with K and N (see Section 3.5.7) and since Ξ's transition matrix has dimension $(\mathcal{S} \times \mathcal{S})$, it needs to be approximated to allow numerical computation of the performance. As an example, even for small values $N = K = 3$, the number of states is already $\mathcal{S} = 373,248$. In this work, we propose a new approach to build the approximated MC $\widetilde{\Xi}$ whose transition matrix is noted $\mathbf{P}$. Note that the approximation $\widetilde{\Xi}$ is built such that it is ergodic as Ξ which means that, $\mathbf{P}$ admits a unique invariant distribution with strictly positive components. With this approximation, formulas (3.8) and (3.9) are still valid if $\mathbf{P}_0$ is replaced by $\mathbf{P}$. It remains to construct $\mathbf{P}$ with justified and motivated arguments. This approach follows two steps: *i*) first we approximate the original Markov chain Ξ by identifying some invariance induced by the utility model defined in Section 3.5.1, *ii*) we then further reduce the MC complexity by neglecting some transitions. Then, from the probability transition matrix of $\widetilde{\Xi}$, we are able to compute the two convergence figures of merit.

3.5.5 Reducing the Markov chain dimensionality

In order to approximate Ξ, we start by considering only the states called *recurrence classes of the unperturbed process* [38], shorten as Recurrence Classes (RC) (see Definition 2.9 of recurrence for details), that were used as the key feature for the TEL and ODL proof of convergence. The system tends to spend naturally a high amount of time in those states which thus play a major role in convergence metrics. The reason comes from the combination of two properties. First, the network needs at least one experimentation to leave an RC, which occurs with small probability ϵ. Secondly, by definition, the network always naturally goes to an RC when no perturbation occurs. These states are characterized by $\mathbf{m} = \mathbf{m}_C := (C, C, \ldots, C)$, $\mathbf{a} = \bar{\mathbf{a}}$, and $\mathbf{u} = \bar{\mathbf{u}}$, *i.e.* , all the players are in the content mood and aligned (*i.e.* $\mathbf{a} = \bar{\mathbf{a}}$, and $\mathbf{u} = \bar{\mathbf{u}}$). We denote by $\mathcal{R}$ the set of these states. We can also drop some notations, and we rewrite a state $\mathbf{z} = (\mathbf{m}_C, \mathbf{a}, \bar{\mathbf{a}}, \mathbf{u}, \bar{\mathbf{u}}) \in \mathcal{R}$ as $\mathbf{z} = (\bar{\mathbf{a}}, \bar{\mathbf{u}})$

To reduce the number of RCs, two invariances induced by the utility model are highlighted. First of all, due to the binary utility values and the utility rules, interchanging actions between players is equivalent to interchange the utility vector components accordingly, thus not modifying the number of 0 and 1 in the utility vector. As such, we can deduce that it does not change the "global" performance of the network. For instance, let consider a network with three players and three resources $\mathbf{z}_1 = ((r_1, r_2, r_1), (0, 1, 0)) \in \mathcal{R}$. Players 1 and 3 have a utility equal to 0 because they use the same resource. If we interchange the actions of player 1 and 2, $\mathbf{z}_1$ is transformed into $\mathbf{z}_2 = ((r_2, r_1, r_1), (1, 0, 0))$. There is always one player with a utility equal to 1 and two players with a utility equal to 0. Nothing has changed from a network perspective, thus, the algorithm performance remains the same from these two states.

Secondly, notice also that interchanging the resource labels does not change at all the utility vector (this is also true for geographical models when orthogonality between resources is assumed). For instance, if we change the resource label 1 with label 3 and, label 2 with label 1, $\mathbf{z}_1$ becomes $\mathbf{z}_3 = ((r_3, r_1, r_3), (0, 1, 0))$ which involves the same observations as the previous modification.

These two invariances have led us to represent any RC with the ordered repartition of players over resources. For any action vector $\bar{\mathbf{a}}$, we build the repartition vector of players over resources $\mathbf{d} := (d_1, d_2, \cdots, d_N)$ where $d_i = \sum_{k=1}^{K} \mathbb{1}_{\{a_k = r_i\}}$ is the number of players that use a resource $r_i \in \mathcal{N}$. For instance, the repartition of player in $\mathbf{z}_1$ is $\mathbf{d}_1 = (2, 1, 0)$ and the repartition vector of $\mathbf{z}_3$ is $\mathbf{d}_3 = (1, 0, 2)$. The ordered repartition vector is $\mathbf{s} = (s_1, \ldots, s_N)$ where $\forall i, j \in [1, N]$, $i < j$, $\exists i', j' \in [1, N]$, $s_i = d_{i'} \geq s_j = d_{j'}$. For instance, the ordered repartition vector of $\mathbf{z}_1$ is $\mathbf{s}_1 = (2, 1, 0)$ and, the ordered repartition of $\mathbf{z}_3$ is $\mathbf{s}_3 = (2, 1, 0)$ which is equal to $\mathbf{s}_1$. Thus, it follows that this representation makes no difference between RC that are invariant with respect to the transformations mentioned. Hence, it is possible to reduce the number of RC in Ξ. Moreover, the utility vector repartition is directly specified by the ordered repartition of players, then we drop this notation and $\mathbf{z} = (\bar{\mathbf{a}}, \bar{\mathbf{u}})$ becomes $\mathbf{z} = \mathbf{s}$.

In what follows, for ease of comprehension we slightly modify RC notations. For each $\mathbf{z} \in \mathcal{R}$ the number of resources employed is noted $n = \sum_{\ell=1}^{N} \mathbb{1}_{\{s_\ell > 0\}}$. In addition, for each $n \in [1, N]$ there exists different possible ordered repartitions of players whose number is noted $\mathrm{I}_N(n)$. It is equal to the number of ways to partition integer N in n parts, *i.e.* $\mathrm{I}_N(n) = \mathrm{Part}(N, n)$ where the recursive formula gives $\mathrm{Part}(N, n) = \mathrm{Part}(N-1, n-1) + \mathrm{Part}(N-n, n)$, and for any integers x, y, $\mathrm{Part}(x, x) = 1$, $\mathrm{Part}(x < y, y) = 0$ and $\mathrm{Part}(x, 1) = 1$ ([46] Chapter 2, Section 2.1, Theorem B). Thus, any $\mathbf{z} \in \mathcal{R}$ can be noted $\mathbf{Z}_n(i)$ where n is the number of resources used and $i \in [1, \mathrm{I}_N(n)]$ is the index of the ordered repartition and, the associated ordered repartition vector is $\mathbf{S}_n(i) = (S^i_{n,1}, S^i_{n,2}, \ldots, S^i_{n,N})$. For instance, in a network with $N = K = 4$, when $n = 2$ there are two possible ordered repartitions $\mathbf{S}_2(1) = (3, 1, 0, 0)$ and $\mathbf{S}_2(2) = (2, 2, 0, 0)$. However, for $n = 3$ there is a unique repartition $\mathbf{S}_3(1) = (2, 1, 1, 0)$. The mapping between indices i and the ordered repartitions is arbitrary and has to be made by the experimenter. The reduced states $\mathbf{Z}_n(i)$ for all $n \in [1, N]$ and for all $i \in [1, I_N(n)]$ are called Reduced Recurrence Classes (RRC).

Notice that the social welfare of RC represented by the same RRC are equal, but we can find different RRC for which their elements have the same social welfare.

3.5.6 Approximated Markov chain

In this section, we build an approximation $\widetilde{\Xi}$ of Ξ that is composed of the RRC and a subset of intermediary states between RRC. More specifically, the construction of the intermediary states considered in each approximations (TEL and ODL) is detailed. These constructions are driven by, *i)* the willingness to conserve the ergodic property of Ξ in order to be able to approach its convergence performance, *ii)* the need to construct a MC with low dimension (*i.e.* with the least number of states). A condition to make property *i)* realizable, consists in constructing intermediary states around each RRC such that, all states of $\widetilde{\Xi}$ (*i.e.* RRC and intermediary states) are accessible from one another. We note $\xi^n(i)$ the set that contains the RRC $\mathbf{Z}_n(i)$ and some associated intermediary states that we define later. The simplest, thus verifying *ii)*, and necessary way to conserve the ergodicity property is to construct intermediary states such that, if the transition between sets $\xi^n(i) \rightsquigarrow \xi^{n+1}(j)$ exists then there also exists a transition from sets $\xi^{n+1}(j) \rightsquigarrow \xi^n(i)$ (the symbol $\rightsquigarrow$ specifies that this transition can involve multiple states in $\widetilde{\Xi}$). This is the consequence of the fact that from every RRC $\mathbf{Z}_n(i)$ where $n < N$, there are players interfered and, it is possible for one of them to find a free resource (*e.g.* an interfered player experiments on a free resource). Repeating this process successively shows that, all RRCs can access $\mathbf{Z}_N(1)$, which is the RRC without interference. Therefore, the condition, if $\xi^n(i) \rightsquigarrow \xi^{n+1}(j)$ exists, then, so does $\xi^{n+1}(j) \rightsquigarrow \xi^n(i)$, implies that all sets communicate. Finally, the previous condition becomes sufficient, if the sets are constructed such that all states in all sets are accessible. In addition to these simplifications, we consider the following hypothesis to build $\widetilde{\Xi}$ completely.

Assumption 3.1. *For each algorithm models, we assume at each iteration of the algorithm that at most one content player can experiment, and such, solely when the system is in an all content mood and aligned state, i.e.* $\mathbf{m} = \mathbf{m}_C$, $\mathbf{u} = \bar{\mathbf{u}}$ *and,* $\mathbf{a} = \bar{\mathbf{a}}$.

The reason to propose this assumption is summarized as follows. When all players are content, the probability that one player experiments (*i.e.* $0 < \epsilon \ll 1$) is larger than the probability that two or more player experiment (*i.e.* $0 < \epsilon^2 \ll \epsilon \ll 1$). Moreover, when the system is not aligned, it goes in less than two steps and with a high probability ($\approx (1-\epsilon)^2$) to a state in which all players are content and aligned or, that contains a discontent player. Thus, most of the time, the system is either in *a)* an all content and aligned state or, *b)* it contains at least one discontent player. In case *a)*, it is most probable that only one player experiments whereas, in case *b)*, the probability that a discontent player experiments is 1 which is much larger than the probability for a content player to experiment ($\epsilon \ll 1$).

Hypothesis 1. *For the TEL model, the probability that a discontent player accepts a new utility u as a benchmark is* $1 \geq \epsilon^{F(u)} \geq \epsilon^{\frac{1}{2K}}$. *We suppose that* $\epsilon^{F(0)} = \epsilon^{\frac{1}{2K}}$ *and that,* $\epsilon^{F(1)} = \epsilon^0 = 1$.

In other words, we suppose that the constants ϕ_1 and ϕ_2 presented in Section 3.3 have been chosen such that $F(.)$ spans the whole available region.

In the next two sections, we present the constructions of sets $\xi^n(i)$ of each algorithm. We start the reasoning by considering all sets $\xi^n(i) = \{\mathbf{Z}_n(i)\}$. Then we add successively intermediary states in all sets to build the approximated MC. When a state is added to $\xi^n(i)$ it is also added to any other set $\xi^{n'}(i')$ where $i \neq i'$ and $n \neq n'$. Figures 3.5a and 3.5b present, for ease of space and comprehension, a resulting partial view of $\widetilde{\Xi}$'s intermediary models with two sets $\xi^n(i)$ and $\xi^{n+1}(j)$, for TEL and ODL respectively. The lines define the oriented connections between states. Plain lines correspond to direct transition inside the same set $\xi^n(i)$ whereas dashed lines correspond to direct transitions between different sets. The connexions are detailed in Appendices A and B for the TEL and ODL respectively. In these figures, without loss of generality, it is supposed that, there exists j such that $\xi^n(i)$ is connected to $\xi^{n+1}(j)$. In such a case, we would also like to have $\xi^{n+1}(j)$ connected to $\xi^n(i)$ for ergodicity. We also suppose that, all intermediary states are present for simplicity of comprehension, whereas as explained in the next two sections, there exists some conditions in which they have to be deleted from their corresponding set to keep $\widetilde{\Xi}$ ergodic.

3.5.6.1 TEL model

This section presents the construction of the intermediary states in the approximated MC based on the TEL algorithm described in Section 3.3. Given any RRC $\mathbf{Z}_n(i)$, a transition where a player interfered finds a free resource, *e.g.* $\mathbf{Z}_n(i) \rightsquigarrow \mathbf{Z}_{n+1}(j)$, does not necessitate additional intermediary state unless one player is left alone on its resource after the experimentation. In this situation, the left alone player sees its utility increases and becomes hopeful. Therefore, we start by considering in $\xi^n(i)$ the state $\xi^n_0(i)$ in addition to $\mathbf{Z}_n(i)$ where

- $\xi^n_0(i)$ corresponds to a player alone in $\mathbf{Z}_n(i)$ that is hopeful.

Thus, at this step, $\forall n, i$, $\xi^n(i) = \{\mathbf{Z}_n(i), \xi^n_0(i)\}$.

A transition in which the network uses one less frequency, *e.g.* $\xi^{n+1}(j) \rightsquigarrow \xi^n(i)$, involves a player that accepts a lower benchmark, which is only possible through a discontent mood. To become discontent, a player passes through a watchful mood. This leads us to consider the two intermediary states $\xi^n_1(i)$ and $\xi^n_2(i)$ where

- $\xi^n_1(i)$ is the state where a player alone in $\mathbf{Z}_n(i)$ is watchful,
- $\xi^n_2(i)$ is the state where a player alone in $\mathbf{Z}_n(i)$ is discontent. It corresponds to the situation where the watchful player in $\xi^n_1(i)$ experiences one more iteration a decrease in utility.

Note that during the transition $\xi^n_1(i) \rightarrow \xi^n_2(i)$ (where → means that the transition is direct), the system is not aligned whereas, a content player experiments. It is not in accordance with Assumption 3.1 but, this is the **only time** that it is overridden in order to keep the MC ergodic. Finally, to avoid any absorbing state, two more intermediary states $\xi_3^n(i)$ and $\xi_4^n(i)$ are considered where

- $\xi_3^n(i)$ is a state where two players that were alone in $\mathbf{Z}_n(i)$ are using the same resource and one of them is watchful. It corresponds to the case where the discontent player from $\xi_2^n(i)$ has updated its benchmark with the resource of a player that was not interfered in $\mathbf{Z}_n(i)$.
- $\xi_4^n(i)$ is a state where two players that were alone in $\mathbf{Z}_n(i)$ are using the same resource and one of them is discontent. It corresponds to the state that follows $\xi_3^n(i)$ where the player watchful becomes discontent.

The base to construct our model for TEL is established with

$$\xi^n(i) = \{\mathbf{Z}_n(i), \xi_0^n(i), \xi_1^n(i), \xi_2^n(i), \xi_3^n(i), \xi_4^n(i)\}.$$

It is said in the introduction, that all intermediary states have to be accessible but, in some cases they are not all present. For instance, when every player in $\mathbf{Z}_n(i)$ is interfered, no one can become discontent and states $\xi_1^n(i)$, $\xi_2^n(i)$, $\xi_3^n(i)$ and $\xi_4^n(i)$ are not present. These absences have to be taken into account to compute the probabilities in Appendix A, and in the simulations in order to build an ergodic chain (an isolated state in a matrix makes the chain not ergodic). These cases are described as follows starting with any given $\xi^n(i) = \{\mathbf{Z}_n(i)\}$:

- If in $\mathbf{Z}_n(i)$ all players are interfered, only the state $\mathbf{Z}_n(i)$ is present
- If in $\mathbf{Z}_n(i)$ only one player is alone on its resource, this player can become discontent or hopeful, however, it cannot make an other player discontent. Therefore, include states $\xi_0^n(i), \xi_1^n(i)$ and $\xi_2^n(i)$ in $\xi^n(i)$. There is one exception, where the distribution $\mathbf{S}_n(i)$ is of the form $(2, \ldots, 2, 1, 0, \ldots, 0)$ and, the state $\xi_0^n(i)$ is removed from $\xi^n(i)$.
- If in $\mathbf{Z}_n(i)$ at least two players are alone on their respective resource, include $\xi_3^n(i)$ and $\xi_4^n(i)$ in $\xi^n(i)$.

The transitions between states and the associated probabilities are detailed in Appendix A.

3.5.6.2 ODL model

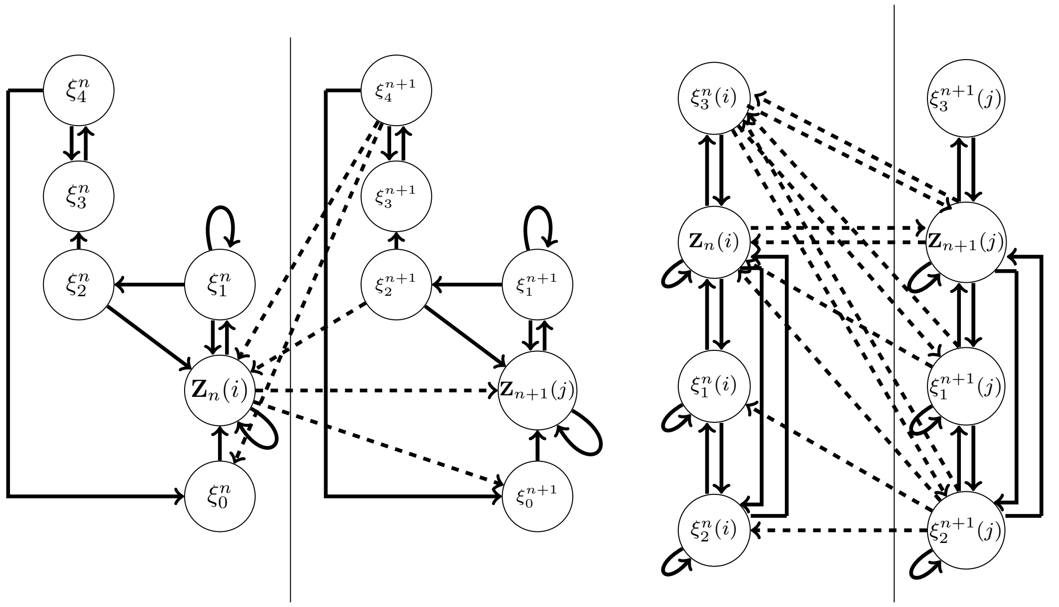
This section presents the construction of the intermediary states in the MC approximation based on the ODL algorithm described in Section 3.3. First of all, the model that contains only the $\xi^n(i) = \{\mathbf{Z}_n(i)\}$ is sufficient to have an ergodic chain $\widetilde{\Xi}$. The transition $\mathbf{Z}_n(i) \rightarrow \mathbf{Z}_n(i)$ occurs if nothing happens. The transition $\mathbf{Z}_n(i) \rightarrow \mathbf{Z}_{n+1}(j)$ represents an interfered player that experiments and finds a free resource. The reversed transition $\mathbf{Z}_{n+1}(i) \rightarrow \mathbf{Z}_n(j)$ occurs if one of the not interfered player in $\mathbf{Z}_{n+1}(j)$ goes back to the position of the experimenter from $\mathbf{Z}_n(i)$. The accuracy of the model can be increased by adding a few more states. The stability of ODL is directly related to the number of discontent players. Such players experiment randomly, which makes the number of possible transitions between states growing very fast with the number of discontent players. It prevents us from describing too many discontent players at the same iteration. In this work, we manage to model the case where at most two players can be discontent at the same time. Going beyond this would require for each additional discontent player a large amount of extra transitions for a small accuracy gain. This model requires three more states $\xi_1^n(i)$, $\xi_2^n(i)$ and $\xi_3^n(i)$ to be added with each RRC $\mathbf{Z}_n(i)$:

- $\xi_1^n(i)$ corresponds to the case where a player alone in $\mathbf{Z}_n(i)$ is discontent,
- $\xi_2^n(i)$ corresponds to the case where two players alone in $\mathbf{Z}_n(i)$ are discontent.
- $\xi_3^n(i)$ is a state where one of the two players that share the same resource in $\mathbf{Z}_n(i)$ is discontent.

As in the previous Section 3.5.6.1, there are some cases, depending on $\mathbf{Z}_n(i)$, where $\xi_1^n(i)$, $\xi_2^n(i)$ and $\xi_3^n(i)$ are not all present simultaneously in $\xi^n(i)$. They have to be removed accordingly to make the resulting MC ergodic. These cases are described as follows starting with $\xi^n(i) = \{\mathbf{Z}_n(i)\}$:

- If there exists a resource played by two players in $\mathbf{Z}_n(i)$, include the state $\xi_3^n(i)$ in the set $\xi^n(i)$.
- If at least one player in $\mathbf{Z}_n(i)$ is alone on its resource, include the state $\xi_1^n(i)$ in $\xi^n(i)$.
- If at least two players in $\mathbf{Z}_n(i)$ are alone on their respective resource, include $\xi_2^n(i)$ in $\xi^n(i)$.

The transitions between states and the associated probabilities are detailed in Appendix B.



(a) TEL (b) ODL

Figure 3.5: Partial view of $\widetilde{\Xi}$ for both algorithms.

3.5.7 Complexity comparison

We compute the Markov chain complexities to highlight the importance of the transformations from Ξ to $\widetilde{\Xi}$ made in this work. The simplifications and approximations are essential in order to be able to predict the algorithms performance. The number of states in Ξ is given by the product of components' dimensions in $\mathbf{z} = (\mathbf{m}, \mathbf{a}, \bar{\mathbf{a}}, \mathbf{u}, \bar{\mathbf{u}})$. The vector of player moods can have M^K values if the mood of each player can take M values. The vector of players' actions and action benchmarks $\mathbf{a}$ or $\bar{\mathbf{a}}$ can take N^K values each one. The utility vector $\mathbf{u}$ is specified by the action vector $\mathbf{a}$ and, the utility benchmark vector $\bar{\mathbf{u}}$ can take 2^K values. Therefore, the complexity of Ξ is $(MN^2 2)^K$. This is obviously intractable and, we have reduced the recurrence states $\mathcal{R}$ into $\mathbf{Z}$ which has a cardinality $|\mathbf{Z}| = \sum_{n=1}^{K} \text{Part}(K, n)$. Afterwards, we have approximated Ξ by keeping some intermediary states as detailed in Section 3.5.6. Figure 3.6 presents the complexity of Ξ and $\widetilde{\Xi}$ for the TEL algorithm with respect to the number of players. The significant complexity reduction allows us to predict performance numerically.

3.5.8 Procedure to compute the transition matrix

Once the states of $\widetilde{\Xi}$ are established, the next step consists in computing the transition probabilities of matrix $\mathbf{P}$. The procedure is described in Algorithm 1

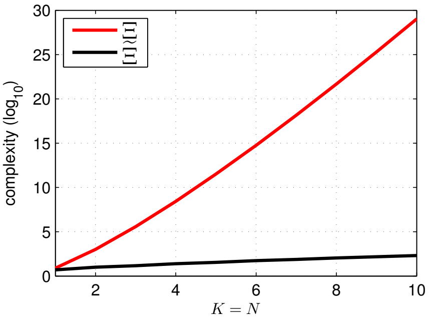


Figure 3.6: Complexity comparison between Ξ and $\widetilde{\Xi}$.

and summarized as follows. The first step necessitates to generate all RRCs. For this purpose, a classical integer partitioning algorithm is used to generate all ordered repartition vector $\mathbf{S}_n(i)$ [47]. The number of RRC using $n \in [1, \ldots, N]$ resources among N is given by $\mathrm{I}_N(n)$ (see Section 3.5.5). In both algorithms, at each RRC $\mathbf{Z}_n(i)$ is associated an intermediary state model $\xi^n(i)$, whose number of states depends on some exceptions specified in Sections 3.5.6.1 and 3.5.6.2 for the TEL and ODL respectively. One has to pay attention to these exceptions when it makes the one-to-one mapping function between the states of $\widetilde{\Xi}$ and the lines of $\mathbf{P}$. Then, the Algorithm 1 goes trough all $\mathbf{Z}_n(i)$ and looks for all $j \in I_N(n+1)$ such that $\mathbf{Z}_{n+1}(j)$ is accessible from $\mathbf{Z}_n(i)$. When $n < N$, there exists at least one such a j and, by construction the set $\xi^{n+1}(j)$ is connected to the set $\xi^n(i)$. The transition probabilities are computed in the algorithm through three consecutive steps. These steps and formulas are highlighted in the same order in Appendices A and B for the TEL and ODL respectively. On the first hand, the probabilities inside the set $\xi^n(i)$ are computed. On the second and third hand, for each j in $I_N(n+1)$ such that $\xi^n(i)$ is connected to $\xi^{n+1}(j)$, the algorithm computes, the probabilities from set $\xi^n(i)$ to set $\xi^{n+1}(j)$ and, the reverse probabilities from set $\xi^{n+1}(j)$ to set $\xi^n(i)$.

The example provided in Figure 3.7 with $K = N = 5$ highlights the links between the sets $\xi^n(i)$, identified by the vector $\mathbf{S}_n(i)$. For instance, the set in the top left corresponds to 5 players interfering on the same resource.

3.5.9 Numerical results

3.5.9.1 Accuracy of the proposed models

We assess the accuracy of our proposed models by comparing the values computed numerically with the proposed approximation $\widetilde{\Xi}$ ($T_{EFHT}(i, j)$ is computed with (3.8) and α_j is computed with (3.9)) with the values obtained through Monte Carlo simulations. We consider three different values for $K = \{3, 5, 7\}$ in two cases

Algorithm 1 Computing the matrix **P**

```
Input: $\mathbf{Z}$; $\forall n \in [1, N]$, $\mathrm{I}_N(n)$; $\forall n \in [1, N]$, $\forall i \in [1, I_N(i)]$
Output: $\mathbf{P}$
Generate $\forall n \in [1, N]$ and $\forall i \in [1, I_N(n)]$, $\mathbf{Z}_n(i) = (\mathbf{S}_n(i))$ with an integer partitioning algorithm
for $n = 1$ to $N$ do
  for $i = 1$ to $\mathrm{I}_N(n)$ do
    Select a distribution $\mathbf{S}_n(i) = [S^i_{n,1}, S^i_{n,2}, \cdots, S^i_{n,n}, 0, 0, \cdots, 0]$
    Compute the following probabilities using appendices A and B for TEL and ODL respectively (check the existence of links using exceptions detailed in Sections 3.5.6.1 and 3.5.6.2), and fill the matrix $\mathbf{P}$:
    For TEL : $p_{\mathbf{Z}_n(i)\xi_1^n(i)}$, $p_{\mathbf{Z}_n(i)\mathbf{Z}_n(i)}$, $p_{\xi_1^n(i)\xi_2^n(i)}$, $p_{\xi_1^n(i)\mathbf{Z}_n(i)}$, $p_{\xi_2^n(i)\mathbf{Z}_n(i)}$, $p_{\xi_2^n(i)\xi_3^n(i)}$, $p_{\xi_2^n(i)\xi_2^n(i)}$, $p_{\xi_3^n(i)\xi_4^n(i)}$, $p_{\xi_4^n(i)\xi_0^n(i)}$, $p_{\xi_4^n(i)\xi_3^n(i)}$, $p_{\xi_4^n(i)\xi_4^n(i)}$, $p_{\xi_0^n(i)\mathbf{Z}_n(i)}$, $p_{\xi_0^n(i)\mathbf{Z}_n(i)}$ using (A.1), (A.2), (A.4), (A.5), (A.6), (A.7), (A.8), (A.10), (A.11), (A.12), (A.13), (A.15) respectively,
    For ODL : $p_{\mathbf{Z}_n(i)\xi_1^n(i)}$, $p_{\mathbf{Z}_n(i)\xi_2^n(i)}$, $p_{\mathbf{Z}_n(i)\xi_3^n(i)}$, $p_{\mathbf{Z}_n(i)\mathbf{Z}_n(i)}$, $p_{\xi_1^n(i)\mathbf{Z}_n(i)}$, $p_{\xi_1^n(i)\xi_2^n(i)}$, $p_{\xi_1^n(i)\xi_1^n(i)}$, $p_{\xi_2^n(i)\mathbf{Z}_n(i)}$, $p_{\xi_2^n(i)\xi_1^n(i)}$, $p_{\xi_2^n(i)\xi_2^n(i)}$, $p_{\xi_3^n(i)\mathbf{Z}_n(i)}$, $p_{\xi_3^n(i)\xi_3^n(i)}$ using (B.1), (B.2), (B.3), (B.4), (B.10), (B.11), (B.12), (B.15), (B.16), (B.17), (B.22), (B.23), respectively.
    for $k = 1$ to $n$ do
      if $S^i_{n,k} > 1$ then
        $w \leftarrow (S^i_{n,1}, \cdots, S^i_{n,k} - 1, \cdots, S^i_{n,n}, 1, 0, \cdots, 0)$
        $\tilde{w} \leftarrow w$ sorted in decreasing order
        Find $j \in \mathrm{I}_N(n+1)$ such that $\mathbf{S}_{n+1}(j) = \tilde{w}$ which corresponds to state $\mathbf{Z}_{n+1}(j)$
        Compute the following probabilities using Appendices A and B for TEL and ODL respectively (check the existence of links using exceptions in Sections 3.5.6.1 and 3.5.6.2), and fill the matrix $\mathbf{P}$:
        For TEL : $p_{\mathbf{Z}_n(i)\mathbf{Z}_{n+1}(j)}$, $p_{\mathbf{Z}_n(i)\xi_0^{n+1}}$, $p_{\xi_2^{n+1}(j)\mathbf{Z}_n(i)}$, $p_{\xi_4^{n+1}(j)\mathbf{Z}_n(i)}$, $p_{\xi_4^{n+1}(j)\xi_0^n(i)}$ using (A.16), (A.17), (A.18), (A.19), (A.20) respectively,
        For ODL : $p_{\mathbf{Z}_n(i)\mathbf{Z}_{n+1}(j)}$, $p_{\mathbf{Z}_n(i)\xi_1^{n+1}(j)}$, $p_{\mathbf{Z}_n(i)\xi_2^{n+1}(j)}$, $p_{\xi_3^n(i)\mathbf{Z}_{n+1}(j)}$, $p_{\xi_3^n(i)\xi_1^{n+1}(j)}$, $p_{\xi_3^n(i)\xi_2^{n+1}(j)}$, $p_{\mathbf{Z}_{n+1}(j)\mathbf{Z}_n(i)}$, $p_{\mathbf{Z}_{n+1}(j)\xi_3^n(i)}$, $p_{\xi_1^{n+1}(j)\mathbf{Z}_n(i)}$, $p_{\xi_1^{n+1}(j)\xi_3^n(i)}$, $p_{\xi_2^{n+1}(j)\mathbf{Z}_n(i)}$, $p_{\xi_2^{n+1}(j)\xi_1^n(i)}$, $p_{\xi_2^{n+1}(j)\xi_2^n(i)}$ and $p_{\xi_2^{n+1}(j)\xi_3^n(i)}$ using (B.27), (B.28), (B.29),(B.24), (B.25), (B.26), (B.30), (B.32), (B.33), (B.34), (B.35), (B.36), (B.37), (B.38) respectively.
      end if
    end for
  end for
end for
return $\mathbf{P}$
```

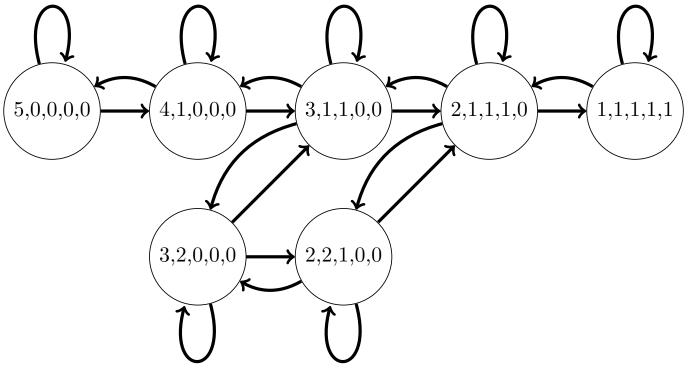
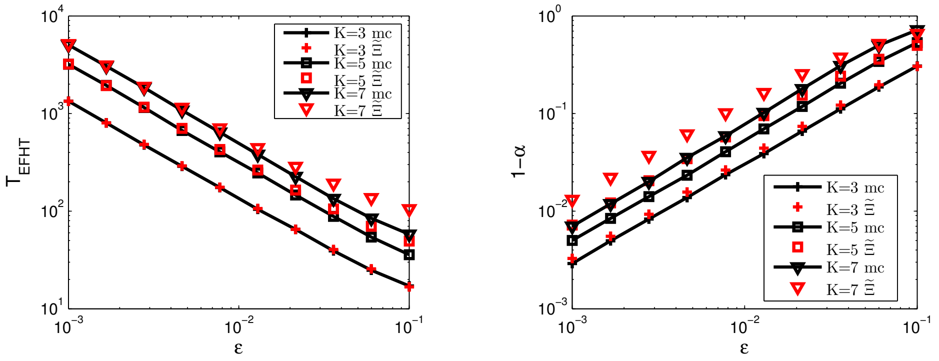


Figure 3.7: Example of transitions considered between RRS in our models for $C = K = 5$.

$N = K$ and, $N = K + 2$. In the ODL, the constant c is equal to K. Both algorithms are compared with respect to the same probability to experiment from a content mood, *i.e.* ϵ in TEL is equal to ϵ^c in ODL. The EFHT is computed from, the state i where all players use the same resource (*e.g.* state with $\mathbf{S}_1(1) = (5, 0, 0, 0, 0)$ in Figure 3.7), to the state j where they are all on different resources (*e.g.* state with $\mathbf{S}_5(1) = (1, 1, 1, 1, 1)$ in Figure 3.7). The stability α_j is computed for the state j where all players use a different resource. In Monte Carlo simulations, denoted by "mc", 5000 trials are used to compute the EFHT and 10^6 trials are used to compute the stability. In what follow, the indices are dropped, the EFHT is noted T_{EFHT} and it is given in number of algorithm iterations and the stability is simply noted α.

For the TEL algorithm, Figures 3.8a and 3.8b present the EFHT and the fraction of time $1 - \alpha$ when $K = N$ respectively. The reason to display $1 - \alpha$ instead of α is to discern the values close to one at low ϵ. For both features, these results are accurate in comparison to Monte Carlo simulations. The EFHT converges to the Monte Carlo results when ϵ decreases. The little gap observed at higher ϵ is caused by an increasing probability to have more than one experiment at a time. Thus, the probability for the system to not be aligned increases and, Assumption 3.1 is less valid. The offset observed in Figure 3.8b is due to the fact that, we are able to represent accurately at most one discontent player at each algorithm iteration. The stability is highly related to the number of discontent players.

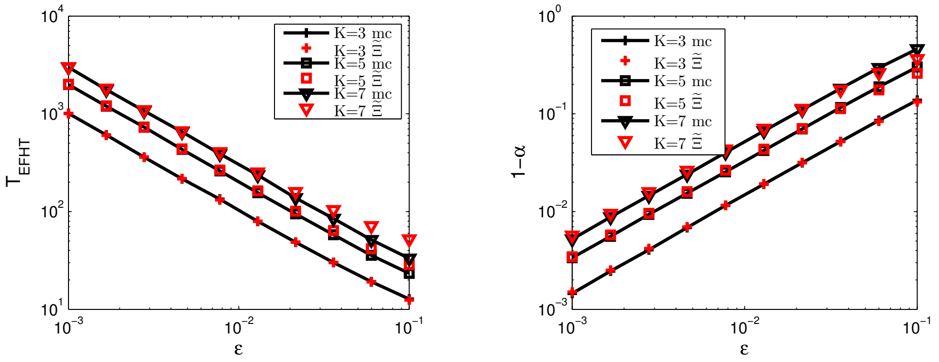
Figures 3.9a and 3.9b present the same results but with $N = K + 2$. The goal is to show the coherence of our approximation. In that scenario, two resources have been added which results in the decrease of the collision probability (a collision is when two players or more try the same resource). Therefore, with respect to the first scenario, Assumption 3.1 is more accurate and, the probability of being



(a) TEL : convergence time in algorithm iterations

(b) TEL : fraction of time

Figure 3.8: Figures of merit comparisons between our approximated models and Monte Carlo simulations when $N = K$.



(a) TEL : convergence time in algorithm iterations

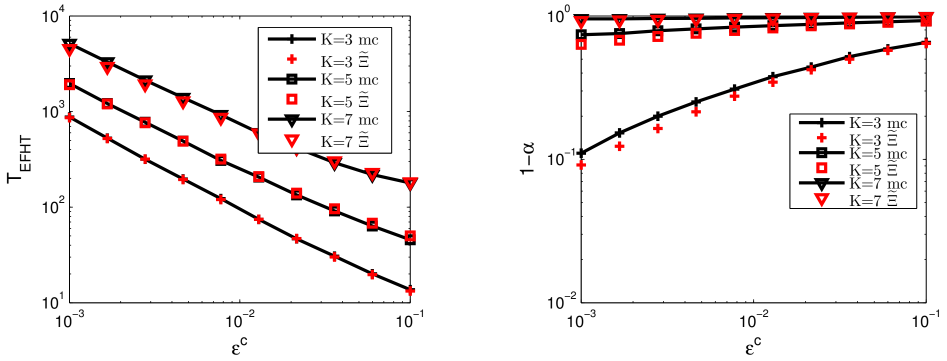
(b) TEL : fraction of time

Figure 3.9: EFHT and stability comparisons between our approximated models and Monte Carlo simulations when $N = K + 2$.

discontent decreases. Consequently, the numerical results of our approximation are closer to Monte Carlo simulations.

In addition, in Figures 3.8 and 3.9, one can check the result detailed in Observation 1, in which the behavior of EFHT is $T_{EFHT} = \mathcal{O}(\frac{1}{\epsilon^{a_2}})$ where $a_2 > 0$ and the behavior of the stability is $1 - \alpha = \mathcal{O}(\epsilon^{a_4})$ where $a_4 > 0$.

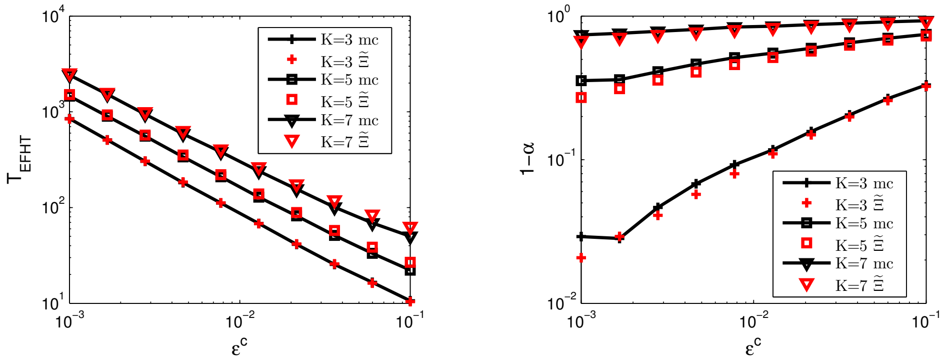
For ODL algorithm, Figures 3.10a and 3.10b present when $K = N$ the EFHT and the fraction of time α_j, respectively. For both features, these results are accurate in comparison to Monte Carlo simulations. The gap observed at low ϵ for stability metric is due to the number of discontent players. We recall that the proposed approximation models accurately at most two discontent players.



(a) ODL : convergence time in algorithm iterations

(b) ODL : fraction of time

Figure 3.10: EFHT and stability comparison between our approximated models and Monte Carlo simulations when $N = K$.



(a) ODL : convergence time in algorithm iterations

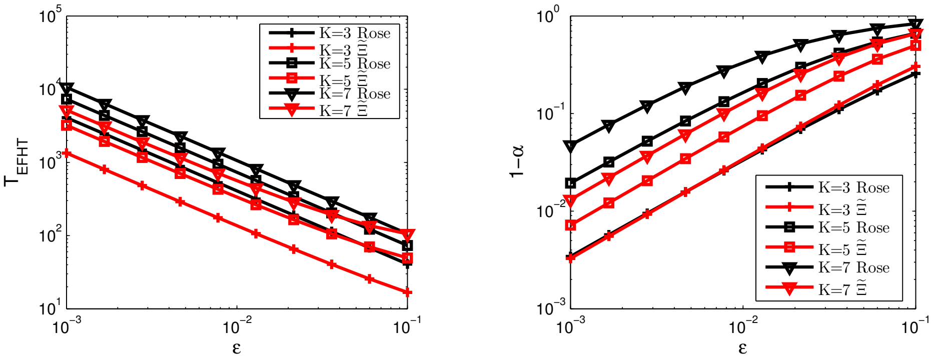
(b) ODL : fraction of time

Figure 3.11: EFHT and stability comparison between our approximated models and Monte Carlo simulations when $N = K + 2$.

When ϵ decreases the number of discontent players increases (a player remains in D with probability $1 - \epsilon$ when $u = 0$) above two with an increasing probability and the model becomes less accurate.

We present in Figures 3.11a and 3.11b the same results but with $N = K + 2$. The accuracy of both features studied is again assessed. The probability to have collisions decreases and so does the probability to have a high number of discontent players. This leads to a better accuracy of the proposed model.

Generally, one can notice how the stability decreases with the number of players and how the convergence time increases. Furthermore, the convergence time decreases when the number of resource increases. In addition, in Figures 3.10



(a) Convergence time in algorithm iterations

(b) Fraction of time

Figure 3.12: Performance comparison between our approximation and Rose approximation when $N = K$.

and 3.11, one can check the result detailed in Observation 2, in which the EFHT is $T_{EFHT} = O(\frac{1}{\epsilon^{cb_2}})$ where $b_2 > 0$ and the behavior of the stability is $1 - \alpha = O(\epsilon^{cb_4})$ where $b_4 > 0$.

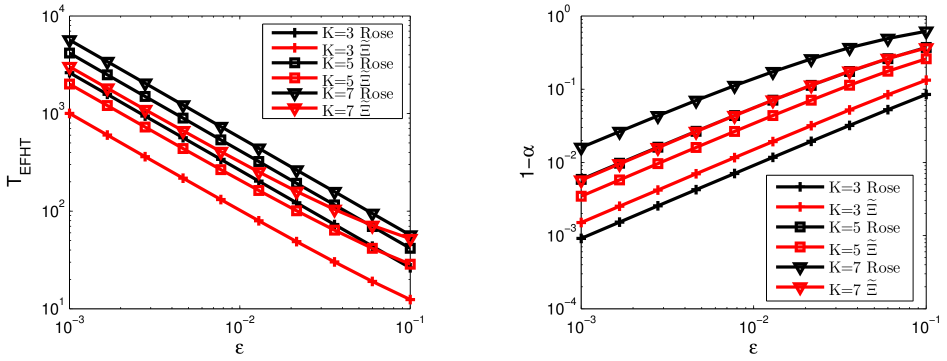
3.5.9.2 Performance comparisons with approximation in the literature

In this section, we compare the results obtained in the previous section with the approximation given in [31], noted model 1 in this work. This last model figures of merit are computed as follows. For the EFHT, the equation (33) in [31] is employed. For the stability, Theorem 5 in [31] gives the stability α but some corrections have been made. For instance, the term $T_{CNE}(k)$ from [31] is replaced with equation (33) in [31] whose sum is started in k instead of 0. The reason for this change is that the variable $T_{CNE}(k)$ diverges when $N = K$ and, it is an upper bound of (33) [31].

Figures 3.12a and 3.13a present the EFHT of the model $\widetilde{\Xi}$ and model 1 when $N = K$ and $K + 2$, respectively. One can observe that both models are quite far from each other except for high ϵ. Knowing that our model converges close to simulations, we immediately deduce the model 1 lack of accuracy.

On the other hand, Figures 3.12b and 3.13b present $1 - \alpha$ when $N = K$ and $K + 2$ respectively. One can notice that, except for $K = 3$, the curves resulting from model 1 are above those of model $\widetilde{\Xi}$. As our model is a tight upper bound on results obtained with Monte Carlo simulations (see Figures 3.8b and 3.9b), it again assesses the accuracy of our model.

For the stability metric, in the case $N = 3$ and $K = N$, the model 1 is as closed to



(a) Convergence time in algorithm iterations

(b) Fraction of time

Figure 3.13: Performance comparison between our approximation and Rose approximation when $N = K + 2$.

Monte Carlo simulations as our proposed approximation. However, contrary to model $\widetilde{\Xi}$, the result for $N = K+2$ shows that model 1 gets away from the simulation contrary to our approximation that gets closer. This proves the coherence of our model in comparison to model 1.

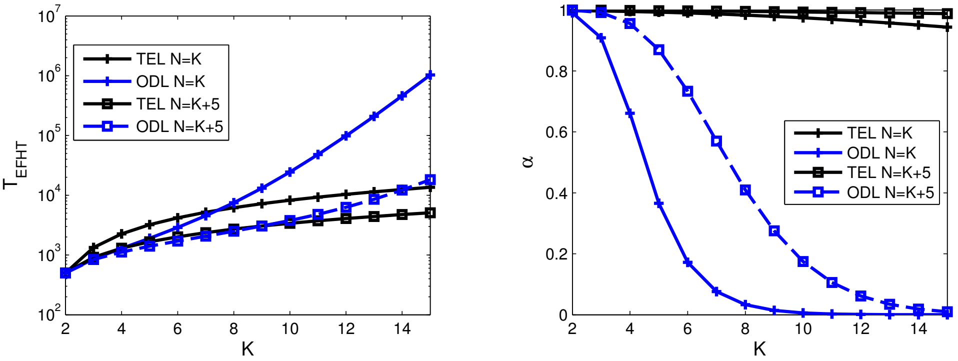
3.5.9.3 Performance comparison between TEL and ODL

In previous sections, we have characterized the accuracy of TEL and ODL proposed models. In this section, we take the advantage of the available approximations that have low complexity, to compare both algorithms and, to analyze their performance in domains hardly reachable with Monte Carlo simulations. Figures 3.14a and 3.14b present, for both algorithms, the EFHT (in logarithmic scale) and the stability α respectively. The number of resources used is $N = K$ and $N = K + 5$ and, the probability to experiment is fixed to $\epsilon = \epsilon^c = 10^{-3}$. The increase of N results for both algorithms, first, in a better stability and, secondly, in a lower convergence time. This result counteracts the argument that the convergence time increases with the alphabet size ([31] Section V. B.). The reason is that players find a free interference state faster and, the probability that two players interfere is less important when the set of free resources is larger. There exists a value of K such that the EFHT of both algorithms is the same. Below this value, ODL is more efficient than TEL with respect to the convergence time and beyond this value the behavior is inverted. More generally, the fact that in some cases TEL converges faster than ODL contradicts the idea that, the larger the algorithm controller (4 moods for TEL and 2 moods for ODL), the slower its convergence, as it is said in [33] (Section IV-B). In addition, Figure 3.14b shows that TEL is more stable than ODL even when N is increased and such, at any K. Figure 3.15 presents the same results as in Figure 3.14 but with $\epsilon = \epsilon^c = 10^{-4}$. The convergence time of both algorithms are increased. This is not a surprise as we deduce in Observations 1 and 2 that T_{EFHT} is inversely proportional to some power of ϵ. The decrease of ϵ increases the stability of both algorithms. As ϵ decreases, so does the probability to experiment of players in state C. Thus, the probability that two players or more collide also decreases with ϵ which results in a higher stability of the state. More generally, the convergence and stability tendencies remain the same in comparison to Figure 3.14. In Figures 3.14a and 3.15a, one can assess the results detailed in Observations 1 and 2. The EFHT of ODL and TEL respectively follow an exponential and a polynomial behavior with respect to K (the y-axis is in logarithmic scale). In Figures 3.14b and 3.15b one can guess the exponential and polynomial decreasing of the ODL and TEL stability respectively. Figure 3.16 presents the stability $1 - \alpha$ and α with respect to K for the TEL and ODL respectively. These two figures confirm the previous guess and assess the convergence results of Observations 1 and 2.

To conclude, in our system model, ODL is less stable than TEL. There exists some region of K for which ODL converges faster. However, the gain in speed convergence is not considerable and, the exponential behavior of ODL with respect to K makes the convergence of this algorithm possibly very long in large systems. This small advantage in convergence speed is compromised by less stability. In view of the results, we would recommend that, the use of ODL algorithm in an environment with large utility variation is preferable when the need in stability is not important and the amount of players is limited.

3.6 Conclusion

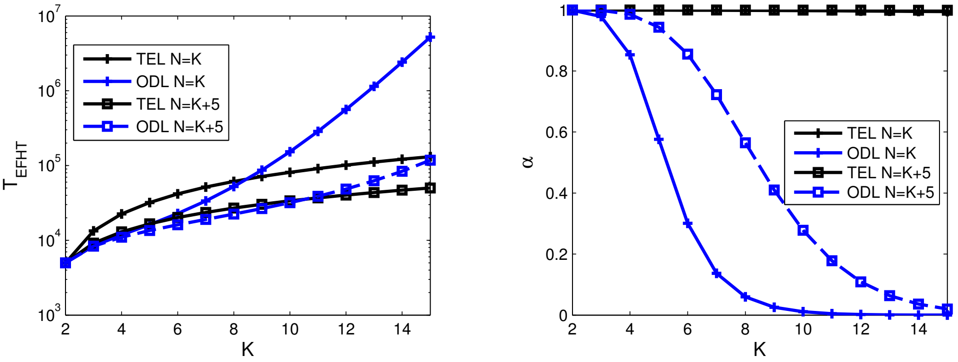
This chapter has first presented the wireless model in a deterministic environment, the utility function and the game model. We then have showed that this game does not necessarily possess a PNE. This is one of the reasons we have selected TE algorithms, that converge in a broad class of games, as candidates to solve the frequency channel allocation problem. The algorithms, TEL and ODL, have been presented as well as their theoretical convergence results. Numerical results obtained in this context have illustrated the good convergence behaviors of both algorithms. Then, we have provided a detailed performance analysis of these learning strategies. To overcome the huge dimension of the inherent MC of the



(a) Convergence time in algorithm iterations

(b) Fraction of time

Figure 3.14: Performance comparison between TEL and ODL when $\epsilon = 10^{-3}$ with respect to K.

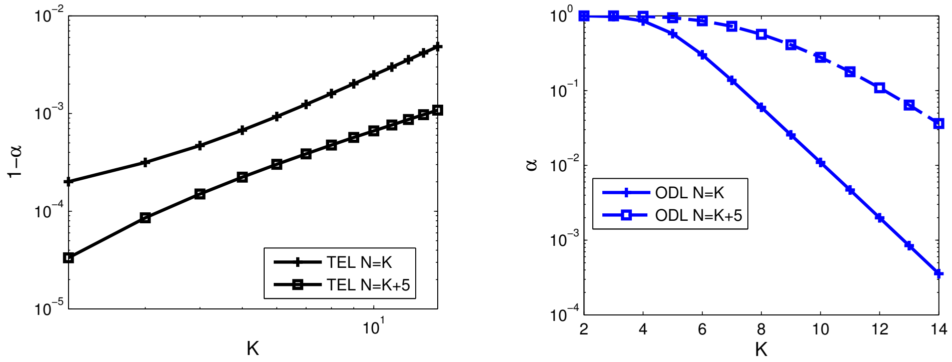


(a) Convergence time in algorithm iterations

(b) Fraction of time

Figure 3.15: Performance comparison between TEL and ODL when $\epsilon = 10^{-4}$ with respect to K.

game, we have provided an approximation of these chains. This has allowed to compute a close approximation of the average time the system remains in a desired state as well as the average time required to achieve that state for the first time. Thanks to the above approximations, a comparison between the performance of TEL and ODL has been provided.



(a) Fraction of time

(b) Fraction of time

Figure 3.16: Stability performance of TEL and ODL with respect to K.

Chapter 4

Distributed channel allocation in presence of disturbances

4.1 Introduction

The theory developed by Young [38] to design TE algorithms such as the TEL and the ODL assumes that for each action vector corresponds a unique utility vector. This context is referred to as *deterministic context* in the sequel. These algorithms are based on FSCs whose states transitions depend on the utility variations that are caused by action changes of the players. When applying these algorithms to the telecommunication context, this assumption is no longer valid because of the stochastic nature of the medium, *i.e.*, for a fixed given action vector the corresponding utility vector is time varying. This context is referred to as *stochastic context* in the sequel. As a consequence, players perceive utility variations that are not caused by action changes, leading to undesired transitions in the FSC. Hence, convergence properties are no longer valid as noticed in [34] for the ODL, causing performance degradations as observed in [35] for the TEL. Thus, the goal of this chapter is to study how to adapt TE algorithms to take into account time variations of the utility so as to improve their performance.

Let $\underline{\mathbf{u}}(\mathbf{a})$ be the utility vector obtained by playing action $\mathbf{a}$ in the deterministic context. In the stochastic context, the utility vector obtained by playing consistently $\mathbf{a}$ varies over time around $\underline{\mathbf{u}}(\mathbf{a})$. We name these variations in the sequel *disturbances* as in [34]. Authors in [34] adapted the ODL algorithm to the stochastic context for ramp coordination in traffic control. We refer the corresponding algorithm to as Robust Optimal Distributed Learning (RODL). Remember that in the FSC of ODL there is only one transition that is triggered by utility variations. In content mood, when there is no experiment, the state may change when

$u \neq \bar{u}$, see Section 3.3.2 page 25. The basic principle of RODL is to introduce one threshold, denoted by ρ, such that the previous condition is changed into $u \notin [\bar{u} - \rho, \bar{u} + \rho]$. The role of the threshold is to absorb the disturbances and it represents its maximum deviation.

To the best of our knowledge the TEL algorithm has not been adapted to a stochastic context. In this thesis, we apply the idea of introducing a threshold as in [34] to the TEL algorithm, that we name Robust Trial and Error Learning (RTEL). We prove under specific disturbance assumptions the convergence of the RTEL to some Stochastic Stable State (SSS). Notice that the idea of using thresholds was evoked in [13], named *tolerance*, as a potential extension to its work. As pointed out in [34] the maximum deviation ρ is not a priori known by players and then needs to be estimated. Notice that in their work this value is set empirically with respect to the simulation context. In the work presented here, we improve the use of the thresholds in three ways:

1. we introduce two different thresholds, one for the lower bound noted δ^-, and one for the upper bound noted δ^+, such that for instance the previous test $u \notin [\bar{u} - \rho, \bar{u} + \rho]$ becomes $u \notin [\bar{u} - \delta^-, \bar{u} + \delta^+]$,

2. we consider that each cluster k has its own pair of thresholds (δ_k^-, δ_k^+) that are set independently between clusters,

3. we provide an on the fly algorithm to learn the thresholds per cluster to adapt to the disturbance distribution.

Since the probability density function (pdf) of the utility belongs to the class of Bernoulli-type distributions which may exhibit strong asymmetry, modification 1 allows to account for the skewness of the utility disturbance. Modification 2 is necessary since each cluster perceives the disturbance differently and is compulsory for a full distributed implementation. In the telecommunication context, notice that as a by-product, the adaptive learning of the thresholds 3 also allows to account for the nodes mobility. These improvements are applied to both the RODL and the RTEL.

This chapter is organized as follows. Section 4.2 presents the model of the utility with disturbances. Section 4.3 presents the RTEL and recall the RODL algorithms. Section 4.4 presents the theoretical convergence results of TEL, ODL, RTEL and RODL in the stochastic context. Section 4.5 presents the algorithm to adapt thresholds on the fly. Section 4.6 presents simulations results of the TE algorithms and their enhanced versions in the stochastic context. Section 4.7 concludes the chapter.

4.2 Disturbed utility model

4.2.1 Disturbance characterization

First, we remind from (3.3) (page 21) that the utility of cluster k is given by

$$u_k(\mathbf{a}) = \frac{1}{|\mathcal{L}_k|} \sum_{i \in \mathcal{L}_k} \ell_i^k(\mathbf{a}), \tag{4.1}$$

where $\ell_i^k(\mathbf{a}) = \mathbb{1}_{\{\mathrm{SINR}_i^k(\mathbf{a})>\Gamma_0\}}$ indicates the link status (0: the link is not working, 1: the link is working) with $\mathrm{SINR}_i^k(\mathbf{a})$ the SINR of link i in cluster k. In Chapter 3 we have considered the so-called deterministic context, meaning that the SINR is assumed to be a constant (not time-varying) for each link as long as $\mathbf{a}$ is played. As a consequence, the values $\ell_i^k(\mathbf{a}), \forall i, \forall k$ are fixed for a given $\mathbf{a}$ and the corresponding utility is a deterministic constant. Then, the deterministic context can be formalized as follows:

Let us define $\underline{\ell}_i^k(\mathbf{a})$ and $\underline{u}_k(\mathbf{a}) := \frac{1}{|\mathcal{L}_k|} \sum_{i \in \mathcal{L}_k} \underline{\ell}_i^k(\mathbf{a})$ the constant deterministic values taken by the links' status and the corresponding deterministic utility respectively, when the SINR of all links are assumed constant. Then, the deterministic context can be formalized as follows:

Definition 4.1 (Deterministic context)**.** $\forall k \in \mathcal{K}$, $\forall t, \forall \mathbf{a}_0 \in \mathcal{N}^K$, $\mathbf{a}_t = \mathbf{a}_0 \Rightarrow u_k(\mathbf{a}_t) = \underline{u}_k(\mathbf{a}_0)$, *where* $\underline{u}_k(\mathbf{a}_0)$ *is the deterministic utility received by cluster* k *when* $\mathbf{a}_0$ *is played.*

Now, we consider the case where the SINR is time-varying and so does the utility (see property *(iv)* in Section 1.2). This may happen for various reasons and two examples are given in Section 4.2.2. As a consequence, the link status are time-varying and can be modeled as:

$$\ell_i^k(\mathbf{a}) = \underline{\ell}_i^k(\mathbf{a}) \oplus b_i^k(\mathbf{a}), \tag{4.2}$$

where $\oplus$ stands for the logical OR operator and $b_i^k(\mathbf{a})$ is a boolean random variable. From (4.2) we deduce that utility (4.1) can be written:

$$u_k(\mathbf{a}) = \underline{u}_k(\mathbf{a}) + \xi_k(\mathbf{a}), \tag{4.3}$$

where the random variable (rv) $\xi_k(\mathbf{a})$ is called the utility disturbance. We can now define the stochastic context as follows:

Definition 4.2 (Stochastic context)**.** $\forall k \in \mathcal{K}$, $\forall t, \forall \mathbf{a}_0 \in \mathcal{N}^K$, $\mathbf{a}_t = \mathbf{a}_0 \Rightarrow u_k(\mathbf{a}_t) = \underline{u}_k(\mathbf{a}_0) + \xi_k(\mathbf{a})$ *where* $\underline{u}_k(\mathbf{a}_0)$ *is the deterministic utility received by cluster* k *when* $\mathbf{a}_0$ *is played, and* $\xi_k(\mathbf{a})$ *is a random variable called the utility disturbance.*

It is important to note that, although time index is omitted in the notation, $\xi_k(\mathbf{a})$ is time-varying. It is also stochastic state and payoff dependent and thus not an independent and identically distributed (iid) noise (*i.e.* averaging over this noise cannot be used to cancel out its effect).

Inserting (4.2) into (4.1), we deduce from (4.3) that

$$\xi_k(\mathbf{a}) = n_k(\mathbf{a}) w_k, \tag{4.4}$$

where $n_k(\mathbf{a}) := \sum_{i \in \mathcal{L}_k} \ell_i^k(\mathbf{a}) - \underline{\ell}_i^k(\mathbf{a})$ is an integer rv and

$$w_k := \frac{1}{|\mathcal{L}_k|}. \tag{4.5}$$

Moreover, since $u_k(\mathbf{a}) \in [0, 1]$, we deduce that $-\underline{u}_k(\mathbf{a}) \leq \xi_k(\mathbf{a}) \leq 1 - \underline{u}_k(\mathbf{a})$, and thus $-\underline{n}_i^k(\mathbf{a}) \leq n_k(\mathbf{a}) \leq |\mathcal{L}_k| - \underline{n}_i^k(\mathbf{a})$, with $\underline{n}_i^k(\mathbf{a}) := \sum_{i \in \mathcal{L}_k} \underline{\ell}_i^k(\mathbf{a})$.

Following [34], one way to characterize the disturbance is to consider its effect on the utility maximum deviation among all clusters, for all action vectors, and all realizations ω_i:

$$\mu := \max_{\mathbf{a}, \omega_1, \omega_2, k} |u_k(\mathbf{a}, \omega_1) - u_k(\mathbf{a}, \omega_2)|, \tag{4.6}$$

where $u_k(\mathbf{a}, \omega)$ is the perceived utility at cluster k at realization ω of the disturbance. According to (4.3), (4.6) is equal to

$$\mu = \max_{\mathbf{a}, \omega_1, \omega_2, k} |\xi_k(\mathbf{a}, \omega_1) - \xi_k(\mathbf{a}, \omega_2)|, \tag{4.7}$$

which shows that the maximum utility deviation is equal to the maximum disturbance deviation. Now, from (4.4), we get

$$\mu = w_k \max_{\mathbf{a}, \omega_1, \omega_2, k} |n_k(\mathbf{a}, \omega_1) - n_k(\mathbf{a}, \omega_2)|, \tag{4.8}$$

which shows that the maximum utility deviation is a multiple integer of w_k, and thus there exists a non negative integer v_k such that

$$v_k := \frac{\mu}{w_k} = \mu |\mathcal{L}_k|. \tag{4.9}$$

Notice that from (4.6), we have $\mu \in [0, 1]$. However, we will see in Section 4.4 that we need to make the following assumption:

Definition 4.3 (Bounded utility disturbance assumption)**.** *The bounded utility disturbance assumption corresponds to the case where the maximum deviation of the disturbance utility* (4.7) *verifies* $0 < \mu < 1$.

Notice that this assumption plays an instrumental role in the derivation of the algorithms in Section 4.3 that restore the convergence properties thanks to the thresholds. However, there is no reason that this assumption is fulfilled in general. This contradiction will be discussed in details in Section 4.5.

4.2.2 Disturbance examples

Disturbances of the utility that constitute the statement of property *(iv)* in Section 1.2 may appear for a lot of reasons depending on the communication system setting. We give here two different cases leading to disturbances in our dynamic frequency allocation context that will be used in the simulations of Section 4.6 .

The first case of disturbance that we consider is the Rayleigh fading. When the propagation channel between transmitters and receivers is random (as for the Rayleigh fading), the SINR becomes a rv even if the same $\mathbf{a}$ is played along time. As a consequence, the working links metrics $\ell_i^k(\mathbf{a}) = \mathbb{1}_{\{\mathrm{SINR}_i^k > \Gamma_0\}}$ become Bernoulli rvs, and the utility a Poisson-Bernoulli one.

The second case that we consider is the effect of the link scheduling inside clusters. In the case of a Time Division Multiple Access (TDMA), the links are scheduled according to Quality of Service (QoS) criteria and to the traffic load. Thus, the order of transmissions inside the cluster (who is transmitting to whom) can be seen as a random process even if the same $\mathbf{a}$ is played along time. As a consequence, the interference perceived by the different links is changing over time and so the SINR, leading to a disturbed utility.

Note that the derivation of the SINR pdf for the case with Rayleigh fading is detailed in Chapter 5.

4.3 Robust trial and error algorithms

This section presents the RTEL and RODL algorithms that extend the TEL and ODL ones to the stochastic context. As said in Section 4.1, these algorithms use thresholds (δ_k^-, δ_k^+) that enable us to restore their convergence properties under the bounded assumption (see Definition 4.3).

4.3.1 RTEL

This section describes the new rules applied to the RTEL FSC for any $k \in \mathcal{K}$ and for any $\epsilon \in (0, 1]$:

- $m_k = \mathrm{C}$, there are two cases to consider :

  1) with probability $1 - \epsilon$, the player keeps playing its benchmark (i.e. $a_k = \bar{a}_k$). The next state changes to H if $u_k > \bar{u}_k + \delta_k^+$ or, it changes to W if $u_k < \bar{u}_k - \delta_k^-$ or, it remains C if $u_k \in [\bar{u}_k - \delta_k^-, \bar{u}_k + \delta_k^+]$.

2) with probability ϵ, the player experiments a new action, i.e. $a_k \in \mathcal{N}\backslash\{\bar{a}_k\}$. The action experimented is selected randomly among $\mathcal{N}\backslash\{\bar{a}_k\}$ (i.e. $\Pr\{a_k = r_i\} = \frac{1}{N-1}$, $\forall r_i \neq \bar{a}_k$) and, the next state remains $m_k = C$. When $u_k > \bar{u}_k + \delta_k^+$, player k updates its benchmark with probability $\epsilon^{G(u_k - \bar{u}_k)}$, where $G(x) = -\nu_1 x + \nu_2$, with $\nu_1 > 0$ and ν_2 such that $0 < G(u_k - \bar{u}_k) < 1/2$. An update consists in changing the benchmark by the played action and the received utility in the next iteration as follows, $\bar{u}_k \leftarrow u_k$ and $\bar{a}_k \leftarrow a_k$.

- $m_k = H$: $a_k = \bar{a}_k$ and the next state changes to C with a utility benchmark update (i.e. $\bar{u}_k \leftarrow u_k$) if $u_k > \bar{u}_k + \delta_k^+$ or, it changes to W if $u_k < \bar{u}_k - \delta_k^-$ or, it changes to C if $u_k \in [\bar{u}_k - \delta_k^-, \bar{u}_k + \delta_k^+]$.

- $m_k = W$: $a_k = \bar{a}_k$ and the next state changes to H if $u_k > \bar{u}_k + \delta_k^+$ or, it changes to D if $u_k < \bar{u}_k - \delta_k^-$ or, it changes to C if $u_k \in [\bar{u}_k - \delta_k^-, \bar{u}_k + \delta_k^+]$.

- $m_k = D$: an action a_k is randomly selected among $\mathcal{N}$ (i.e. $\Pr\{a_k = r_i\} = \frac{1}{N}$, $\forall r_i \in \mathcal{N}$) with probability 1. The next state m_k changes to C with probability $\epsilon^{F(u_k)}$, where $F(u) = -\phi_1 u + \phi_2$ with, $\phi_1 > 0$ and ϕ_2 such that $0 < F(u) < 1/2K$, with a benchmark update (i.e. $\bar{u}_k \leftarrow u_k$ and $\bar{a}_k \leftarrow a_k$), otherwise, with probability $1 - \epsilon^{F(u_k)}$, $m_k = D$.

Note that when we set $\delta_k^- = 0$ and $\delta_k^+ = 0$, the RTEL reduces to the TEL.

4.3.2 RODL

This section describes the new rules applied to the RODL FSC. This algorithm extends the one given in [34] by considering two different thresholds δ^- and δ^+ per cluster, and also the fact that these thresholds are different from one cluster to another:

- $m_k = C$, there are two cases to consider :

1) with probability $1 - \epsilon^c$, where $c > K$ is a real constant, $a_k = \bar{a}_k$. If $u_k \notin [\bar{u}_k - \delta_k^-, \bar{u}_k + \delta_k^+]$ then the state m_k changes to D with probability $1 - \epsilon^{1-u_k}$. Otherwise, with probability ϵ^{1-u_k}, the cluster updates its benchmark (i.e. $\bar{u}_k \leftarrow u_k$) and remains C.

2) with probability $\epsilon^c > 0$, a new action is experimented, $a_k \in \mathcal{N}\backslash\{\bar{a}_k\}$. The new action is selected randomly in the set $\mathcal{N}\backslash\{\bar{a}_k\}$. The state m_k changes to D with probability $1 - \epsilon^{1-u_k}$. Otherwise, with probability ϵ^{1-u_k}, the cluster updates its benchmark (i.e. $\bar{u}_k \leftarrow u_k$ and $\bar{a}_k \leftarrow a_k$) and remains in C.

- $m_k = D$: an action a_k is randomly chosen among $\mathcal{N}$. The cluster switches to C with probability ϵ^{1-u_k} and updates its benchmark (i.e. $\bar{u}_k \leftarrow u_k$ and $\bar{a}_k \leftarrow a_k$), otherwise with probability $1 - \epsilon^{1-u_k}$, it remains D.

Note that when we set $\forall k, \delta_k^- = \delta$ and $\delta_k^+ = \delta$, the RODL reduces to the algorithm given in [34], and when we set $\delta_k^- = 0$ and $\delta_k^+ = 0$, the RODL reduces to the ODL.

4.4 Convergence properties of TE based algorithms in the stochastic context

In this section, we address the convergence of the TE based algorithms in the stochastic context. First, we show that the TEL and ODL do not converge to a desirable state in a disturbed environment. Then, we prove the convergence of the RTEL and RODL algorithms under the bounded assumption (see Definition 4.3). As in the deterministic context, an additional assumption required in the stochastic context to get those results is the *interdependence* property (see Definition 2.4 in page 13). In the case of robust algorithms, this property ensures that given any state of the network, there exists an appropriate action change such that a set of clusters modifies the utility of at least one different cluster more than its tolerance level. Thus, in any state there always exists an appropriate group of clusters that can make the network leave the current state.

We now define the non-cooperative stochastic game in normal form that constitutes the framework adapted to the study of the convergence of RTEL and RODL algorithms in the stochastic context.

Definition 4.4 (stochastic game in normal form). *A normal form stochastic game with K players is defined by the quadruplet $\mathcal{G}_s = (\mathcal{K}, (\mathcal{N}_k)_{k\in\mathcal{K}}, (\Omega_k)_{k\in\mathcal{K}}, (u_k(\mathbf{a}, \omega)_{k\in\mathcal{K},\mathbf{a}\in\mathcal{N},\omega\in\Omega_k}))$ where $\mathcal{K}$ is the set of players, $\mathcal{N}_k$ is the set of player k actions $\forall k \in \mathcal{K}$, $\mathcal{N} = \prod_{k=1}^{K} \mathcal{N}_k$, Ω_k is the set of possible disturbance realizations for players $k \in \mathcal{K}$, $u_k : \mathcal{N} \times \Omega_k \to [0, 1]$ is the disturbed utility function of player i given by model (4.3).*

Let us first present the proposition that states the loss of convergence properties for both algorithms (we remind that the proofs in the deterministic context are presented in Section 3.4.1).

Proposition 4.1 (Instability in presence of disturbances). *Let an interdependent stochastic game $\mathcal{G}_s$ in which all clusters use the TEL or the ODL. The state $\mathbf{z}$ in which all clusters are discontent is the SSS.*

The proof of this proposition is given in Appendix C.1. It says that the network spends a high amount of time in a state (the discontent state) in which all clusters behave randomly. This is an undesirable state which motivated us to propose the robust TEL algorithms.

We are now addressing the convergence of the RTEL and RODL in the game $\mathcal{G}_s$. Before stating the corresponding theorems, we need to introduce some notions and notations that are required to understand them and whose formal definitions are provided in Appendix C.2. Notice that we have stated and derived the proofs of these theorems assuming that the number of links in each cluster is a constant and thus $\forall k, w_k = w$ and $v_k = v$, and also that $\delta_k^- = \delta$ and $\delta_k^+ = \delta$. These assumptions have been made in order to simplify the notations for the sake of proof clarity, but these results can be easily extended to the general case.

- C_δ^0: denotes the set of states in which all clusters are content (*i.e.* $\forall k \in \mathcal{K}$, $m_k = C$) and aligned. In a deterministic context a cluster is said to be aligned if $u = \bar{u}$ and $a = \bar{a}$. In a disturbed environment and with the introduction of tolerance thresholds this notion is slightly modified. A cluster is aligned if when $a = \bar{a}$ then $u \in [\bar{u} - \delta, \bar{u} + \delta]$ (see (C.3)).

- E_δ^0: denotes the set in which no cluster can increase its benchmark by more than δ when comparing the utility with the benchmark (referred to as δ-PNE in the sequel; see (C.4) for the definition). We have $E_\delta^0 \subset C_\delta^0$.

- $S_\delta(\mathbf{z})$: for $\mathbf{z} \in \Xi$ (see Section 3.4.1 page 26), it represents the maximum utility increase that one of the player can obtain by deviating unilaterally. In other word, the larger this gap, the more probable the network to observe an action change. It is the propensity that the network has to leave a state $\mathbf{z}$.

- $\delta^* := \left\lceil \frac{v-1}{2} \right\rceil w$, where w and v are defined in (4.5) and (4.9) respectively, and $\lceil x \rceil$ is the ceiling function of x.

- $\overline{W}(\mathbf{z})$ represents the benchmark social welfare given in Definition 3.2 in page 28. This notion is important here since, unlike the TEL, the convergence of the robust algorithms is derived in terms of benchmark social welfare instead of instantaneous social welfare.

We can now state the following theorems for which proofs are given in Appendix C.2 and Appendix C.3 respectively.

Theorem 4.1 (Convergence result of RTEL)**.** *If the K persons game $\mathcal{G}_s$ is interdependent and all players use the algorithm RTEL with $\delta \geq \delta^*$,* *(i) if $E_\delta^0 \neq \emptyset$, every state in E_δ^0 that maximizes the benchmark social welfare is an SSS,*

(ii) otherwise, every SSS is a state $\mathbf{z}^* \in C_\delta^0$ that maximizes a trade-off between the benchmark social welfare and the instability as follows

$$\phi_1 \overline{W}(\mathbf{z}^*) - \nu_1 S_\delta(\mathbf{z}^*). \tag{4.10}$$

Theorem 4.2 (Convergence result of RODL)**.** *If the K persons game $\mathcal{G}_s$ is interdependent and all players use the algorithm RODL with $\delta \geq \delta^*$, every state in C_δ^0 that maximizes the benchmark social welfare is an SSS.*

Theorems 4.1 and 4.2 show that the RTEL and RODL converge in the stochastic context in terms of benchmark social welfare, under the bounded disturbance assumption. They can be seen as an extension of the convergence results for the TEL and the ODL obtained in the deterministic context expressed in terms of social welfare (using the utility). Theorem 4.1 states that if a δ-PNE exists then RTEL spends a high amount of time in a state that maximizes the benchmark social welfare whereas, otherwise, it maximizes a trade-off between the benchmark social welfare and a stability function. Theorem 4.2 states that in any case, the algorithm spends a high amount of time in a state that maximizes the benchmark social welfare. Notice that a proof of Theorem 4.2 is given in [34], but our proof is simpler and shorter.

4.5 Adaptive robust algorithms

We have seen in the previous section that the convergence of the robust algorithms is ensured provided the tolerance threshold is such that $\delta \geq \delta^*$. The disturbance distribution characteristic δ^* is supposed unknown to the clusters. Its value depends on the actions of the players but also, in our application, on the network topology that is possibly time-varying due to mobility, and the propagation channel conditions between nodes. Moreover, when the players change their actions, the disturbance distributions change accordingly. It is clear that an adaptive solution is required to learn the tolerance levels at each cluster. Thus, the purpose of this section is to provide solutions in order to learn and adapt of the tolerance thresholds for each cluster.

For the sake of clarity, we expose our solution in two steps. First, we present the tolerance thresholds estimation algorithm assuming that the action vector $\mathbf{a}$ of the game and the network topology are fixed. In that case, the disturbance distribution is assumed to be fixed at each cluster and the data flow that is used to estimate the thresholds are rvs belonging to the same distribution. This case will be referred to as *stationary* and on the fly conventional estimators can be used. Second, we tackle the case (referred to as *non-stationary*) when $\mathbf{a}$ is changing over time (due to experimentations of clusters) or the network topology is changing over time. In that case the disturbance distribution is no longer fixed along time at each cluster, and an adaptation needs to be done to detect these changes and to cope with it accordingly.

4.5.1 Assuming that the action vector is fixed

In order to present the procedure to learn the thresholds, we start by considering the topology and the action vector $\mathbf{a}$ fixed which involves that the rvs such as the utility or working link metrics are stationary. The later condition is important for a convergence without bias of thresholds estimators.

The thresholds are set such that a cluster is not sensitive to disturbances which is given by

$$\forall \xi,\ u(\mathbf{a};\xi) \in [\bar{u} - \delta^{-}, \bar{u} + \delta^{+}]. \tag{4.11}$$

With the model of the utility presented in Section 4.2, the disturbance deviation is bounded and one can set targeted positive tolerance thresholds $\delta^{-} = \delta_0^{-*}$ and $\delta^{+} = \delta_0^{+*}$ as follows

$$\delta_0^{-*} := \bar{u} - L_0^*, \tag{4.12}$$

where $L_0^* = \min_{\xi}(u(\mathbf{a};\xi))$ is a low bound on the utility distribution and,

$$\delta_0^{+*} := U_0^* - \bar{u}, \tag{4.13}$$

where $U_0^* = \max_{\xi}(u(\mathbf{a};\xi))$ is an upper bound on the utility distribution. Note that with previous definitions $[\bar{u} - \delta^{-}, \bar{u} + \delta^{+}] = [L_0^*, U_0^*]$.

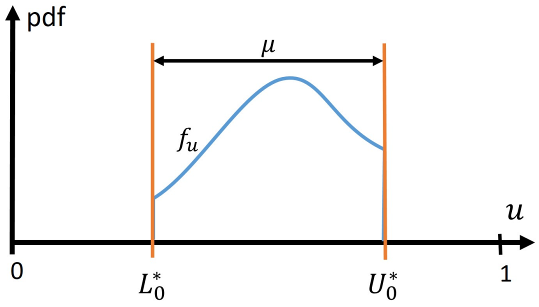


Figure 4.1: Selection of bounds L_0^* and U_0^* in case of bounded disturbance.

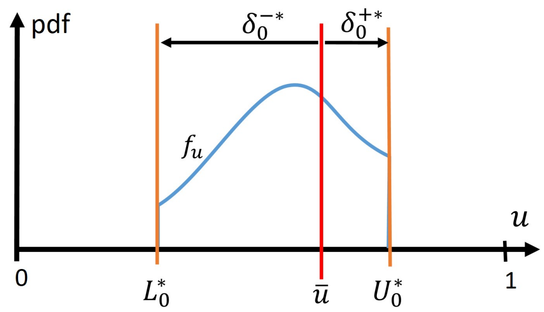


Figure 4.2: Selection of thresholds δ_0^{-*} and δ_0^{+*} in case of bounded disturbance.

Figure 4.1 presents the selection principle of bounds L_0^* and U_0^* when the deviation of the utility is bounded (*i.e.* $\mu < 1$, see Section 4.2). For simplicity, we represent the distribution of the utility as a continuous variable with a pdf denoted by $f_u(.)$ whereas in this thesis it takes discrete values. Then, in the same context, Figure 4.2 illustrates the threshold computations (4.12) and (4.13) when the benchmark utility $\bar{u} \in [L_0^*, U_0^*]$.

It happens that $\bar{u} \notin [L_0^*, U_0^*]$. In this case the benchmark utility is considered "outdated". This arises, for instance, when $\bar{u}$ has been updated and the network state has changed meantime. When the network state changes, the utility distribution also changes and the benchmark utility selected during the previous network state does not represent anymore the new received utility at the cluster. We propose two modifications in order for the cluster to change its benchmark in such case. The first modification is to impose the thresholds δ^- and δ^+ to be positive. The second modification is to prevent the region covered by thresholds δ^+ and δ^- from being larger than the range covered by L_0^* and U_0^*. Figure 4.3 represents this case when $\bar{u}$ is outdated. If the two previous modifications are applied, then $\delta^+ = 0$ because with (4.13) it must be negative whereas with the first modification $\delta^+ = \max\{0, \delta_0^{+*}\}$. In addition, with the second modification $\delta^- = \min\{U_0^* - L_0^*, \delta_0^{-*}\}$ so in this scenario $\delta^- = U_0^* - L_0^*$. It follows that the cluster is sensitive to the dashed part of received utilities and there is a positive probability that the state of the cluster's FSC changes. This later change results in a benchmark update such that with some probability the new benchmark belongs to $[L_0^*, U_0^*]$.

In practice, the bounded assumption (Definition 4.3) is not valid. For instance, in this thesis when the utility is disturbed, it is a Poisson-Binomial process (see Section 5.1.1 in page 100) and when all the Bernoulli parameters are such that $0 < p_i < 1$ (5.2) (note that if $p_i = 0$ or 1 the link is deterministic as it always works or does not work), then $\min_{\xi}(u(\mathbf{a};\xi)) = 0$ and $\max_{\xi}(u(\mathbf{a};\xi)) = 1$ (*i.e.* $\mu = 1$ (4.6) which violates the bounded assumption). As a consequence, with the given bounds L_0^*

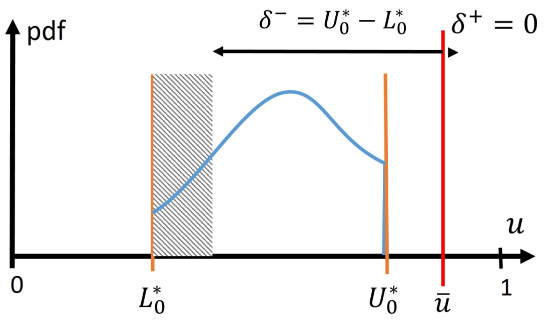


Figure 4.3: Selection of thresholds δ^- and δ^+ in case of bounded disturbance when the benchmark utility $\bar{u}$ does not well represent the perceived utility.

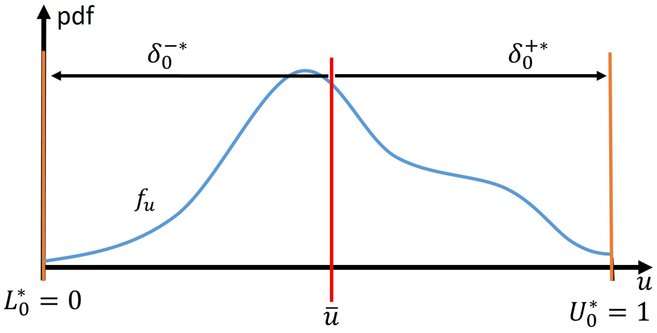


Figure 4.4: Problem for the selection of thresholds in case of unbounded disturbance, *i.e.* $\mu = 1$.

and U_0^*, condition (4.11) is always satisfied since $\bar{u} - \delta^- = L_0^* = 0$ and $\bar{u} + \delta^+ = U_0^* = 1$ and, the cluster is insensitive to any action change or disturbance which freezes the system. Figure 4.4 illustrates a scenario of unbounded disturbance. It presents the resulting thresholds L_0^* and U_0^* and, the associated tolerance thresholds. One can clearly see that the cluster becomes insensitive to any received utility value.

In order to avoid the algorithm from being frozen in this case, one needs to release the constraint to set the bounds L_0^* and U_0^* and to allow a few quantity of the disturbances to be outside the interval $[\bar{u} - \delta^-, \bar{u} + \delta^+]$ (4.11). The idea is that if the quantity of utility values allowed outside the interval is small enough, the algorithm is going to behave as there were no disturbance in addition to be sensitive to action changes in the network. Therefore, one needs to set different targeted positive tolerance thresholds $\delta^- = \delta^{-*}$ and $\delta^+ = \delta^{+*}$ where

$$\delta^{-*} := \bar{u} - L^*, \tag{4.14}$$

$$\delta^{+*} := U^* - \bar{u}, \tag{4.15}$$

where bounds L^* and U^* are defined such that the utility can vary a small amount of time outside $[\bar{u} - \delta^-, \bar{u} + \delta^+] = [L^*, U^*]$. The tolerance thresholds δ^- and δ^+ are computed with (4.14) and (4.15) with L^* and U^* defined as follows

$$L^* := \arg\max_x \left\{ F_u(x) \leq \frac{R_1}{2} \right\}, \tag{4.16}$$

and

$$U^* := \arg\min_x \left\{ F_u^c(x) \leq \frac{R_1}{2} \right\}, \tag{4.17}$$

where $R_1 > 0$ is the small proportion of utility values allowed outside the set $[L^*, U^*]$, $F_u(x) = \Pr\{u \leq x\}$ and $(F_u^c(x) = \Pr\{u \geq x\})$ is the probability of u to be less (greater) than x computed with the pdf of u noted f_u in the sequel. Therefore, as u takes discrete values in a set $[x_0, x_1, \ldots, x_{|\mathcal{L}|}]$, we compute the cumulative functions $F_u(x) = \Pr\{u \leq x\}$ and $F_u^c(x) = \Pr\{u \geq x\}$ as follows

$$F_u(x) := \sum_{x_i \leq x} f_u(x_i), \tag{4.18}$$

and

$$F_u^c(x) := \sum_{x_i \geq x} f_u(x_i), \tag{4.19}$$

Knowing the pdf f_u, it is possible to compute $F_u(.)$ $(F_u^c(.))$ and hence the two targeted bounds L^* and U^* that are necessary to estimate the thresholds. The above procedure is illustrated in Figure 4.5. The dashed section of the utility distribution is the small part of disturbed utility (R_1 percent here) allowed outside the set of utilities $[L^*, U^*]$ from which the cluster is insensitive to. Consequently, the bounds $L^* > 0$ and/or $U^* < 1$ and the cluster is sensitive to action changes when the utility value falls in $[0, L^*]$ or $[U^*, 1]$. It is interesting to note that if $R_1 = 0$, we go back to the problem of threshold selection illustrated in Figure 4.4 which is that the cluster becomes insensitive to any utility variation. On the other side, if $R_1 = 1$, then $L^* = U^*$ and $\delta^{*-} + \delta^{*+} = 0$ (see the sum of (4.14) and (4.15)). It means that the cluster becomes sensitive to almost all utility variations as in the TEL and we do not take advantage of the proposed modification. Besides, the convergence of the algorithm is not satisfying as stated in Proposition 4.1. Therefore, the setting of R_1 relies on a trade-off such that a cluster is robust to disturbances, $R_1 < 1$, and sensitive to action changes, $R_1 > 0$. If there is no action change, R_1 defines the probability for a cluster to leave the content state due to disturbances. Hence, with a probability proportional to R_1^2 the cluster becomes discontent and perturbs the network by experimenting randomly. If the perturbation $\epsilon > 0$, there is a high probability that this cluster goes back to content in few iterations (it is proportional to $\epsilon^{1/2K} \gg 0$) so even if a few clusters change their state to discontent

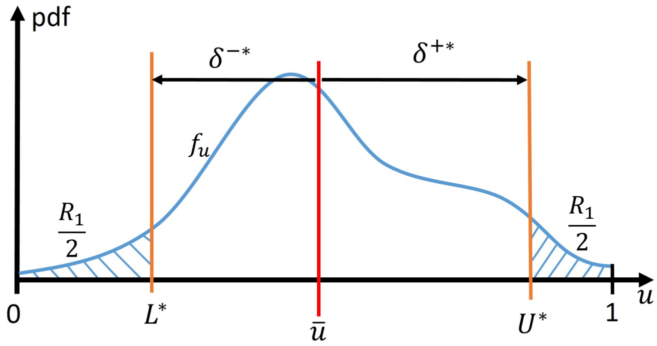


Figure 4.5: Solution for the selection of thresholds in case of unbounded disturbance.

the network is not destabilized that much due to the fast transit from discontent to content mood. If R_1 is too high, the number of discontent cluster rises due to their increase sensitivity to disturbances (*i.e.* the sensitive range $[0, L^*]$ and $[U^*, 1]$ has increased). Thus, the idea to set R_1 is to make it close to 0 and positive such that the probability to get simultaneously many discontent clusters is low in order to keep the network stable, while being sensitive enough to action changes.

Note that when $\bar{u} \notin [L^*, U^*]$ we can use the same modifications proposed in the bounded case in order for the cluster to update its benchmark utility.

The utility u is the sum of independent Bernoulli rvs ℓ_i whose parameters are p_i (see Section 5.1.1 in page 100 for details). When the parameters p_i are known for all users, one can compute the density f_u and then obtain the tolerance thresholds using (4.14) and (4.15) with (4.16) and (4.17). Unfortunately, the parameters p_i cannot be known exactly since their estimation relies on a finite set of observations (*e.g.* ℓ_i values received at each algorithm iteration). It follows that the pdf of u can only be approximated and so the bounds L^* and U^* which implies that $\delta^- \neq \delta^{-*}$ and $\delta^+ \neq \delta^{+*}$. To tackle this problem, a first idea consists in using the maximum likelihood estimate of p_i to construct an estimation of f_u, denoted $\hat{f}_u$. With this later estimate, one finds L and U the estimates of L^* and U^* using (4.16) and (4.17) respectively by replacing $F_u(.)$ and $F_u^c(.)$ with $F_{\hat{u}}(.)$ and $F_{\hat{u}}^c(.)$ computed with $\hat{f}_u$.

Figure 4.6 presents a realization of the previous procedure. The main issue of this method is that for low number of observations, the estimated p_i denoted by $\hat{p}_i$ has a large estimation error. An example of this problem is illustrated in Figure 4.7 where the red curve is one given value of p_i and the blue curve is its maximum likelihood estimate. One can observe that at low number of observations (less than 20), p_i is largely overestimated. In addition, even for more than 20 observation an estimation error regularly arises.

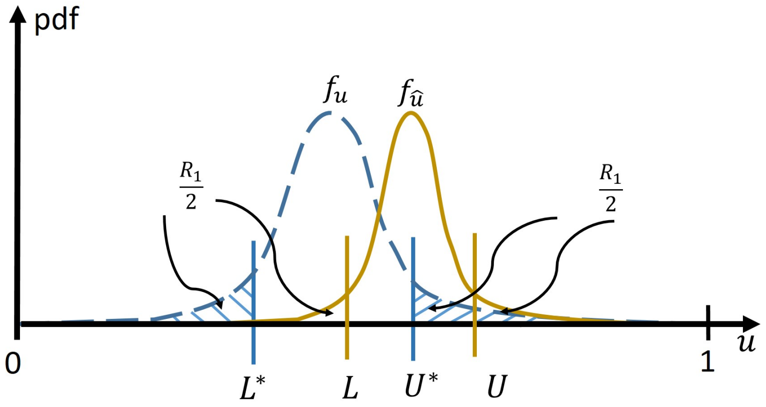


Figure 4.6: Estimation of the pdf bounds based on $f_{\hat{u}}$.

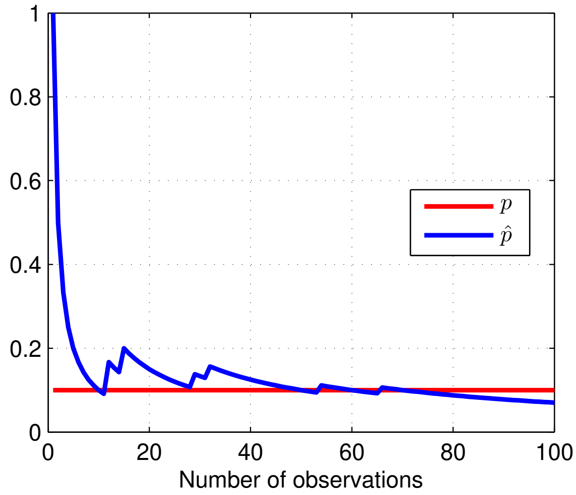


Figure 4.7: Estimation of the Bernoulli parameter using the maximum of the likelihood.

Consequently, the pdf $f_{\hat{u}}$ is a coarse estimate of f_u as presented in the example in Figure 4.6. Therefore, this often results in wrong tolerance thresholds estimations as presented in Figure 4.8. For clarity, f_u is withdrawn and the bounds L^* are U^* are kept. The targeted tolerance thresholds are computed using (4.14) and (4.15) but recall that they are not available because f_u is not known to the cluster. Instead we compute estimated tolerance thresholds δ^- and δ^+ similarly by replacing L^* with L and U^* with U in (4.14) and (4.15). One can see in this example that $\delta^- < \delta^{-*}$ which makes a cluster too sensitive to disturbances as more than R_1 percent of utility values impact transitions in the FSC. These changes lead to more experimentations which in turn perturb other clusters and the overall network in the end. More generally, this highlights the difficulty to estimate robust tolerance thresholds (*i.e.* $\delta^- \geq \delta^{-*}$ and $\delta^+ \geq \delta^{+*}$) with one estimation of f_u. Figure 4.9 presents

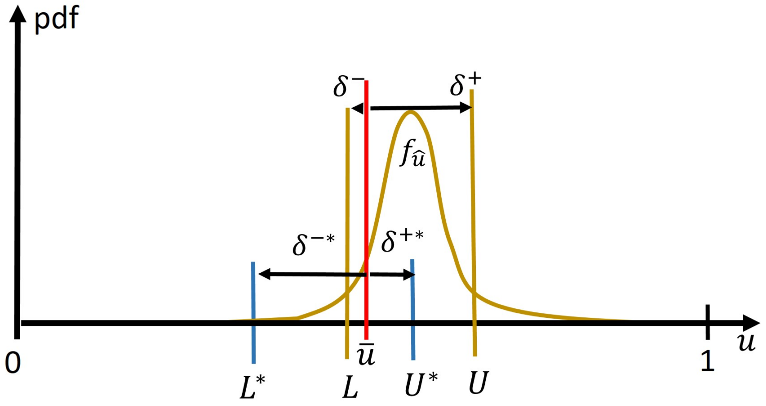


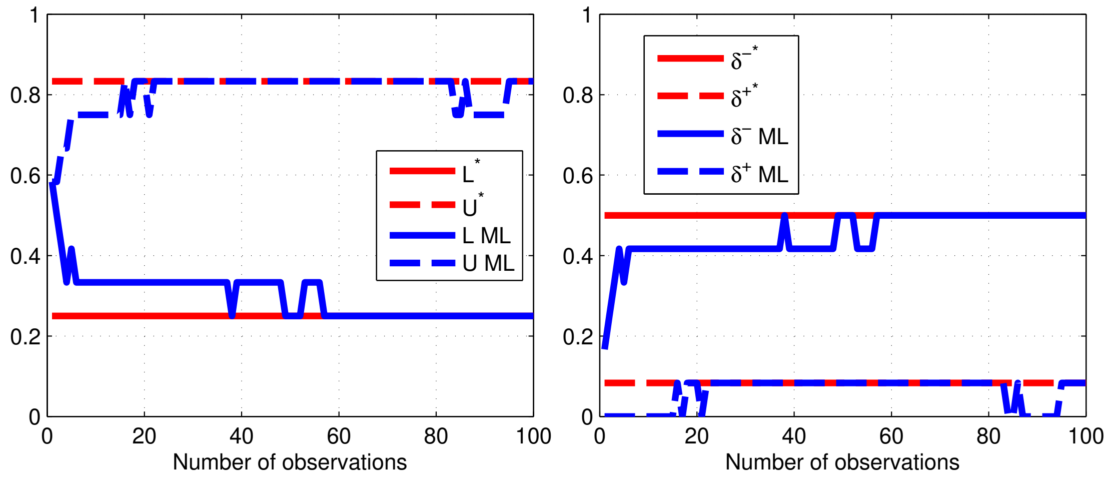
Figure 4.8: Estimation of the tolerance thresholds based on $f_{\hat{u}}$.

on the left the estimation of bounds L and U using the maximum likelihood method noted "ML" and on the right the estimation of tolerance thresholds based on L and U. The red curves are the targeted value that are not known to the cluster and the blue curves are the estimated values. We model a Poisson Binomial rv with 12 links with their respective Bernoulli parameters uniformly drawn. One can observe in Figure 4.9a the wrong estimation of bounds at low number of observations. More specifically, the problem is that L^* is overestimated and U^* is underestimated. The opposite (*e.g.* L^* is underestimated) would not have been a problem as one will see in the sequel. Thus, on the right in Figure 4.9b both thresholds are underestimated at low number of observations whereas we would have preferred these two thresholds to be overestimated (*i.e.* $\delta^+ \geq \delta^{+*}$ and $\delta^- \geq \delta^{-*}$) in order for the cluster to be robust to disturbances. For instance, $\delta^+ = 0$ under 20 observations whereas it must be at least $\delta^{+*} = 0.1$ to ensure the cluster from not being perturbed by more than R_1 percent of disturbances. The consequences on the network performance of cluster sensitivity to disturbances are highlighted in Section 4.6.2.

In this thesis, we thus propose a different solution such that, instead of estimating p_i, we estimate an interval $[p_i^-, p_i^+]$ that contains p_i. Let f_{u^-} and f_{u^+} be the pdf obtained with all p_i replaced with p_i^- and p_i^+ respectively (then $p_i^- \leq p_i \leq p_i^+$). Let also $F_{u^-}(.)$ and $F^c_{u^+}(.)$ be the distribution functions computed as in (4.18) and (4.19) but with $f_{u^-}(.)$ and $f_{u^+}(.)$ respectively. We conjecture the following result.

Conjecture 1. *If for all $i \in \mathcal{L}$, $p_i \geq p_i^-$, then $\forall x$, $F_{u^-}(x) \geq F_u(x)$ and $F^c_u(x) \geq F^c_{u^-}(x)$.*

This result can be proved in a Binomial case where $\forall i$, $p_i = p$ but it remains an open question when dealing with a Poisson Binomial distribution. Based on this conjecture, Figure 4.10 presents the three pdfs f_{u^-}, f_u and f_{u^+} and the bounds L^*



(a) Bounds L and U

(b) Tolerance thresholds

Figure 4.9: Illustration of the error estimation using a maximum likelihood procedure to compute the tolerance thresholds.

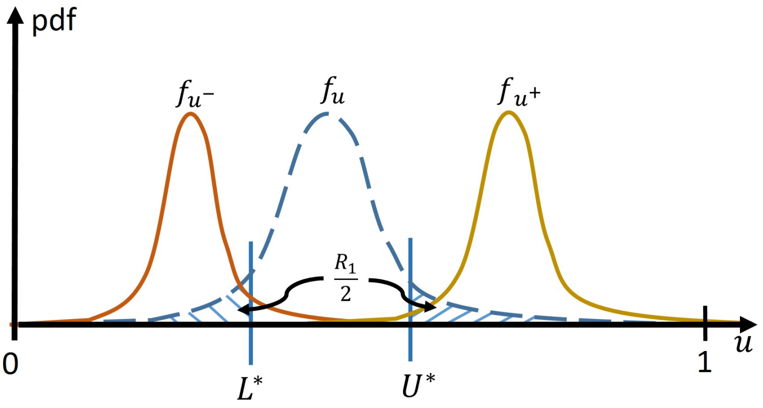


Figure 4.10: Representation of the pdfs f_{u^-} and f_{u^+} based on the Conjecture 1.

and U^* computed using f_u. The pdf f_u is dashed because in practice we cannot compute it. The pdf f_{u^-} provides a cumulative function F_{u^-} that is greater than F_u. Similarly, the pdf f_{u^+} provides a complementary cumulative function $F^c_{u^+}$ that is greater than F^c_u.

From these observations, we deduce that the estimated bounds L and U of L^* and U^* are computed based on f_{u^-} and f_{u^+} as illustrated in Figure 4.11. For clarity, the pdf f_u is withdrawn and the bounds L^* and U^* are kept. The arrows mean that when the number of observations increases both pdfs tends to f_u, hence the bounds L and U respectively tend to L^* and U^*.

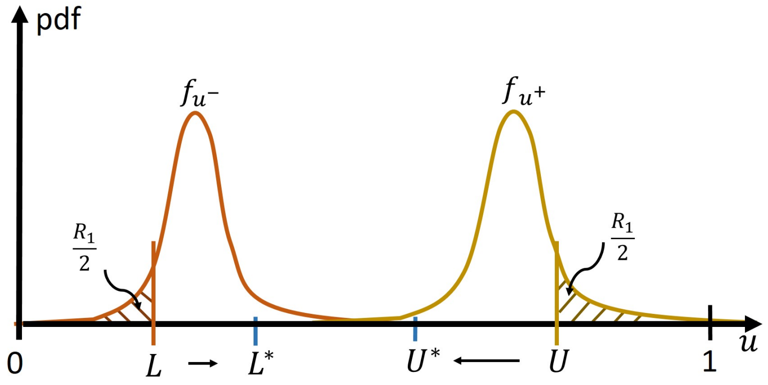


Figure 4.11: Bounds selection based on pdfs f_{u^-} and f_{u^+}.

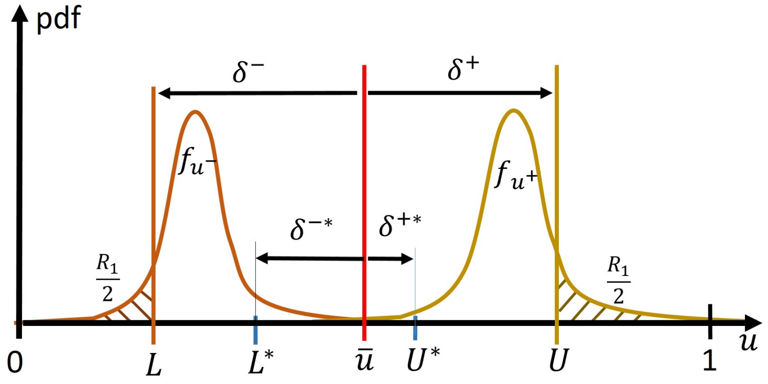


Figure 4.12: Tolerance thresholds selection based on pdfs f_{u^-} and f_{u^+}.

More Formally, we compute L and U using $F_{u^-}(.)$ and $F^c_{u^+}(.)$ as follows

$$L = \arg\max_x \left\{ F_{u^-}(x) \leq \frac{R_1}{2} \right\}, \tag{4.20}$$

$$U = \arg\min_x \left\{ F^c_{u^+}(x) \leq \frac{R_1}{2} \right\}, \tag{4.21}$$

where with Conjecture 1, $L \leq L^*$ and $U \geq U^*$. With the bounds L and U it is possible to compute the tolerance thresholds δ^- and δ^+ as illustrated in Figure 4.12. The targeted tolerance thresholds δ^{-*} and δ^{+*} are computed using (4.14) and (4.15) respectively. However, we know that these values are not available because we cannot compute f_u. Therefore, a similar procedure is employed to compute δ^- and δ^+ that is based on L and U. In Figure 4.12, one can see that $\delta^+ \geq \delta^{+*}$ and that $\delta^- \geq \delta^{-*}$ hence, clusters do not react untimely to utility variations. Thus, the estimated tolerance thresholds δ^- and δ^+ are given by

$$\delta^- = \bar{u} - L, \tag{4.22}$$

$$\delta^{+} = U - \bar{u}, \tag{4.23}$$

and do not underestimate the targeted tolerance thresholds δ^{-*} and δ^{+*} given by (4.12) and (4.13) respectively (*i.e.* $\delta^{-} \geq \delta^{-*}$ and $\delta^{+} \geq \delta^{+*}$). Note that to deal with the case in which $\bar{u} \notin [L, U]$ we use the same modifications as proposed in the bounded and unbounded case which consists in keeping the thresholds positives and such that $\delta^{+} + \delta^{-} \leq U - L$.

However, remind that the interval $[p_i^-, p_i^+]$ is also computed using a finite number of observations so there is a probability such that $\exists i, p_i \leq p_i^-$ or that $p_i \geq p_i^+$. In such case, the hypothesis of Conjecture 1 is not verified and it is possible that $F_{u^-}(x) \leq F_u(x)$ or $F_u^c(x) \leq F_{u^-}^c(x)$. Consequently, the estimated tolerance thresholds underestimate the targeted thresholds which results in a cluster that becomes too sensitive to disturbances.

In this thesis, we estimate the confidence interval $[p_i^-, p_i^+]$ of p_i with a Bayesian method described afterwards. The reason to select a Bayesian procedure is that confidence interval obtained by directly inverting the binomial distribution is overly conservative [48]. In addition, other confidence interval approximations just as the adjusted Wald or Score intervals [48] while better than the exact interval, either perform badly at low number of sample or do not provide the requested coverage in average. We choose a Bayesian approach for its simplicity to update through time and its performance for any value of the Bernoulli parameter [49]. This approach is described as follows.

The confidence interval $C_i(R_2) = [p_i^-, p_i^+]$ contains the parameter p_i with a confidence $(1 - R_2)$, *i.e.* R_2 is the probability that $p_i \notin C_i(R_2)$. This interval is computed with the posterior distribution of p_i. The Bayesian approach relies on the good choice of a prior for p_i, noted $p_{p_i}(.)$. Let a $\mathbf{x} = (x_1, \ldots, x_M)$ be the M measurements of a Bernoulli variable with the parameter p_i. It follows that $y_M = \sum_{\ell=1}^{M} x_\ell$ is a binomial random variable with parameters M and Mp_i. The likelihood of observing $y_M = k$ is

$$p(y_M = k | p_i) = \binom{M}{k} (p_i)^k (1 - p_i)^{M-k}. \tag{4.24}$$

It is important to note that $p(y_M = k|p_i)$ is a beta distribution, so $p(y_M = k|p_i) = \mathrm{Beta}(p_i, k+1, M-k+1)$ [50] (26.1.33 page 930) where

$$\mathrm{Beta}(p, a, b) = \frac{1}{\mathrm{B}(a, b)} p^{a-1} (1 - p)^{b-1}, \tag{4.25}$$

where $\mathrm{B}(a, b)$ is the beta function ([50] 6.2.1). It is practical to employ a prior that leads to a posterior that is easy to deal with. In this case, a prior that follows a beta law

$$p_{p_i}(x) \propto (x)^{\alpha-1}(1-x)^{\beta-1}. \tag{4.26}$$

leads to a posterior that is also a beta law as follows [51]

$$p(p_i|k, M) = \mathrm{Beta}(p_i, k+\alpha, M-k+\beta) \tag{4.27}$$

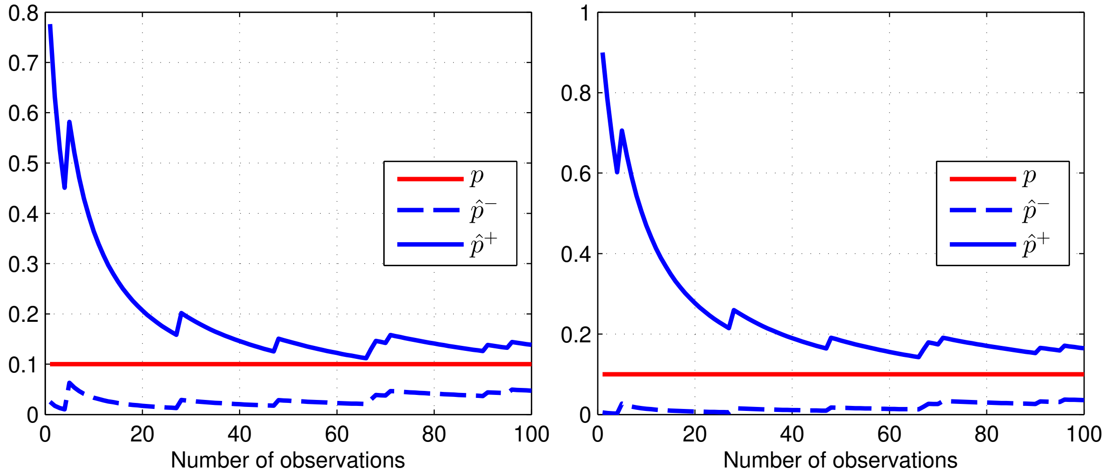
If there is no information on p_i, the prior can be chosen such that it is a uniform distribution, *i.e.* $p_{p_i}(x) = \mathrm{Beta}(x, 1, 1)$. However, with this procedure we have observed in the simulations, that the convergence speed of thresholds is slow when the Bernoulli parameter is either close to 0 or 1. This leads us to use instead a Jeffrey's prior which is defined by $p_{p_i}(x) = \mathrm{Beta}(x, 1/2, 1/2)$ [51] (*i.e.* $\alpha = \beta = 1/2$). This prior distribution increases slightly the probability for p_i to be close to 0 or 1. Consequently, the convergence speed of the tolerance interval estimation is increased when p_i is close to 0 or 1. An equal-tailed confidence interval for p_i with a confidence $(1-R_2)$ is obtained by solving the following two equations

$$\int_0^{p_i^-} \mathrm{Beta}(p, k+1/2, M-k+1/2)dp = \frac{R_2}{2}, \tag{4.28}$$

$$\int_{p_i^+}^{1} \mathrm{Beta}(p, k+1/2, M-k+1/2)dp = \frac{R_2}{2}. \tag{4.29}$$

It is equivalent to invert the incomplete beta function ([50] 26.5.1) in $\frac{R_2}{2}$ and $1-\frac{R_2}{2}$ and this provides p_i^- and p_i^+ respectively. Figure 4.13 presents the result of the above procedure to estimate interval that contains the Bernoulli parameter with a confidence $1-R_2$. The red curve is the Bernoulli parameter to find, the dashed blue curve is p_i^- obtained after solving (4.28) and the plain blue curve is p_i^+ computed using (4.29). The difference between Figures 4.13b and 4.13a is a change in R_2. Note that in both case the parameter p_i is well bounded. The parameter R_2 can be seen as the risk that $p_i \notin [p_i^-, p_i^+]$. Thus one can observe that in Figure 4.13a the interval defined by $[p_i^-, p_i^+]$ seems narrower than the one in Figure 4.13b. The reason is that, the greater R_2, the narrower the interval $[p_i^-, p_i^+]$ and, the more probable to not bound p_i. This interval converges faster as R_2 increases but at the expense of accuracy to bound p_i which in our case is very important.

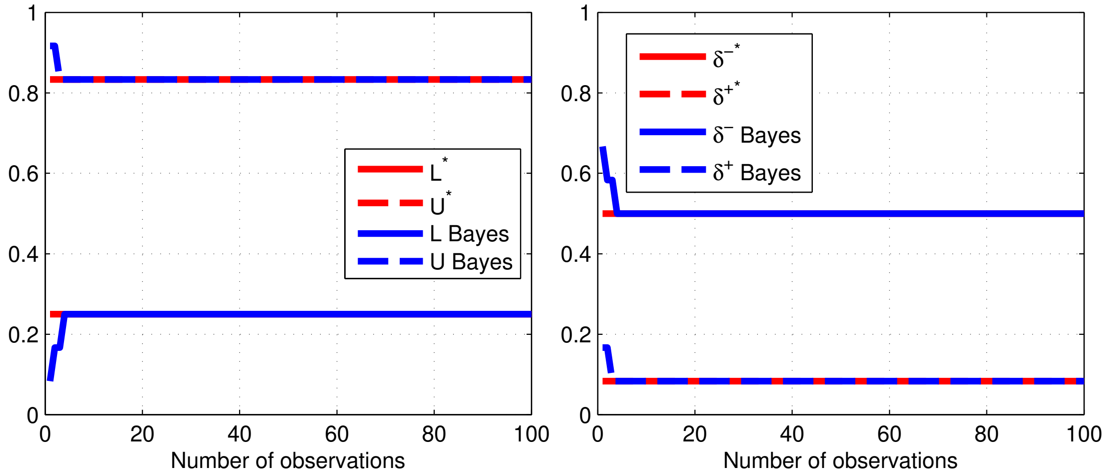
Finally, with all p_i^- and p_i^+ one computes $F_{u^-}(.)$ and $F_{u^+}^c(.)$ to get L and U with (4.20) and (4.21) respectively. Figure 4.14a presents a realization of this procedure on a specific example with 12 links whose Bernoulli parameters are uniformly drawn. In comparison to the maximum likelihood estimation presented in Figure 4.9a one can see that the bounds $L \leq L^*$ and $U \geq U^*$ which ensures a robust estimate of tolerance thresholds. Then, it remains to compute δ^- and δ^+ with (4.22)



(a) $R_2 = 5\%$

(b) $R_2 = 1\%$

Figure 4.13: Illustration of the interval $[p_i^-, p_i^+]$ estimation using a Bayes method.



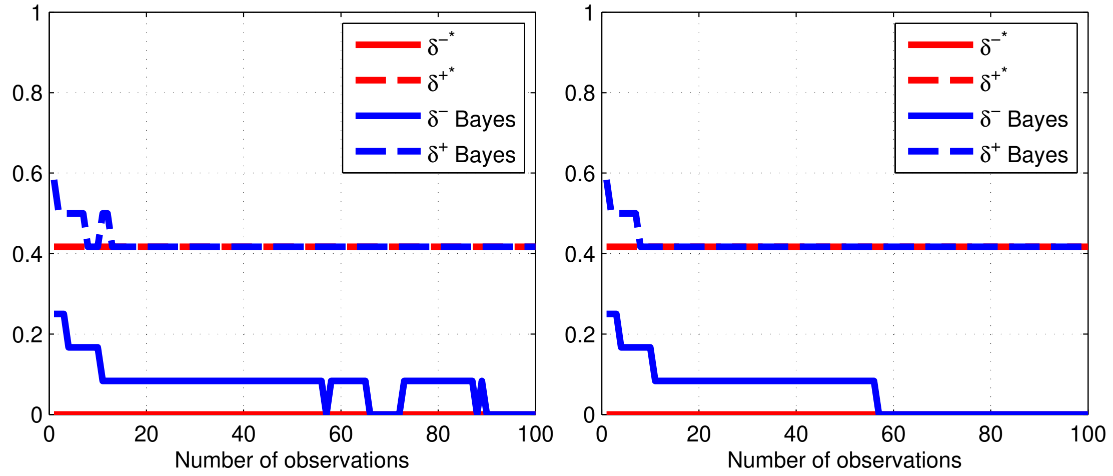
(a) Bounds L and U

(b) Tolerance thresholds

Figure 4.14: Illustration of the bounds L and U and the tolerance estimations using a Bayes method.

and (4.23). This procedure is illustrated in Figure 4.14b in which we observe the desired behavior of robust tolerance thresholds, *i.e.* $\delta^- \geq \delta^{-*}$ and $\delta^+ \geq \delta^{+*}$. In addition, in this specific realization one can note the better convergence speed of the proposed procedure in comparison to the maximum likelihood one in Figure 4.9b.

Figure 4.15a presents a second example of tolerance threshold estimation but for a second draw of Bernoulli parameters. One can notice that the thresholds sometimes increase again. This behavior can have severe consequences on the



(a) Without the solution

(b) With the proposed solution

Figure 4.15: A solution to tolerance thresholds jumps.

stability. For instance, a cluster can increase the thresholds such that, in a worst case, it becomes progressively insensitive to a disturbance whose deviation increases due to network action changes. This would hence prevent the cluster from changing its benchmark whereas the network state has changed. One of the solution is to not allow the thresholds to increase once the estimation process has started as presented in Figure 4.15b. With the proposed solution, the tolerance thresholds converge to the targeted value without jumping untimely.

Finally, the procedure to adapt the tolerance thresholds on the fly is presented in Algorithm 2 and summarized as follows. Let note $\boldsymbol{\ell}_i := (\ell_i(1), \ldots, \ell_i(M))$ the M measurements of the i working link metric which are used to estimate the thresholds. The $\ell_i(.)$ are Bernoulli rvs with parameter p_i which is not known to the cluster. The first step is to compute the pdfs f_{u^-} and f_{u^+} with the Bernoulli parameter p_i^- and p_i^+ obtained after bounding p_i using a $1 - R_2$ confidence interval and measurements. With these densities, one can compute a lower bound L of L^* and an upper bound U of U^* using (4.20) and (4.21). These bounds are used to obtain the thresholds δ_0^+ and δ_0^- but it remains several steps before getting the estimated thresholds δ^- and δ^+. The first step is to keep the thresholds positive and less than $U - L$ in order for the cluster to change its benchmark utility when it is outdated as illustrated in Figure 4.3. The second and final step is to decrease the thresholds if the new computed value is smaller than the previous one only when the number of measurement $M > 1$. If $M = 1$ the tolerance thresholds are initialized with the computed value.

Algorithm 2 Compute the thresholds δ^- and δ^+

```
Input: $R_1, R_2, \bar{u}, |\mathcal{L}|, \forall i \in \mathcal{L}, \boldsymbol{\ell}_i = (\ell_i(1), \ldots, \ell_i(M)), \delta^-(t), \delta^+(t)$
Output: $\delta^-(t+1), \delta^+(t+1)$
for $i = 1$ to $|\mathcal{L}|$ do
    $p_i^- \leftarrow$ solve (4.28) using $R_2$ and, $k = \sum_{j=1}^{M} \ell_i(j)$,
    $p_i^+ \leftarrow$ solve (4.29) using $R_2$ and, $k = \sum_{j=1}^{M} \ell_i(j)$,
end for
$f_{u-}(.) \leftarrow$ this pdf is computed using convolution of the pdf of Binomial variables with parameter $p_i^-$ (e.g. a convolution method is given in [52]),
$f_{u+}(.) \leftarrow$ as in the previous line but with $p_i^-$ replaced with $p_i^+$
$F_{u^-}(.) \leftarrow$ computed using (4.18) with $f_u(.)$ replaced by $f_{u-}(.)$,
$F^c_{u^+}(.) \leftarrow$ computed using (4.19) with $f_u(.)$ replaced by $f_{u+}(.)$,
$L \leftarrow$ use (4.20) with $R_1$,
$U \leftarrow$ use (4.21) with $R_1$,
$\delta_0^- = \bar{u} - L$ as in (4.22),
$\delta_0^+ = U - \bar{u}$ as in (4.23),
$\delta_2^- \leftarrow \min\{\delta_1^-, U - L\}$ with $\delta_1^- \leftarrow \max\{0, \delta_0^-\}$, this condition is to avoid $\bar{u}$ from being outside $[L, U]$,
$\delta_2^+ \leftarrow \min\{\delta_1^+, U - L\}$ with $\delta_1^+ \leftarrow \max\{0, \delta_0^+\}$, this condition is to avoid $\bar{u}$ from being outside $[L, U]$,
if $M > 1$ then {If there are more than one measurement to process the estimation}
    $\delta^-(t+1) \leftarrow \min\{\delta^-(t), \delta_2^-\}$, this line and the next one avoid jumps in thresholds estimations as illustrated in Figure 4.15
    $\delta^+(t+1) \leftarrow \min\{\delta^+(t), \delta_2^+\}$,
else {If there are only one measurement the thresholds are initialized}
    $\delta^-(t+1) \leftarrow \delta_2^-$,
    $\delta^+(t+1) \leftarrow \delta_2^+$,
end if
return $\delta^-(t+1), \delta^+(t+1)$.
```

4.5.2 General case

The estimation process of tolerance thresholds described above must be based on stationary observations to converge without bias. That is why we have fixed the action vector **a**. In practice, the action vector is time varying since clusters experiment new actions, and they react to network changes (*e.g.* clusters mobility that modify the interference and hence the utility, experimentations from other clusters) by modifying their own action. This section presents a procedure to keep only stationary observations so as to update the thresholds. Note that as long as cluster do not move and do not experiment the received measurements are assumed to be stationary. In our case, the observations are the one bit feedback from each link $i \in \mathcal{L}$, $(\ell_i(1), \ldots, \ell_i(M))$, where M is the number measurements available. In the following, we consider that these observations are stored in a memory, denoted by $\boldsymbol{\ell}_i(t) = (\ell_i(1), \ldots, \ell_i(M))$ at time t for link i. This memory provides the input for the estimation procedure presented in Algorithm 2 in the previous section. Thus, the difficulty is to fill this memory with only stationary observations. For instance, if a cluster experiments a new action, the received ℓ_i for all i are not stationary with previous measurements as the network action has changed. Thus, the received measurement do not have to be taken into account in the thresholds estimation process and it is not added to the memory $\boldsymbol{\ell}_i(t+1)$.

The procedure to fill $\boldsymbol{\ell}_i(t)$ at time t is presented in Algorithm 3 and is independent of the RODL or RTEL algorithms. We consider three modes to update the memory: *(i)* add the received information to the memory, *(ii)* keep the memory unchanged, *(iii)* reset the memory and add the current received measurement to it. The reasoning contains steps based on each algorithm state controller that are summarized as follows. Let suppose initially that the memory of the cluster k, $\boldsymbol{\ell}_i(t)$, is filled with stationary information.

The first step consists in testing if the cluster experiments a new action or not. If it experiments, the state of the network changes and the received information, ℓ_i for all i, is not stationary with respect to the information stored in $\boldsymbol{\ell}_i(t)$ and must not be added to it (*i.e.* either update mode *(ii)* or *(iii)* will be selected depending on further conditions described in the sequel). If it does not experiment, the cluster needs to check if the received information is stationary with what has been stored into $\boldsymbol{\ell}_i(t)$. In this case, the cluster cannot know if the network state has changed (it is not aware of other clusters' actions) but, if $u \in [\bar{u} - \delta^-, \bar{u} + \delta^+]$ the cluster considers that the network state has not changed. Therefore, from a cluster point of view, the received ℓ_i are stationary with respect to the information contained in $\boldsymbol{\ell}_i(t)$. This information is stored which is noted $\boldsymbol{\ell}_i(t+1) = (\ell_i(1), \ldots, \ell_i(M), \ell_i(M+1))$

with $\ell_i(M+1) = \ell_i$ (*i.e.* update mode *(i)*).

If the cluster experiments or, if it does not experiment but $u \notin [\bar{u} - \delta^-, \bar{u} + \delta^+]$, then the observation received, ℓ_i, are not stationary with respect to the information in $\boldsymbol{\ell}_i(t)$ (*i.e.* either update mode *(ii)* or *(iii)* will be selected). In this scenario there are two possibilities, either the benchmark is updated, noted case *(a)* or it is not, noted case *(b)*. In case *(a)*, the new state is the future benchmark therefore the information stored in $\boldsymbol{\ell}_i(t)$ is outdated as it corresponds to the observations related to the former state. Consequently, the memory $\boldsymbol{\ell}_i(t+1)$ is reset and the new information is stored which is noted $\boldsymbol{\ell}_i(t+1) = (\ell_i(1))$ with $\ell_i(1) = \ell_i$ (*i.e.* update mode *(iii)*). In case *(b)*, the observation received is not stationary with respect to $\boldsymbol{\ell}_i(t)$ an it cannot be added to it. In addition, the state has not necessarily changed (*i.e.* no benchmark update) so the memory cannot be refreshed that is why, we keep it unchanged (*i.e.* update mode *(ii)*).

It is worth mentioning that the memory $\boldsymbol{\ell}_i(t)$ can be fulfilled with any type of observations using this procedure as long as either RTEL or the RODL are employed. However, in our case we are specifically interested in the bits feedback ℓ_i from each link.

4.6 Numerical results

This section presents the numerical results related to the robust TE algorithms presented in Section 4.3, namely the RTEL and RODL (page 55). It also presents the numerical results associated to the enhancement proposed in Section 4.5 that are related to the capacity for each cluster to adapt online their tolerance thresholds. In addition, we provide numerical results to validate in average the performance of the adaptive RTEL. More precisely, the proposed algorithms are random in the sense that the choice of channel is governed by a probability distribution. In this sense, the performance of algorithm varies from one realization to the other. It is of much importance to study the average performance of the proposed modifications.

Section 4.6.1 presents the different disturbance models considered. Section 4.6.2 provides results to valid, in case of fading, the assumption on the bounded distribution of the utility (see Section 4.2). The second step of this section is to sustain the Proposition 4.1 that presents the impact of the stochastic process on TEL and ODL algorithms. Then, in Section 4.6.3, we present results involving the proposed robust algorithms where it is shown that their performance coincide with the non modified algorithms in a deterministic environment. Section 4.6.4, presents re-

Algorithm 3 Update of memory $\mathbf{s}_k$

```
Input: $\boldsymbol{\ell}_i(t) = (\ell_i(1), \dots, \ell_i(M))$, $a$, $\bar{a}(t)$, $\bar{a}(t-1)$, $u$, $\bar{u}(t)$, $\bar{u}(t-1)$, $\delta^-$, $\delta^+$, $\forall i\ \ell_i$,
Output: $\boldsymbol{\ell}_i(t+1)$
STEP 1
if $a \neq \bar{a}$ then
    go to STEP 3
else
    go to STEP 2
end if
STEP 2
if $u \in [\bar{u} - \delta^-, \bar{u} + \delta^+]$ then
    $\forall i$, $\boldsymbol{\ell}_i(t+1) = (\ell_i(1), \dots, \ell_i(M), \ell_i(M+1))$ with $\ell_i(M+1) = \ell_i$, the new observation is added to the memory.
    go to END
else
    go to STEP 3
end if
STEP 3
if $(\bar{a}(t), \bar{u}(t)) \neq (\bar{a}(t-1), \bar{u}(t-1))$ (i.e. the benchmark has been updated) then
    $\forall i$, $\boldsymbol{\ell}_i(t+1) = (\ell_i(1))$ with $\ell_i(1) = \ell_i$ (i.e. $M = 1$), the memory is refreshed with the new observations.
    go to END
else
    $\boldsymbol{\ell}_i(t+1) = \boldsymbol{\ell}_i(t)$, nothing happens,
    go to END
end if
END
return $\boldsymbol{\ell}_i(t+1)$
```

sults related to the enhanced adaptive algorithms. Finally, Section 4.6.5 provides thorough results that compare the TEL and the enhanced RTEL.

4.6.1 Clusters communication modelling

The section presents the SINR formulas that will be used in numerical simulations to compute the utility functions. These formulas are derived in Chapter 5 which is dedicated to the study of the utility statistics and the SINR modeling in the Rayleigh fading context for the NB and WB cases. Depending on the context (deterministic/stochastic and types of scheduling), we consider three cases: *(i)* worst case of the deterministic context, *(ii)* worst case of the stochastic context named *configuration 1*, *(iii)* multiple access with random scheduling in the stochastic context named *configuration 2*.

4.6.1.1 Worst case in the deterministic context

In the first case, the utility is deterministic. The SINR formula is given by (3.1) which is a constant (*i.e.* we do not consider averaging errors). As a worst case, we assume that receivers in a cluster are interfered by all nodes belonging to a different clusters on the same channel (*i.e.* the set $\mathcal{M}_n$ in (3.1) contains all nodes of cluster n). Note that this worst case does not exist in practice since transmitters are scheduled to avoid intra cluster interference. With the SINR, one can compute the working link metrics (3.2) and then the utility with (3.3) in page 21. This case is denoted by AWGN.

4.6.1.2 *configuration 1*: worst case fading

In *configuration 1*, we consider a worst case with Rayleigh fading in which we assume that receivers in a cluster are interfered by all nodes belonging to different clusters on the same channel (*i.e.* the set $\mathcal{M}_n$ of transmitting nodes in cluster n is composed of all of its nodes). The SINR formula that takes into account the Rayleigh fading are presented in details in Chapter 5. We consider flat fading (noted NB fading) and frequency selective fading (noted WB fading). The SINR is given by (5.5) in NB (page 101) and (5.14) in WB (page 105). Therefore, the working link metric in (3.2) can be written for link i in cluster k

$$\ell_i^k(\mathbf{a}) = \mathbb{1}_{\{\Gamma_i^k(\mathbf{a})>\Gamma_0\}}, \tag{4.30}$$

with

$$\Gamma_i^k(\mathbf{a}) = \begin{cases} \frac{\alpha_i^k \mathrm{SNR}_i^k}{1+\sum_{n\in\mathcal{I}_k}\sum_{j\in\mathcal{M}_n}\alpha_{j,i_r}^{n,k}\mathrm{INR}_{j,i_r}^{n,k}}, \text{(5.5) in NB fading,} \\ -\beta\log\left(\frac{1}{N_c}\left(\sum_{m=1}^{N_b-1} N_{cc} e^{-\frac{\mathrm{SINR}_i^k(m)}{\beta}} + N'_{cc} e^{-\frac{\mathrm{SINR}_i^k(N_b)}{\beta}}\right)\right), \text{(5.14) in WB fading,} \end{cases} \tag{4.31}$$

where the SINR in each block m of coherent subcarriers is given by the NB SINR in (4.31) with the indices of the block added as follows

$$\mathrm{SINR}_i^k(m) = \frac{\alpha_i^k(m)\mathrm{SNR}_i^k}{1+\sum_{n\in\mathcal{I}_k}\sum_{j\in\mathcal{M}_n}\alpha_{j,i_r}^{n,k}(m)\mathrm{INR}_{j,i_r}^{n,k}}. \tag{4.32}$$

4.6.1.3 *configuration* 2: multiple access with scheduling and fading

In *configuration 2*, we consider Rayleigh fading and that nodes are scheduled in a TDMA scheme with several possibilities to transmit during an iteration of algorithms. An algorithm iteration is divided into Q time slots and, the node that transmits at slot q in cluster k is noted $i_q(k)$. The sequence i_q, for $q = 1, \dots, Q$, is drawn randomly from one iteration to the other such that each node can transmit M times during the iteration. Consequently, we propose a new definition of a working link in this context as follows

$$\ell_i^k(\mathbf{a}) := \mathbb{1}_{\{L_i^k(\mathbf{a})>\Gamma_1\}}, \tag{4.33}$$

where Γ_1 is a threshold that belongs to $[0,1]$ and $L_i^k(\mathbf{a})$ is the ratio of successful transmissions of link i. This ratio is given by

$$L_i^k(\mathbf{a}) := \frac{1}{M}\sum_{q=1}^{Q}\delta_{i_q,i_t}\mathbb{1}_{\left\{\Gamma_{i_q,i_r}^k(\mathbf{a})>\Gamma_0\right\}}, \tag{4.34}$$

where $\delta_{i,j}$ is the dirac function, $\Gamma_{i_q,i_r}^k(\mathbf{a})$ is the SINR of link (i_q, i_r) computed during slot q. One can modify slightly the SINR defined in (4.31) to compute the one in slot q except that the interference depends solely on nodes that transmit on the same resource (*i.e.* $\mathcal{M}_n = \{i_q(n)\}$):

$$\Gamma_{i_q,i_r}^k(\mathbf{a}) = \begin{cases} \frac{\alpha_{i_q,i_r}^k \mathrm{SNR}_{i_q,i_r}^k}{1+\sum_{n\in\mathcal{I}_k}\alpha_{i_q(n),i_r}^{n,k}\mathrm{INR}_{i_q(n),i_r}^{n,k}}, \text{(5.5) in NB fading,} \\ -\beta\log\left(\frac{1}{N_c}\left(\sum_{m=1}^{N_b-1} N_{cc} e^{-\frac{\mathrm{SINR}_{i_q,i_r}^k(m)}{\beta}} + N'_{cc} e^{-\frac{\mathrm{SINR}_{i_q,i_r}^k(N_b)}{\beta}}\right)\right), \text{(5.14) in WB fading,} \end{cases} \tag{4.35}$$

Table 4.1: Simulation parameters values

N	8	k_b	$1.38\ 10^{-23}\ WK^{-1}s$	Γ_0	4.3 dB
P_{Tx}	1 W	T	238 K	n_{it}	15000
B	250 KHz	P_b	$7.9\ 10^{-15}$ W		

where the SINR in each block m of coherent subcarriers is the NB SINR in (4.35) with the indices of the block added as follows

$$\mathrm{SINR}^{k}_{i_q,i_r}(m) = \frac{\alpha^{k}_{i_q,i_r}(m)\mathrm{SNR}^{k}_{i_q,i_r}}{1 + \sum_{n\in\mathcal{I}_k} \alpha^{n,k}_{i_q(n),i_r}(m)\mathrm{INR}^{n,k}_{i_q(n),i_r}}. \tag{4.36}$$

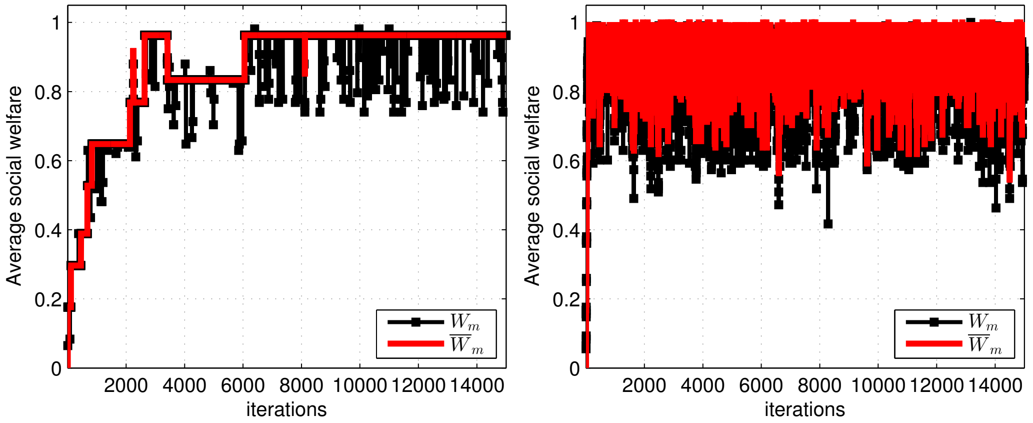
Note that the set $\mathcal{M}_n$ of interfering nodes in cluster n corresponds to the transmitting node on the same resource $i_q(n)$ which is why the sum in (4.35) and in (4.36) is removed in comparison to (4.31) and (4.32).

4.6.2 Disturbed utility and its impact on TE algorithms

This section presents numerical results related to the impact of disturbance on the utility and on the TEL and ODL algorithms. More specifically, we present results that assess Proposition 4.1 (page 57) which states the loss of efficient convergence properties of both algorithms in a disturbed environment.

The ad hoc clustered network used in simulation is given in Figure 3.3. Parameters are contained in Table 4.1. The parameter Γ_0 is set such that the BER is 10^{-2}. Unless specified, experimentation probability in TEL and RTEL is $\epsilon = 10^{-3}$ whereas for ODL and RODL one need a lower value to get similar performance $\epsilon^c = 10^{-4}$ (i.e. $\epsilon = 0.36$). We consider one less channel resource than the number of clusters so the results are expected to be lower than in Section 3.4.2 (page 27) where there are as many cluster as channels. Such a configuration is set to force reuse of frequency channels. In addition to this effect, the fading also decreases the maximum available social welfare.

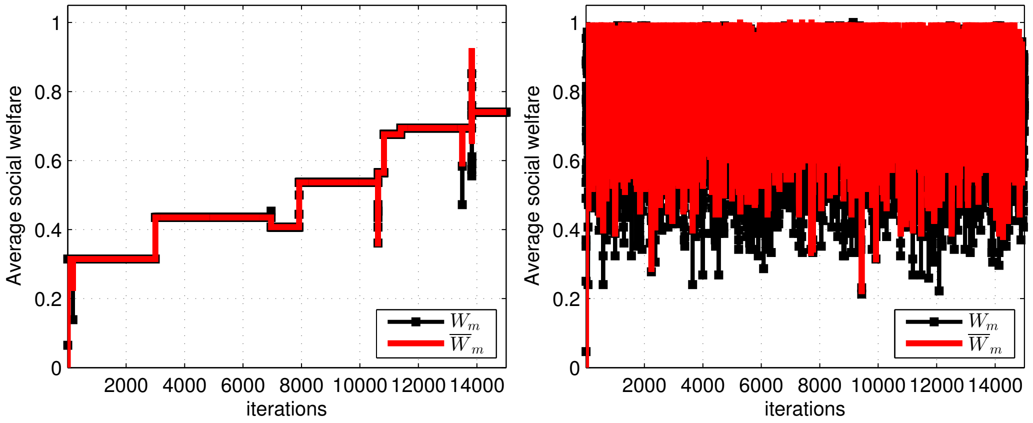
Figures 4.16 present the different average social welfare behavior of TEL in an AWGN channel on the left and with Rayleigh fading on the right. The black curve is the average social welfare (see Definition 2.3 in page 12) noted W_m. It is the normalized sum of all clusters' utilities. The red curve is the average benchmark social welfare (see Definition 3.2 in page 28) noted $\overline{W}_m$. It is the normalized sum of all clusters' benchmark utilities. A stable red curve means that the network does not change its benchmark and is relatively stable. The black curve can be subject to high variations due to experimentations and fading. Clearly, there is a loss of



(a) TEL with AWGN channel.

(b) TEL in *configuration 1*.

Figure 4.16: Impact of disturbances on the average social welfare for the TEL.



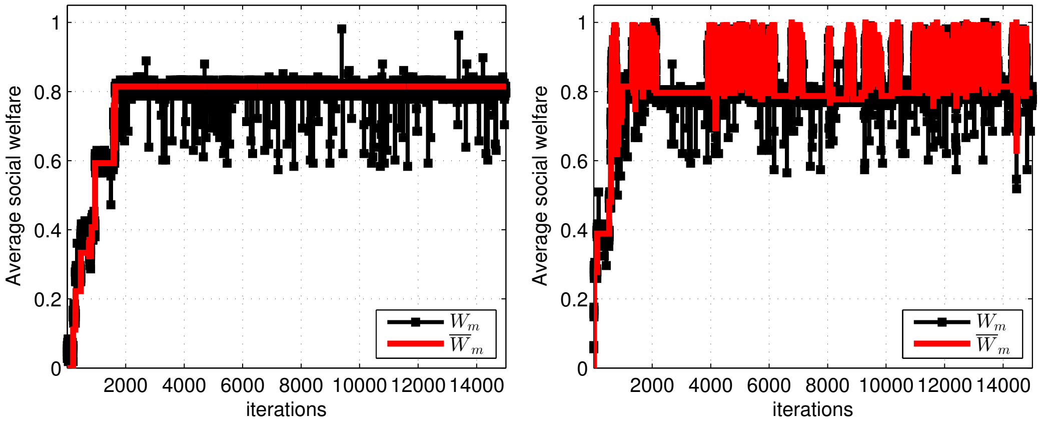
(a) ODL with AWGN channel.

(b) ODL in *configuration 1*.

Figure 4.17: Impact of disturbances on the average social welfare for the ODL.

stability when a stochastic payoff is considered. With TEL, most of the clusters are discontent as explained by Proposition 4.1, so they act randomly which leads to the observed behavior with large outages. However, with the disturbance, the average of W_m over time is good despite outages. The reason is that clusters select channels uniformly when discontent. So they are spread over all available channels which, in average, provides almost good performance.

Figures 4.17 present the same results than in Figures 4.16 but with the ODL algorithm. The conclusion are similar in comparison to the TEL case. One can note that the convergence rate in Figure 4.17a is slower than in Figure 4.16a. In addition, outages in Figure 4.17b are larger than in 4.16b.



(a) Constant tolerance threshold $\delta = 0.25$. (b) Constant tolerance threshold $\delta = 0.15$.

Figure 4.18: Average social welfare for RTEL in *configuration 1*.

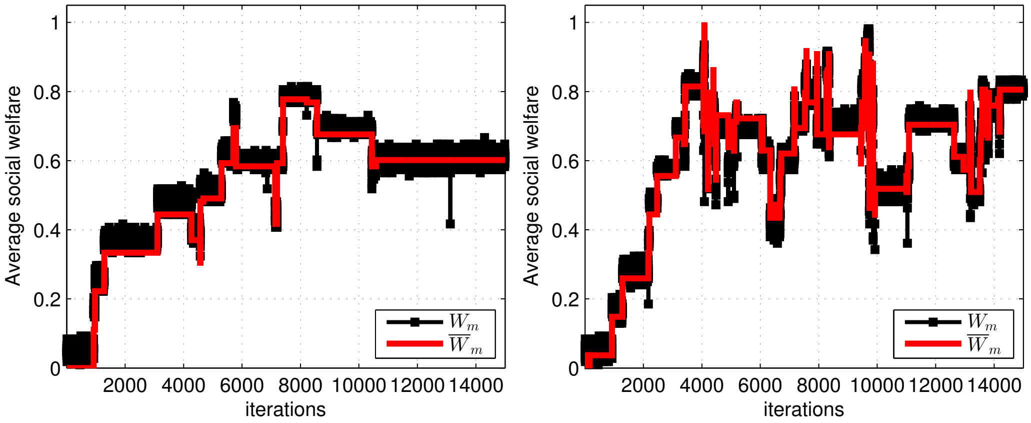
4.6.3 Robust algorithms with fixed tolerance δ

This section presents results related to the RTEL and RODL algorithms described in Section 4.3 page 55. The network used for simulations is the same as the one given in Figure 3.3. Parameters are contained in Table 4.1.

Figures 4.18 present the average social welfare with the RTEL for different threshold values. As in Section 4.6.2, the black curve is the average social welfare noted W_m and the red curve is the average benchmark social welfare noted $\overline{W}_m$. The global decrease in average social welfare when comparing results in Figure 4.16a with results in Figure 4.18a is due to fading. W_m varies a lot because of the fast fading but, one can notice that $\overline{W}_m$ is quite stable as in Figure 4.16a. The tolerance stabilizes the network. On the right, in Figure 4.18b, the tolerance threshold is decreased. The stability also decreases from Figure 4.18a to Figure 4.18b (see $\overline{W}_m$ variations) but, during instabilities, the social welfare is above the stable benchmark social welfare. This highlights that making the algorithm more robust (*i.e.* high tolerance level) can also prevent it from finding a state with a high social welfare. Globally, in these realizations, the modifications improve, in term of social welfare, the behavior of the algorithm in presence of disturbances.

Figures 4.19 present results with RODL algorithm in presence of Rayleigh fading. Note that the tolerance level is increased in comparison to 4.18. The tolerance in RODL also stabilizes the behavior of the network and the $\overline{W}_m$ behaves similarly to the case without disturbance in Figure 4.17a. In Figure 4.19b, the tolerance level is decreased and leads to some instability. However, the network converges faster to an average good social welfare in comparison to Figure 4.19a.

Numerical results presented in Figures 4.18 and 4.19 highlight the following



(a) Constant tolerance threshold $\delta = 0.34$.

(b) Constant tolerance threshold $\delta = 0.25$.

Figure 4.19: Average social welfare for RODL in *configuration 1*.

observations. The RODL seems less stable than the RTEL and this involves the presence of strong outages. In addition, in these realizations, the RODL algorithm has more difficulties than the RTEL to find and remain in a high social welfare state. Generally, when there are disturbances, the presence of a tolerance threshold can stabilize the benchmark social welfare similarly to the case without disturbances. In addition, decreasing the tolerance leads the network to an average good social welfare faster but at the expense of the stability. On the contrary, a large tolerance level can freeze the network in a suboptimal state (*e.g.* see Figures 4.18a or 4.19a). Consequently, the tolerance threshold needs to be set carefully and accurately.

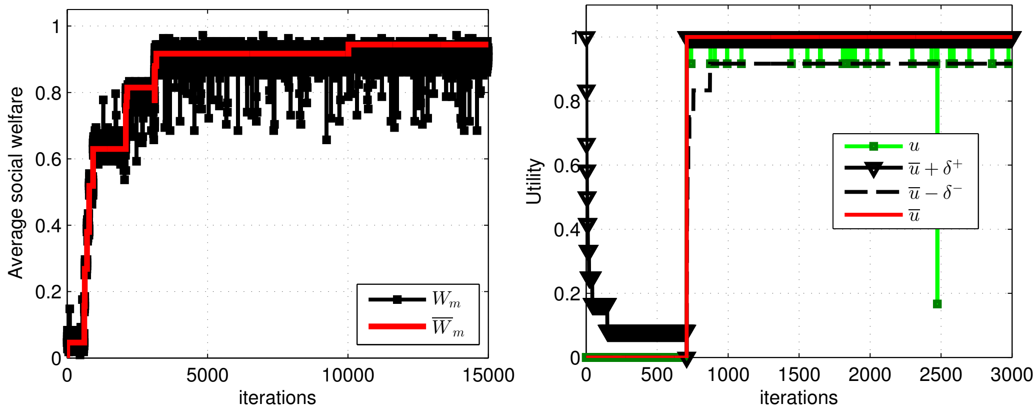
4.6.4 Tolerance adaptation in RTEL and RODL

In this section, we provide numerical results with the enhancement proposed previously. The network is the same as presented in Figure 3.3. The parameters are listed in Table 4.2. The parameter Γ_0 is set such that the BER is 10^{-2}. With $R_1 = 0.01$, the tolerance thresholds should cover at least 99 % of the utility distribution. Parameter R_2 is chosen such that the convergence of the confidence intervals of the Bernoulli parameters are quite fast while the tolerance thresholds estimation is robust (*i.e.* the tolerance thresholds are not too small, see Section 4.5 for details).

Figures 4.20 present results associated with the use of RTEL algorithm and the adaptive thresholds. More specifically, Figure 4.20a presents the average social welfare where the black curve is the average social welfare (see Definition 2.3) noted W_m and the red curve is the average benchmark social welfare (see Definition 3.2) noted $\overline{W}_m$. Figure 4.20b presents, for cluster 6, the utility, the benchmark utility

Table 4.2: Simulation parameters values

N	8	T	238 K	R_1	0.01
P_{Tx}	1 W	P_b	$7.9\ 10^{-15}$ W	R_2	0.4
B	250 KHz	Γ_0	13.4 dB		
k_b	$1.38\ 10^{-23}\ WK^{-1}s$	n_{it}	15000		



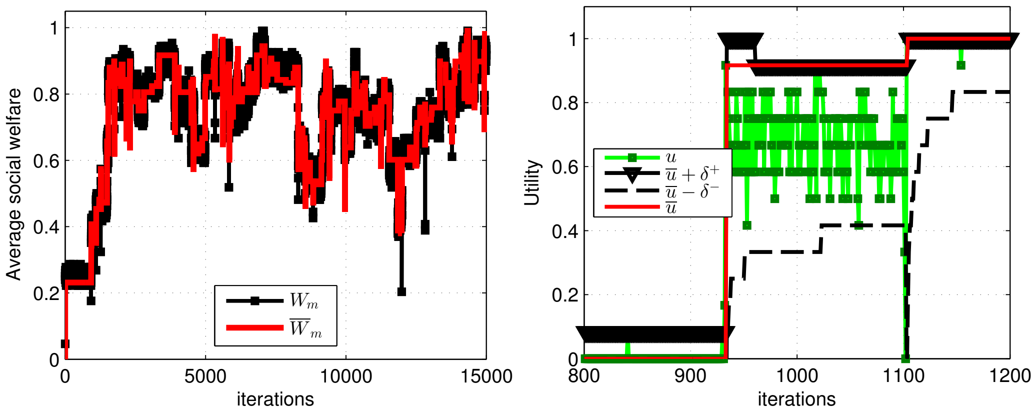
(a) Social welfare

(b) Utility of cluster 6

Figure 4.20: Behavior of the network with RTEL algorithm in *configuration 1*.

and thresholds variations during 3000 iterations. One can observe that adaptive thresholds manage to keep the network stable even in the presence of disturbances as in the fixed threshold case presented in Figures 4.18. On the right, Figure 4.20b shows the capacity for the thresholds to adapt online to the utility distribution. The tolerance thresholds are sometimes reset and converge to a new value (*e.g.* see around iteration 700). In addition, one can see the capacity for the cluster to not be perturbed by experiments (*e.g.* after iteration 1000). There is a strong outage due to experiments in the network and, the thresholds are not modified because this value is outside the range defined by the tolerance levels. The thresholds keep the cluster away from becoming discontent and stabilize the network. In comparison to the case with a fixed tolerance level in Figure 4.18a, here the network is able to find a better state with a higher social welfare. It is due to the fact, that clusters adapt dynamically their thresholds and can avoid more often from being stuck into a state because of a large tolerance level.

Figures 4.21 present the same results as in Figure 4.20 but with the enhanced RODL algorithm. As usual, the average social welfare seems more unstable in comparison to the RTEL case. Similarly to the fixed thresholds scenario in Figure 4.19, the adaptive thresholds manage to stabilize the network in the presence



(a) Social welfare

(b) Utility of cluster 6

Figure 4.21: Behavior of the network with RODL algorithm in *configuration 1*.

of disturbances. In addition, it seems to have a better average social welfare in comparison to the fixed tolerance case in Figures 4.19. Figure 4.21b illustrates the tolerance level variations of cluster 6 between iterations 800 and 1200. One can observe the capacity of thresholds to adapt to the utility variations (in green) despite the instability of RODL.

We have been able to show the importance of the presence of thresholds to stabilize the network in presence of disturbances. In addition, we have shown that these tolerance levels should be adaptive and, a procedure to set the tolerance online has been presented with motivated arguments and mathematical justifications. This enhancement shows the particularity to keep the performance of both algorithms in a disturbed environment similar to a non one (from a stability perspective). It even seems to outperform the fixed tolerance scenario in some conditions. So far, we have considered the social welfare as a leading performance metric to implement modifications of the TEL and ODL and each time, we have observed one realization of the allocation process over 15 000 iterations. From a telecommunication perspective, there are some other metrics to be observed. In addition, one need also to test the proposed modifications over several run of the same scenario to completely analyze the advantages and disadvantages of these modifications.

4.6.5 Comparison of TEL with adaptive RTEL

This section provides numerical results to compare the enhanced RTEL with the TEL. The goal is to highlight the advantages and disadvantages of the proposed solution in a context with disturbances. TE algorithms behavior is driven by

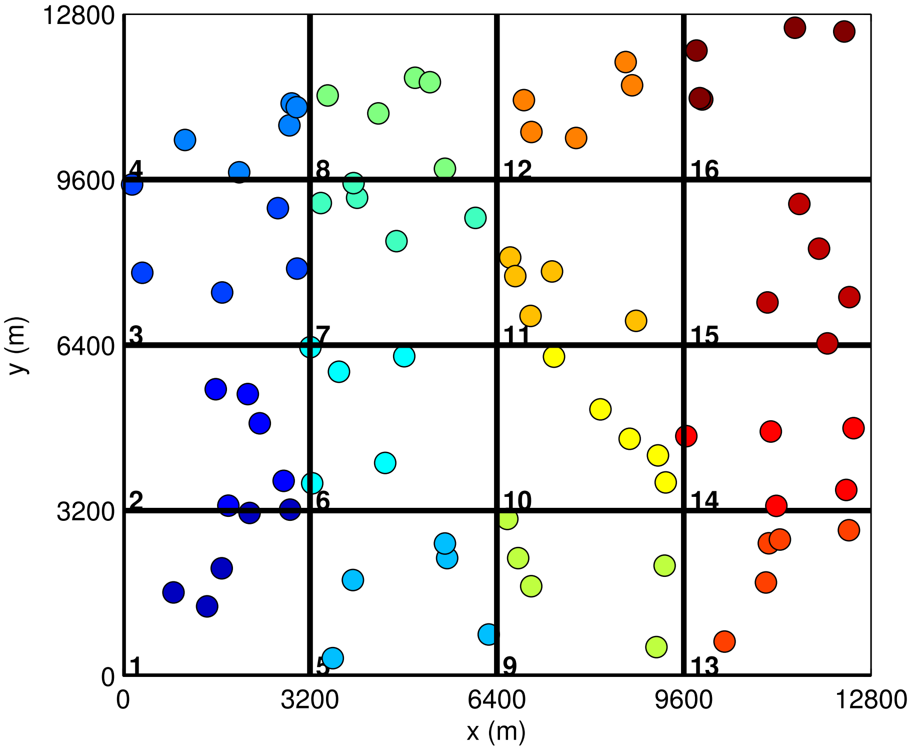


Figure 4.22: Example of a clustered ad hoc network with 16 clusters and 5 nodes per cluster. Each cluster is highlighted by square containing nodes with a specific color and an identification number on the bottom left.

stochastic processes. Players try new actions and modify their states following a probability distribution (see RTEL description in Section 4.3 in page 55). In this sense, their performance change depending on the network and on the realization considered. They spend a high proportion of time in efficient states but they also can spend some time in undesirable states. It is important to study their average performance over several realizations and networks. We are going to test both algorithms in *configuration 1* and then in *configuration 2*. Performance metrics are presented in Section 4.6.5.1. They are averaged over 5 realizations of 5 different networks maps as the one presented in Figure 4.22. More specifically, each map is a square of 12800 m aside separated in squares of 3200 m long. Each network is composed of 16 clusters of 5 nodes each. Nodes are distributed uniformly in squares.

4.6.5.1 Average ad hoc network metrics

This section presents the different metrics considered in Section 4.6.5.3. We average metrics over algorithm iterations, networks and realizations of simulations that are repeated several times.

- Average utility per cluster

No matter what is the configuration considered, the utility is the normalized sum of indicator functions ℓ_i^k (see Section 4.6.1 in page 77) and is noted $u_k(\mathbf{a})$ for cluster k. For each configuration one can compute the average utility per cluster

$$\overline{U} = \frac{1}{K(n_f - n_0)} \sum_{t=n_0}^{n_k} \sum_{k=1}^{K} u_k(\mathbf{a}_t), \tag{4.37}$$

where $\mathbf{a}_t$ is the network action at instant t and, n_0 and n_f are the first and last iterations considered respectively.

- Average channel changes per cluster

The average channel changes of a network is the average number of time the action changes. It reflects the global perturbation of the network. A lot of action changes means that it does not find an appropriate action and can lead to severe collisions. A player k has changed action from iteration t to $t+1$ if

$$J^k(t) = \mathbb{1}_{\{a_k(t) \neq a_k(t+1)\}}, \tag{4.38}$$

where $a_k(t)$ is the action played by cluster k at instant t. The average number of channel changes per cluster is given by

$$\bar{J} = \frac{1}{K(n_f - n_0)} \sum_{t=n_0}^{n_f} \sum_{k=1}^{K} J^k(t). \tag{4.39}$$

- Cluster's topology based metrics

The utility defines the global performance of each cluster but does not provide details on their structure. For instance, it is really important to keep nodes in cluster connected (in the sense of the working link metric ℓ). In addition, one can require that a message cannot be relayed more than a given number of times (*e.g.* 2 times) to avoid delay. Topology metrics based on graph theory are a efficient tool to measure these constraints.

In this thesis, the topology of a cluster is defined by the adjacency matrix, named $A(k)$ for cluster k, and computed using the working link metrics ℓ^k as follows

$$\forall i = (m,n) \in \mathcal{L}_k,\ (A(k))_{mn} = \begin{cases} 1, \text{ if } \ell_i^k = 1 \text{ or } m = n, \\ 0, \text{ otherwise }, \end{cases} \tag{4.40}$$

where, as a reminder, $\mathcal{L}_k$ is the set of links in cluster k.

With this matrix, one can compute the minimum number of hops necessary to allow a communication between two nodes. It is given by a matrix, noted $D(k)$ for cluster k, that can be computed as follows

$$\forall i = (m, n) \in \mathcal{L}_k,\ (D(k))_{mn} = \inf\{c \in \mathbb{N} \big| (A(k)^c)_{mn} > 0\}, \tag{4.41}$$

where $(D(k))_{kk} = 0$ and, two non connected nodes have an infinite distance. It is then possible to compute the diameter of each cluster k that is the maximum number of hops to connect two nodes in cluster k, and it is given by

$$d_k = \max D(k). \tag{4.42}$$

It is worthwhile to remark that when $d_k = \infty$, there are at least two nodes that cannot communicate at the requested QoS. In practice, we never test $d_k > N_k$, where N_k is the number of nodes in cluster k, as N_k is the length of the longest path in the graph. In addition, matrix D is computed with a depth first search algorithm.

4.6.5.2 Perturbation profile and satisfaction

The numerical results presented so far are obtained using a constant perturbation ϵ. In practice, this leads to very slow increasing social welfare (see results in Sections 3.4.2, 3.5.9, 4.6.4 and 4.6). In Chapter 3, we have shown that the convergence time can be of the order of $1/\epsilon^a$ where $a > 0$. Usually, when dealing with TE algorithms one need to start with high value of perturbation to converge fast to an average good social welfare and then decreases slowly ϵ. In [41] authors prove that to ensure convergence of TE algorithms with probability 1, the perturbation ϵ must decrease with a profile such that $\lim_{T\to\infty} \sum_{t=1}^{T} \epsilon_t = \infty$, where ϵ_t is the perturbation at instant t. Example of profiles that fulfill this condition are of the form $\epsilon_t \approx \frac{\epsilon_0}{t^b}$ with $b \in]0, 1[$, where t is the iteration number and, ϵ_0 is the initial perturbation value. This profile rises two problems. The first problem is that it decreases extremely slowly which involves a very slow convergence. The second problem is that once ϵ has started to decrease, the network is less and less able to adapt to changes. Solutions encountered in the literature consider first a fast decrease of the perturbation and secondly, a possible increase of ϵ as soon as some specific constraints are not satisfied.

In [35, 39], authors propose a way to decrease ϵ very fast if the utility of a player is equal to 1. If the utility is less than its maximum value, the perturbation is set to an initial value. While the utility is maximum, a player divides by two the perturbation at each iteration. Such a quick decrease prevents the algorithm from converging with probability 1, however, it shows good results in practice [35]. This process of decreasing very fast the perturbation once a given criterion is reached reminds the notion of satisfaction [53, 54]. The idea behind satisfaction is to satisfy each player instead of trying to maximize each player's utility. A player can be satisfied though the utility is not maximum.

The satisfaction can have many forms and using the utility can be one of them [31]. In this thesis, we decided to slightly modify this point of view by taking into account the topological structure of clusters. The diameter in (4.42) is considered as a satisfaction function and more specifically, a cluster is satisfied as long as its diameter is less than 2. The main advantage is that even if a link is not working because of interference or disturbance (which means that the utility is strictly less than 1), the diameter can still be 2 and the cluster is able to operate (*i.e.* we consider that a message can be relayed once). Hence, the perturbation is initiated less untimely than if the cluster was satisfied with a utility equal to one. Note that the satisfaction based on the utility equal to its maximum could be relaxed by considering clusters satisfied if their utility is greater than a threshold strictly less than the maximum but, this would still not take into account the topological cluster's structure.

The perturbation profile of each cluster is generally defined as follows

$$\epsilon_{t+1} = \begin{cases} \epsilon_0, \text{ if } s(\mathbf{a}_t) = 0, \\ \max\{\epsilon_{\min}, v\epsilon_t\}, \text{ otherwise,} \end{cases} \tag{4.43}$$

where ϵ_t is the value of the perturbation at instant t, ϵ_0 is the initial perturbation value, $\epsilon_{\min}$ is the minimum perturbation value, $\mathbf{a}_t$ is the network action at instant t, v is positive constant strictly less than 1 (*e.g.* in [31], $v = 0.5$) and, $s(.)$ is the satisfaction function that can take two values here 0 or 1 (*e.g.* in [31], $s(\mathbf{a}_t) = \mathbb{1}_{\{u(\mathbf{a}_t)=1\}}$). These parameters are specified in the following numerical results section.

4.6.5.3 Results

- Numerical results : *configuration 1*

We start by presenting the numerical results associated with *configuration 1*. The simulation parameters are presented in Table 4.3. We consider that transmissions are NB and subject to Rayleigh fading (see Chapter 5 for details on the fading model). Both algorithms, namely RTEL and TEL, are compared with an epsilon profile given by (4.43). We set $R_1 = 0.98$ in order to cover 98 % of utility values.

Table 4.3: Parameters values for simulations in *configuration 1*

N	$12, \ldots, 16$	P_{Tx}	1 W	B	250 KHz
k_b	$1.38\ 10^{-23}\ WK^{-1}s$	T	238 K	P_b	$7.9\ 10^{-15}$ W
Γ_0	13.2 dB	ϵ_0	$0.01\frac{N}{K}$	ϵ_{min}	$\frac{0.01}{K}$
n_0	2000	n_f	5000	v	0.61
R_1	0.02	R_2	0.4		

This value is a fair trade-off between covering a large amount of the utility and let some out in order to be able to react to experiments or network action changes. The parameter $R_2 = 0.4$ is low because the estimation process is quite conservative. It follows that this value enables a fast convergence of thresholds in any case while keeping robust the convergence by above values to ideal thresholds. Parameter is set to $v = 0.61$ in order that a cluster, when satisfied, goes from ϵ_0 to ϵ_{min} in five iterations.

Figure 4.23 presents the average utility (4.37) computed over 25 simulations. The red curve is the metric obtained using the RTEL algorithm and the black curve is the one obtained using TEL algorithm. The vertical bars are one standard deviation above and below the average. First of all, one can observe that the average utility increases with the number of available channels. This is an expected behavior as there are more possibilities for the algorithm to select states with less interference. Second, the RTEL algorithm improves performance of the clustered ad hoc network in comparison to the TEL. The solution of adding tolerance thresholds enables the algorithm to reach a better average performance. This improvement is not that significant when the number of channels becomes larger. In this case, there exists states with low or no interference. It follows that the SNR is high enough to avoid strong outages and consequently utility disturbances. The working condition of the TEL are recovered and the advantages of adding tolerance thresholds are less significant.

Note that the standard deviation of measurements is larger when using RTEL than when using the TEL. In the RTEL, the utility variations are greater than the thresholds levels and the network is stable in each states visited. In the contrary, without thresholds any utility variations are possible and the network state changes very quickly around an average constant social welfare. Consequently, the resulting average utility value with RTEL can be quite different (though very good) from one simulation to the other in comparison the one obtained with the TEL. This phenomenon can be clearly observed in Figure 4.16. In addition, when the average utility increases, utility values are closer to one and the measurement

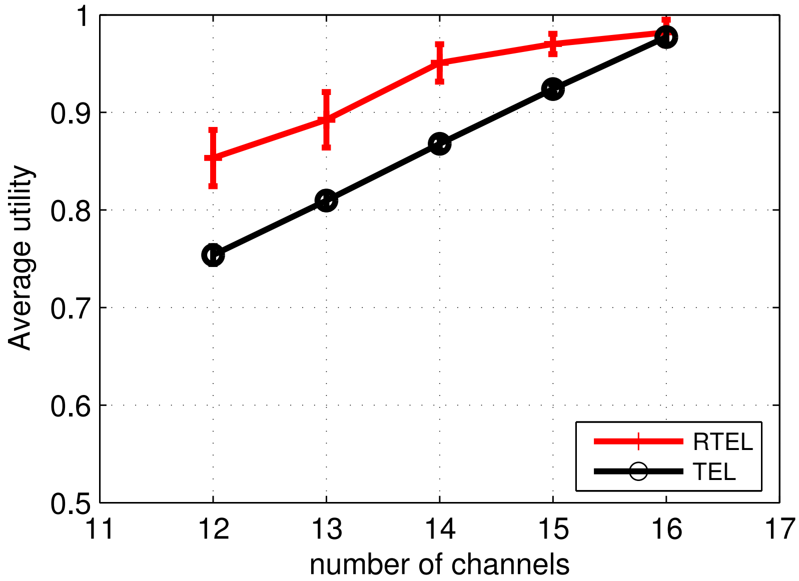


Figure 4.23: Average utility computed over 5 realizations of 5 clustered ad hoc networks with 12 to 16 channels in *configuration 1*. Bars represent one standard deviation up and below the average value.

deviation decreases.

Figure 4.24 presents the average diameter per cluster. This metric decreases when the number of channel increases. The reason is that the less interference, the better the communication links and, hence the better the diameter. One can notice that the diameter with the RTEL is lower than with the TEL. It highlights the fact that a greater utility does not imply necessarily a better cluster topology. However, both global behavior are linked. When the utility increases, the diameter decreases. One of the reasons of the gap observed is that the tolerance thresholds can stabilize clusters in states in which the diameter, is sometimes higher than 1. Without thresholds as soon as the diameter is greater than two, it is highly probable that clusters change their actions and, in average, this leads to the small gap observed.

Figure 4.25 presents the average number of channels changes per cluster. In this figure, the impact of the proposed solution is clearly visible. The network is impressively stabilized with the solution in comparison to the TEL. It follows that some control has been recovered as there are few channel changes and a high average utility. In the TEL case however, there are a lot of channel changes due to the high number of discontent clusters. These players act totally randomly and the consequence is a lost in control and a lower average utility. Again, when the number of channels increases, the average channel changes with TEL algorithm

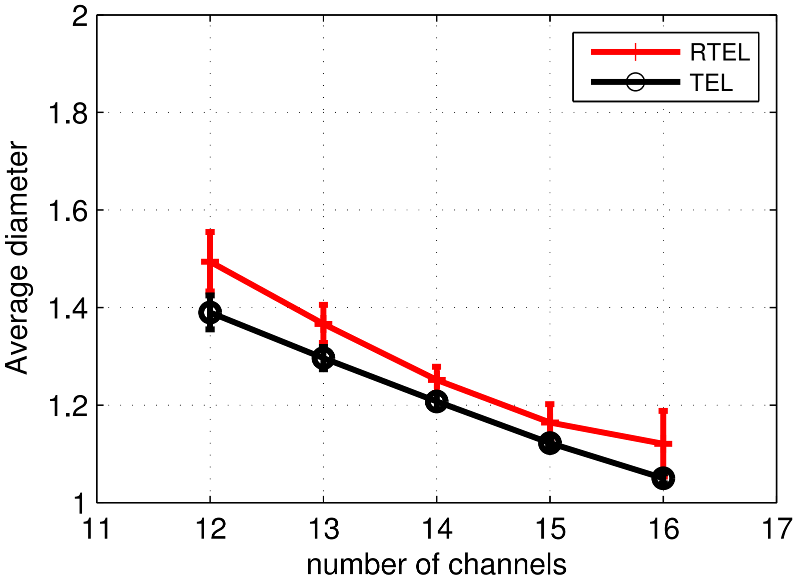


Figure 4.24: Average diameter computed over 5 realizations of 5 clustered ad hoc networks with 12 to 16 channels in *configuration 1*. Bars represent one standard deviation up and below the average value.

comes close to the one obtained with RTEL as the hypothesis needed to get the TEL works are recovered (*i.e.* there exists states with no disturbances).

The average channel change is never null because the perturbation profile never goes lower than 10^{-3}. It means that players try new channels at least every 1000 iterations even if they are satisfied.

Figure 4.26 presents the average 10th percentile of the utility. It is the utility value that is above the 10 % of the lowest utility values. It represents the importance of outages or strong decreases in the utility. The reason to highlight this result is to show that a low number of channel changes implies less collisions and thus lower outage utilities. One can see that the red curve is, except for $N = 16$ above the dark one. This means that strong utility outages are more significant when there are no tolerance thresholds which justifies the use of the modification in such case. It appears that this effect is reduced when the number of channel increases and can even be reversed. When $N = 16$ the red curve is below the black one. As it is explained in previous numerical results, in this situation, hypotheses needed for the TEL are recovered and it performs well. In addition, this also shows that the add of tolerance threshold can prevent the network from reaching better performance states. This phenomenon is explained more accurately in the sequel when considering *configuration 2*.

- Numerical results : *configuration 2*

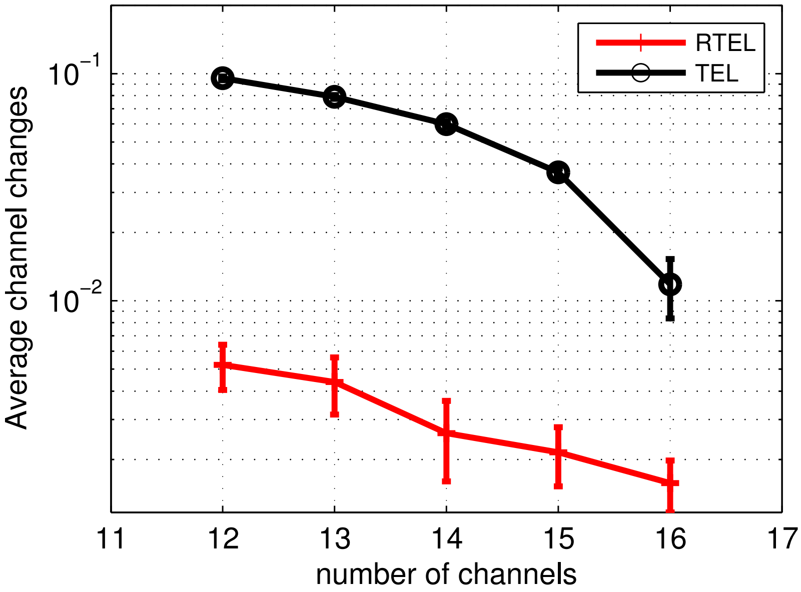


Figure 4.25: Average frequency of channel changes per cluster computed over 5 realizations of 5 clustered ad hoc networks with 12 to 16 channels in *configuration 1*. Bars represent one standard deviation up and below the average value.

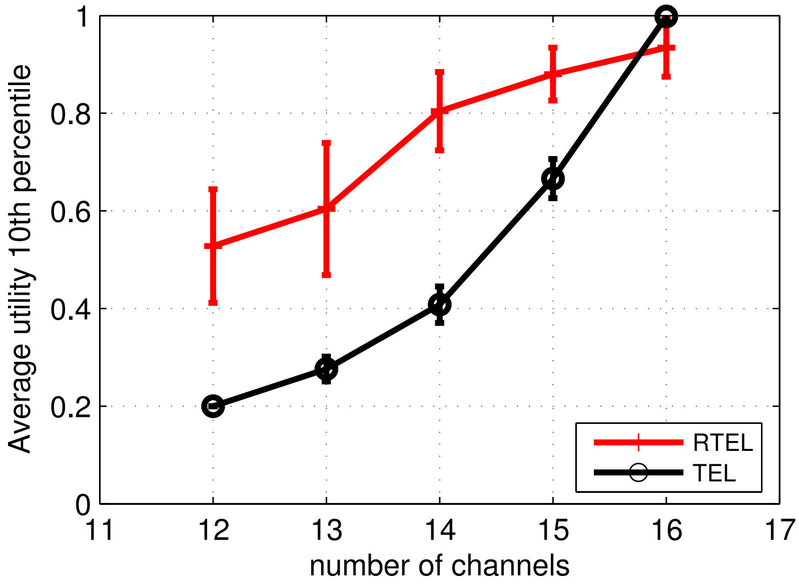


Figure 4.26: Average utility 10th percentile computed over 5 realizations of 5 clustered ad hoc networks with 12 to 16 channels in *configuration 1*. Bars represent one standard deviation up and below the average value.

This section deals with the results obtained considering *configuration 2*. The simulations parameters are similar to those presented in Table 4.3. In addition, we consider that each node transmits $M = 5$ times during a TE iteration and that the threshold $\Gamma_1 = 0.8$. In this case, for each node at most one message can be lost among all transmissions during an iteration.

Figure 4.27 presents the average utility (4.37) as in Figure 4.23 but with *configuration 2*. First of all, one can notice that at lower number of channels one get similar or better performance in comparison to results obtained under *configuration 1* in Figure 4.23 and this for both algorithms. This is an expected behavior because under *configuration 2* the interference generated is lower than in *configuration 1* (see Section 4.2). In addition, one can notice two different regions. The first one before $N = 15$ and the second one above $N = 15$. The first region corresponds to what has been observed in *configuration 1* results in Section 4.6.5.3. In this region, the RTEL performs better because no matter what is the state, there are disturbances on the utility so the tolerance thresholds have an important role. However in the second region, results are less convincing. It appears that the average utility is stuck just below one in comparison to results obtained with TEL that reach one (black curve). The reason is that, in the RTEL, once the benchmark plus the threshold of a cluster is equal to one, it is stuck in the state unless it goes to discontent which can take a long time (*e.g.* a cluster needs two consecutive strong enough collisions to go through discontent).

Figure 4.28 illustrates the latter phenomenon. It shows the utility of cluster 5 with $N = 16$ during 600 iterations of one of the simulation's realizations. One can notice that, after a few hundred iterations, $\bar{u} + \delta_+ = 1$ but the average utility is less than one. This cluster is interfered with cluster 4 and both are then sensitive to fast fading. Cluster 5 cannot, by its own, select an action that improves the average utility and is stuck in a suboptimal state. To accept a new action, the experimentation must result in a higher utility than $\bar{u} + \delta_+$ which is already maximum. Note that in the simulations, we have observed that both clusters are satisfied (*i.e.* diameter less than 2) though the utility is not equal to 1.

Figure 4.29 presents the average diameter per cluster. The average diameter is less than 2 in both cases which fulfills the requirement of the satisfaction. In addition, as in the previous configuration it decreases with N. There are some remarks to add. Both regions described in Figure 4.27 are also visible. The diameter of both algorithms is almost equal until $N = 13$ which is slightly different from what has been observed in Figure 4.24. In the latter figure the diameter of the TEL is slightly lower than the one of RTEL. After $N = 13$ a gap start to increase between both curves. It is due to the robustness induced by tolerance levels which prevents cluster from finding state with better utility and with better diameter.

Figure 4.30 presents the average number of channels changes per cluster.

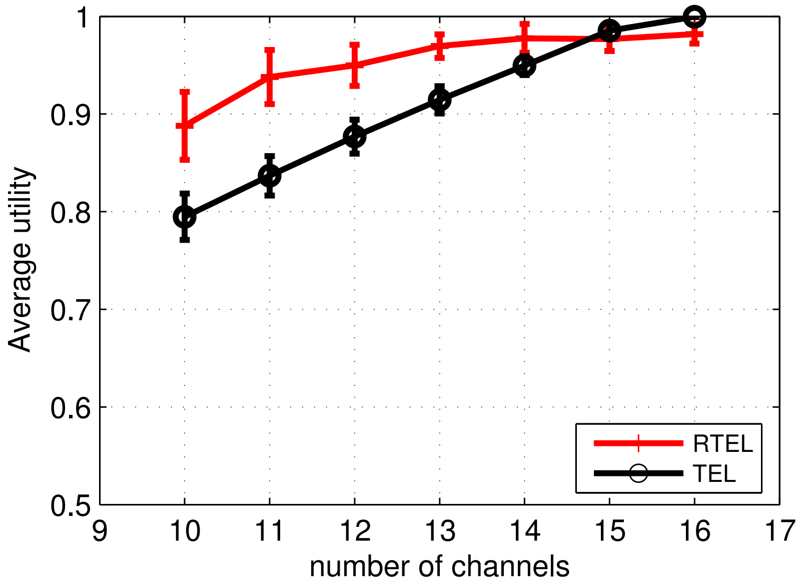


Figure 4.27: Average utility computed over 5 realizations of 5 clustered ad hoc networks with 12 to 16 channels in *configuration 2*. Bars represent one standard deviation up and below the average value.

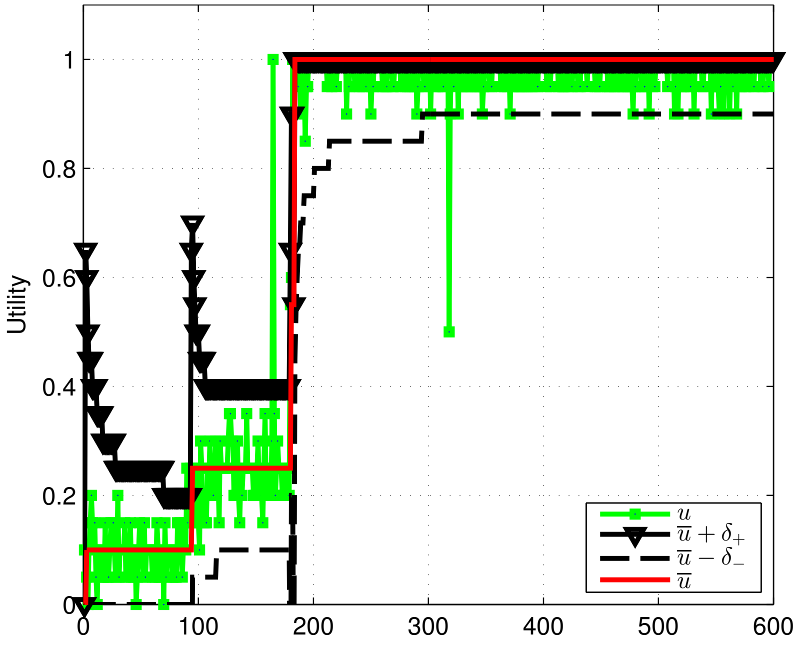


Figure 4.28: Instantaneous utility of cluster 5 during a realization of the simulation with $N = 16$ channels.

As in Figure 4.25 the stabilization induced by the proposed solution is clearly visible. Note that in this case when $N = 16$ average channel changes of both algorithms are almost the same. This means that TEL has found a stable state

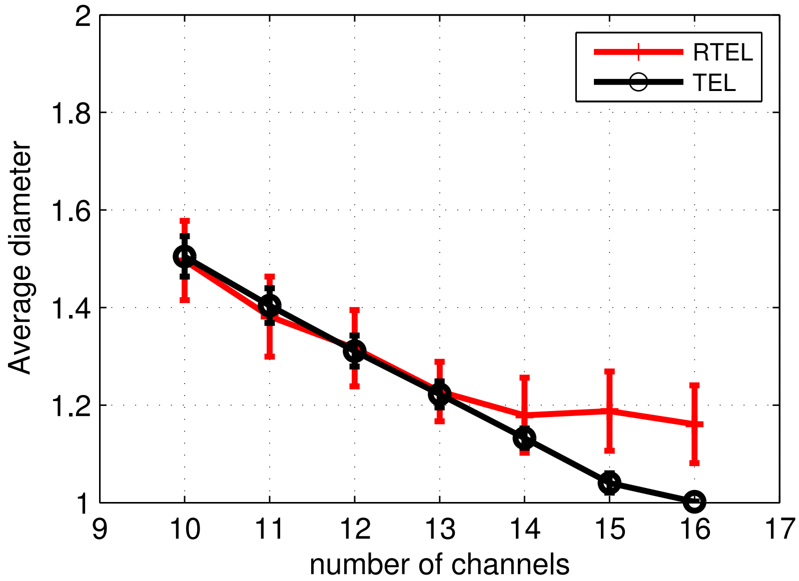


Figure 4.29: Average diameter computed over 5 realizations of 5 clustered ad hoc networks with 12 to 16 channels in *configuration 2*. Bars represent one standard deviation up and below the average value.

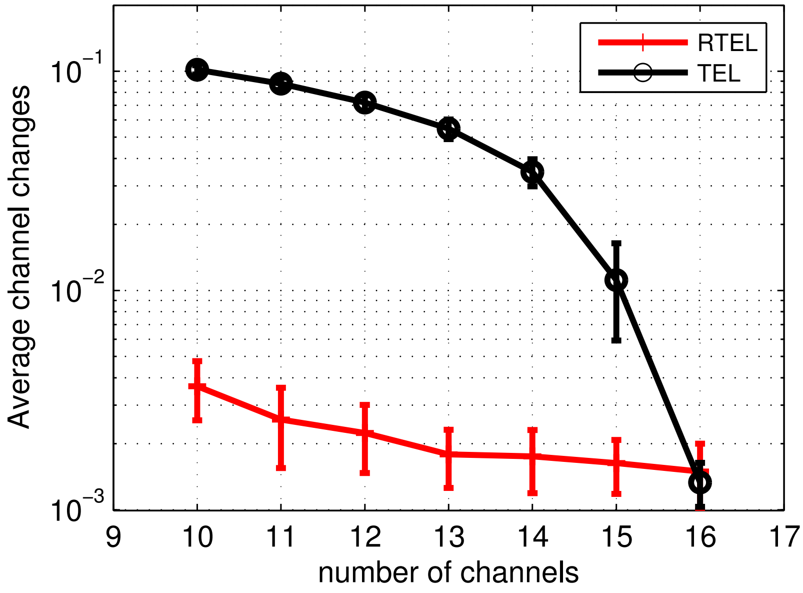


Figure 4.30: Average frequency of channel changes per cluster computed over 5 realizations of 5 clustered ad hoc networks with 12 to 16 channels in *configuration 2*. Bars represent one standard deviation up and below the average value.

with no disturbance that verifies the satisfaction. The existence of this state is more probable when there are more and more channels as there exists states with less and less interference.

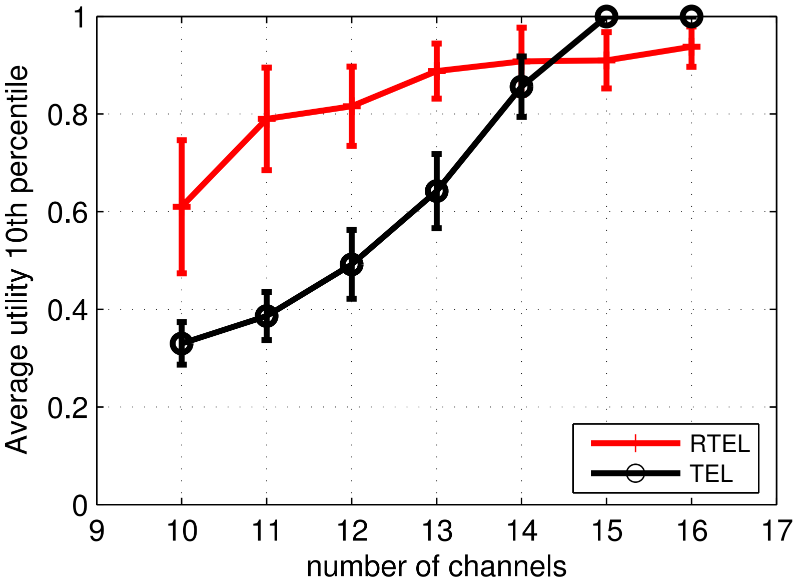


Figure 4.31: Average utility 10th percentile computed over 5 realizations of 5 clustered ad hoc networks with 12 to 16 channels in *configuration 2*. Bars represent one standard deviation up and below the average value.

Figure 4.31 presents the average 10th percentile of the utility. It is the utility value that is above the 10 % of the lowest values. Again, two working regions are highlighted. Below $N = 14$ with TEL, outages are more significant than the one with RTEL . Above $N = 14$ the behavior is reversed. This still justifies the use of the tolerance when the utility is disturbed.

4.7 Conclusion

This chapter has presented modifications of the TEL and ODL algorithms, named RTEL and RODL, that are able to operate in environment with a utility disturbed by a stochastic process. One of the reasons to propose these modifications is that TEL and ODL have been shown, theoretically and numerically, to loose their good convergence properties in such disturbed environment. The convergence proof of the RTEL has been derived under a general disturbance model. The convergence proof of the RODL was already been proposed in the literature [34]. Both robust algorithms have then been design to adapt to varying disturbance distribution. For instance, the distribution depends on the network action and structure that could vary over time. We therefore have proposed an enhancement of the RTEL and RODL that have made them able to adapt to highly varying environment.

We have provided a numerical analysis that sustains theoretical results. We have shown the loss of convergence of the TEL and ODL in a disturbed environment in which we have considered Rayleigh fading. We also have illustrated that the RODL and RTEL were able to operate in disturbed environment. However, despite these good results we have shown that the RODL is quite unstable just as the ODL. We have compared the results obtained with the RTEL and the initial algorithm TEL in two configurations for a high number of simulations. Different metrics have revealed the capacity for the robust algorithm to improve performance when the utility was subject to disturbances. We have highlighted the drawback that the presence of thresholds could sometimes prevent the network from improving its performance. Thresholds have stabilized the network but they could prevent some clusters from learning better actions and especially when they had a utility close to the maximum. These scenarios have arisen when there were almost as many channels as players. This has created states with no disturbances which, for a reminder, are the stable states of the TEL which in addition here maximized the social welfare. However, when the utility was constantly disturbed, at low number of available channels, the RTEL has operated efficiently in comparison to TEL. Consequently, the two different behaviors that have been observed, have appeared from the fact that simulations contained a mix of states with disturbed and non-disturbed utility values. An idea to operate in such environment would be to find rules that deactivate the thresholds when necessary, because the RTEL without thresholds becomes the TEL.

Chapter 5

Statistics of the disturbed utility

In this chapter, we derive the statistics of the disturbed utility that are needed to study its behavior (*e.g.* bounded assumption, variance,...). Since these statistics depend on the SINR ones, we first derive the statistics of the SINR when the propagation is modeled by Rayleigh fading.

We consider both the NB and WB fading cases modeled by flat Rayleigh fading and frequency selective Rayleigh fading respectively. In the WB context, the SINR is computed using a link to system abstraction using the effective SNR. In order to get a closed form expression of the pdf of the utility, we select the EESM abstraction since we use OFDM for the physical layer. However, since the frequency fading coefficients are correlated over all subcarriers, we simplify the fading model using the coherence bandwidth concept where subset of subcarriers are subject to independent fadings. From this model, we succeed to compute the close to exact pdf of the utility, using numerical convolution, in a simplified case in which the interference do not experiment fading. In the general case in which interference are subject to Rayleigh fading, although the pdf of the terms to be convolved can be derived in closed form, we need an approximation to realize tractable numerical convolutions.

In order to apply the previous model in simulations, one need to set the coherence bandwidth value parameter, *i.e.* the number of consecutive subcarriers that are subject to the same flat fading. However, the definition of the coherence bandwidth (that may differ between authors) is not accurate enough to feed our model. In addition, this bandwidth needs to be adjusted with respect to the channel parameters (path amplitudes, path delays, ...). In this thesis, we provide a theoretical framework to find the “right” coherence bandwidth value to perform pdf computations and simulations. We select the coherence bandwidth such that the pdf of the effective SNR obtained using the coherence bandwidth assumption fits as close as possible the one obtained using the real channel. This approach gives a meaningful interpretation of the coherence bandwidth with respect to the block fading model. Application of this framework is applied to a typical urban standard channel. The relevance and accuracy of the proposed framework is assessed by simulations comparing the Packet Error Rate (PER) obtained by both abstractions.

Section 5.1 presents the statistical model of the utility when the SINR is a rv and, proposes models of the SINR when transmissions are subject to different types of Rayleigh fading such as the NB fading and the WB fading. The later model necessitates the computation of the coherence bandwidth for which we propose a methodology described in Section 5.2. Based on this model, we highlights in Section 5.3 numerical results that illustrate the steps of the methodology. We also present a validation of the procedure by predicting with high accuracy performance in WB OFDM schemes and we derive the utility distribution is some cases.

5.1 Pdf of the SINR and of the utility

5.1.1 Pdf of the utility

This section, presents the general model of the utility distribution when the SINR is time varying. For clarity, we remind some notations from Chapter 3. We consider a wireless network composed of K clusters with possibly different sizes. The set of clusters is $\mathcal{K}$. The set of N possible resources is denoted by $\mathcal{N}$. In a cluster, a link i is said to work if its SINR at the receiver side is above a predefined threshold $\Gamma_0 \geq 0$. Remind that, the utility function is, according to (3.3),

$$u_k(\mathbf{a}) = \frac{1}{|\mathcal{L}_k|} \sum_{i \in \mathcal{L}_k} \ell_i^k(\mathbf{a}), \tag{5.1}$$

where $\ell_i^k(\mathbf{a}) = \mathbb{1}_{\{\text{SINR}_i^k(\mathbf{a}) > \Gamma_0\}}$. When the SINR is a rv, then $\ell_i^k(\mathbf{a})$ is a binomial rv equal to one when link i in cluster k is working and zero otherwise. The Bernoulli parameter of $\ell_i^k(\mathbf{a})$, noted p_i^k, is given by

$$p_i^k(\Gamma_0) := \Pr\left\{\text{SINR}_i^k > \Gamma_0\right\}, \tag{5.2}$$

which can be seen as the complementary Cumulative Density Function (cdf) of $\text{SINR}_i^k(\mathbf{a})$. Define the vector $\mathbf{b} := (b_1, ..., b_{|\mathcal{L}_k|})$ where each component can take its value in $\{0, 1\}$ and the set $\mathfrak{B}_\ell := \{\mathbf{b} | \sum_{i=1}^{|\mathcal{L}_k|} b_i = \ell\}$ which is the set of all combinations

b such that the sum of its terms is ℓ. Since u_k is a discrete rv, its distribution is equal to its probability mass

$$p_{u_k}(\ell) = \Pr\left\{u_k = \frac{\ell}{|\mathcal{L}_k|}\right\}, \tag{5.3}$$

and the closed form of that distribution is a Poisson-Binomial one

$$p_{u_k}(\ell) = \sum_{\mathbf{b}\in\mathfrak{B}_\ell} \prod_{i=1}^{|\mathcal{L}_k|} (p_i^k)^{b_i}(1-p_i^k)^{b_i}. \tag{5.4}$$

Using the definition given by (3.3) and (5.2), the mean of the process is $\mathbb{E}\left[u_k\right] = \frac{1}{|\mathcal{L}_k|}\sum_{i\in\mathcal{L}_k} p_i^k$ which can be interpreted as the arithmetic mean of p_i^k and the variance is $V(u_k) = \frac{1}{|\mathcal{L}_k|}\sum_{i\in\mathcal{L}_k} p_i^k(1-p_i^k)$. Finally, to compute the utility distribution, one need to find parameters p_i^k (5.2) that are directly linked to the cdf of the SINR. Then, a practical way to compute the exact Poisson-Binomial distribution numerically is to use convolution methods as presented in [52]. We consider two cases, first we deal with the standard NB case and then, we derive the WB one.

5.1.2 NB fading

Interference is caused by nodes belonging to clusters using the same resource. Let $\mathcal{M}_n$ be the set of transmitting nodes at time t in cluster $n \in \mathcal{K}$ and, $\mathcal{I}_k$ is set of clusters interfering (operating on the same resource) with cluster k. The instantaneous SINR expression of link i in cluster k considering only AWGN channels is given by (3.1) (page 20). In a NB scheme, the useful signal as well as the interference are subject to Rayleigh flat fading. The SINR at the receiver i_r of link $i = (i_t, i_r)$ in cluster k becomes

$$\mathrm{SINR}_i^k = \frac{\alpha_i^k \mathrm{SNR}_i^k}{1 + \sum_{n\in\mathcal{I}_k}\sum_{j\in\mathcal{M}_n} \alpha_{j,i_r}^{n,k}\mathrm{INR}_{j,i_r}^{n,k}}, \tag{5.5}$$

where α_i^k and α_{j,i_r}^{nk} are the fading coefficient that impact the link $i = (i_t, i_r)$ in cluster k and the interfering link (j, i_r) with j in cluster n and i_r in cluster k. When considering Rayleigh fading, signals' power is modulated by an exponential rv [55]. In addition, a parameter $\lambda = 1$ for the exponential rv means that there is no power loss through Rayleigh fading. Consequently, the fading coefficient α_i^k and α_{j,i_r}^{nk} follow an exponential distribution with parameter $\lambda = 1$. Here the fading impacts at the same time the SNR and the Interference to Noise Ratio (INR) terms. The formula (5.5) is employed in Chapter 4 as the input of the utility function in case of NB fading.

When the system is NB, it is possible to compute an exact expression of the success probability p_i^k (5.2). In Appendix D, we prove that for a link $i = (i_t, i_r)$, the Bernoulli parameter is given by the complementary cdf of the SINR as follows

$$p_i^k(\Gamma_0) = \frac{\exp\left(-\frac{\Gamma_0}{\mathrm{SNR}_i^k}\right)}{\prod_{n\in\mathcal{I}_k}\prod_{j\in\mathcal{M}_n}\left(1+\frac{\Gamma_0\ \mathrm{INR}_{j,i_r}^{n,k}}{\mathrm{SNR}_i^k}\right)}. \tag{5.6}$$

5.1.3 WB fading

In this section, we present a WB formula of the SINR in an OFDM system. When such a system is subject to frequency selective fading, symbols suffer from different SNR over each subcarrier. This has motivated the use of link to system mapping techniques such as EESM [56, 57]. This solution, employed in OFDM level simulations, for instance in LTE [58, 59], maps the different SNRs over all the subcarriers into one single effective value $\mathrm{SNR}_{\mathrm{eff}}$ (so-called *effective SNR*) as follows

$$\mathrm{SNR}_{\mathrm{eff}} := -\beta \log\left(\frac{1}{N_c}\sum_{i=1}^{N_c} e^{-\frac{\mathrm{SNR}_i}{\beta}}\right), \tag{5.7}$$

where N_c is the number of subcarriers, SNR_i is the SNR of channel i and, β is a scaling parameter depending on the Modulation and Coding Scheme (MCS). The basic principle to model WB OFDM system in this thesis consists in replacing the NB SINR used in ℓ_i^k (*e.g.* see (5.1)) by the effective SINR computed in (5.7). Then the computation of the utility (5.1) remains the same.

The EESM metric is useful, for instance, to evaluate the performance of wireless systems (*e.g.* PER) or to build adaptive modulation and coding schemes [60]. However, due to the nonlinearity form of (5.7), there exists no exact closed form for its statistics. It follows that the success probability of each link i in cluster k, $p_i^k(\Gamma_0)$ (5.2), cannot be obtained analytically and the utility distribution cannot be derived accordingly. The proposed approaches in the literature are based on moment matching approximation with different distributions. Gaussian, Generalized Extreme value and Pearson distributions are employed in [61], Log-normal distribution is used in [58] and, a more accurate approximation based on beta distribution is presented in [62].

Our approach is totally different from what has been proposed in the literature [63] and use the coherence bandwidth assumption [64]. Under this assumption, the SNR of subcarriers lying in the same coherence bandwidth are the same whereas the SNR of subcarriers in different coherence bandwidth are iid. This changes the sum of correlated variable in (5.7) into the sum of iid rvs. It follows that one can use numerical convolutions and a change of variables to approximate the statistics of EESM. In comparison to moment matching methods that try to approach the EESM distribution, this method provides the exact statistics of the effective SNR generated with the coherence bandwidth assumption.

The benefits of this approach are manifolds. First, the independence property between coherent bandwidth allows the realization of simple calculation to predict accurately system performance in specific cases as one will see in the sequel. In these cases it is also possible to compute success probability $p_i^k(\Gamma_0)$ from which, the utility distribution can be derived. Then, it provides a low computational method to generate effective SINR in order to simulate WB OFDM systems in Chapter 4. Furthermore, in practice, the knowledge of channel coefficient necessitates one pilot training per subcarrier which involves a large amount of signaling. In future wireless systems, the coherence bandwidth gives insights on the number of pilots to use to recover the statistics of the channel when using EESM metric. To take advantage of these benefits, one need to set the coherence bandwidth parameter or the number of subcarriers to be grouped in this bandwidth.

The rest of this section is organized as follows: Section 5.1.3.1 presents how to compute EESM when considering a real channel, Section 5.1.3.2 describes the computation of EESM when considering the proposed block fading channel, Section 5.1.3.3 presents the calibration of the EESM, Section 5.1.3.4 describes the procedure to compute the statistics of EESM with the block fading channel in the simplified model where the interference are no subject to fading and, Section 5.1.3.5 details the similar procedure but in the full model where the interference are subject to Rayleigh fading.

5.1.3.1 EESM model with the real channel

We consider an OFDM system with N_c subcarriers and Δf the frequency shift between two consecutive subcarriers. We consider the following specular channel model that we refer to as "real" channel in the sequel and defined as,

$$h(t) = \sum_{i=1}^{N_p} a_i \delta(t - \tau_i), \tag{5.8}$$

where $\delta(.)$ is the Dirac function, N_p is the number of paths, a_i is the amplitude of path i and τ_i is the associated delay. We consider a Rayleigh fading in the rest of the work so, coefficients a_i follow a circular complex, white and Gaussian with zero mean and variance σ_i^2 distribution (*e.g.* $\mathcal{CN}(0, \sigma_i^2)$). The path amplitudes are supposed to be iid and, $\sum_{i=1}^{N_p} \sigma_i^2 = 1$ (the energy loss is already taken into account in the path loss coefficient).

We start the study by considering the SNR without interference which are introduced later in the model. The received SNR at subcarrier i, after removing the cyclic prefix and applying a match filtering is given by

$$\mathrm{SNR}_i = |H(f_i)|^2 \gamma, \tag{5.9}$$

where γ is the expected SNR per subcarrier, and $H(f_i)$ is the Fourier transform of (5.8) at the subcarrier i with frequency f_i:

$$H(f_i) = \sum_{k=1}^{N_p} a_i e^{-2j\pi f_i \tau_k}. \tag{5.10}$$

According to Definition (5.7), the EESM metric obtained with the real channel model is

$$\mathrm{SNR}_{\mathrm{eff}}(\gamma) := -\beta \log \left(\frac{1}{N_c} \sum_{i=1}^{N_c} e^{-\frac{|H(f_i)|^2 \gamma}{\beta}} \right). \tag{5.11}$$

5.1.3.2 EESM model with the block fading channel

This section presents the model of EESM in which we take into account the coherence bandwidth assumption. The main idea consists in considering a block fading channel such that, the SNR is the same for each block of N_{cc} consecutive subcarriers and, the fading coefficients are iid between the different blocks of N_{cc} subcarriers. Consequently, the model of EESM considering the block fading channel is given by

$$\gamma_{\mathrm{eff}}(\gamma) := -\beta \log \left(\frac{1}{N_c} \left(\sum_{i=1}^{N_b - 1} N_{cc} e^{-\frac{\alpha_i \gamma}{\beta}} + N'_{cc} e^{-\frac{\alpha_{N_b} \gamma}{\beta}} \right) \right), \tag{5.12}$$

where N_b is the number of independent blocks of subcarriers, $N'_{cc} = N_c - (N_b - 1) N_{cc}$ is the number of remaining coherent subcarriers after gathering them by group of N_{cc}, and, α_i for $i = 1, \ldots, N_b$ are iid rvs following an exponential distribution with parameter $\lambda = 1$. The reason to model α_i as an exponentially distributed rv comes from the exponential behavior of $|H(f_i)|^2$ in (5.9).

Since the SNR over consecutive subcarriers are not exactly equal in the real channel, the grouping represents an approximation of the original EESM model given in (5.11). The goal is then to find the number N_{cc} such that the statistics of γ_{eff} in (5.12) obtained with the block fading channel are close to the one of $\mathrm{SNR}_{\mathrm{eff}}$ in (5.11) obtained with the "real" channel.

The previous model is extended to consider interference. With the same principle, the fading coefficients are considered iid from one coherence block to another. The SINR at the receiver of link i in cluster k and in each block m is computed as follows

$$\mathrm{SINR}_i^k(m) = \frac{\alpha_i^k(m)\mathrm{SNR}_i^k}{1 + \sum_{n\in\mathcal{I}_k}\sum_{j\in\mathcal{M}_n}\alpha_{j,i_r}^{nk}(m)\mathrm{INR}_{j,i_r}^{nk}}, \tag{5.13}$$

with the difference with (5.5) that $\alpha_i^k(m)$ and $\alpha_{j,i_r}^{nk}(m)$ are the fading coefficients of block m that impact the link i in cluster k and the interfering link (j, i_r) with j in cluster n and i_r in cluster k. These coefficients are iid from one block to another. Considering the coherence bandwidth assumption, the EESM of link i in a cluster k is modeled as follows

$$\gamma_{\mathrm{eff}}^{i,k}(\gamma) = -\beta\log\left(\frac{1}{N_c}\left(\sum_{j=1}^{N_b-1} N_{cc}e^{-\frac{\mathrm{SINR}_i^k(j)}{\beta}} + N'_{cc}e^{-\frac{\mathrm{SINR}_i^k(N_b)}{\beta}}\right)\right), \tag{5.14}$$

where $\mathrm{SINR}_j^k(m)$ is given by (5.13). This formula is used in Chapter 4 as the SINR in a WB OFDM system which is inputed in the utility. To complete the model it remains two steps. The first step presents a procedure in order to set the scaling factor β in Section 5.1.3.3. The second step presents a methodology to find N_{cc} in Section 5.2. Between these two steps, we present how the block fading assumption can be used to obtain the statistics of EESM in Sections 5.1.3.4 and 5.1.3.5.

5.1.3.3 Calibration of EESM

In formulas (5.11), (5.12) and (5.14) the scaling factor β needs to be calibrated for each considered MCS. We employ the standard calibration procedure in [65] which is presented in Algorithm 4 and summed up briefly here. We use the model described in Section 5.1.3.1 to realize the calibration. The parameter β is optimized such that, the PER obtained for various $\mathrm{SNR}_{\mathrm{eff}}$ is as close as possible to the PER obtained for AWGN channel with the given MCS. The PER obtained in AWGN channel serves as a reference called the AWGN-PER Look Up Table (LUT) and is noted $\mathrm{PER}_{\mathrm{LUT}}$. For each realization of a channel (*e.g.* fixed paths amplitude and AWGN density), the PER, noted $\mathrm{PER}_{\mathrm{sim}}$ is obtained through simulation. The $\mathrm{SNR}_{\mathrm{eff}}$ is computed with (5.11). Note that $\mathrm{SNR}_{\mathrm{eff}}$ depends on β. Using the LUT, one can map a $\mathrm{PER}_{\mathrm{LUT}}$ value for each $\mathrm{SNR}_{\mathrm{eff}}$. With these both PER values, the square difference of the logarithm of the PER, $|\log_{10}(\mathrm{PER}_{\mathrm{LUT}}) - \log_{10}(\mathrm{PER}_{\mathrm{sim}})|^2$, is computed. The above procedure is repeated over several channel realizations.

Then, we sum these differences over all channel realizations and, β is tuned so as to minimize this sum.

Algorithm 4 Calibration of β

```
Input: $\mathrm{PER}_{\mathrm{LUT}}$, $\boldsymbol{\gamma} = (\gamma_1, \dots, \gamma_{N_{snr}})$,$\boldsymbol{\beta} = (\beta_1, \dots, \beta_{N_\beta})$,$N_\beta$, $N_{snr}$, $N_{ch}$
Output: $\beta$
for $i_{snr} = 1$ to $N_{snr}$ do
    $\gamma \leftarrow \boldsymbol{\gamma}(i_{snr})$, select the expected SNR per subcarrier
    for $i_{ch} = 1$ to $N_{ch}$ do
        $h = ((a_1, \tau_1), \dots, (a_{N_p}, \tau_{N_p})) \leftarrow$ set the channel $h$ (see Section 5.1.3.1)
        $\mathrm{PER}_{\mathrm{sim}}(h, \gamma) \leftarrow$ Monte Carlo simulation of the OFDM system with the
            given channel $h$
        $\mathrm{SNR}_{\mathrm{eff}}(h, \gamma, \boldsymbol{\beta}) \leftarrow$ compute the effective SNR with (5.9), (5.10) and (5.11),
    end for
end for
$\beta = \arg\min_{\boldsymbol{\beta}} \left\{ \sum_h \sum_\gamma \left| \log_{10}(\mathrm{PER}_{\mathrm{LUT}}(\mathrm{SNR}_{\mathrm{eff}}(h, \gamma, \boldsymbol{\beta}))) - \log_{10}(\mathrm{PER}_{\mathrm{sim}}(h, \gamma)) \right|^2 \right\}$
return $\beta$
```

5.1.3.4 EESM statistics and pdf of the utility in a simplified model

The main advantage to model the channel as independent blocks in Section 5.1.3.2 is that convolution can be employed to get the pdf (or the cdf) of γ_{eff} (5.12). This computation is of interest since it helps in evaluating the statistics of the utility (see Section 5.1.1) and the performance of wireless broadband systems using the EESM technique. The computation requires some technical steps/details that are provided in this section. We start by a considering a simplified model in which there are no interference or if there are, they are no subject to fading. Hence in the later case, the SINR over each block of coherent subcarriers (5.13) becomes

$$\mathrm{SINR}_i^k(m) = \alpha_i^k(m) \frac{\mathrm{SNR}_i^k}{1 + \sum_{n \in \mathcal{I}_k} \sum_{j \in \mathcal{M}_n} \mathrm{INR}_{j,i_r}^{nk}}, \tag{5.15}$$

which can be seen as a constant multiplied by a rv that follows an exponential distribution. The distribution of the utility requires to compute the cdf of EESM (see Section 5.1) with the block fading channel (5.12) that is given by

$$\Pr\{\gamma_{\mathrm{eff}}(\gamma) \leq \Gamma\} = \Pr\left\{ \sum_{i=1}^{N_b - 1} e^{-\frac{\alpha_i \gamma}{\beta}} + \frac{N'_{cc}}{N_{cc}} e^{-\frac{\alpha_{N_b} \gamma}{\beta}} \geq \frac{N_c}{N_{cc}} e^{\frac{-\Gamma}{\beta}} \right\}. \tag{5.16}$$

We change the formulation of $\Pr\{\gamma_{\text{eff}}(\gamma) \leq \Gamma\}$ as follows

$$\Pr\{\gamma_{\text{eff}}(\gamma) \leq \Gamma\} = 1 - \Pr\left\{\sum_{i=1}^{N_b-1} X_i + X'_{N_b} \leq f(\Gamma)\right\}, \tag{5.17}$$

where for i from 1 to $N_b - 1$, $X_i = e^{-\frac{\alpha_i \gamma}{\beta}}$ are iid rvs with density $p_X(x)$, $X'_{N_b} = \frac{N'_{cc}}{N_{cc}} e^{-\frac{\alpha_{N_b}\gamma}{\beta}}$ is a rv independent from X_i with density $p_{X'}(x)$, and

$$f(\Gamma) := \frac{N_c}{N_{cc}} e^{\frac{-\Gamma}{\beta}} \tag{5.18}$$

is a positive constant. The problem of computing the cdf of γ_{eff} (5.16) can be seen as the problem of finding the cdf of the rv $Y = \sum_{i=1}^{N_b-1} e^{-\frac{A_i \gamma}{\beta}} + \frac{N'_{cc}}{N_{cc}} e^{-\frac{A_{N_b}\gamma}{\beta}}$ in (5.17). The later is the sum of independent rvs and its density is the convolution of rvs' densities involved into the sum as follows

$$p_Y(y) = (\underbrace{p_X * p_X * \cdots * p_X}_{N_b-1} * p_{X'})(y), \tag{5.19}$$

where $*$ is the convolution symbol and, with $y \in [0, N_c/N_{cc}]$. The next step consists in computing the densities p_X and $p_{X'}$. These variables have the general form $X = C_1 e^{-C_2 A}$ where A is exponentially distributed with parameter 1 (*i.e.* $p_A(a) = e^{-a}$), C_1 and C_2 are two strictly positive real constants. For X, $C_1 = 1$ and $C_2 = \frac{\gamma}{\beta}$. Then, for X', $C_1 = \frac{N'_{cc}}{N_{cc}}$ and $C_2 = \frac{\gamma}{\beta}$. Hence, the density of X or X', denoted by $p_X(x)$ and $p_{X'}(x)$, can be obtained after applying the change of variable $x = g(\alpha)$, where $g(\alpha) = C_1 e^{-C_2 \alpha}$ is strictly monotone with the appropriate C_1 and C_2. Thus, we only present the general pdf p_X obtained with the following formula ($p_{X'}$ is obtained by setting C_1 and C_2 appropriately in the next formulas)

$$p_X(x) = \frac{p_A(g^{-1}(x))}{|g'(g^{-1}(x))|}, \tag{5.20}$$

where $g^{-1}(x) = -\frac{1}{C_2}\log(\frac{x}{C_1})$ and $g'(\alpha) = -C_1 C_2 e^{-C_2 \alpha}$. Thus, the pdf of X becomes

$$\begin{aligned} p_X(x) &= \frac{1}{C_1^{1/C_2}} \frac{1}{C_2} x^{1/C_2 - 1}, \\ &= \frac{x^{1/C_2-1}}{\mathcal{B}(C_1; 1/C_2, 1)}, \end{aligned} \tag{5.21}$$

with $x \in]0, C_1]$, $\mathcal{B}(x; a, b) := \int_0^x t^{a-1}(1-t)^{b-1} dt$ is the beta incomplete function ([50] 6.6.1) and, $\mathcal{B}(C_1; 1/C_2, 1) = C_1^{1/C_2} C_2$. This means that X follows a Beta law. An analytical form of (5.19) is available for $N_b = 1$ or 2 (if $N'_{cc} = 0$) but we are not able for $N_b > 2$ to derive it as the computation involves the integral of incomplete beta functions. Thus, it is necessary to proceed with numerical convolutions.

In this case, we sample the densities p_X and $p_{X'}$. Let a regular sampling $\tilde{x} = x_0, \ldots, x_{N_s-1}$ where $x_{k+1} - x_k = \Delta_x$ with $\Delta_x = 1/N_s$, N_s the number of samples and $x_k = k\Delta_x$. The sampled distribution is noted $\tilde{p}$ such that $\tilde{p}_X[k] = p_X(x_k)$. We realize with this sampling that the probability is not conserved anymore due to the approximation or,

$$\sum_{i=0}^{N_s-1} \tilde{p}_X[k]\Delta_x \neq 1. \tag{5.22}$$

In addition, the density p_X possesses a singularity in 0 when $C_2 > 1$ (see (5.21)) which means that previous sum diverges in some cases. In order to conserve probability through the convolutions to compute Y and to avoid the singularity from being propagated numerically, we have found the following trick to be very useful and efficient. We replace the sampled term of the density in 0 by a specific term as follows

$$\tilde{p}_X[k] := \begin{cases} \frac{1}{\Delta_x} - \sum_{i=1}^{N_s-1} p_X(x_i),\ k = 0, \\ p_X(x_k),\ \text{otherwise}. \end{cases} \tag{5.23}$$

One can check that $\sum_{i=0}^{N_s-1} \tilde{p}_X[i]\Delta_x = 1$ with the proposed trick. The numerically approximated pdf of Y, noted $\tilde{p}_Y$, is computed using the sampled version of (5.19) with the proposed $\tilde{p}_X$ (5.23) as follows

$$\tilde{p}_Y[k] = (\underbrace{\tilde{p}_X * \tilde{p}_X * \cdots * \tilde{p}_X}_{N_b-1} * \tilde{p}_{X'})[k], \tag{5.24}$$

where the sampling of y is $\tilde{y} = (y_0, \ldots, y_{N_s'})$ with a step $\Delta_y = \Delta_x = 1/N_s$ and $N_s' = (N_b - 1)N_s + \lfloor(\frac{N_c}{N_{cc}} - N_b)N_s\rfloor$ and $N_b = \lfloor\frac{N_c}{N_{cc}}\rfloor$. In addition, we can prove that with the modified pdf in (5.23), the resulting cdf of Y obtained numerically (cumulative sum of (5.24) multiplied with Δ_y) still converges to the right cdf of Y as N_s increases. The theoretical cdf of γ_{eff} (5.17), knowing the density p_Y, is given by

$$\Pr\{\gamma_{\text{eff}}(\gamma) \leq \Gamma\} = 1 - \int_0^{f(\Gamma)} p_Y(y)dy. \tag{5.25}$$

Thus, the cdf can be approximated numerically using $\tilde{p}_y$ as follows

$$\Pr\{\gamma_{\text{eff}}(\gamma) \leq \Gamma\} \approx 1 - \sum_{i=0}^{N'(\Gamma)} \tilde{p}_Y[i]\Delta_y. \tag{5.26}$$

where

$$\begin{aligned} N'(\Gamma) &:= \left\lfloor \frac{f(\Gamma)}{\Delta_y} \right\rfloor, \\ &= \lfloor N_s f(\Gamma) \rfloor, \end{aligned} \tag{5.27}$$

with $f(\Gamma)$ given in (5.18). The value $N'(\Gamma)$ is the number of point in the sampling $\tilde{y}$ to be taken into account in the sum in (5.26). It is important to note that when Γ increases, $f(\Gamma)$ decreases and so does $N'(\Gamma)$. In addition, we require $N'(\Gamma) \geq 1$. The reason is that, when we compute numerically the cdf of γ_{eff}, we link each sample y_i to its counterpart in the effective SNR domain using the following transformation (see (5.17) and (5.18))

$$\begin{aligned} \Gamma_i &= f^{-1}(y_i), \\ &= -\beta \ln\left(\frac{N_{cc}}{N_c} y_i\right) \end{aligned} \tag{5.28}$$

Therefore, when $N'(\Gamma) = 0$ it is equivalent to compute the cdf in $\Gamma_0 = f^{-1}(0) = \infty$ and we cannot illustrate numerically the result. Consequently, the maximum effective SNR, $\Gamma_{\max}$ at which we can compute the cdf of γ_{eff} is in y_1 (the lowest y_i is for the maximum Γ) and it is given by

$$\begin{aligned} \Gamma_{\max} &= -\beta \ln\left(\frac{N_{cc}}{N_c} \Delta_y\right), \\ &= -\beta \ln\left(\frac{N_{cc}}{N_c} \frac{1}{N_s}\right). \end{aligned} \tag{5.29}$$

Figure 5.1 illustrates previous observation with $N_s = 1000$, $\gamma = 15$ dB, $N_b = 8$, $\gamma = 7$ dB and $\beta = 1.8$. The red curve is the cdf of γ_{eff} computed numerically with (5.26) and the black curve is the cdf of γ_{eff} (5.12) obtained with Monte Carlo simulations. The red curve stops at $\Gamma_{\max} = 12.1$ dB in the figure which is the value obtained with the above reasoning.

For instance, if one wants to compute the cdf at $\Gamma_{\max} = 16$ dB, the reverse reasoning can be applied to find the right sampling step. In this case, one needs $N_s = \frac{N_{cc}}{N_c} e^{\frac{\Gamma_{\max}}{\beta}} = 2 \times 10^9$ samples which requires a lot of numerical resources to be convolved. The computation of the cdf beyond $\Gamma_{\max}$ with low numerical complexity can be obtained with the following proposed numerical procedure that combines *(i)* the fact that the high effective SNR correspond to small y_i in the sampling with *(ii)* the fact that the first terms of $\tilde{p}_Y$ require to realize the convolution of $\tilde{p}_X$ over a limited number of points. State *(ii)* can be explained as follows. Let be a discrete function $\tilde{p}_X[i]$ for $i = 0, \ldots, N_s - 1$ that we convolve with itself to get $\tilde{p}_Y[i]$ with $i = 0, \ldots, 2N_s - 2$ (*i.e.* $Y = X_1 + X_2$ in this toy example). Then to obtain $\tilde{p}_Y[1]$, no matter how large is N_s, one needs to compute

$$\tilde{p}_Y[1] = 2\tilde{p}_X[0]\tilde{p}_X[1].$$

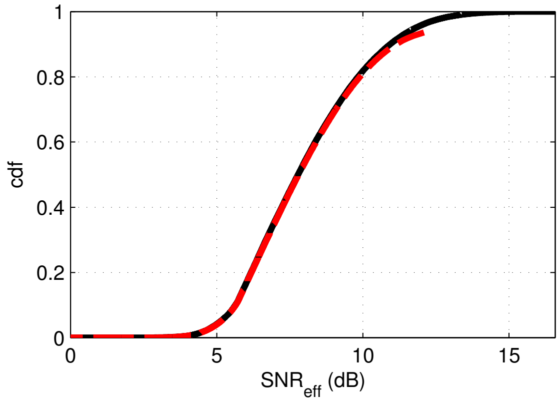


Figure 5.1: The black curve is the cdf of γ_{eff} obtained with Monte Carlo simulation using 10^5 realizations. The red curve is the cdf of γ_{eff} computed numerically with (5.26) with $N_s = 1000$.

Previous equation requires only two samples and provides the value $\tilde{p}_Y[1]$. In the same way, as long as $m < N_s - 1$, computing $\tilde{p}_Y[m]$ requires the $(m+1)$ first terms of $\tilde{p}_X$. Thus, one can compute the cdf of γ_{eff} in $\Gamma_{\max}$, by setting the sampling step $\Delta_y^{\max} = 1/N_s^{\max}$ such that $N'(\Gamma_{\max}) = 1$ in (5.27), *i.e.* $N_s^{\max} = \frac{N_{cc}}{N_c} e^{\frac{\Gamma_{\max}}{\beta}}$. Hence, $N'(\Gamma_{\max}) = 1$ and to compute (5.17) using the approximation (5.26), one needs solely to compute $\tilde{p}_Y$ in y_0 and y_1 whose convolutions require respectively the first and second term of $\tilde{p}_X$ as described previously.

However, note that the approximation (5.26) becomes accurate when $N'(\Gamma_{\max}) \gg 1$. Therefore, we propose to oversample with a factor $M \gg 1$ previous sampling such that the new sampling step is $\Delta_x = \Delta_y = \frac{1}{MN_s^{\max}}$. Consequently, $N'(\Gamma_{\max}) = \lfloor \frac{f(\Gamma)}{\Delta_y} \rfloor = \lfloor MN_s^{\max} f(\Gamma) \rfloor = M$. Equation (5.26) requires the computation of terms $\tilde{p}_Y[m]$ for $m = 0, \ldots, M$ which are obtained by convolving at most the M first terms of $\tilde{p}_X$. We have found that $M > 20$ provides good results.

Before showing numerical results, it remains to compute the initial point $\tilde{p}_X[0]$ in (5.23). According to Definition (5.23),

$$\tilde{p}_X[0] = \frac{1}{\Delta_x} - \sum_{i=1}^{N_s-1} p_X(x_i), \tag{5.30}$$

and when N_s is large computing the sum requires a lot of numerical resources (*e.g.* $N_s > 10^9$), therefore we look for an approximation of $\tilde{p}_X[0]$. With (5.21), and a regular sampling $\tilde{x} = (x_1, \ldots, x_{N_s-1})$ such that $\Delta_x = 1/N_s$, the initial point is given by

$$\tilde{p}_X[0] = \frac{1}{\Delta_x}\left(1 - \frac{\Delta_x}{\mathcal{B}(C_1; 1/C_2, 1)} \sum_{i=1}^{N_s-1} x_i^{1/C_2-1}\right). \tag{5.31}$$

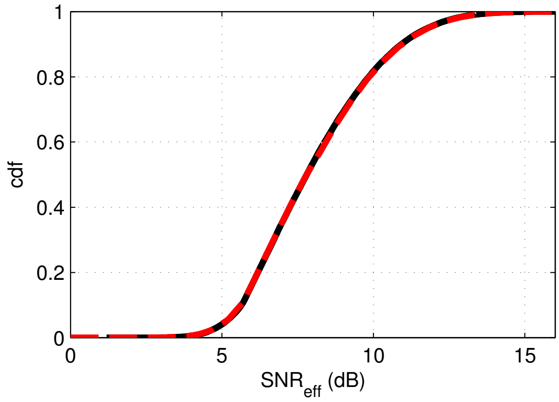


Figure 5.2: The black curve is the cdf of γ_{eff} obtained with Monte Carlo simulation using 10^5 realizations. The red curve is the cdf of γ_{eff} computed numerically with $N_s = 1000$.

One can rewrite $x_i = i\Delta_x$ so

$$\tilde{p}_X[0] = \frac{1}{\Delta_x}\left(1 - \frac{\Delta_x^{1/C_2}}{\mathcal{B}(C_1; 1/C_2, 1)} \sum_{i=1}^{N_s-1} i^{1/C_2-1}\right). \tag{5.32}$$

Using the following asymptotic approximation ($x < 1$):

$$\sum_{i=1}^{N_s-1} i^{x-1} \approx \frac{(N_s)^x}{x} + \zeta(1-x), \tag{5.33}$$

where $\zeta(.)$ is the Riemann zeta function, $\tilde{p}_X[0]$ can be approximated by,

$$\begin{aligned}\tilde{p}_X[0] \approx & \frac{1}{\Delta_x}\left(1 - \frac{\Delta_x^{1/C_2}}{\mathcal{B}(C_1; 1/C_2, 1)}\left(C_2 N_s^{1/C_2} + \zeta(1 - 1/C_2)\right)\right), \\ = & \frac{1}{\Delta_x}\left(1 - \frac{C_2}{\mathcal{B}(C_1; 1/C_2, 1)}\left(1 + \frac{\Delta_x^{1/C_2}\zeta(1-1/C_2)}{C_2}\right)\right).\end{aligned} \tag{5.34}$$

Finally, with the modified sampled density (5.23) and the computed initial point (5.34) and with the convolution procedure explained above, we can compute numerically the cdf of γ_{eff}. Figure 5.2 presents the convergence of our procedure with the same simulation conditions used to obtain results in Figure 5.1. One can observe that the cdf can be computed for all the SNR$_{\text{eff}}$ range.

This cdf is of importance to compute the Bernoulli parameters involved in the distribution of the utility (see (5.2)). The above procedure is developped in the case with no interference which means that the SNR over each subcarrier can be seen as a constant multiplied by an exponential rv. Thus this can be adapted to the case with interference that are not subject to fading (see (5.15)). In this case, one can use the above procedure using

$$\gamma = \frac{\mathrm{SNR}_i^k}{1 + \sum_{n\in\mathcal{I}_k}\sum_{j\in\mathcal{M}_n} \mathrm{INR}_{j,i_r}^{n,k}}, \tag{5.35}$$

to compute the Bernoulli parameters of rvs involved in the utility function and to get its distribution.

5.1.3.5 EESM statistics with interference subject to fading

In this section, we present how to compute the pdf of EESM in a more general case where the interference are subject to block Rayleigh fading. This case is more complex in the sense that at each subcarrier the SINR cannot be seen as a constant multiplied by a rv as in the previous section. This involves further computations. For clarity, with the block fading channel, we recall that the SINR of link i in cluster k over each block m of coherent subcarriers is given by (5.13) as follows

$$\mathrm{SINR}_i^k(m) = \frac{\alpha_i^k(m)\mathrm{SNR}_i^k}{1 + \sum_{n\in\mathcal{I}_k}\sum_{j\in\mathcal{M}_n} \alpha_{j,i_r}^{n,k}(m)\mathrm{INR}_{j,i_r}^{n,k}}, \tag{5.36}$$

where $\alpha_i^k(m)$ and $\alpha_{j,i_r}^{n,k}(m)$ are exponential rvs with parameter 1. The first step to get the density of EESM consists in computing the pdf of the SINR over each block of coherent subcarrier. We prove in Appendix E that this density, denoted by $p_{SINR_i^k}(.)$, is given by

$$p_{SINR_i^k}(\gamma) = \frac{1}{\mathrm{SNR}_i^k} e^{-\frac{\gamma}{\mathrm{SNR}_i^k}} \sum_{n\in\mathcal{I}_k}\sum_{j\in\mathcal{M}_n} \frac{\pi_j^n}{\mathrm{INR}_{j,i_r}^{n,k}} \frac{1}{\frac{\gamma}{\mathrm{SNR}_i^k} + \frac{1}{\mathrm{INR}_{j,i_r}^{n,k}}} \left(1 + \frac{1}{\left(\frac{\gamma}{\mathrm{SNR}_i^k} + \frac{1}{\mathrm{INR}_{j,i_r}^{n,k}}\right)}\right), \tag{5.37}$$

where

$$\pi_j^n = \prod_{n'\neq n\in\mathcal{I}_k}\prod_{j'\neq j\in\mathcal{M}_n} \frac{\mathrm{INR}_{j,i_r}^{n,k}}{\mathrm{INR}_{j,i_r}^{n,k} - \mathrm{INR}_{j',i_r}^{n',k}}. \tag{5.38}$$

The second step of the process to compute the pdf of EESM is to get the density of terms involved in the logarithms of EESM denoted by $X = C_1 e^{-\frac{SINR_i^k}{\beta}}$ with $C_1 = 1$ or $\frac{Ncc}{N_c}$ depending on which pdf is considered in the computation of (5.19) (X or X'). This pdf denoted by $p_X(.)$ can be obtained using the same change of variable as defined in (5.20) with $C_2 = \frac{1}{\beta}$. The proof is presented in Appendix E where we show that the pdf of X is given by

$$p_X(x) = \frac{\nu}{C_1^\nu} x^{\nu-1} \sum_{n\in\mathcal{I}_k}\sum_{j\in\mathcal{M}_n} \frac{\pi_j^n}{\mathrm{INR}_{j,i_r}^{n,k}} \frac{1}{\left(\frac{1}{\mathrm{INR}_{j,i_r}^{n,k}} - \nu\ln(\frac{x}{C_1})\right)} \left(1 + \frac{1}{\frac{1}{\mathrm{INR}_{j,i_r}^{n,k}} - \nu\ln(\frac{x}{C_1})}\right), \tag{5.39}$$

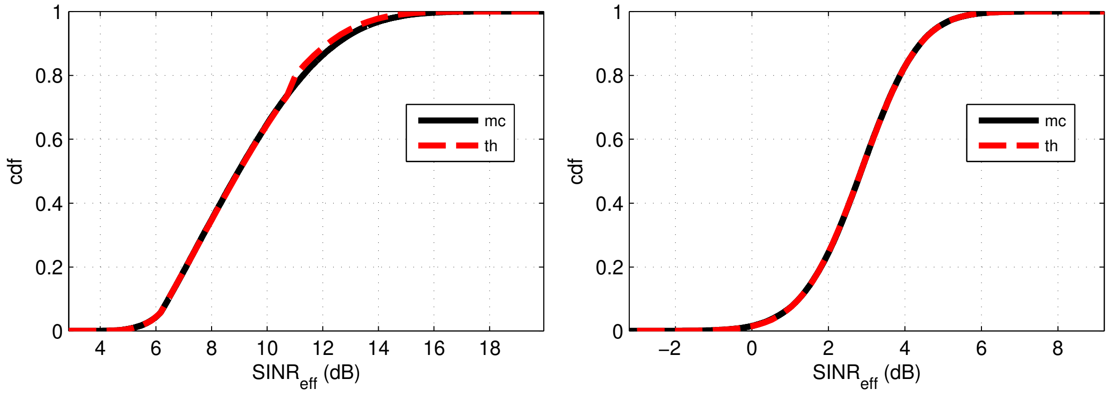
where $\nu = \frac{\beta}{\mathrm{SNR}_i^k}$. To compute the pdf of EESM one needs to convolve a sampled version of (5.39) denoted by $\tilde{p}_X[.]$ and defined as in (5.23). Note that this pdf is proportional to $\nu x^{\nu-1}$, as in (5.21), except that here there is a second multiplying term involving highly nonlinear combination of the interference. In the current case, the nonlinear term that involves the interference prevents us from finding an accurate approximation of $\tilde{p}_X[0]$ (as in (5.30) when the interference are not subject to Rayleigh fading) for large N_s. It follows that, with this method, computing the cdf of γ_{eff} at high SNR per subcarrier remains an open question.

However, we propose the following trick to approach the cdf at high effective SINR. When the number of sample N_s is too large to compute the term $\tilde{p}_X[0]$ in (5.23), we propose to replace it by the term we would have obtained if the interference were no subject to Rayleigh fading. Hence, $\tilde{p}_X[0]$ is replaced by (5.34) with $C_1 = 1$ or $\frac{N_{cc}}{N_c}$ and $C_2 = \frac{\mathrm{SINR}^*}{\beta}$ with $\mathrm{SINR}^* = \frac{\mathrm{SNR}_i^k}{1+\sum_{n\in\mathcal{I}_k}\sum_{j\in\mathcal{M}_n}\mathrm{INR}_{j,i,r}^{n,k}}$. We justify this approach as follows. At high effective SINR, the INR are much lower than the SNR, thus, the impact of fading on the interference is lower than the impact of fading on the useful signal. Therefore, we neglect the fading that impact the interference and we approximate the SINR per subcarrier as a Rayleigh rv multiplied by SINR^*.

Figure 5.3 presents in two cases the comparison between the cdf of EESM obtained through Monte Carlo simulations in black and with the numerical method proposed in this section in red. In both case, the SNR = 25 dB. The difference are the number of interfering signals and their INRs. In Figure 5.3a there are two interference sources with INR = $[1, 5]$ dB and in Figure 5.3b there are 5 interference sources with INR = $[1, 2, 5, 10, 20]$ dB. One can note that, there are less interference in the left figure than in the right one so, the effective SINR is, in average, higher in the left figure than in the right one. Hence, we can observe on the left figure the slight gap between the two curves (for $\mathrm{SINR}_{\text{eff}} > 11$ dB) due to the use of the proposed trick to approximate the cdf at high effective SINR. In addition, we observe that the proposed modification leads to an accurate approximation of the cdf. In Figure 5.3b the cdf of the effective SINR is computed without the use of the approximation and leads to a close to perfect match. In both case, the accuracy of the numerical approach is illustrated and assessed.

5.2 Proposed methodology to find N_{cc}

In this section, we present a methodology to find N_{cc} such that the statistics of the proposed model γ_{eff} (5.12) obtained with the block fading channel are similar



(a) 2 interferers

(b) 5 interferers

Figure 5.3: cdf of EESM when interference are subject to Rayleigh fading with different number of interferer and $N_b = 10$.

to the statistics of $\mathrm{SNR}_{\mathrm{eff}}$ (5.11) obtained with the real channel (5.8). The idea is that if we are able to reproduce the statistics of $\mathrm{SNR}_{\mathrm{eff}}$ using the coherence bandwidth approximation (with the right value of N_{cc}) then we can use γ_{eff} to compute statistics of the effective SNR and of the utility. Therefore, N_{cc} is selected such that the cdf of both models are as close as possible. This is realized with a Goodness of Fit (GoF) test which is a procedure comparing the cdf of two rvs. The principle of this procedure is presented in Figure 5.4. We consider a standard OFDM transmission with a bit generator, a convolutional code ("CC") and the standard OFDM procedure. Then, the signal goes through a channel which is either the real channel or the block fading channel whose length of blocks N_{cc} is the key parameter to set. A white Gaussian noise is added to the signal such that the average SNR per subcarrier is set to γ. With the resulting signal we compute the effective SNR metric using EESM. If we repeat this procedure, one can obtain the cdf of $\mathrm{SNR}_{\mathrm{eff}}$ obtained through the real channel, and the cdf of γ_{eff} obtained through the block fading channel. Then, it remains to adjust N_{cc} such that both cdf are as close as possible, which is the role of the GoF test presented in Section 5.2.3.

5.2.1 Procedure

The methodology to select N_{cc} is detailed in Algorithm 5 and is summarized as follows. The first step consists in drawing N_{snr} realizations of the real channel to compute N_{snr} values of $\mathrm{SNR}_{\mathrm{eff}}(\gamma)$ for a given γ. The number of samples N_{snr} must be large enough to consider the empirical cdf as the theoretical reference (see Section 5.2.3). Then, we draw N_{snrmod} realizations of the block fading channel

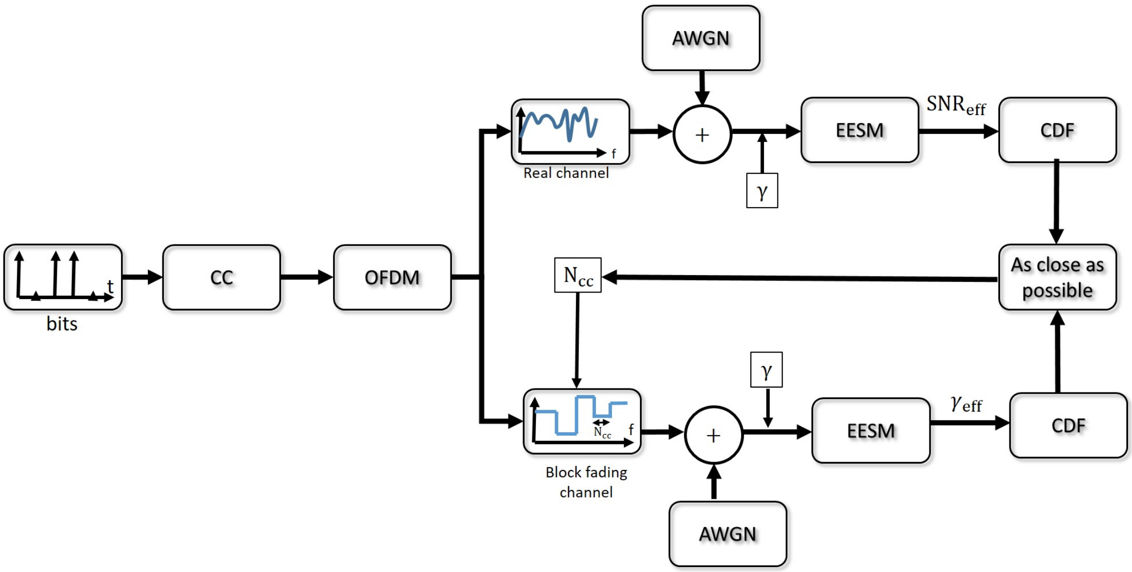


Figure 5.4: Principle of the procedure to select N_{cc}.

for a given N_{cc} to compute N_{snrmod} values of $\gamma_{\text{eff}}(\gamma)$ less than a threshold b whose role is detailed in Section 5.2.2. We repeat this procedure N_ω times for N_{cc} between 1 and N_c. Afterwards, we proceed to the statistical test presented afterward in Section 5.2.3. Finally, for every realization i_ω, we find N_{cc} between 1 and N_c that minimizes the metric of the test and, we average over all N_ω realizations.

5.2.2 SNR bound b computation

This section explains the reason to introduce the threshold b in Algorithm 5 and proposes a method to compute it. We have observed through simulations particular cdf behaviors of SNR_{eff} statistics at high SNR_{eff}. Figure 5.5 compares the cdf of SNR_{eff} (black curve) with the cdf of γ_{eff} (red curve). The real channel is a standardized channel "Typical case for urban area" (TU) with 6 taps from [66]. The bandwidth of the signal is $W_s = 1$ MHz with $N_c = 256$, $\gamma = 20$ dB and $\beta = 1.8$ (*i.e.* QPSK modulation). Clearly, the red curve cannot fit the black curve for SNR_{eff} higher than 10 dB.

In addition, if we wanted to realize the statistical test in line 5 of Algorithm 5 over all the SNR_{eff} range, it would return a value N_{cc} that makes both cdf as close as possible over the whole range of SNR_{eff} as illustrated in Figure 5.6. This is an unwanted result because the cdf of γ_{eff} obtained with the block fading channel would never fit correctly the cdf of SNR_{eff}.

Therefore, in order to model the performance, (*e.g.* PER prediction) the distribution of SNR_{eff} at low SNR specifies the most important part of the PER value

Algorithm 5 Compute the number of coherent carriers N_{cc}

```
Input: $\gamma, N_c, N_{snr}, N_\omega, N_{snrmod}, b$
Output: $N_{cc}$
SNR$_{\text{eff}}(\gamma) \leftarrow$ generate $N_{snr}$ samples of effective SNR with the real channel that are sorted ($N_{snr}$ must be large e.g. 40000)
for $i_\omega = 1$ to $N_\omega$ do {Test over all realizations}
    for $i_{cc} = 1$ to $N_c$ do {Try all possible coherence bandwidths}
        $\gamma_{eff}(\gamma, i_{cc}) \leftarrow$ generate $N_{snrmod}$ samples of effective SNR less than $b$ with the channel model,
        $T_{\text{KS}}(i_\omega, i_{cc}) \leftarrow$ statistical comparison between $\gamma_{eff}(\gamma, i_{cc})$ and SNR$_{\text{eff}}(\gamma)$ ((5.46) in Section 5.2.3) for values less than $b$,
    end for
    $N'_{cc}(i_\omega) \leftarrow \arg\min(T_{\text{KS}}(i_\omega, :))$,
end for
$N_{cc}(\gamma) \leftarrow \frac{1}{N_\omega}\sum_{i_\omega=1}^{N_\omega} N'_{cc}(i_\omega)$
return $N_{cc}(\gamma)$
```

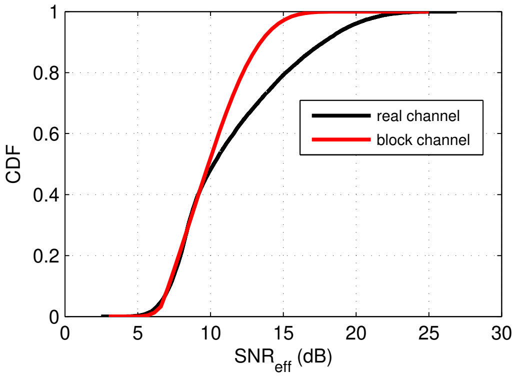


Figure 5.5: Comparison of the cdf of SNR$_{\text{eff}}$ and γ_{eff} at high γ for $N_{cc} = 20$.

[67]. We are hence interested in fitting the beginning of the cdf with our model as presented in Figure 5.5. In this specific case, one can see that the threshold b to limit the test is around 10 dB.

In the following, we first study the cause of this specific behavior in Section 5.2.2.1 in order to propose a procedure that returns a given threshold b in Section 5.2.2.2.

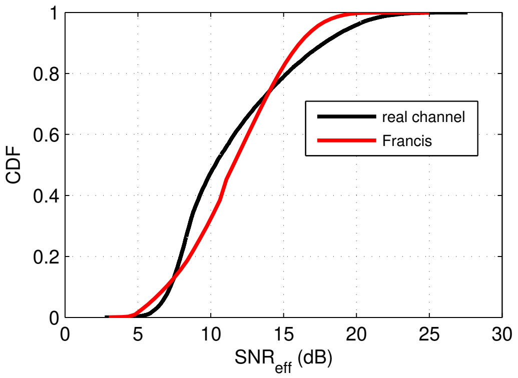


Figure 5.6: Comparison of the cdf of SNR_{eff} and γ_{eff} at high γ for $N_{cc} = 50$.

5.2.2.1 cdf mismatch at high SNR

One of the reason of the possible mismatch between the cdf of SNR_{eff} and γ_{eff} at high SNR_{eff} is due to the combination of the exponential decrease of the terms inside the logarithm of the EESM (5.11) and the correlation between channel coefficients in the real channel model.

The exponential terms decrease fast to zeros as the exponent decreases (it is a negative exponent). Therefore, for low average SNR γ, the exponents are close to 0 and all terms in the logarithm of EESM have a low probability to be null simultaneously. Consequently, they all have a statistical contribution in the density of EESM. On the other hand, for high γ, the exponential terms are not null only when the fading coefficient is close to 0, which happens more rarely as γ increases. Consequently, a few terms in the logarithm of EESM have a strong influence on the statistical behavior of EESM. As one will see in the following, this phenomenon prevents the EESM statistics from being approximated, for high γ, by either models in the literature as in [62] or our model.

This phenomenon is highlighted in Figure 5.7 where the pdf of SNR_{eff} is presented in two cases. We consider a TU channel and the same parameter as in the introduction of this section. In the first case in black, we compute SNR_{eff} with all the exponential terms as in (5.11) as follows

$$\text{SNR}_{\text{eff}}(\gamma) = -\beta \log\left(\frac{1}{N_c}\sum_{i=1}^{N_c} e^{-\frac{|H(f_i)|^2\gamma}{\beta}}\right). \tag{5.40}$$

In the second case in red, a different version of SNR_{eff} is computed, where we keep in the logarithm the three largest terms at each realization of the channel. It is given by

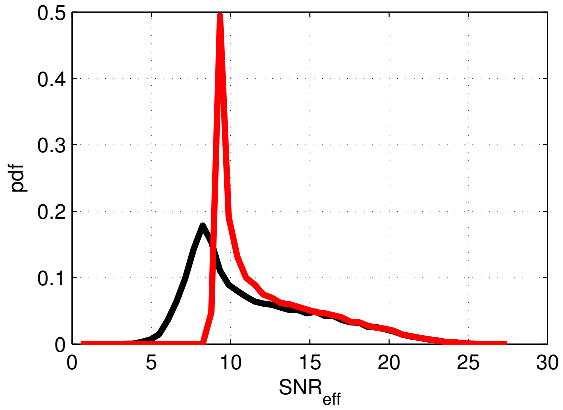


Figure 5.7: Comparison of EESM pdfs considering all terms (black curve) and the three most influential terms (red curve) into the logarithm of (5.11).

$$\mathrm{SNR}_{\mathrm{eff}} = -\beta \log\left(\frac{1}{N_c}\sum_{j=1}^{3} e^{-\frac{\mathrm{SNR}_j^*}{\beta}}\right), \tag{5.41}$$

where SNR_j^* for $j = 1$ to 3 are the three lowest SNR among all $\mathrm{SNR}_i = |H(f_i)|^2\gamma$ in (5.40) for $i = 1$ to N_c. Note that the three most influential terms in the logarithm of (5.40) have the smallest channel fading coefficients. It is clear that the statistical behavior of EESM changes around 10 dB. In addition, for $\mathrm{SNR}_{\mathrm{eff}} > 15$ dB the distribution can be described with a limited number of subcarriers (less than 3 with the red curve), whereas for $\mathrm{SNR}_{\mathrm{eff}} < 15$ dB, one need more subcarriers to describe completely the distribution.

Consequently, we assume that the sudden change in the statistic behavior around 10 dB observed in Figures 5.5 and 5.7 is due to the fact that the number of terms in the logarithm of (5.40) that have statistical impact is less than N_c.

The exponential behavior of terms inside the logarithm explains the sudden change in the pdf but not the mismatch between the cdf of γ_{eff} and $\mathrm{SNR}_{\mathrm{eff}}$. This mismatch is due to the correlation between consecutive subcarriers when considering the real channel. Let take the example in which EESM can be described with the two most influential subcarriers. Then (5.11) becomes

$$\mathrm{SNR}_{eff}(\gamma) = -\beta \log\left(\frac{1}{N_c}\left(e^{-\frac{\nu_1^*\gamma}{\beta}} + e^{-\frac{\nu_1^*\gamma}{\beta}}\right)\right), \tag{5.42}$$

where ν_1^* and ν_2^* are the two lowest channel coefficients. In a WB case, fading coefficients are correlated thus, the two lowest coefficients are necessarily correlated as well. With the block fading channel, EESM in (5.12) becomes

$$\gamma_{\mathrm{eff}}(\gamma) = -\beta \log\left(\frac{1}{N_c}\left(N_{cc}e^{-\frac{\mu_1^*\gamma}{\alpha}} + N_{cc}e^{-\frac{\mu_1^*\gamma}{\beta}}\right)\right), \tag{5.43}$$

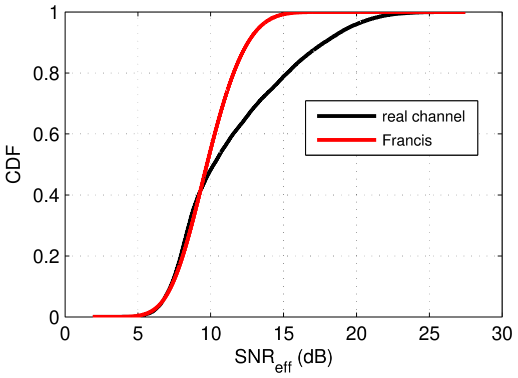


Figure 5.8: Comparison with model proposed by [62]

where N_{cc} is the number of coherent subcarriers that are in the same coherence bandwidth and μ_1^* and μ_2^* are the two lowest channel coefficients. Even if N_{cc} goes to 1, the remaining two terms in (5.43) are independent whereas in (5.42) they are highly correlated. This shows the limitation for the proposed approach to fit the curves at high effective SNR.

Before getting into details of the procedure to set b it is interesting to show that one of the approximations found in the literature (that are all principally based on moment matching methods) has a similar limitation. The existing approaches in the literature try to approximate EESM statistics by **one** given distribution. These distributions are well adapted to approximate the distribution of EESM for low SNR_{eff} (or high SNR_{eff}) but not for the entire range of SNR_{eff} as it exhibits two different statistical behaviors. One would need a mixture of distributions to do so. Figure 5.8 presents the cdf of EESM obtained with the model in [62] in red and with the real channel in black. One can see again the impossibility for the approximation to fit the black curve at high SNR_{eff}.

5.2.2.2 Procedure to compute b

In this section, we propose a procedure to set the threshold b. In the previous section, we have highlighted that this threshold separates the SNR_{eff} domain in two regions. Below b, one needs the N_c terms in the logarithm of (5.40) to describe the statistical behavior of SNR_{eff}. Above b, less than N_c terms in the previous logarithm are necessary to describe the statistical behavior of SNR_{eff}.

The procedure to select b is presented in Algorithm 6 and is summarized as follows. The objective is to detect when the statistical behavior of SNR_{eff} can be described completely with $(N_c - 1)$ terms in the sum in the logarithm of (5.40). We start by drawing a real channel and then we compute SNR_{eff} taking into account all channel coefficients $|H(f_i)|^2$ as in (5.40) that we note $\mathrm{SNR}_{\mathrm{eff}}^{(1)}(\gamma)$. Then, we compute a second $\mathrm{SNR}_{\mathrm{eff}}$, noted $\mathrm{SNR}_{\mathrm{eff}}^{(2)}(\gamma)$ using the same realization of the channel. The particularity is that we withdraw the term with the highest channel coefficient that we call j here. This effective SNR is given by

$$\mathrm{SNR}_{\mathrm{eff}}^{(2)}(\gamma) = -\beta \log\left(\frac{1}{N_c}\sum_{i=1,\neq j}^{N_c} e^{-\frac{|H(f_i)|^2\gamma}{\beta}}\right). \tag{5.44}$$

With these two values, we compute an error term $\epsilon(\mathrm{SNR}_{\mathrm{eff}}^{(1)}) = |10\log_{10}(\mathrm{SNR}_{\mathrm{eff}}^{(1)}(\gamma)) - 10\log_{10}(\mathrm{SNR}_{\mathrm{eff}}^{(2)}(\gamma))|$. If the statistic of the $\mathrm{SNR}_{\mathrm{eff}}$ can be described with less subcarrier than N_c, the error will be close to 0. This procedure is repeated N_{snr} times. Then we look for the minimum $\mathrm{SNR}_{\mathrm{eff}}^{(1)}$ such that this error decreases below a given threshold that we have set to $\tau = 10^{-6}$.

Algorithm 6 Compute the threshold b

```
Input: $\gamma, \tau, N_c, N_{snr}$,
Output: $b$
$\mathrm{SNR}_{\mathrm{eff}}^{(1)}(\gamma) \leftarrow$ generate $N_{snr}$ samples of effective SNR with the real channel that are sorted ($N_{snr}$ must be large e.g. 40000)
$\mathrm{SNR}_{\mathrm{eff}}^{(2)}(\gamma) \leftarrow \mathrm{SNR}_{\mathrm{eff}}^{(1)}(\gamma)$ except that the less influential subcarrier has been withdrawn in the sum of (5.11) (see (5.44))
$\epsilon \leftarrow |10\log_{10}(\mathrm{SNR}_{\mathrm{eff}}^{(1)}(\gamma)) - 10\log_{10}(\mathrm{SNR}_{\mathrm{eff}}^{(2)}(\gamma))|$
$b \leftarrow \min\left\{\mathrm{SNR}_{\mathrm{eff}}^{(1)} \middle| \epsilon < \tau\right\}$
return $b$
```

5.2.3 GoF test

GoF tests are non-parametric hypothesis tests whose objective is to compare the distribution of a set of data with a theoretical distribution. The main advantages of these tests are their selectivity with a few data samples in addition to their simplicity. The test compares a theoretical cdf that one wants to fit with the empirical cdf of the data set. In this work in Line 5 of Algorithm 5, γ_{eff} is considered as the data sample and $\mathrm{SNR}_{\mathrm{eff}}$ as the theoretical data to fit. Among the different GoF tests, we find the Kolmogorov-Smirnov (KS) one the more appropriate since we compute the test on a bounded SINR domain (up to b; see the previous section).

The empirical cdf of a given set $(x_1, \ldots, x_N)$ being defined by

$$F_N(y) = \sum_{i=1}^{N} \mathbb{1}_{\{x_i \leq y\}}, \tag{5.45}$$

where N is the number of sample in the data set, the KS metric is given by [68]

$$T_{\mathrm{KS}} = \sup_{y} |F_N(y) - F(y)|, \tag{5.46}$$

where $F(.)$ is the theoretical cdf to fit.

Notice that although in GoF tests this metric is compared to a given threshold, for our procedure we use only the metric value.

5.2.4 Average PER computation

This section presents how to estimate the theoretical prediction of PER in order to validate the procedure to select N_{cc}. This prediction uses the approximation of the pdf p_Y, $\tilde{p}_Y$ (5.24), that is obtained in Section 5.1.3.4. First, we start with the definition of the average PER given by

$$\mathrm{PER}(\gamma) = \int_0^{\infty} \mathrm{PER}_{\mathrm{LUT}}(z) p_{\gamma_{\mathrm{eff}}(\gamma)}(z) dz, \tag{5.47}$$

where $p_{\gamma_{\mathrm{eff}}(\gamma)}(z)$ is the density of $\gamma_{\mathrm{eff}}(\gamma)$ and, $\mathrm{PER}_{\mathrm{LUT}}(z)$ is the LUT of the codeword error rate [65]. Note that, we are able to compute a numerical approximation of $p_Y(y)$ (5.19) which is linked to γ_{eff} as follows $y = f(\gamma_{\mathrm{eff}}) = \frac{N_c}{N_{cc}} e^{-\frac{\gamma_{\mathrm{eff}}}{\beta}}$ (5.18). With the change of variable $y = \frac{N_c}{N_{cc}} e^{-\frac{z}{\beta}}$ and $|dy| = \frac{N_c}{N_{cc}\beta} e^{-\frac{z}{\beta}} |dz|$, (5.47) becomes

$$\mathrm{PER}(\gamma) = \int_0^{\infty} \mathrm{PER}_{\mathrm{LUT}}(z) p_Y\left(\frac{N_c}{N_{cc}} e^{-\frac{z}{\beta}}\right) \frac{N_c}{N_{cc}} \frac{1}{\beta} e^{-\frac{z}{\beta}} dz. \tag{5.48}$$

It is known that the variations of the effective SNR are well represented on a logarithmic scale. In addition, the LUT curve is often provided in a decibel scale rather than in a linear one. This means that $\mathrm{PER}_{\mathrm{LUT}}(z)$ is not available and we have instead $\mathrm{PER}_{\mathrm{LUT}}^{dB}(v)$ where $v = 10\log_{10}(z)$. Using $z = 10^{\frac{v}{10}}$ and the fact that $|dz| = \frac{\ln(10)}{10} 10^{\frac{v}{10}} |dv|$, (5.48) becomes

$$\mathrm{PER}(\gamma) = \int_{-\infty}^{\infty} \mathrm{PER}_{\mathrm{LUT}}^{\mathrm{dB}}(v) \frac{\ln(10)}{10} 10^{\frac{v}{10}} p_Y\left(\frac{N_c}{N_{cc}} e^{-\frac{10^{\frac{v}{10}}}{\beta}}\right) \frac{N_c}{N_{cc}} \frac{1}{\beta} e^{-\frac{10^{\frac{v}{10}}}{\beta}} dv. \tag{5.49}$$

In practice, previous formula is computed over a restricted range of effective SNR denoted by $[\mathrm{SNR}_{\mathrm{eff,dB}}^{\min}, \mathrm{SNR}_{\mathrm{eff,dB}}^{\max}]$ here, therefore (5.49) is approached by

$$\mathrm{PER}(\gamma) \approx \int_{\mathrm{SNR}_{\mathrm{eff,dB}}^{\min}}^{\mathrm{SNR}_{\mathrm{eff,dB}}^{\max}} \mathrm{PER}_{\mathrm{LUT}}^{\mathrm{dB}}(v) \frac{\ln(10)}{10} 10^{\frac{v}{10}} p_Y\left(\frac{N_c}{N_{cc}} e^{-\frac{10^{\frac{v}{10}}}{\beta}}\right) \frac{N_c}{N_{cc}} \frac{1}{\beta} e^{-\frac{10^{\frac{v}{10}}}{\beta}} dv. \tag{5.50}$$

Then, we approximate previous formula numerically using (5.24) as follows,

$$\mathrm{PER}(\gamma) \approx \frac{\ln(10)}{10} \frac{N_c}{N_{cc}} \frac{1}{\beta} \sum_{i=0}^{N_v - 1} \mathrm{PER}_{\mathrm{LUT}}^{\mathrm{dB}}(v_i) 10^{\frac{v_i}{10}} p_Y\left(\frac{N_c}{N_{cc}} e^{-\frac{10^{\frac{v_i}{10}}}{\beta}}\right) e^{-\frac{10^{\frac{v_i}{10}}}{\beta}} \Delta_v, \tag{5.51}$$

Table 5.1: MCS

MCS	coding	modulation
1	CC 1/2	QPSK
2	CC 1/2	16-QAM
3	CC 3/4	16-QAM

Table 5.2: Calibrated β

MCS	1	2	3
$N_c = 512$	1.8	6.4	8.25

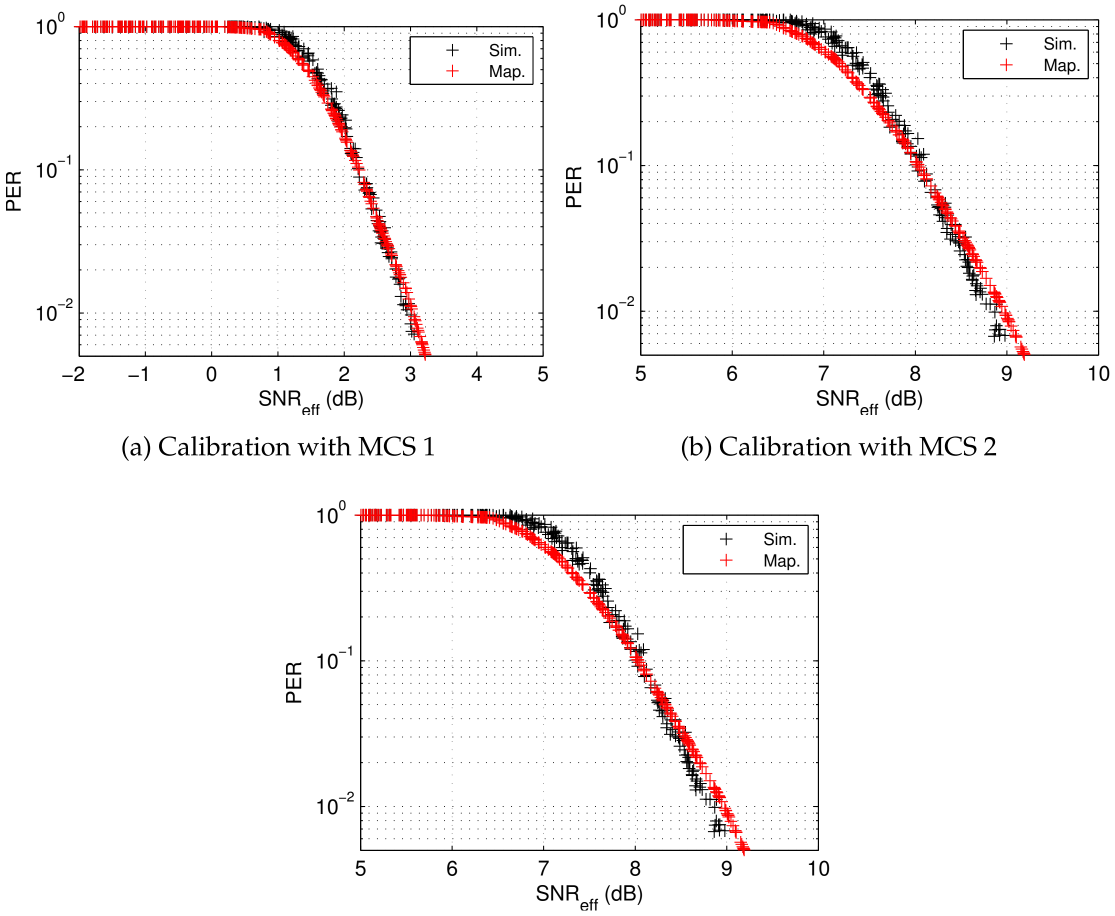
where $\boldsymbol{v} = (v_0, \ldots, v_{N_v-1})$ is regular sampling of the effective SNR in decibels with $v_0 = \mathrm{SNR}^{\min}_{\mathrm{eff,dB}}$, $v_{N_v-1} = \mathrm{SNR}^{\max}_{\mathrm{eff,dB}}$ and $v_{k+1} = v_k + \Delta_v$ where $\Delta_v = \frac{\mathrm{SNR}^{\max}_{\mathrm{eff,dB}} - \mathrm{SNR}^{\min}_{\mathrm{eff,dB}}}{N_v}$.

5.3 Numerical results

In this section, we analyze the numerical results obtained by applying the algorithm 5. OFDM symbols are coded independently and, the interleaver is modified randomly at every packet. We implement the two different convolutional codes described in [69]: a rate 1/2 and a rate 3/4 with a conventional punctured pattern [55]. The number of subcarriers is $N_c = 512$ and, the signal bandwidth is $W_s = N_c \Delta f = 10$ MHz. The numerical results are obtained for three MCSs that are presented in table 5.1. We consider a standardized channel TU with 6 taps described in [66]. In this case, $\mathbf{a} = [-3, 0, -2, -6, -8, -10]$ dB are the relative paths' powers and $\boldsymbol{\tau} = [0, 0.2, 0.5, 1.6, 2.3, 5]$ μs are the paths' delays.

5.3.1 Calibration

The first step is to calibrate the scaling factor β with the method in [65] presented in Algorithm 4. Figure 5.9a presents the result of β calibration for the three MCSs described in Table 5.1. The black curve, denoted by "Sim." is the real PER obtained for a fixed $\mathrm{SNR}_{\mathrm{eff}}$ and the red curve is the PER predicted for this $\mathrm{SNR}_{\mathrm{eff}}$ using the LUT curve defined in Section 5.1.3.3. The calibration of β makes these curves as close as possible. The calibration results are presented in Table 5.2 for each MCS.



(a) Calibration with MCS 1

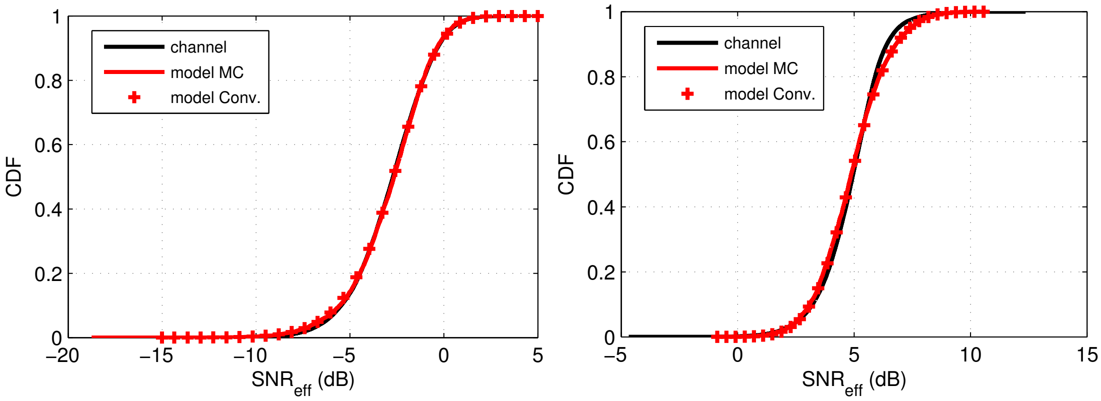
(b) Calibration with MCS 2

(c) Calibration with MCS 3

Figure 5.9: Calibration for all MCS.

5.3.2 Application of the procedure to a standardized channel

We apply the procedure described in Algorithm 5 with $N_{snr} = 20000$, $N_{\omega} = 2000$ and $N_{snrmod} = 100$. Figures 5.10a and 5.10b present an example of a comparison between the cdfs obtained with the channel and with the model for the selected N_{cc} with MCS 1. The value N_{cc} are presented afterwards. The cdf of the model is generated with Monte Carlo method denoted by "model MC", and with the convolution method described in Section 5.1.3.4 denoted by "model Conv.". First of all, one can observe that the Monte Carlo cdf and the numerical cdf are almost similar which assesses the accuracy of the convolution method. The step size taken is $\Delta x = 10^{-3}$. Second, the choice of N_{cc} makes the cdf obtained by the model very close to the real cdf of the channel (the black curve). At low γ, in Figure 5.10a, the matching is almost perfect for the whole range of SNR$_{\text{eff}}$. At higher γ, in Figure 5.10b, the matching starts to be less accurate at high SNR$_{\text{eff}}$ because of the



(a) $\gamma = -2$ dB

(b) $\gamma = 9$ dB

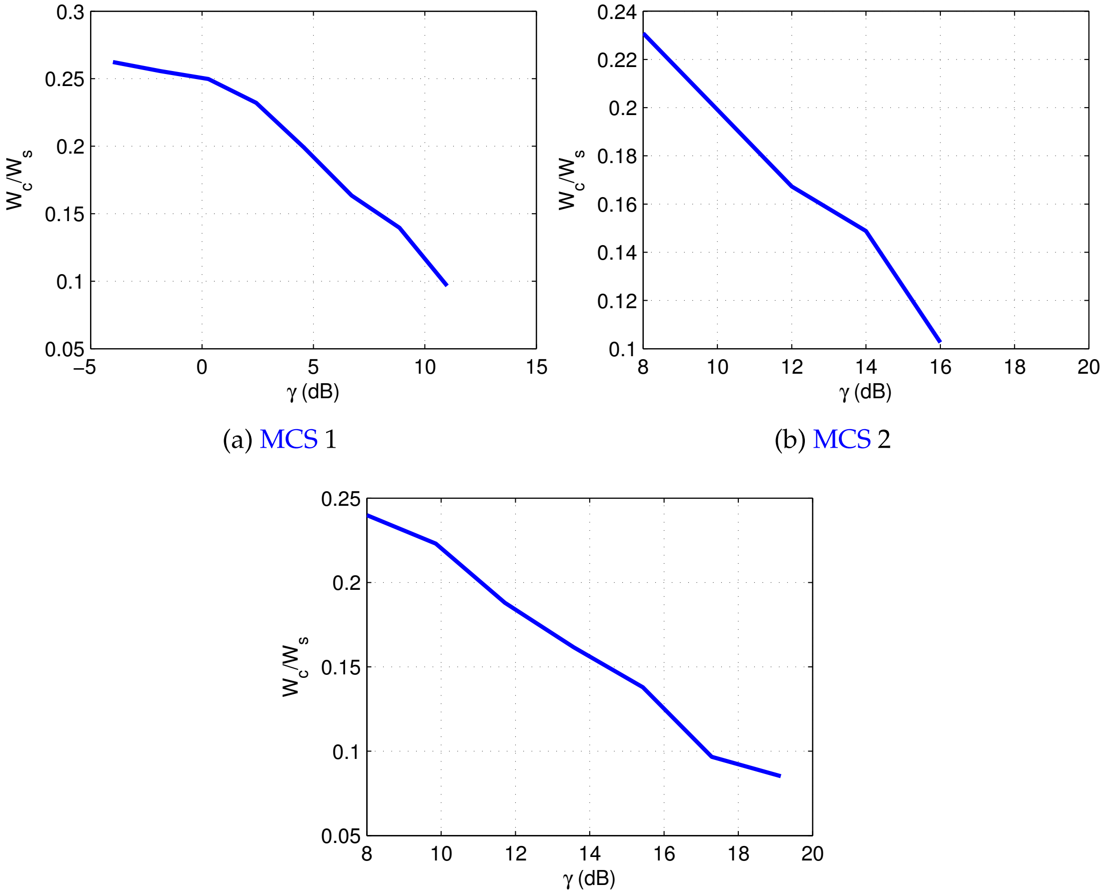
Figure 5.10: cdf comparison of EESM for different average SNR values with MCS 1.

limitation of the proposed approximation as explained in Section 5.2.2.

Let $W_c = \Delta f N_{cc}$ be the bandwidth of each block of subcarriers which is supposed to represent the coherence bandwidth of the channel. Then Figures 5.11a, 5.11b and 5.11c present the equivalent relative coherence bandwidth $\frac{W_c}{W_s} = \frac{N_{cc}}{N_c}$ found by the Algorithm 5 for the three MCSs. Surprisingly, these figures show that the number of coherent subcarriers decreases when the per-carrier SNR increases. One would have thought that the estimated coherence bandwidth is a constant with respect to the average SNR. Instead, the bandwidth of each block varies between 0.1 and 0.25 times the signal bandwidth. In addition, at low γ, the number of subcarriers to be considered in the same bandwidth seems to converge to a constant value a least with MCS 1. One of the possible reasons to observe this change in coherence bandwidth is that the larger γ, the larger the variations of the SNR between consecutive subcarriers. Consequently, close subcarriers appear to be more independent from a statistical perspective and the estimated coherence bandwidth decreases.

5.3.3 Validation of the proposed methodology to select the coherence bandwidth

In this section, we compare the performance of the point-to-point communication obtained with the TU channel and with the block fading channel using N_{cc} subcarriers as length of blocks that are selected with Algorithm 5. In addition, we compare previous performance obtained with Monte Carlo simulations with the



(a) MCS 1

(b) MCS 2

(c) MCS 3

Figure 5.11: Normalized coherence bandwidth estimation using algorithm 5 with $N_c = 512$.

numerical computation of the PER presented in Section 5.2.4. The procedure to valid the methodology to compute N_{cc} is presented in Figure 5.12 and composed of three parts. There are three illustrated chains denoted by (a), (b) and (c). Chain (a) represents the PER computation through Monte Carlo simulations when considering the TU channel and whose result is denoted by $\mathrm{PER}_{\mathrm{RC}}$. The block *CC* means that we use a Convolutional Code and CC^{-1} represents the decoding procedure. The block *OFDM* illustrates the standard OFDM procedure applied to the signal and $OFDM^{-1}$ the standard associated demodulation. The first block of the chain are the bits transmitted. The block with the real channel is a representation of the TU channel in frequency. Note that we add a white Gaussian noise such that the average SNR per subcarrier is set to γ. Thus, the PER obtained in the end depends on γ. The second chain (b) is similar to the first chain except that the real channel is replaced with the block fading channel for a specific length of blocks given by N_{cc}

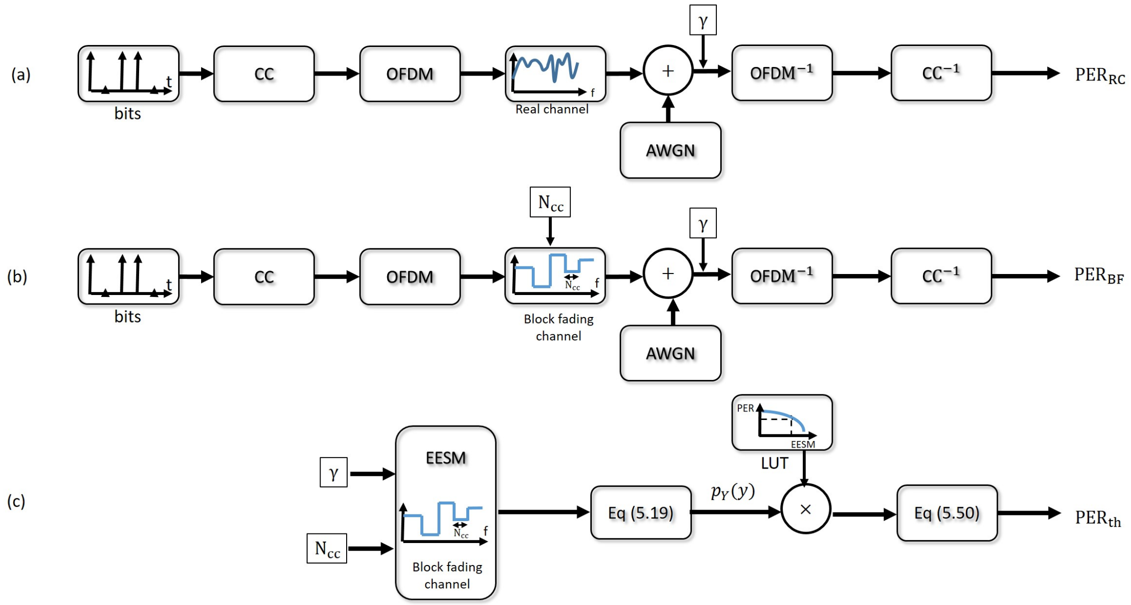
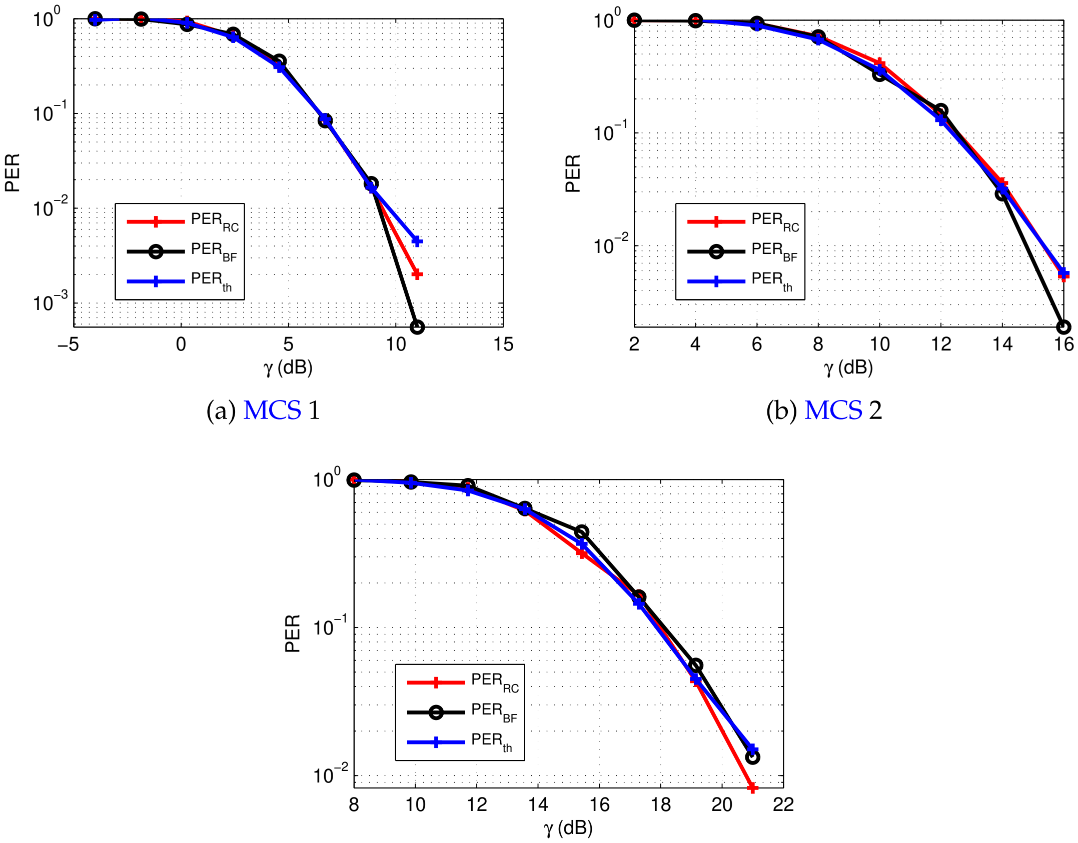


Figure 5.12: Validation of the procedure scheme.

consecutive subcarriers. This length is computed using the methodology given in Section 5.2.1. The resulting PER is noted PER_{BF}. Note also that the average SNR per subcarrier is set to γ as in chain (a). The last chain (c) represents the numerical computation whose steps are detailed in Section 5.2.4. The resulting PER is noted PER_{th}. As in (b), the length of blocks in the channel are set to N_{cc} consecutive subcarriers and the average SNR per subcarrier is set to γ. With the EESM metric and the specific channel model we can use a convolution technique to obtain the pdf of Y as described in Section 5.1.3.4. Note that we assume the EESM metric to be calibrated with the procedure presented in Section 5.1.3.3 (*i.e.* β is set). Then after multiplying this pdf with the LUT curve (see Section 5.1.3.3 in page 105 for details about LUT) it remains to integrate over the effective SNR to obtain the desired PER.

Figures 5.13a, 5.13b and 5.13c present the PER comparison for the three MCSs considered (see Table 5.1). The red curves are obtained through Monte Carlo simulations with the TU channel as presented in chain (a) in Figure 5.12. In this case, the SNR over each subcarrier is computed using (5.9) in page 104. The dark curves are also obtained through Monte Carlo simulations where the fading is flat over blocks of N_{cc} consecutive subcarriers and independent between blocks as presented in chain (b) in Figure 5.12. The length of blocks, N_{cc}, for the TU channel are provided in Section 5.3.2 and depends on γ. The blue curves are obtained using the theoretical approach described in Section 5.2.4 and illustrated in chain (c) in Figure 5.12. For the three MCSs, the three methods presented are



(a) MCS 1

(b) MCS 2

(c) MCS 3

Figure 5.13: PER comparison with the channel and with the model for the three MCSs.

very close, which confirms the accuracy of the proposed method to select the coherence bandwidth and the validity of the numerical method to predict the performance.

Note that there is a slight underestimation of the PER at higher SNR for MCSs 1 and 2. For large values of γ, the statistical test, presented in Section 5.2.3, is proceeded over a restricted range of effective SNR to avoid fitting problems (see Section 5.2.2). When γ becomes large, the range over which the test is proceeded decreases and the proposed approximation becomes less accurate which explains the gap observed.

5.3.4 Analysis of the pdf of the utility

The goal of this section is threefold, *(i)* we check if the pdf of the utility obtained by simulation is close to the one obtained theoretically (we consider both the NB and the WB fading cases), *(ii)* if the answer to *(i)* in the WB fading case is positive, we check if the modeling of the real channel by the block fading channel with the right value of N_{cc} provides similar pdfs of the utility, *(iii)* we highlight the difference on the pdfs of the utility between the case where the interference are not subject to fading, that we name "simplified" model in the sequel, and the case where the interference are subject to fading, that we name "full" model in the sequel. We illustrate these results using the network presented in Figure 5.14. More specifically, we observe the pdf of the utility at cluster 7 in three cases denoted by *Scenario 1*, *Scenario 2* and *Scenario 3* in the sequel. These scenarios are illustrated in Figure 5.15 where a schematic view of the network in Figure 5.14 is presented with the respective clusters' indices. In the three figures, the green square represents the cluster at which we observe the utility and the red boxes are the interfering clusters. In Scenario 1, cluster 7 is interfered by clusters 4 and 8 which we can be seen as a bad channel allocation, in Scenario 2, cluster 7 is interfered by clusters 4 and 3 which can be seen as an average scenario as one of the interferer is farther in comparison to Scenario 1, finally, in Scenario 3, cluster 7 is interfered by cluster 1 and 3 which is a better resource allocation than in Scenario 1 and 2.

In the sequel, when the pdf of the utility is obtained through Monte Carlo simulations it is denoted by "mc" in the legend of the figures. When it is obtained using the numerical method presented in this chapter, it is denoted by "th" (for theoretical) in the legend. In addition, for clarity, remind that the principle to compute the pdf of the utility is presented in Section 5.1.1 and it requires the cdf of the SINR in the NB case or the cdf of the effective SINR in the WB case. In the former case, the closed form of this cdf is presented in Section 5.1.2 whereas in the later, we have presented how to approach this value numerically with the block fading channel in Section 5.1.3.4 for the simplified model and in Section 5.1.3.5 for the full one.

5.3.4.1 NB fading

Figure 5.16 presents the pdf of the utility in the NB fading case described in Section 5.1.2. The red bars are obtained through Monte Carlo simulations and the black bars are obtained through numerical convolution. The later is computed as described in Section 5.1.1. One can observe the perfect matching between the red and black bars which assesses the accuracy of the numerical computation.

One can also note that from Scenario 1 to 3, the pdf moves also from the left to the right. Figure 5.16a corresponds to the bad allocation scheme and the resulting

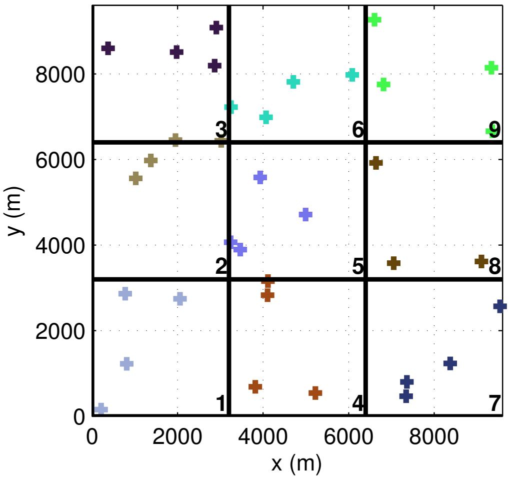


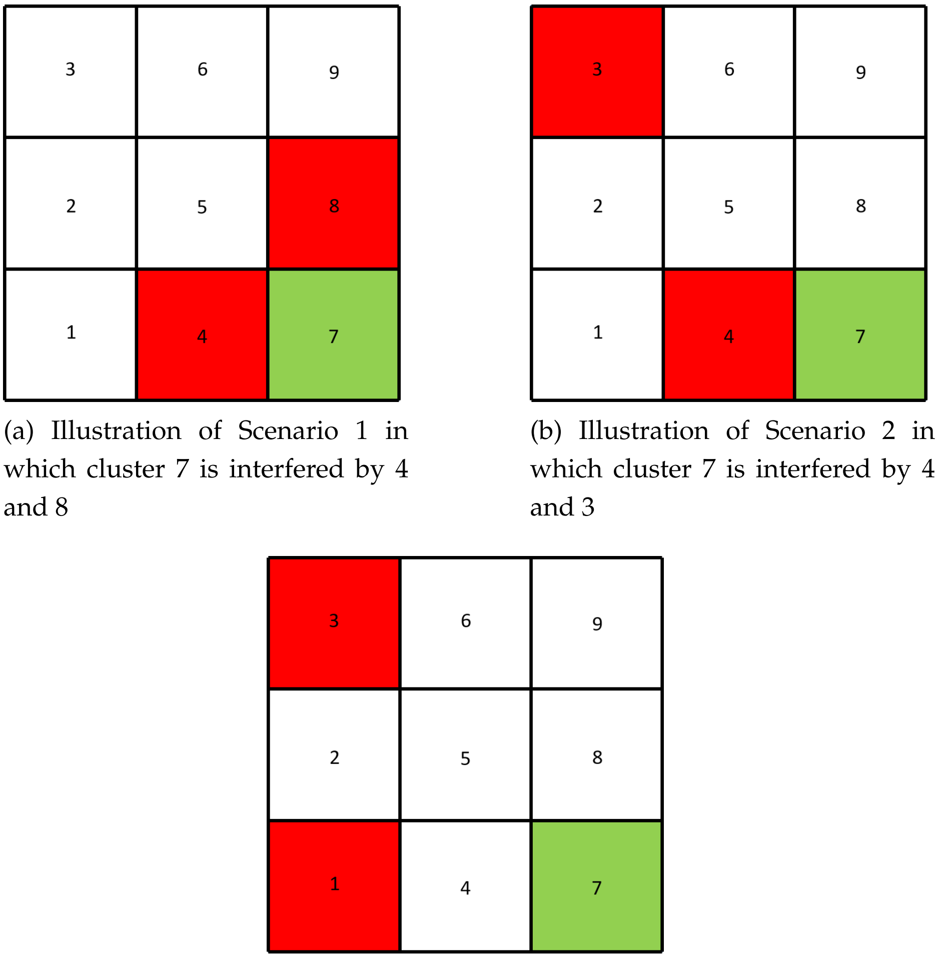
Figure 5.14: Wireless ad hoc clustered network with 9 clusters composed of 4 nodes each. Crosses represent mobile terminals, and colors the affiliations of terminals to cluster identified by the number in the bottom right of each square.

utility is closer to 0 than in the better allocation scheme in Figure 5.16c where it is closer to 1. Figure 5.16b is an average case and the utility is approximatively centered. In addition, we observe that while the utility can take all values in the set $[0, 1]$, the probability decreases very fast when moving away from the maximum of the pdf. It never takes, with high probability, all possible utility values. This observation sustains the choice made in Section 4.5 which consists in setting the tolerance thresholds such that a small amount of utility R_1 can lie outside the tolerance interval. By doing so, we see here that the thresholds will not cover the overall set $[0, 1]$. Therefore, the cluster is going to be sensitive to action changes while being robust to disturbances. Furthermore, one can see how the pdf of the utility changes from one Scenario to the other and its asymmetry in Scenario 1 and 3. On the first hand, this assesses the need for tolerance thresholds to be adaptive on the fly. On the second hand, this assesses the need for possibly asymmetric thresholds.

5.3.4.2 WB fading: simplified model

In this section, we present the pdf of the utility in a simplified model where the interference are not subject to fading. The SINR is given in (5.15) considering the block fading channel. In this case it is possible to compute numerically the pdf of the utility despite the highly nonlinear form of the effective SNR (5.12). The method to compute this pdf is provided in Section 5.1.3.4.

Figure 5.17 presents the pdf of the utility in Scenarios 1, 2 and 3. Note that



(a) Illustration of Scenario 1 in which cluster 7 is interfered by 4 and 8

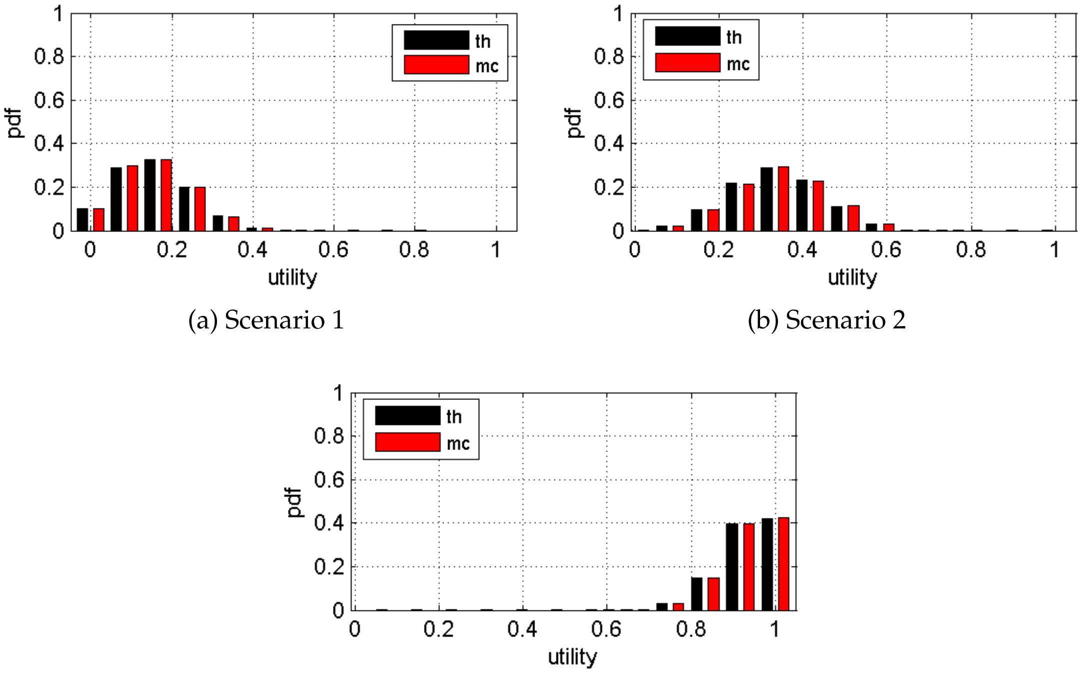
(b) Illustration of Scenario 2 in which cluster 7 is interfered by 4 and 3

(c) Illustration of Scenario 3 in which cluster 7 is interfered by 1 and 3

Figure 5.15: Illustration of the different scenarios considered.

we have set a given number of blocks to $N_b = 10$. The perfect match between the black and red bars assesses the accuracy of the procedure to compute the pdf of the utility when the EESM metric is used.

Furthermore, one can note that from the NB (results presented in the previous section) to the WB case the deviation of the distribution is decreased. In the WB case, the EESM metric sums the contribution of SINRs subject to iid Rayleigh fading which alleviates its variations. This reinforces the fact that the adaptive tolerance thresholds will not cover the overall set of utility values in Chapter 4. In the WB case, the tolerance will be smaller than in the NB one. Hence, the cluster is going to be more sensitive to action changes while more robust to disturbances at the same time in comparison to the NB case.



(a) Scenario 1

(b) Scenario 2

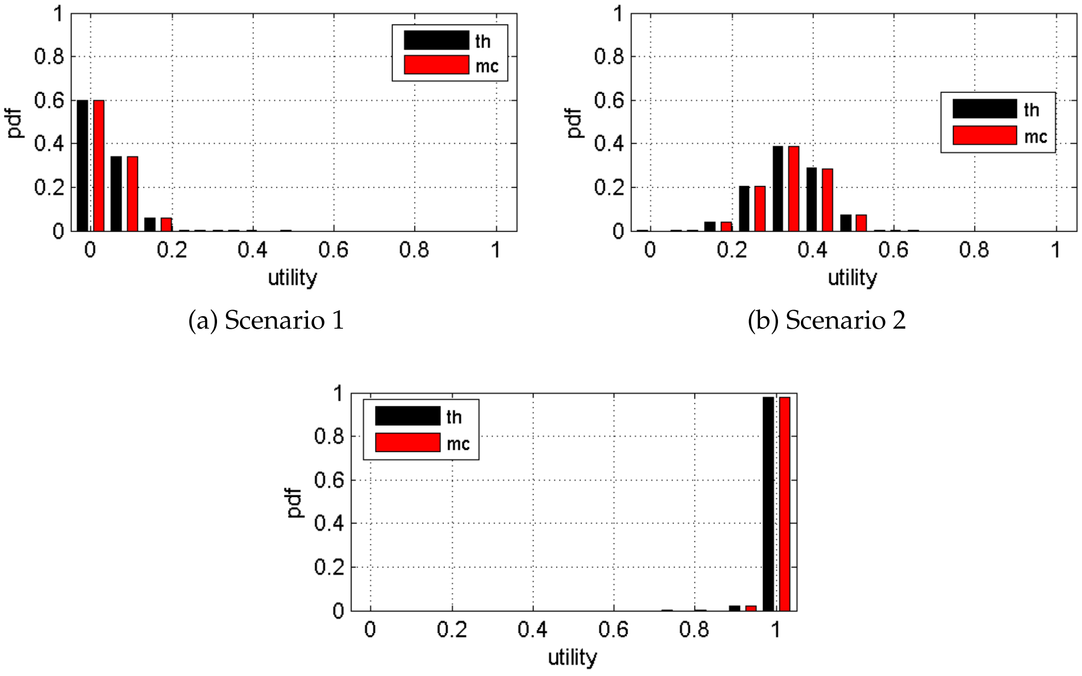
(c) Scenario 3

Figure 5.16: Pdf of the utility in a NB fading case.

5.3.4.3 WB fading: simplified model applied to a real channel

The aim of this section is to check the validity of the procedure to find the coherence bandwidth of an OFDM system that uses the EESM metric as presented in Section 5.2. We compare the pdf of the utility obtained with the block fading channel denoted by "BF" with the one obtained with a real channel model. We consider as a real channel the Typical Urban channel denoted by "TU". The effective SINR in the later is computed as described in Section 5.1.3.1. More specifically, we consider the MCS 1 described in Table 5.1 which is composed of a QPSK modulation and a convolutional code with rate 1/2, hence, the scaling factor $\beta = 1.8$. The pdf of the utility considering the block fading channel requires the use of the right coherence bandwidth. To that end we use the length of coherence bandwidth provided in Section 5.3.2 for different MCSs that are obtained with the procedure described in Section 5.2. The particularity here is that this bandwidth depends on the average SINR per subcarrier so it is going to vary from one link to another and we have to take it into account when computing the pdf of the utility.

Figure 5.18 presents the pdf of the utility with the block fading channel in black and with the real channel in red. Because of the channel coefficients' correlations between subcarriers, this later pdf can only be obtained using Monte Carlo sim-



(a) Scenario 1

(b) Scenario 2

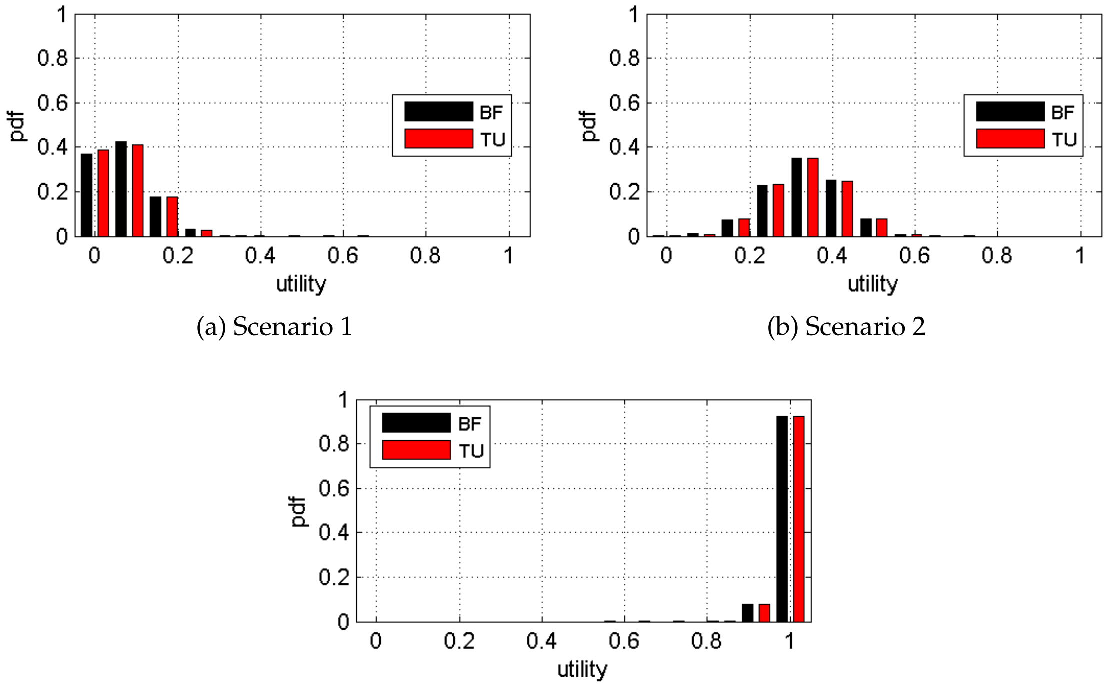
(c) Scenario 3

Figure 5.17: Pdf of the utility with the block fading channel (WB fading and simplified model).

ulations in comparison to the BF case in which it is computed numerically. The close to perfect match between red and black bars in the three scenarios assesses at the same time the fact that we can model the real channel by its abstracted counterpart the block fading channel and, the relevance of the procedure to select the length of blocks in the block fading channel.

5.3.4.4 WB fading : full model

In this section, we present the results obtained with the full model that is described in Section 5.1.3.5, and in which we consider the interference subject to fading (5.36). In this case, the complex form of the pdf involved (5.39) prevents a numerical computation of the EESM pdf at high average SINR per subcarrier. We have hence proposed a trick to approach the cdf in such case. To check the validity of the approximation in addition to the numerical method, Figure 5.19 presents the pdf of the utility obtained through Monte Carlo in red in comparison to the pdf of the utility obtained through numerical convolution proposed in Section 5.1.3.5 in black. The results are obtained considering $\beta = 1.8$, $N_c = 500$ and a constant size of blocks of coherent subcarriers with $N_{cc} = 50$. The close to perfect match



(a) Scenario 1

(b) Scenario 2

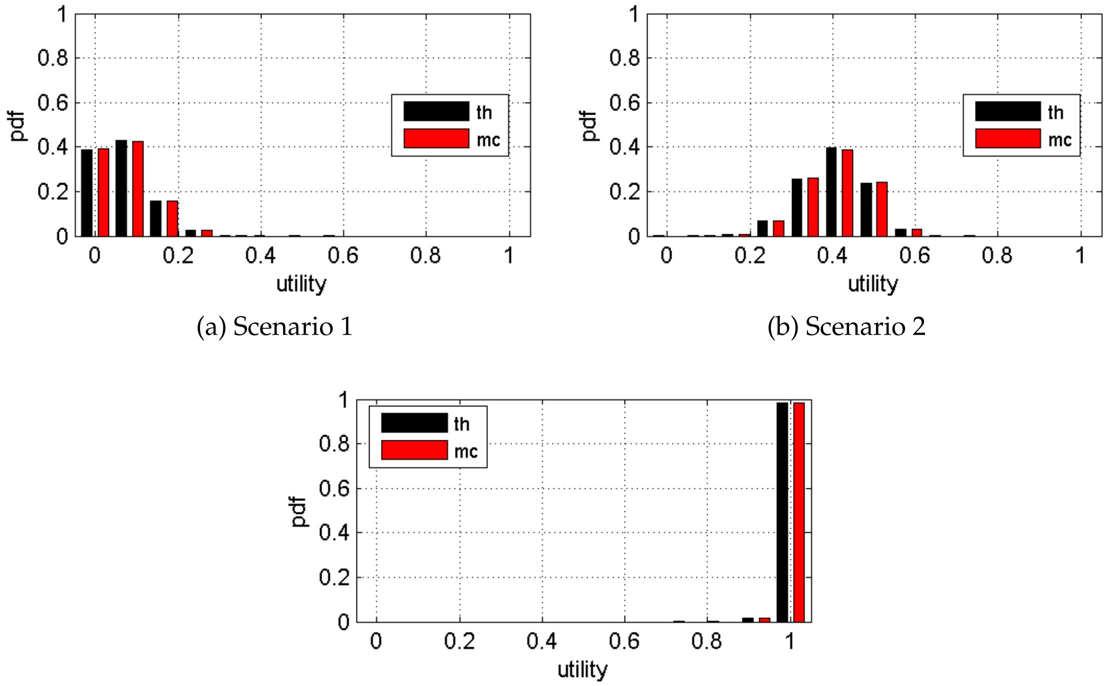
(c) Scenario 3

Figure 5.18: Pdf of the utility obtained with the real TU channel in comparison to the one obtained using the block fading channel.

between red and black bars assesses the accuracy of the proposed method.

5.3.4.5 WB fading : comparison between the simplified and full model

In this section, we compare the results obtained when using the simplified model described in Section 5.3.4.2 and the one obtained when using the full model described in Section 5.3.4.4 to understand the impact of interference subject to fading on the utility. Figure 5.20 aggregates these results in the three scenarios considered. We observe that when the interference are strong as in Scenario 1 or 2 (Figures 5.20a and 5.20b), the pdfs obtained using the simplified model are really different from the one obtained using the full model in red. The reason is that the faded interference have a strong impact on the statistics of the utility which are not taken into account when using the simplified model. In addition, one can observe that with the full model, the utility has a pdf with a better average value when comparing to the simplified model. The reason is that interference are subject to fading, thus, it happens that their impact can also be subject to outage from one realization to the other which, in such case, increases the effective SINR. These results can be predicted by comparing the cdf of EESM obtained with the



(a) Scenario 1

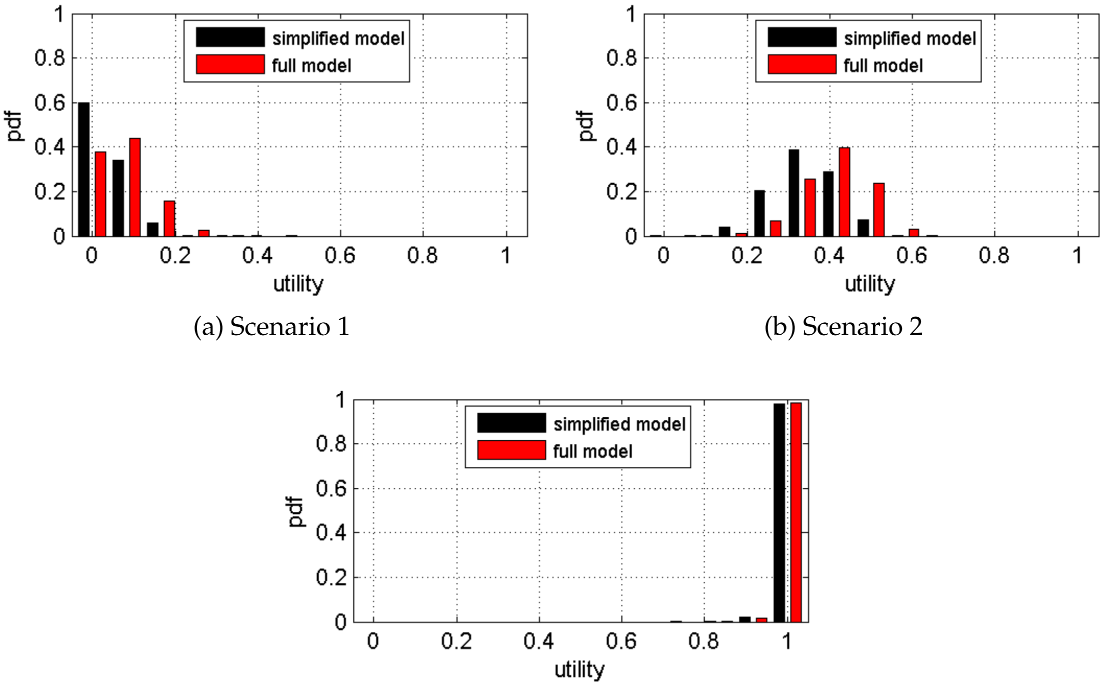
(b) Scenario 2

(c) Scenario 3

Figure 5.19: Pdf of the utility obtained when interference are subject to fading, comparison between Monte Carlo results and numerically computed results.

simplified model and with the full model. Figure 5.21 presents the typical behavior of these cdfs in the specific context with $SNR = 25$ dB and 5 interference sources with $\text{INR} = [1, 2, 5, 10, 20]$ dB. One can observe that the cdf obtained using the full model noted F_{cpl} is lower than the cdf obtained using the simplified model noted F_s, *i.e.* $\forall x, F_{cpl}(x) \leq F_s(x)$. The Bernoulli parameters of each link, that are required to derive the pdf of the utility, are computed using (5.2) (page 100) which can be seen as the complementary cdf of EESM. Therefore, $p_{cpl} = 1 - F_{cpl}(\Gamma_0) \geq p_s = 1 - F_s(\Gamma_0)$ are the links' Bernoulli parameters of the full and simplified model respectively that are computed in Γ_0 the threshold which specifies if a link is working or not. It follows that the links' Bernoulli parameters computed with the full model are all greater or equal to the one computed with the simplified model. Using the Conjecture 1 this sustains the results observed in Figure 5.20 which is that the probability to receive higher utility values with the full model is greater than in the simplified model, or equivalently, the cdf of the utility with the full model is lower than the cdf of the utility obtained with the simplified model.

In Scenario 3 (Figure 5.20c), interference have less statistical importance because the SINR per subcarrier is larger than in Scenario 1 and 2. It follows that,



(a) Scenario 1

(b) Scenario 2

(c) Scenario 3

Figure 5.20: Comparison of the pdf of the utility obtained with and without ("simplified model") interference subject to fading.

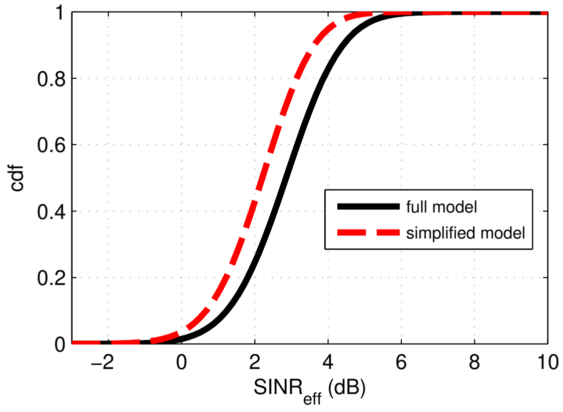


Figure 5.21: Comparison of the cdf of EESM between the simplified and full model.

both pdfs have closer similarities in comparison to Scenarios 1 and 2. This later observation further justifies the trick we use in Section 5.1.3.5 to approach the cdf obtained with the full model at high effective SINR with the cdf obtained with the simplified model.

5.4 Conclusion

The aim of this chapter has been to derive the pdf of the utility when taking into account the Rayleigh fading so as to understand better the impact of fading on the utility and on resource allocation algorithms. In order to derive the pdf of the utility in the WB context, we have approximated the correlated fading over frequencies as successive bands with independent fading. The real channel has been hence modeled by a block fading channel. To complete this abstraction, we have proposed a methodology based on statistical tests to set the coherence bandwidth such that the statistics of the effective SNR that have been obtained with the block fading channel fit the statistics of the effective SNR that have been obtained with the real channel model. We have provided a description to compute numerically the statistics of the effective SNR based on the block fading channel in two different cases.

Numerical results have shown that the effective SNR with the block fading channel could well approximate the statistics of the effective SNR with the real channel if the coherence bandwidth had been well chosen. We have highlighted that with the proposed abstraction, we could predict performance numerically in an OFDM system using EESM. In addition, we have shown numerically the relevance of such model to compute the pdf of the utility in a specific case. The results have sustained the arguments that we had used in the reasoning in Section 4.5 in order to make adaptive and asymmetric the tolerance thresholds on the fly. Moreover, we have observed that with the right value of coherence bandwidth which had been selected with the procedure, we could reproduce numerically the statistics of the utility obtained using the real channel. This again has assessed the capacity for the block fading channel to approximate a real channel and the relevance of the coherence bandwidth selection procedure.

The proposed methodology to compute the coherence bandwidth has to be realized off-line so as to provide reference tables to compute the desired statistics (*e.g.* for performance prediction) at low computational cost. In this chapter, we have applied the methodology on one real channel. It would be interesting to realize this method over many channels to observe if a general underlying model appears. For instance, we have highlight a relationship of the average SNR per subcarrier with the estimated coherence bandwidth. Such a relationship may exist with the maximum delay of the channel's paths or with τ_{RMS}. Moreover, this relationship could eventually be described by a relatively simple model (*e.g.* polynomials of low order). This would simplify further the build of reference tables of the coherence bandwidth, in addition to provide insightful meaning of the impact of various physical layer parameters on the performance of the system.

Chapter 6

Conclusion and perspectives

6.1 Conclusion

This thesis has addressed the fully distributed channel allocation problem in clustered ad hoc networks. Game theory has been employed as a mathematical framework for modeling the problem and, in this context, we have selected TE based algorithms to tackle the challenge. In this paradigm, players are either able to experiment rarely new actions or they experiment all the time random actions. In both cases, they accept with higher probability an outcome that provides a higher utility. These algorithms have shown the particularity to exhibit efficient cooperative behavior despite the non-cooperative nature of the problem (fully distributed) which has been one of the reasons they were selected as candidates to solve the allocation problem considered in this thesis.

The convergence proofs of these algorithms rely on perturbed MC whose dimension grows exponentially with the size of the problem. Therefore, the performance analysis of these algorithms is intractable and constitutes an important challenge. To overcome the huge dimension of the inherent MC, we have provided an approximation of these chains. This has allowed us to compute a close approximation of the average time the system remains in a desired state as well as the average time required to achieved that state for the first time. This has provided the possibility to compare the performance of two algorithms in addition to give new insights on their convergence.

Toward the convergence study, we have noted that the main drawback of TE algorithms was their sensitivity to disturbance. In wireless telecommunication systems, it is common to assume the presence of time varying measurement in radio channel propagation (fading, shadowing). Adapting these algorithms to disturbance was hence a major goal of this thesis. To do so, we have proposed modifications of the above algorithms, with supporting theoretical proofs, which have consisted in introducing tolerance thresholds into the allocation decision. Then, we have proposed a method to adapt dynamically these thresholds in order to make the algorithm operating in different time varying contexts.

We have especially studied the impact of Rayleigh fading on the utility by deriving its pdf in various contexts and in specific cases. The utility function requires SINR measurements as inputs whose formulas have been provided. Note that simulations in Chapter 4 relied on this physical layer abstraction. We have considered an OFDM transmission as it is one of the most widely used technology nowadays. In case of frequency selective Rayleigh fading, the correlated SNRs over subcarriers have been mapped into one effective SNR, noted EESM, that was representing the performance of the transmission. The nonlinear form of EESM in addition to the correlated SNR over subcarriers did not enable theoretical analysis that is why we have simplified further the modeling. We have considered that the fading was flat over consecutive subcarriers inside a bandwidth known as the coherence bandwidth. The channel has been hence modeled as a block fading channel. The idea has been then to use the independence property of fading between bandwidths to realize a theoretical analysis. Then, we have provided a method, based on statistical tests, to find the coherence bandwidth such that the statistics of EESM with the coherence bandwidth assumption would fit the statistics of EESM obtained with the real channel. We have observed that this bandwidth depended on several parameters such as the channel to be simulated, the signal bandwidth and the MCS. With the proposed assumption, we have been able to derive in a specific case the pdf of the utility, to predict accurately the performance of an OFDM system and, as a by-product, this abstraction could be used as input in the system level simulations to avoid the implementation of detailed physical layer.

6.2 Perspectives

As future work, several extensions of the current thesis can be addressed.

In this thesis, we have focused on the allocation of frequency channels among clusters. It should be interesting to allocate the power and the frequency channels simultaneously. In some cases, decreasing the power will reduce the interference while keeping good communications capabilities of the nodes. This would increase the capacity to reuse frequency channels among clusters.

We also have assumed that the nodes always transmit information. In practice, the load in clusters varies due to the time varying nature of the traffic. It thus could be interesting to include traffic load variations as it could involve less interference during periods with low communication demands. Hence, the capacity of the network to reuse frequency could be increased.

In this thesis, we have considered that each user employs only one MCS during the whole communication period. In practice, the choice of MCS depends on the required QoS and on the SINR. Thus, it would be of interest to propose strategies that include adaptive modulation and coding so as to adapt the transmission to interference and load variations.

We have supposed that the clusters do not communicate among each other. This has been motivated by the ad hoc type of the network. In some contexts in wireless networks (*e.g.* small cells) some exchanging of signaling could be possible between the clusters. In such context, it would be interesting to study the amount of information necessary to increase the performance and make possible almost sure convergence. In addition, one could study other type of approaches to allocate frequency channels that require more information such as regret based methods or distributed gradient methods for instance.

In Chapter 4, we have considered relaying capabilities of the nodes through the possibility for the cluster to have a diameter greater than 1. It could be of interest to develop further relaying strategies. For instance, different nodes repeat several times the message to transmit so as to reach a farther node. It would provide robust communications' capabilities and less disturbed utility measurements.

In addition, one of the major problem which leads to strong outages of the utility is due to the fact that a cluster tries a frequency that is used by another cluster close to the first one. It thus could be interesting to allow frequency channel sensing by clusters. In this case, for instance, a cluster would try among frequencies which have a low measured power. This would reduce the number of trials that induce strong interference and it could lead to better performance.

Furthermore, another possible method to improve the performance is to use successive interference cancellation (SIC) receivers. In practice, nodes are interfered by few transmitting nodes from different clusters. A node that is subject to high interference would first decode the interfering message, and then it could remove it from the signal so as to decode the message that is intended to it. This allows for a better satisfaction of the users and hence a lower fluctuations of average utility due to channel allocation.

In this thesis, we have assumed that the trial and error procedures are synchronized (*e.g.* clusters measure and try during the same periods). This assumption can be difficult to meet especially when inter clusters communications capabilities are limited. It would be then of interest to study the performance of algorithms when the nodes are not synchronized. For instance, a cluster receives the links metrics during one phase, transmits decisions orders to nodes in a second one, and realizes measurements of metrics during a third phase. Therefore, if a cluster realizes its measurements during a phase different than the third one, it observes a utility that does not correspond to the current state of the network. Therefore, it is going to perturb the cluster future decision which can result in performance decreases.

Finally, the performance analysis of TE algorithms in Chapter 3 has been conducted for a specific network and utility model. This choice has been motivated by the willing to be able to simplify the huge Markov chain induced by the TE processes. Thus, extension of this analysis to a more general utility model is an interesting question that remains open.

Appendix A

Probabilities involved in TEL approximation

This appendix describes all the possible transitions and probabilities of TEL algorithm model presented in Figures 3.5a.

Remind with Hypothesis 1 that, a discontent player accepts a free resource with probability $\epsilon^0 = 1$ and, it accepts a resource already interfered with probability $\epsilon^{F(0)} = \epsilon^{\frac{1}{2K}}$. In addition, with Assumption 3.1, we approximate the probability that there is one experimentation among K content players by the probability that at least one experimentation happens $P_\epsilon(K) = 1 - (1 - \epsilon)^K$.

A.1 Notations and preliminaries

As in Figure 3.5a, we consider two sets $\xi^n(i)$ and $\xi^{n+1}(j)$. We assume the presence of all the intermediary states in order to derive the most general transition probabilities. In practice, using Section 3.5.6.1, the reader must check the existence of the intermediary states involved in sets before computing the probabilities.

The probability computation require the knowledge of the players repartition over resources. For each set $\xi^n(i)$, the number of resources having p players in $\mathbf{Z}_n(i)$ is noted $M_n^i(p) = \sum_k \mathbb{1}_{\{S_{n,k}^i = p\}}$ and the number of players that share a resource with $p - 1$ other players in $\mathbf{Z}_n(i)$ is noted $m_n^i(p) = pM_n^i(p)$.

During a transition from $\xi^n(i)$ to $\xi^{n+1}(j)$, a resource is decremented by one player and a free resource is incremented. We note $k(i, j)$ the resource decremented such that the term $S_{n,k(i,j)}^i$ of $\mathbf{S}_n(i)$ is decremented by one. Meanwhile, the term $S_{n,n+1}^i$ is incremented by one. During the reverse transition from $\xi^{n+1}(j)$ to $\xi^n(i)$, the most left column of $\mathbf{S}_{n+1}(j)$ that has $S_{n,k(i,j)}^i - 1$ players is incremented by one and, $S^{j}_{n+1,n+1}$ is decremented by one.

A.2 Transitions inside each $\xi^n(i)$

We start by describing the transitions inside $\xi^n(i)$ that is to say between the states $\mathbf{Z}_n(i)$, $\xi^n_0(i)$, $\xi^n_1(i)$, $\xi^n_2(i)$, $\xi^n_3(i)$ and $\xi^n_4(i)$. State $\mathbf{Z}_n(i)$ is connected to $\xi^n_1(i)$ and itself. The transition $\mathbf{Z}_n(i) \rightarrow \xi^n_1(i)$ happens when a player that is not interfered becomes watchful. It is given by probability

$$p_{\mathbf{Z}_n(i)\xi^n_1(i)} = \overbrace{P_\epsilon(K)\frac{K-1}{K}}^{(a)}\overbrace{\frac{M^i_n(1)}{N-1}}^{(b)}, \tag{A.1}$$

where (a) is the probability that there is an experimentation from any player except the one that is going to be interfered and, (b) is the probability to select the frequency of a player not interfered. The probability of transition $\mathbf{Z}_n(i) \rightarrow \mathbf{Z}_n(i)$ is computed using the conservation probability property

$$p_{\mathbf{Z}_n(i)\mathbf{Z}_n(i)} = 1 - p_{\mathbf{Z}_n(i)\xi^n_1(i)} - p_{\mathbf{Z}_n(i)\mathbf{Z}_{n+1}}, \tag{A.2}$$

where $p_{\mathbf{Z}_n(i)\mathbf{Z}_{n+1}}$ is the probability for the network to find a new resource. It is given by

$$p_{\mathbf{Z}_n(i)\mathbf{Z}_{n+1}} = \overbrace{P_\epsilon(K)\frac{(K-m^i_n(1))}{K}}^{(a)}\overbrace{\frac{M^n_i(0)}{N-1}}^{(b)}, \tag{A.3}$$

where (a) is the probability that an interfered player experiments and, (b) is the probability that it finds a free resource.

From the state $\xi^n_1(i)$, the network can directly go in $\mathbf{Z}_n(i)$ or $\xi^n_2(i)$. With Assumption 3.1, we do not consider any experimentation in $\xi^n_1(i)$ except the one needed to make the MC ergodic in transition $\xi^n_1(i) \rightarrow \xi^n_2(i)$. This later happens if the watchful player (which cannot experiment) is subject to a second experiment on its resource by an other player. This is given by the following probability

$$p_{\xi^n_1(i)\xi^n_2(i)} = \mathcal{P}_\epsilon(K-1)\frac{1}{N-1}. \tag{A.4}$$

Otherwise, we do not consider any other case from $\xi^n_1(i)$ and, the system goes naturally from $\xi^n_1(i)$ to $\mathbf{Z}_n(i)$ with probability

$$p_{\xi^n_1(i)\mathbf{Z}_n(i)} = 1 - p_{\xi^n_1(i)\xi^n_2(i)}. \tag{A.5}$$

State $\xi_2^n(i)$ is connected to $\mathbf{Z}_n(i)$ and $\xi_3^n(i)$. During transition $\xi_2^n(i) \rightarrow \mathbf{Z}_n(i)$, on the first hand, the discontent player selects a free resource. There are $M_i^n(0)$ free resources in addition to the discontent player resource. Secondly, it accepts it as a new benchmark with probability $\epsilon^{F(1)} = 1$ (Hypothesis 1). The probability of this transition is thus given by

$$p_{\xi_2^n(i)\mathbf{Z}_n(i)} = \frac{M_i^n(0)+1}{N}, \tag{A.6}$$

The transition $\xi_2^n(i) \rightarrow \xi_3^n(i)$ happens if the discontent player selects a resource already occupied by only one player and, it updates its benchmark with probability $\epsilon^{F(0)} = \epsilon^{\frac{1}{2K}}$ (Hypothesis 1). The new player interfered becomes watchful in the next iteration. From the discontent player point of view, there are $M_i^n(1)-1$ players alone on their resource. The transition happens with the following probability

$$p_{\xi_2^n(i)\xi_3^n(i)} = \frac{M_i^n(1)-1}{N}\epsilon^{\frac{1}{2K}}. \tag{A.7}$$

The probability of transition $\xi_2^n(i) \rightarrow \xi_2^n(i)$ is computed using the conservation probability property

$$p_{\xi_2^n(i)\xi_2^n(i)} = 1 - p_{\xi_2^n(i)\mathbf{Z}_n(i)} - p_{\xi_2^n(i)\xi_3^n(i)} - p_{\xi_2^n(i)\mathbf{Z}_{n-1}}, \tag{A.8}$$

where $p_{\xi_2^n(i)\mathbf{Z}_{n-1}}$ is the probability that the discontent player selects and accepts a resource already occupied by two players or more, which has probability

$$p_{\xi_2^n(i)\mathbf{Z}_{n-1}} = \frac{N - M_n^i(1) - M_n^i(0)}{N}\epsilon^{\frac{1}{2K}}. \tag{A.9}$$

In state $\xi_3^n(i)$, the system is not aligned and there is no discontent player. Thus, with Assumption 3.1, no experiment is proceeded and, the system moves directly to state $\xi_4^n(i)$ with probability

$$p_{\xi_3^n(i)\xi_4^n(i)} = 1. \tag{A.10}$$

The state $\xi_4^n(i)$ is connected to $\xi_0^n(i)$ and $\xi_3^n(i)$. For the following transitions, it useful to note that in $\xi_4^n(i)$, the number of resources with players that do not interfere is $M_i^n(1)-2$ and, the number of free resources is $M_i^n(0)+1$. The transition $\xi_4^n(i) \rightarrow \xi_0^n(i)$ corresponds to the situation where the discontent player chooses a free resource, that it accepts with probability $\epsilon^{F(1)} = 1$ (Hypothesis 1). The player left alone sees its utility increases and becomes hopeful. This happens with probability

$$p_{\xi_4^n(i)\xi_0^n(i)} = \frac{M_i^n(0)+1}{N}. \tag{A.11}$$

Transition $\xi_4^n(i) \rightarrow \xi_3^n(i)$ occurs when the discontent player selects an occupied resource with one player (there are $M_i^n(1)-2$ of them in $\xi_4^n(i)$) and, it accepts this resource as a new benchmark with probability $\epsilon^{\frac{1}{2K}}$ according to Hypothesis 1. Thus, the new player interfered becomes watchful with probability

$$p_{\xi_4^n(i)\xi_3^n(i)} = \frac{M_i^n(1) - 2}{N}\epsilon^{\frac{1}{2K}}. \tag{A.12}$$

The probability to remain in $\xi_4^n(i)$ is

$$p_{\xi_4^n(i)\xi_4^n(i)} = 1 - p_{\xi_4^n(i)\xi_0^n(i)} - p_{\xi_4^n(i)\xi_3^n(i)} - p_{\xi_4^n(i)\mathbf{Z}_{n-1}}, \tag{A.13}$$

where $p_{\xi_4^n(i)\mathbf{Z}_{n-1}}$ is the probability that, the network use one less frequency with all players content and aligned. This happens if the discontent player selects and accepts one of the $N - M_n^i(1) - M_n^i(0)$ resources already occupied by two players or more or, if it selects and accepts its current resource where the content player interfered is aligned (*i.e.* it is going to accept the choice of the discontent player). Consequently, the system uses one less frequency with probability

$$p_{\xi_4^n(i)\mathbf{Z}_{n-1}} = \frac{N - M_n^i(1) - M_n^i(0) + 1}{N}\epsilon^{\frac{1}{2K}}, \tag{A.14}$$

and the probability that it remains one more step in $\xi_4^n(i)$ is $p_{\xi_4^n(i)\xi_4^n(i)} = 0$. Once the network is in $\xi_0^n(i)$, one player is hopeful and with Assumption 3.1, no player experiments. Thus, in the next step this player becomes content with a benchmark update and the network goes to $\mathbf{Z}_n(i)$ with the following probability

$$p_{\xi_0^n(i)\mathbf{Z}_n(i)} = 1. \tag{A.15}$$

A.3 Transitions from $\xi^n(i)$ to $\xi^{n+1}(j)$

The only way for the network to find a new resource is to go through a RC $\mathbf{Z}_n(i)$. If the network is not in $\mathbf{Z}_n(i)$ either one player is discontent or the network is not aligned. In the later, Assumption 3.1 tells us that no player experiments, whereas in the former, the discontent player cannot discover a free resource because it is necessarily not interfered in $\mathbf{Z}_n(i)$.

During transition $\mathbf{Z}_n(i) \to \mathbf{Z}_{n+1}(j)$, a player experiments on a free resource with probability

$$p_{\mathbf{Z}_n(i)\mathbf{Z}_{n+1}(j)} = \begin{cases} \overbrace{P_\epsilon(K)\frac{m_n^i\left(S_{n,k(i,j)}^i\right)}{K}}^{(a)}\overbrace{\frac{M_i^n(0)}{N-1}}^{(b)}, \text{ if } S_{n,k(i,j)}^i > 2, \\ 0, \text{ if } S_{n,k(i,j)}^i = 2, \end{cases} \tag{A.16}$$

where (a) is the probability to have an experimentation from any player interfered on resources with $S^i_{n,k(i,j)} > 2$ players and, (b) is the probability to select a free resource. The second line corresponds to an other transition $\mathbf{Z}_n(i) \rightarrow \xi_0^{n+1}(j)$, in which the player left alone after the experimentation sees its utility increase and becomes hopeful. The probability of transition $\mathbf{Z}_n(i) \rightarrow \xi_0^{n+1}(j)$ is thus complementary to the previous one and, it is given by

$$p_{\mathbf{Z}_n(i)\xi_0^{n+1}(j)} = \begin{cases} 0, \text{ if } S^i_{n,k(i,j)} > 2, \\ P_\epsilon(K)\frac{m_n^i(S^i_{n,k(i,j)})}{K}\frac{M_i^n(0)}{N-1}, \text{ if } S^i_{n,k(i,j)} = 2. \end{cases} \tag{A.17}$$

A.4 Transitions from $\xi^{n+1}(j)$ to $\xi^{n}(i)$

The approximation is constructed such that if it is possible to go from $\xi^n(i)$ to $\xi^{n+1}(j)$, it is also possible to go from $\xi^{n+1}(j)$ to $\xi^n(i)$ (see Section 3.5.6.1). The way for the system to go in a set where one less resource is employed only happens in states with a discontent player, *i.e.* $\xi_2^{n+1}(j)$ and $\xi_4^{n+1}(j)$ for transition $\xi^{n+1}(j)$ to $\xi^n(i)$. In practice, to compute the transitions inside $\xi^{n+1}(j)$, we use the formulas in Appendix A.2 by replacing the indices appropriately ($\xi^{n+1}(j)$ is similar to $\xi^n(i)$ with one less free resource). In this section, the starting state is in $\xi^{n+1}(j)$. Therefore, we use the functions $m^j_{n+1}(.)$ and $M^j_{n+1}(.)$ instead of $m^i_n(.)$ and $M^i_n(.)$. Moreover, during a transition from $\xi^n(i)$ to $\xi^{n+1}(j)$, the resource that contained $S^i_{n,k(i,j)}$ in $\mathbf{S}_n(i)$ has been decremented by one. Thus, the transition from $\xi^{n+1}(j)$ to $\xi^n(i)$ occurs if any resource that contains $S^i_{n,k(i,j)} - 1$ players is incremented by one.

The transition $\xi_2^{n+1}(j) \rightarrow \mathbf{Z}_n(i)$ happens if the discontent player selects a frequency with $S^i_{n,k(i,j)} - 1$ players and accept the new benchmark. The probability of $\xi_2^{n+1}(j) \rightarrow \mathbf{Z}_n(i)$ is thus given by

$$p_{\xi_2^{n+1}(j)\mathbf{Z}_n(i)} = \begin{cases} \frac{M_j^{n+1}(S^i_{n,k(i,j)}-1)}{N}\epsilon^{\frac{1}{2K}}, \text{ if } S^i_{n,k(i,j)} - 1 \geq 2, \\ 0, \text{ if } S^i_{n,k(i,j)} - 1 = 1, \end{cases} \tag{A.18}$$

where, the second line is null because it is represented by transition $\xi_2^{n+1}(j) \rightarrow \xi_3^{n+1}(j)$ (see (A.7) with appropriate indices changes).

The transition $\xi_4^{n+1}(j) \rightarrow \mathbf{Z}_n(i)$ happens when the discontent player selects an occupied resource and all players are content and aligned in the end. This is given by probability

$$p_{\xi_4^{n+1}(j)\mathbf{Z}_n(i)} = \begin{cases} 0, \text{ if } S^i_{n,k(i,j)} - 1 \geq 2, \\ \frac{1}{N}\epsilon^{\frac{1}{2K}}, \text{ if } S^i_{n,k(i,j)} - 1 = 1, \end{cases} \tag{A.19}$$

where, the first line is null because this corresponds to the transition $\xi_4^{n+1}(j) \to \xi_0^n(i)$ described afterwards. The second line is the probability for the discontent player to select and to accept the current resource. The transition $\xi_4^{n+1}(j) \to \xi_0^n(i)$ corresponds to the case where the discontent player selects and accepts a resource with $S^i_{n,k(i,j)} - 1 \geq 2$ players. Consequently, the player that is left alone becomes hopeful with probability

$$p_{\xi_4^{n+1}(j)\xi_0^n(i)} = \begin{cases} \frac{M_j^{n+1}(S^i_{n,k(i,j)}-1)}{N}\epsilon^{\frac{1}{2K}}, \text{ if } S^i_{n,k(i,j)} - 1 \geq 2, \\ 0, \text{ if } S^i_{n,k(i,j)} - 1 = 1, \end{cases} \tag{A.20}$$

where, the second line corresponds to previous transition $\xi_4^{n+1}(j) \to \mathbf{Z}_n(i)$.

Appendix B

Probabilities involved in ODL approximation

This Appendix describes all the possible transitions and probabilities of ODL algorithm model presented in Figure 3.5b. We also use the same notations and preliminaries detailed in Appendix A.1.

In ODL, a player which perceives a utility or an action change accepts the new benchmark with probability ϵ^{1-u} or, it refuses it and becomes discontent with probability $1-\epsilon^{1-u}$.

B.1 Transitions inside $\xi^n(i)$

We start by describing the transitions between the states $\mathbf{Z}_n(i)$, $\xi_1^n(i)$, $\xi_2^n(i)$ and $\xi_3^n(i)$. The state $\mathbf{Z}_n(i)$ is connected to $\xi_1^n(i)$, $\xi_2^n(i)$, $\xi_3^n(i)$. Transition $\mathbf{Z}_n(i) \rightarrow \xi_1^n(i)$ happens if an alone player becomes discontent after perceiving a utility change that it does not accept. This situation arises with probability

$$p_{\mathbf{Z}_n(i)\xi_1^n(i)} = \overbrace{\mathcal{P}_\epsilon(K)\frac{(K-m_n^i(1)-m_n^i(2))}{K}\frac{M_n^i(1)}{N-1}}^{(a)}\overbrace{(1-\epsilon)}^{(b)}. \tag{B.1}$$

where (a) is the probability that any player interfered by two players or more experiments on a resource with solely one player, (b) is the probability that this alone player becomes discontent.

The transition $\mathbf{Z}_n(i) \rightarrow \xi_2^n(i)$ represents the situation where two players alone in $\mathbf{Z}_n(i)$ become discontent in one step, whose probability is

$$p_{\mathbf{Z}_n(i)\xi_2^n(i)} = \overbrace{\mathcal{P}_\epsilon(K)\frac{m_n^i(1)}{K}\frac{M_n^i(1)-1}{N-1}}^{(a)}\overbrace{(1-\epsilon)^2}^{(b)}, \tag{B.2}$$

where, (a) is the probability that an alone player experiments on a resource with an other alone player and, (b) is the probability that both players become discontent.

The transition $\mathbf{Z}_n(i) \rightarrow \xi_3^n(i)$ represents the situation where, from a resource with two players, one of them experiments on an other resource with one player and, one of them ends in discontent mood. This happens with probability

$$p_{\mathbf{Z}_n(i)\xi_3^n(i)} = \overbrace{\mathcal{P}_\epsilon(K)\frac{m_n^i(2)}{K}}^{(a)}\overbrace{\frac{M_n^i(1)}{N-1}2\epsilon(1-\epsilon)}^{(b)}, \tag{B.3}$$

where (a) is the probability that a player experiments from a resource with two of them, (b) is the probability to interfere with one player and, one of the two players involved becomes discontent. The presence of multiplier 2 in term (b) means that, inverting player's label is a different event that results in the same state $\xi_3^n(i)$ and with the same probability.

The transition $\mathbf{Z}_n(i) \rightarrow \mathbf{Z}_n(i)$ is computed using probability conservation as follows,

$$\begin{aligned} p_{\mathbf{Z}_n(i)\mathbf{Z}_n(i)} &= 1 - p_{\mathbf{Z}_n(i)\xi_1^n(i)} - p_{\mathbf{Z}_n(i)\xi_2^n(i)} - p_{\mathbf{Z}_n(i)\xi_3^n(i)} - p_{\mathbf{Z}_n(i)\mathbf{Z}_{n+1}} - p_{\mathbf{Z}_n(i)\mathbf{Z}_{n-1}} \\ &\quad - p_{\mathbf{Z}_n(i)\xi_1^{n+1}} - p_{\mathbf{Z}_n(i)\xi_2^{n+1}} - p_{\mathbf{Z}_n(i)\xi_3^{n-1}}, \end{aligned} \tag{B.4}$$

where $p_{\mathbf{Z}_n(i)\mathbf{Z}_{n+1}}$ is the probability for any interfered player to select a free resource, $p_{\mathbf{Z}_n(i)\mathbf{Z}_{n-1}}$ is the probability for any not interfered player to become interfered, $p_{\mathbf{Z}_n(i)\xi_1^{n+1}}$ and $p_{\mathbf{Z}_n(i)\xi_2^{n+1}}$ are similar to $p_{\mathbf{Z}_n(i)\mathbf{Z}_{n+1}}$ with one and two players ending in discontent mood respectively and, $p_{\mathbf{Z}_n(i)\xi_3^{n-1}}$ is the probability that two players alone in $\mathbf{Z}_n(i)$ finish on the same resource with one of them discontent. The first probability is given by

$$p_{\mathbf{Z}_n(i)\mathbf{Z}_{n+1}} = P_\epsilon(K)\frac{(K - m_n^i(1))}{K}\frac{M_i^n(0)}{N-1}, \tag{B.5}$$

which is the same as (A.3) in the TEL model. The second probability is given by

$$p_{\mathbf{Z}_n(i)\mathbf{Z}_{n-1}} = \overbrace{\mathcal{P}_\epsilon(K)\frac{m_n^i(1)}{K}}^{(a)}\left(\overbrace{\frac{N - M_n^i(1) - M_n^i(0)}{N-1}\epsilon}^{(b)} + \overbrace{\frac{M_n^i(1)-1}{N-1}\epsilon^2}^{(c)}\right), \tag{B.6}$$

where, (a) is the probability for a player that is not interfered to experiment, (b) is the probability that it experiments on resource with two players or more and that it updates its benchmark, (c) is the probability to select the resource of a player not interfered and that both accept this new benchmark. Probability $p_{\mathbf{Z}_n(i)\xi_1^{n+1}}$ is given by

$$p_{\mathbf{Z}_n(i)\xi_1^{n+1}} = P_\epsilon(K)\frac{K - m_n^i(1)}{K}\overbrace{\frac{N - M_n^i(1) - M_n^i(0) - 1}{N-1}(1-\epsilon)}^{(a)}, \tag{B.7}$$

where (a) is the probability that the player interfered selects a frequency with two players or more, except its own resource, and, that it ends in discontent mood.

The probability to end in ξ_2^{n+1} is

$$p_{\mathbf{Z}_n(i)\xi_2^{n+1}} = P_\epsilon(K)\frac{K - m_n^i(1)}{K}\overbrace{\frac{M_n^i(1)}{N-1}(1-\epsilon)^2}^{(a)}, \tag{B.8}$$

where (a) is the probability that the experimenter selects the resource of player not interfered and that both end up in discontent mood.

Finally, the probability to go from $\mathbf{Z}_n(i)$ to ξ_3^{n-1} is given by

$$p_{\mathbf{Z}_n(i)\xi_3^{n-1}} = P_\epsilon(K)\frac{m_n^i(1)}{K}\overbrace{\frac{M_n^i(1) - 1}{N-1}2\epsilon(1-\epsilon)}^{(a)}, \tag{B.9}$$

where (a) is the probability that the experimenter selects a resource with a player not interfered and, one of them ends in discontent mood. The multiplier 2 has a similar role than in (B.3).

The state $\xi_1^n(i)$ is connected to $\mathbf{Z}_n(i)$ and $\xi_2^n(i)$ inside the set $\xi^n(i)$. During transition $\xi_1^n(i) \to \mathbf{Z}_n(i)$ the discontent player either chooses a free resource or its current benchmark with probability

$$p_{\xi_1^n(i)\mathbf{Z}_n(i)} = \frac{M_n^i(0) + 1}{N}. \tag{B.10}$$

During transition $\xi_1^n(i) \to \xi_2^n(i)$, the discontent player makes an other player discontent in addition to itself. This is given by probability

$$p_{\xi_1^n(i)\xi_2^n(i)} = \overbrace{\frac{M_n^i(1) - 1}{N}}^{(a)}(1-\epsilon)^2, \tag{B.11}$$

where (a) is the probability that the discontent player selects a resource that contains a player alone, except its own resource.

The probability $p_{\xi_1^n(i)\xi_1^n(i)}$ is obtained using the conservation property:

$$p_{\xi_1^n(i)\xi_1^n(i)} = 1 - p_{\xi_1^n(i)\mathbf{Z}_n(i)} - p_{\xi_1^n(i)\xi_2^n(i)} - p_{\xi_1^n(i)\mathbf{Z}_{n-1}} - p_{\xi_1^n(i)\xi_3^{n-1}}, \tag{B.12}$$

where $p_{\xi_1^n(i)\mathbf{Z}_{n-1}}$ and $p_{\xi_1^n(i)\xi_3^{n-1}}$ are the probability for the system starting in $\xi_1^n(i)$ to end for all $j \in I_N(n-1)$ in states $\mathbf{Z}_{n-1}(j)$ and $\xi_3^{n-1}(j)$ respectively. The first probability is given by

$$p_{\xi_1^n(i)\mathbf{Z}_{n-1}} = \frac{N - M_n^i(0) - M_n^i(1)}{N}\epsilon + \frac{M_n^i(1) - 1}{N}\epsilon^2, \tag{B.13}$$

which are similar to terms (b)+(c) in (B.6) except the choice is made over all resources as the player is in state D.

The second probability is given by

$$p_{\xi_1^n(i)\xi_3^{n-1}} = \frac{M_n^i(1) - 1}{N} 2\epsilon(1-\epsilon), \tag{B.14}$$

which is similar to (a) in (B.9) except that the choice is made among N resources.

The state $\xi_2^n(i)$ is connected to $\xi_1^n(i)$ and $\mathbf{Z}_n(i)$. The probability of transition $\xi_2^n(i) \rightarrow \mathbf{Z}_n(i)$ is given by

$$p_{\xi_2^n(i)\mathbf{Z}_n(i)} = \overbrace{\frac{M_n^i(0)+2}{N}}^{(a)} \overbrace{\frac{M_n^i(0)+1}{N}}^{(b)}, \tag{B.15}$$

where (a) is the probability that the first discontent player selects a free resource. The number of free resource is $M_n^i(0)$ in addition to the 2 resources left by the discontent players. Term (b) is the probability that the other discontent player selects a free resource given that, the first discontent player has already selected a free resource.

The probability of a transition $\xi_2^n(i) \rightarrow \xi_1^n(i)$ is given by

$$p_{\xi_2^n(i)\xi_1^n(i)} = 2\frac{M_n^i(0)+2}{N} \overbrace{\frac{N - M_n^i(1) - M_n^i(0)}{N}}^{(a)} (1-\epsilon), \tag{B.16}$$

where (a) is similar to the term (a) in (B.7) except there is one more resource available.

The probability to remain in $\xi_2^n(i)$ is given by probability conservation

$$p_{\xi_2^n(i)\xi_2^n(i)} = 1 - p_{\xi_2^n(i)\mathbf{Z}_n(i)} - p_{\xi_2^n(i)\xi_1^n(i)} - p_{\xi_2^n(i)\mathbf{Z}_{n-1}} - p_{\xi_2^n(i)\xi_1^{n-1}} - p_{\xi_2^n(i)\xi_2^{n-1}} - p_{\xi_2^n(i)\xi_3^{n-1}}, \tag{B.17}$$

where $p_{\xi_2^n(i)\mathbf{Z}_{n-1}}$, $p_{\xi_2^n(i)\xi_1^{n-1}}$ and $p_{\xi_2^n(i)\xi_2^{n-1}}$ represent the probability that the system uses one less resource and, that, respectively, all player are content and aligned, one player ends discontent and two players end discontent. The probability $p_{\xi_2^n(i)\xi_3^{n-1}}$ corresponds to the event where two players not interfered end on the same resource with one of them discontent. A transition $\xi_2^n(i) \rightarrow \mathbf{Z}_{n-1}$ happens if one of the two discontent players selects a resource already occupied and the system ends in an all content and aligned state. The probability of the first events is given by

$$p_{\xi_2^n(i)\mathbf{Z}_{n-1}} = \frac{M_n^i(0)+2}{N}\frac{M_n^i(1)-1}{N}\epsilon^2 + 2\frac{M_n^i(0)+2}{N}\frac{N-M_n^i(1)-M_n^i(0)}{N}\epsilon, \tag{B.18}$$

The transition $\xi_2^n(i) \rightarrow \xi_1^{n-1}$ happens if one of the two discontent players selects a resource already occupied and the system ends with one player discontent. The probability of all these possible events is given by

$$\begin{aligned} p_{\xi_2^n(i)\xi_1^{n-1}} &= 2\frac{N-M_n^i(1)-M_n^i(0)}{N}\epsilon\frac{N-M_n^i(1)-M_n^i(0)}{N}(1-\epsilon) \\ &+ 2\frac{M_n^i(1)-2}{N}\epsilon^2\frac{N-M_n^i(1)-M_n^i(0)+1}{N}(1-\epsilon), \end{aligned} \tag{B.19}$$

where the first terms in the sum deals with the cases in which one of the two discontent players accepts a resource with two players or more and, the second term deals with the case in which one of the two players selects a resource with one player solely.

The probability of transition from $\xi_2^n(i)$ to ξ_2^{n-1} is the probability that one player updates its benchmark with a resource already occupied and that, the systems ends with two discontent players. It is given by

$$p_{\xi_2^n(i)\xi_2^{n-1}} = 2\frac{N-M_n^i(1)-M_n^i(0)}{N}\epsilon\frac{M_n^i(1)-2}{N}(1-\epsilon)^2 + \frac{M_n^i(1)-2}{N}\epsilon^2\frac{M_n^i(1)-3}{N}(1-\epsilon)^2, \tag{B.20}$$

where the first term deals with the case in which, one of the discontent players updates its benchmark with a resource that contains two players or more and, the second term is about the case where both discontent players select a resource with one player. More specifically, in the second term, once the first discontent player has selected a resource with one player and, both have accepted the new benchmark, there is now one less resource with one player, *i.e.* $M_n^i(1)-3$.

The transition $\xi_2^n(i)$ to ξ_3^{n-1} happens if one of the discontent players finds a free resource and, the other selects the resource of a player not interfered. In this last situation one of the two players interfered becomes discontent. This is given by the following probability

$$p_{\xi_2^n(i)\xi_3^{n-1}} = \frac{M_n^i(0)+2}{N}\frac{M_n^i(1)-1}{N}2\epsilon(1-\epsilon), \tag{B.21}$$

The state $\xi_3^n(i)$ is linked to $\mathbf{Z}_n(i)$ and itself in the set $\xi_n(i)$. During transition $\xi_3^n(i) \rightarrow \mathbf{Z}_n(i)$ the system comes back to the all content and aligned state with probability

$$p_{\xi_3^n(i)\mathbf{Z}_n(i)} = \overbrace{\frac{\epsilon}{N}}^{(a)} + \overbrace{\frac{M_n^i(1)}{N}\epsilon^2}^{(b)}, \tag{B.22}$$

where (a) is the probability that the discontent player tries the current resource and that it updates its benchmark. Note that, the player interfered is already aligned in $\xi_3^n(i)$. Term (b) is the probability that the discontent player tries an other resource with one player and both accept the new benchmark.

The probability to remain in $\xi_3^n(i)$ is given by

$$p_{\xi_3^n(i)\xi_3^n(i)} = 1 - p_{\xi_3^n(i)\mathbf{Z}_n(i)} - p_{\xi_3^n(i)\mathbf{Z}_{n+1}} - p_{\xi_3^n(i)\xi_1^{n+1}} - p_{\xi_3^n(i)\xi_2^{n+1}}, \tag{B.23}$$

where $p_{\xi_3^n(i)\mathbf{Z}_{n+1}}$, $p_{\xi_3^n(i)\xi_1^{n+1}}$ and $p_{\xi_3^n(i)\xi_2^{n+1}}$ are the probabilities for the system to use one more frequency from $\xi_3^n(i)$ and, respectively, the system ends with all players content, one player discontent and two players discontent. Note that, these transitions lead to one unique state j in $I_N(n+1)$, $\mathbf{Z}_{n+1}(j)$, $\xi_1^{n+1}(j)$ and $\xi_2^{n+1}(j)$. The transition $\xi_3^n(i) \rightarrow \mathbf{Z}_{n+1}(j)$ occurs if the system ends in an all content mood and aligned with one more frequency used after the experimentation. It happens if the discontent player chooses a free resource with probability

$$p_{\xi_3^n(i)\mathbf{Z}_{n+1}(j)} = \begin{cases} \frac{M_n^i(0)}{N}, \text{ if } S_{n,k(i,j)} = 2, \\ 0, \text{ otherwise.} \end{cases} \tag{B.24}$$

The term $p_{\xi_3^n(i)\mathbf{Z}_{n+1}}$ is the sum over all possible j of $p_{\xi_3^n(i)\mathbf{Z}_{n+1}(j)}$. Consequently, $p_{\xi_3^n(i)\mathbf{Z}_{n+1}} = \frac{M_n^i(0)}{N}$.

The transition $\xi_3^n(i) \rightarrow \xi_1^{n+1}(j)$ occurs if the discontent cluster remains discontent. It happens with probability

$$p_{\xi_3^n(i)\xi_1^{n+1}(j)} = \begin{cases} \frac{N-M_n^i(1)-M_n^i(0)-1}{N}(1-\epsilon), \text{ if } S_{n,k(i,j)} = 2, \\ 0, \text{ otherwise.} \end{cases} \tag{B.25}$$

where the first line is the probability that the discontent player experiments on a resource, with two players or more, except the current one and, that it remains discontent. After this event, the player left alone by the discontent player is no more interfered and accepts the new benchmark with probability 1. The total probability to go in ξ_1^{n+1} is $p_{\xi_3^n(i)\xi_1^{n+1}} = \frac{N-M_n^i(1)-M_n^i(0)-1}{N}(1-\epsilon)$.

The event that leads to transition $\xi_3^n(i) \rightarrow \xi_2^{n+1}(j)$ is realized if the discontent player experiments on an other resource with a cluster not interfered and both end in discontent. This happens with probability

$$p_{\xi_3^n(i)\xi_2^{n+1}(j)} = \begin{cases} \frac{M_n^i(1)}{N}(1-\epsilon)^2, \text{ if } S_{n,k(i,j)} = 2, \\ 0, \text{ otherwise.} \end{cases} \tag{B.26}$$

Using the same reasoning, $p_{\xi_3^n(i)\xi_2^{n+1}} = \frac{M_n^i(1)}{N}(1-\epsilon)^2$.

B.2 Transition from $\xi^n(i)$ to $\xi^{n+1}(j)$

The only states in $\xi^n(i)$ from which the system can use one more frequency are $\mathbf{Z}_n(i)$ and $\xi_3^n(i)$. In other states, the discontent players are alone on their resource, which mean that they cannot discover a new one.

The transitions $\xi_3^n(i) \rightarrow \mathbf{Z}_{n+1}(j)$, $\xi_3^n(i) \rightarrow \xi_1^{n+1}(j)$ and $\xi_3^n(i) \rightarrow \xi_2^{n+1}(j)$ have been described in equations (B.24), (B.25) and (B.26).

The transition $\mathbf{Z}_n(i) \rightarrow \mathbf{Z}_{n+1}(j)$ happens if a player on a resource with $S^i_{n,k(i,j)}$ experiments on a free resource. It is given by probability

$$p_{\mathbf{Z}_n(i)\mathbf{Z}_{n+1}(j)} = P_\epsilon(K)\frac{m_n^i(S^i_{n,k(i,j)})}{K}\frac{M_i^n(0)}{N-1}, \tag{B.27}$$

which is term j of the sum that gives the total probability $p_{\mathbf{Z}_n(i)\mathbf{Z}_{n+1}}$ (B.5). The term $K - m_n^i(1)$ in (B.5) is decomposed as follows $\sum_j m_n^i(S^i_{n,k(i,j)}) = K - m_n^i(1)$.

The transition from $\mathbf{Z}_n(i) \rightarrow \xi_1^{n+1}(j)$ corresponds to the term j of the sum that gives probability $\mathbf{Z}_n(i) \rightarrow \xi_1^{n+1}$ in (B.7). Using the same procedure

$$p_{\mathbf{Z}_n(i)\xi_1^{n+1}(j)} = P_\epsilon(K)\frac{m_n^i(S^i_{n,k(i,j)})}{K}\frac{N-M_n^i(1)-M_n^i(0)-1}{N-1}(1-\epsilon), \tag{B.28}$$

Again with the same decomposition, probability of transition $\mathbf{Z}_n(i) \rightarrow \xi_2^{n+1}(j)$ is obtained using (B.8) as follows

$$p_{\mathbf{Z}_n(i)\xi_2^{n+1}(j)} = P_\epsilon(K)\frac{m_n^i(S^i_{n,k(i,j)})}{K}\frac{M_n^i(1)}{N-1}(1-\epsilon)^2, \tag{B.29}$$

B.3 Transitions from $\xi^{n+1}(j)$ to $\xi^n(i)$

The system can employ one less resource when an alone player selects a resource already occupied as a new benchmark. These transitions are possible from states $\mathbf{Z}_{n+1}(j)$, $\xi_1^{n+1}(j)$ and $\xi_2^{n+1}(j)$. In practice, probability transitions inside $\xi^{n+1}(j)$, are computed with formulas detailed in Section B.1 by replacing the indices appropriately. For example, in this section, the starting state is in $\xi^{n+1}(j)$. Therefore, we use the functions $m^j_{n+1}(.)$ and $M^j_{n+1}(.)$ instead of $m^i_n(.)$ and $M^i_n(.)$. Moreover, during a transition from $\xi^n(i)$ to $\xi^{n+1}(j)$, the resource that has $S^i_{n,k(i,j)}$ is decremented by one. Thus, from $\xi^{n+1}(j)$, any resource that contains $S^i_{n,k(i,j)} - 1$ players can be incremented by one to make the transition to $\xi^n(i)$ occurs.

The transition $\mathbf{Z}_{n+1}(j) \to \mathbf{Z}_n(i)$ happens with probability

$$p_{\mathbf{Z}_{n+1}(j)\mathbf{Z}_n(i)} = \begin{cases} \mathcal{P}_\epsilon(K)\frac{m^j_{n+1}(1)}{K}\frac{M^j_{n+1}(S^i_{n,k(i,j)}-1)}{N-1}\epsilon,\ S^i_{n,k(i,j)} - 1 > 1 \\ \mathcal{P}_\epsilon(K)\frac{m^j_{n+1}(1)}{K}\frac{M^j_{n+1}(1)-1}{N-1}\epsilon^2,\ S^i_{n,k(i,j)} - 1 = 1. \end{cases} \tag{B.30}$$

The first line corresponds to the probability that an alone player experiments on a resource with $S^i_{n,k(i,j)} - 1 > 1$ players, and that it accepts the decrease in utility. The second line corresponds to the probability that an alone player experiments on a resource with $S^i_{n,k(i,j)} - 1 = 1$ player, and that both accept the decrease in utility. From the experimenter point of view, there are $M^j_{n+1}(1) - 1$ resources with one player.

Note that $p_{\mathbf{Z}_{n+1}(j)\mathbf{Z}_n(i)}$ is the term i of the sum that gives $p_{\mathbf{Z}_{n+1}(j)\mathbf{Z}_n} = \sum_i p_{\mathbf{Z}_{n+1}(j)\mathbf{Z}_n(i)}$ in (B.6). Therefore, the first and second line of (B.30) are, with respect to the right indices changes, the term i of the sum that gives the first term and the second term of (B.6) respectively. These similarities are used in what follows.

The transition probabilities $p_{\mathbf{Z}_{n+1}(j)\xi_3^n(i)}$, $p_{\xi_1^{n+1}(j)\mathbf{Z}_n(i)}$, $p_{\xi_1^{n+1}(j)\xi_3^n(i)}$, $p_{\xi_2^{n+1}(j)\xi_1^n(i)}$, $p_{\xi_2^{n+1}(j)\xi_2^n(i)}$ and $p_{\xi_2^{n+1}(j)\xi_3^n(i)}$ correspond to the term i of the sum that gives $p_{\mathbf{Z}_{n+1}(j)\xi_3^n}$ $p_{\mathbf{Z}_{n+1}(j)\mathbf{Z}_n}$, $p_{\xi_1^{n+1}(j)\mathbf{Z}_n}$, $p_{\xi_1^{n+1}(j)\xi_3^n}$, $p_{\xi_2^{n+1}(j)\xi_1^n}$ and $p_{\xi_2^{n+1}(j)\xi_3^n}$ respectively. After changing the indices $n+1$ into n, n into $n-1$ and j into i, one can realize that these probabilities have already been computed. They correspond to $p_{\mathbf{Z}_n(i)\xi_3^{n-1}}$, $p_{\mathbf{Z}_n(i)\mathbf{Z}_{n-1}}$, $p_{\xi_1^n(i)\mathbf{Z}_n}$, $p_{\xi_1^n(i)\xi_3^{n-1}}$, $p_{\xi_2^n(i)\xi_1^{n-1}}$ and $p_{\xi_2^n(i)\xi_3^{n-1}}$ from (B.9), (B.13), (B.14),(B.18), (B.19) and (B.21) respectively. Thus, to obtain $p_{\mathbf{Z}_{n+1}(j)\xi_3^n(i)}$, $p_{\xi_1^{n+1}(j)\mathbf{Z}_n(i)}$, $p_{\xi_1^{n+1}(j)\xi_3^n(i)}$, $p_{\xi_2^{n+1}(j)\xi_1^n(i)}$, $p_{\xi_2^{n+1}(j)\xi_2^n(i)}$ and $p_{\xi_2^{n+1}(j)\xi_3^n(i)}$, we use previous probabilities by changing the indices appropriately and then, the i th term of the sum that results in $N - M^j_{n+1}(0) - M^j_{n+1}(1)$ is selected. This term corresponds to $M^j_{n+1}(S^i_{n,k(i,j)} - 1)$ as

$$\sum_{i, S^i_{n,k(i,j)}-1>1} M^j_{n+1}(S^i_{n,k(i,j)} - 1) = N - M^j_{n+1}(0) - M^j_{n+1}(1). \tag{B.31}$$

With these modifications and, using (B.9), the transition $\mathbf{Z}_{n+1}(j) \to \xi_3^n(i)$ has a probability

$$p_{\mathbf{Z}_{n+1}(j)\xi_3^n(i)} = \begin{cases} 0, \; S^i_{n,k(i,j)} - 1 > 1, \\ P_\epsilon(K)\tfrac{m^j_{n+1}(1)}{K}\tfrac{M^j_{n+1}(1)-1}{N-1}2\epsilon(1-\epsilon), \; S^i_{n,k(i,j)} - 1 = 1. \end{cases} \tag{B.32}$$

From state $\xi_1^{n+1}(j)$ it is possible to go to $\mathbf{Z}_n(i)$ and $\xi_3^n(i)$. During transition $\xi_1^{n+1}(j) \to \mathbf{Z}_n(i)$ the discontent player chooses a frequency that contains $S^i_{n,k(i,j)} - 1$ players. It happens with the following probabilities

$$p_{\xi_1^{n+1}(j)\mathbf{Z}_n(i)} = \begin{cases} \tfrac{M^j_{n+1}(S^i_{n,k(i,j)}-1)}{N}\epsilon, \; S^i_{n,k(i,j)} - 1 > 1, \\ \tfrac{M^j_{n+1}(1)-1}{N}\epsilon^2, \; S^i_{n,k(i,j)} - 1 = 1. \end{cases} \tag{B.33}$$

The first line is similar to the term i of the sum that results in the first term of (B.13). The second line corresponds to the second term of (B.13).

With the same reasoning, using (B.14), $\xi_1^{n+1}(j) \to \xi_3^n(i)$ happens with probability

$$p_{\xi_1^{n+1}(j)\xi_3^n(i)} = \begin{cases} 0, \; S^i_{n,k(i,j)} - 1 > 1, \\ \tfrac{M^j_{n+1}(1)-1}{N}2\epsilon(1-\epsilon), \; S^i_{n,k(i,j)} - 1 = 1. \end{cases} \tag{B.34}$$

The state $\xi_2^{n+1}(j)$ is connected to $\mathbf{Z}_n(i)$, $\xi_1^n(i)$, $\xi_2^n(i)$ and $\xi_2^n(i)$. The probability of transition $\xi_2^{n+1}(j) \to \mathbf{Z}_n(i)$ is obtained using (B.18) as follows

$$p_{\xi_2^{n+1}(j)\mathbf{Z}_n(i)} = \begin{cases} 2\tfrac{M^j_{n+1}(0)+2}{N}\tfrac{M^j_{n+1}(S^i_{n,k(i,j)}-1)}{N}\epsilon, \text{ if } C^i_{n,k(i,j)} - 1 > 1, \\ \tfrac{M^j_{n+1}(0)+2}{N}\tfrac{M^j_{n+1}(1)-1}{N}\epsilon^2, \text{ if } S^i_{n,k(i,j)} - 1 = 1. \end{cases} \tag{B.35}$$

The first line is similar to the term i of the sum that results in the second term of (B.18). The second line is similar to the first term of (B.18).

The probability of transition $\xi_2^{n+1}(j) \to \xi_1^n(i)$, noted $p_{\xi_2^{n+1}(j)\xi_1^n(i)}$, is given by

$$\begin{cases} 2\tfrac{M^j_{n+1}(S^i_{n,k(i,j)}-1)}{N}\epsilon\tfrac{N-M^j_{n+1}(1)-M^j_{n+1}(0)}{N}(1-\epsilon), S^i_{n,k(i,j)} - 1 > 1, \\ 2\tfrac{M^j_{n+1}(1)-2}{N}\epsilon^2\tfrac{N-M^j_{n+1}(1)+1-M^j_{n+1}(0)}{N}(1-\epsilon), S^i_{n,k(i,j)} - 1 = 1. \end{cases} \tag{B.36}$$

which is obtained using (B.19).

The probability of transition $\xi_2^{n+1}(j) \to \xi_2^n(i)$ is given by

$$p_{\xi_2^{n+1}(j)\xi_2^n(i)} = \begin{cases} 2\tfrac{M^j_{n+1}(S^i_{n,k(i,\ell)}-1)}{N}\epsilon\tfrac{M^j_{n+1}(1)-2}{N}(1-\epsilon)^2, \text{ if } S^i_{n,k(i,j)} - 1 \geq 2, \\ \tfrac{M^j_{n+1}(1)-2}{N}\epsilon^2\tfrac{M^j_{n+1}(1)-3}{N}(1-\epsilon)^2, \text{ if } S^i_{n,k(i,j)} - 1 = 1. \end{cases} \tag{B.37}$$

which is obtained using (B.20).

Finally, the probability of transition $\xi_2^{n+1}(j) \to \xi_3^n(i)$ is obtained using (B.21) as follows

$$p_{\xi_2^{n+1}(j)\xi_3^n(i)} = \begin{cases} 0, \text{ if } S^i_{n,k(i,j)} - 1 \geq 2, \\ \frac{M^j_{n+1}(0)+2}{N}\frac{M^j_{n+1}(1)-1}{N}2\epsilon(1-\epsilon), \text{ if } S^i_{n,k(i,j)} - 1 = 1. \end{cases} \tag{B.38}$$

Appendix C

Proofs of theoretical results in Section 4.4

This appendix presents the proofs of Proposition 4.1, Theorems 4.1 stated in Section 4.4 in page 57. In addition, it also provides a shorter proof of Theorem 4.2 than the one given in [34]. They are based on perturbed MC theory whose main concept is presented in Section 2.2. Thus, the convergence proofs of Proposition 4.1, Theorems 4.1 and 4.2 involve the computation of the stochastic potential and its minimization. Before proving these results, we need to show that the MC induced by either the RTEL or the RODL algorithms is a regular perturbed MC. Recall that the matrix representation of these MCs is noted $\mathbf{P}_0$ (as in Section 3.5.4 page 33). For clarity, the reference to the algorithm in the notation is dropped when there is no ambiguity. We distinguish $\mathbf{P}_0^\epsilon$ in which the perturbation $\epsilon > 0$ and $\mathbf{P}_0^0$ in which the process is not perturbed. The convergence results are valid if $\mathbf{P}_0^\epsilon$ is a regular perturbation (see Definition 2.12 page 15) of matrix $\mathbf{P}_0^0$ for each algorithm. For clarity, this

Lemma C.1. *For both algorithms RTEL and RODL, $\mathbf{P}_0^\epsilon$ is a regular perturbation of $\mathbf{P}_0^0$.*

Proof of Lemma C.1. To prove this result, we show the three properties of the Definition 2.12. For the first property, we use the Claim 1 in [10] whose proof is similar here assuming valid the interdependence condition (see Definition 2.4 for the TEL of the ODL). In the case of robust algorithms, this property ensures that given any state of the network, there exists an appropriate action change such that a set of clusters modifies the utility of at least one different cluster more than its tolerance level. We recall this Claim in [10] as follows:

Claim C.1. *Given any state $\mathbf{z} \in \Xi$ in which at least one player is discontent, there exists a sequence of transitions in the unperturbed process to $\overline{D}$.*

where $\overline{D}$ is the set in which all clusters are discontent and, which is recalled for clarity as follows

$$\overline{D} := \{\mathbf{z} = (\mathbf{m}, \bar{\mathbf{a}}, \bar{\mathbf{u}}) \,|\forall i \in \mathcal{K}, m_i = D\} \,. \tag{C.1}$$

With the previous Claim, as long as one player is discontent, the process ends in the all discontent state $\overline{D}$. In addition, all states in Ξ are accessible from $\overline{D}$ by definition of algorithms transitions in Section 4.3. With the interdependence property, there always exists a positive probability for one player to become discontent (*e.g.* two successive experimentations that decrease the utility of a player more than the threshold δ). Consequently, $\mathbf{P}_0^{\epsilon}$ is irreducible since all states communicate (see Definition 2.8 in page 14).

Let prove the aperiodicity of MCs required in Definition 2.12 *(i)*. The periodicity is a class property [70] so as $\mathbf{P}_0^{\epsilon}$ is irreducible, one needs to show that one state is aperiodic. In practice, when the probability to remain in a state is not null it is aperiodic. Since there exists some states in which it is possible to remains at least two successive algorithm iterations (*e.g.* no player experiments two times in a row), this condition is true and *(i)* in Definition 2.12 is valid.

It remains to show conditions *(ii)* and *(iii)*. Condition *(ii)* is the consequence of the algorithms descriptions in Section 4.3. When the the perturbation ϵ tends to 0, one obtains $\mathbf{P}_0^0$ by definition of the perturbation in the algorithm. The last condition *(iii)* is verified because either a transition has a probability proportional to ϵ^0 or $1 - \epsilon^r$ with $r \geq 0$ (*e.g.* no experimentation) or, a transition occurs with a probability that is proportional to ϵ^r with $r > 0$ (*e.g.* experimentation). □

C.1 Proof of Proposition 4.1

Proposition 4.1 is proved by showing that the Recurrence Classes (RC)s of $\mathbf{P}_0^0$ are reduced to only one RC in which all clusters are discontent (*i.e.* the set $\overline{D}$). It follows that this state is the only Stochastic Stable State (SSS) to which the network converges to which is not a desirable state.

Proof of Proposition 4.1. Suppose that all clusters are in a content mood and aligned. In the deterministic context, if no one experiments a new action, each cluster receives a deterministic and constant utility value (see Definition 4.1). However, due to the presence of disturbances (see Definition 4.2), a cluster can see its utility changing two times in a row such that, either it becomes discontent or, content after an update through the hopeful state. In the last case, if the utility comes back to the initial value because of disturbances, the cluster can see its utility decreasing two times in a row and becomes discontent. With the interdependence property and Claim C.1, only one discontent cluster is needed to make others going to D with probability one. Hence, all RCs are reduced to the all discontent state $\overline{D}$ which becomes the SSS. □

C.2 Proof of Theorem 4.1

Theorem 4.1 is proved by analyzing spanning trees over the RCs, $\mathcal{R}$, of the unperturbed process $\mathbf{P}^0_0$. The first part of the proof consists in finding the RCs of $\mathbf{P}^0_0$ which is provided in Lemma C.2 and, the second part of the proof deals with the computation of edge resistances between these RCs through Lemmas C.3 to C.5. The final part of the proof presents the stochastic potential computation and the final results. We follow a procedure similar to the proof in [10] and we highlight the main differences involved.

C.2.1 Recurrence classes of $\mathbf{P}^0_0$

The notion of alignment in [10] is redefined in the disturbed case as follows

$$Z^0_\delta := \{\mathbf{z} = (\mathbf{m}, \bar{\mathbf{a}}, \bar{\mathbf{u}}) | \forall i \in \mathcal{K}, \forall \omega_i, u_i(\bar{\mathbf{a}}; \omega_i) \in [\bar{u}_i - \delta, \bar{u}_i + \delta]\}, \tag{C.2}$$

and the set of content and aligned clusters becomes

$$C^0_\delta := \left\{\mathbf{z} = (\mathbf{m}, \bar{\mathbf{a}}, \bar{\mathbf{u}}) | \mathbf{z} \in Z^0_\delta, \forall i \in \mathcal{K}, m_i = C\right\}. \tag{C.3}$$

Furthermore, the notion of $\boldsymbol{\delta}$-Nash equilibrium with respect to the benchmark with all content and aligned clusters is

$$E^0_\delta := \left\{\mathbf{z} = (\mathbf{m}, \bar{\mathbf{a}}, \bar{\mathbf{u}}) \middle| \mathbf{z} \in C^0_\delta, \forall i \in \mathcal{K}, \forall a_i \neq \bar{a}_i, \bar{u}_i > \max_{\omega_i}(u_i((a_i, \bar{\mathbf{a}}_{-i}), \omega_i)) - \delta\right\}. \tag{C.4}$$

Lemma C.2. *The recurrence classes, $\mathcal{R}$, of the unperturbed process are $\overline{D}$, and all singletons $\mathbf{z} \in C^0_\delta$ such that $\forall i$, $\delta \geq \delta^*$ with $\delta^* = \left\lceil \frac{v-1}{2} \right\rceil w$.*

Proof of Lemma C.2. In the unperturbed process $\mathbf{P}^0_0$, if every player is in D, then the network remains in $\overline{D}$ as the probability to leave D is null. Therefore, $\overline{D} \in \mathcal{R}$. Moreover, if the network is in state $\mathbf{z} \in Z^0_\delta$ without a cluster discontent, then every cluster ends in content mood if $\forall i \in \mathcal{K}$ and $\forall \omega$, $|u_i(\bar{\mathbf{a}}, \omega) - \bar{u}_i| \leq \delta$. Suppose that there exists δ^* such that if $\delta \geq \delta^*$, all clusters go to the content mood in this case and $C^0_\delta \in \mathcal{R}$.

Table C.1: Resistances between states

No.	Transition	Resistance
1	$E^0_\delta \to \bar{D}$	$r(e \to \bar{D}) = 2$
2	$C^0_\delta \backslash E^0_\delta \to \bar{D}$	$r(\mathbf{z} \to \bar{D}) = 1 + G(S_\delta(\mathbf{z}))$
3	$\bar{D} \to C^0_\delta$	$r(\bar{D} \to \mathbf{z}) = \sum_{i \in \mathcal{K}} F(\bar{u}_i)$

The value δ^* is computed afterwards. The condition $\forall i \in \mathcal{K}, \forall \omega, |u_i(\bar{\mathbf{a}}, \omega) - \bar{u}_i| \leq \delta$ means that $\forall i \in \mathcal{K}, \bar{u}_i - \min_\omega(u_i(\bar{\mathbf{a}}, \omega)) \leq \delta$ and $\max_\omega(u_i(\bar{\mathbf{a}}, \omega)) - \bar{u}_i \leq \delta$. In a general case, one of the two left hand side (lhs) inequalities is lower than the other and, we can find an integer $k \in \mathbb{N}$, such that adding kw in one of them equals both lhs. Assume without loss of generality, that this term is added to the lhs of the first inequality such that, there exists always the same δ verifying both equations at the same time. We get the following system

$$\begin{cases} \bar{u}_i - \min_\omega(u_i(\bar{\mathbf{a}}, \omega)) + kw \leq \delta, \\ \max_\omega(u_i(\bar{\mathbf{a}}, \omega)) - \bar{u}_i \leq \delta, \end{cases} \tag{C.5}$$

In this situation, we can add both inequalities which provides

$$\delta \geq \frac{\max_\omega(u_i(\bar{\mathbf{a}}, \omega)) - \min_\omega(u_i(\bar{\mathbf{a}}, \omega)) - 1 + kw}{2}, \tag{C.6}$$

where with (4.8), we obtain,

$$\delta \geq \frac{v - 1 + k}{2} w, \tag{C.7}$$

Thus, the minimum value for the tolerance level δ occurs when $k = 0$. In addition, as $(v - 1)/2$ may not belong to $\mathbb{N}$, we get $\delta \geq \delta^* = \left\lceil \frac{v-1}{2} \right\rceil w$ where $\lceil x \rceil$ is the closest integer above x.

It remains to prove that other states in Ξ are not contained in recurrence classes. To that end, we use the Claim C.1 with similar arguments as in [10]. □

C.2.2 Resistances, trees and stochastic potential

This section presents complementary Lemmas C.3 to C.5 required for a complete tree analysis. We present their proofs only when a major difference with [10] exists. In summary, these Lemmas present the resistances of the tree composed of the RCs of the unperturbed process $\mathbf{P}^0_0$. These resistance values are summarized in Table C.1. With these results, a tree analysis is possible as in [10].

Lemma C.3. *$\forall e \in E^0_\delta$, $r^*(e) = 2$ and $e \to \bar{D}$ is an easy edge.*

First of all we recall the definition of easy edge from [10]:

Definition C.1 (Easy edge)**.** *An edge* $\mathbf{x} \rightarrow \mathbf{y}$ *is called easy if its resistance* $r(\mathbf{x} \rightarrow \mathbf{y})$ *is equal to the minimum resistance needed to leave the starting state* $\mathbf{x}$*. This resistance is noted* $r^*(\mathbf{x})$*.*

Proof of Lemma C.3. The proof is similar to the one in [10] but with definition of E^0 in [10] replaced with E^0_δ (C.4) here. The reason of this change is that in the RTEL, a cluster that experiments can accept the new benchmark only if the received utility is greater than δ. □

Lemma C.4. $\forall \mathbf{z} \in C^0_\delta \backslash E^0_\delta, r^*(\mathbf{z}) = 1 + G(S_\delta(\mathbf{z}))$*, if for* $\mathbf{z}' \in \mathcal{R}$*,* $\mathbf{z} \rightarrow \mathbf{z}'$ *is an easy edge then,* $\overline{W}(\mathbf{z}) < \overline{W}(\mathbf{z}')$*, otherwise,* $\mathbf{z} \rightarrow \bar{D}$ *is an easy edge.*

Proof of Lemma C.4. The main differences with [10] are, the definition of the stability $S_\delta(\mathbf{z})$ described afterwards which results from the presence of the tolerance level and, the increase in benchmark social welfare $\overline{W}(.)$ instead of social welfare $W(.)$. If a new player i has the possibility to increase its payoff benchmark by more than δ into the state $\mathbf{z}'$, the new transition has a resistance $r(\mathbf{z} \rightarrow \mathbf{z}') = 1+G(\bar{u}'_i - \bar{u}_i)$. Thus, the minimum resistance to escape $\mathbf{z}$ is by definition,

$$\begin{aligned} r^*(\mathbf{z}) =& 1 + \min_i \min_{a_i} \min_{\omega_i} (G((u_i((a_i, \bar{\mathbf{a}}_{-i}), \omega_i) - \bar{u}_i))), \\ &= 1 + G(S_\delta(\mathbf{z})), \end{aligned} \tag{C.8}$$

with the decreasing property of G and where, $S_\delta(\mathbf{z}) = \min_{i \in \mathcal{K}} \left\{ S^i_\delta(\mathbf{z}) \right\}$ with,

$$S^i_\delta(\mathbf{z}) = \max_{a_i, \omega_i} \{ u_i((a_i, \bar{\mathbf{a}}_{-i}), \omega_i) - \bar{u}_i | u_i(a_i, \bar{\mathbf{a}}_{-i}, \omega_i) - \bar{u}_i \geq \delta \}.$$

A transition with two experimenting players cannot be an easy edge as the resistance of the transition would be greater than $1 + G(S_\delta(\mathbf{z}))$ due to the design of function $G(.)$ in Section 4.3 as justified in [10]. □

Lemma C.5. $\forall \mathbf{z} \in C^0_\delta$*,* $r(\bar{D} \rightarrow \mathbf{z}) = \sum_{i \in \mathcal{K}} F(\bar{u}_i)$*.*

The difference with [10] comes from the use of $\bar{u}_i$ instead of $u_i(\bar{\mathbf{a}})$.

Lemma C.6. *There exists a* $\overline{D}$*-tree* $T^*_{\overline{D}}$ *that is easy.*

A tree is said to be easy, if it is composed of easy edges. The proof of Lemma C.6 is similar to Lemma 5 in [10] except that in this thesis, the no cycle property is proven using the strict increase in $\overline{W}$ instead of W. Recall that this property is shown in Lemma C.4. In the following, we note ρ^* the stochastic potential of the easy tree $T^*_{\overline{D}}$.

C.2.3 Proof of the main result

The main result presented in Theorem 3.1 can be obtained by looking for the states in $\mathcal{R}$ that minimize the stochastic potential. The stochastic potential of a recurrence class $\mathbf{z} \in \mathcal{R}$ is obtained by computing the resistance of the easy tree rooted at $\mathbf{z}$ because it has the less resistance among all trees rooted at $\mathbf{z}$ (see Lemma 6 in [10]). In this case, starting from the tree $T^*_{\overline{D}}$, it suffices to remove the easy edge $\mathbf{z} \to \overline{D}$ and to add the easy edge $\overline{D} \to \mathbf{z}$ to get an easy tree that is rooted at $\mathbf{z}$. Consequently, the stochastic potential of $\mathbf{z}$ is given by

$$\rho(\mathbf{z}) = \rho^* - r(\mathbf{z} \to \bar{D}) + r(\bar{D} \to \mathbf{z}). \tag{C.9}$$

Therefore, if $\mathbf{c} \in C^0_\delta \backslash E^0_\delta$ then, with resistances given in Table C.1, (C.9) becomes

$$\rho(\mathbf{c}) = (\rho^* - K\phi_2 - \nu_2) - \phi_1 \overline{W}(\mathbf{z}) - \nu_1 S_\delta(\mathbf{z}). \tag{C.10}$$

For a state $\mathbf{e} \in E^0_\delta$, with the resistances given in Table C.1, (C.9) becomes

$$\rho(\mathbf{e}) = (\rho^* - 2) - \phi_1 \overline{W}(\mathbf{z}), \tag{C.11}$$

With the function F and G given in Section 4.3.1 (page 55), it is proved in [10] that $\rho(\mathbf{e}) \leq \rho(\mathbf{c})$. So if there exists a δ-PNE (*i.e.* $E^0_\delta \neq \emptyset$), the SSSs are those which minimize (C.11) or equivalently that minimize the benchmark social welfare among PNEs. Otherwise, if $E^0_\delta = \emptyset$, the SSSs are those that minimize (C.10) or equivalently that maximize a trade-off between the benchmark social welfare and the stability. The proof of Theorem 4.1 is complete.

C.3 Proof of Theorem 4.2

With the same sets Z^0_δ, C^0_δ and $\overline{D}$ defined in (C.2), (C.3) and (C.1) respectively, one can prove that the RCs of the unperturbed process are all singletons in C^0_δ and the set $\overline{D}$ as in the RTEL proof provided in the previous section. The proof relies on similar arguments than in Section C.2.1. Then, with the same arguments employed in [12], the resistances between RCs are similar to the one in [12] after replacing Z^0, C^0 by their counterpart in our work Z^0_δ, C^0_δ respectively ($\overline{D}$ remains unchanged). Consequently, the proof can be derived in the same way and is shorter than the one provided in [34].

In [34], the main difference with our proof is that authors consider the set C^0_δ separated in different sets $C^0, C^1, \ldots, C^m$ with $m < K$. Briefly, in the set C^0, all players have chosen their benchmark utility with respect to a received utility

$u(\bar{\mathbf{a}})$. The set C^1 is composed of a subset of players as in C^0 and a subset of players that keep the same benchmark from C^0 whereas the action has changed. These second subset of players do not change their benchmark because with the tolerance thresholds they do not see an enough significant change in utility to react. Similarly, the set C^2 is composed of a subset of players from C^0, a subset of player from C^1 and a subset of player from C^1 that is not sensitive to a new action change because of the tolerance levels. The definitions of other sets is an iteration of previous description.

In comparison to our proof the simplification is to gather together the sets C^0, $C^1, \dots, C^m$ into one set C^0_δ. The simple idea behind this reduction is to only focus on whether the selected benchmark is stable to disturbances or not. Consequently, it reduces the number of RCs to study and, the complexity of the tree analysis.

Appendix D

Success probability in NB single carrier scheme

We compute the success probability p_i^k defined in Section 5.1.1. Using Definition (5.2) (page 100), p_i^k is given by

$$p_i^k(\Gamma_0) = \Pr\left\{\mathrm{SINR}_i^k > \Gamma_0\right\}, \tag{D.1}$$

where the SINR is a rv modeled by (5.5). Thus, for any link $i = (i_t, i_r)$ in cluster k,

$$p_i^k(\Gamma_0) = \Pr\left\{\frac{\alpha_i^k \mathrm{SNR}_i^k}{1 + \sum_{n\in\mathcal{I}_k}\sum_{j\in\mathcal{M}_n} \alpha_{j,i_r}^{nk}\mathrm{INR}_{j,i_r}^{nk}} > \Gamma_0\right\}. \tag{D.2}$$

Then,

$$p_i^k(\Gamma_0) = \Pr\left\{\alpha_i^k > \frac{\Gamma_0}{\mathrm{SNR}_i^k}\left(1 + \sum_{n\in\mathcal{I}_k}\sum_{j\in\mathcal{M}_n} \alpha_{j,i_r}^{nk}\mathrm{INR}_{j,i_r}^{nk}\right)\right\}. \tag{D.3}$$

In order to compute (D.3), we use the law of total probability with respect to the α_{j,i_r}^{nk} as follows

$$p_i^k(\Gamma_0) = \int_{\mathbf{x}} \Pr\left\{\alpha_i^k > \frac{\Gamma_0}{\mathrm{SNR}_i^k}\left(1 + \sum_{n\in\mathcal{I}_k}\sum_{j\in\mathcal{M}_n} x_{n,j}\mathrm{INR}_{j,i_r}^{nk}\right)\middle|\mathbf{x}\right\} p_{\boldsymbol{\alpha}}(\mathbf{x})d\mathbf{x}, \tag{D.4}$$

where $\boldsymbol{x} = ((x_{n,j})_{n\in\mathcal{I}_k, j\in\mathcal{M}_n})$, $\boldsymbol{\alpha} = ((\alpha_{j,i_r}^{nk})_{n\in\mathcal{I}_k, j\in\mathcal{M}_n})$ and $p_{\boldsymbol{\alpha}}(\mathbf{x})$ is the joint density function of $\boldsymbol{\alpha}$ given by

$$p_{\boldsymbol{\alpha}}(\mathbf{x}) = \prod_{n\in\mathcal{I}_k}\prod_{j\in\mathcal{M}_n} \lambda \exp\left(-\lambda x_{n,j}\right). \tag{D.5}$$

The α_{j,i_r}^{nk} are exponential rvs iid with parameter $\lambda = 1$. In that case, for any fixed $\mathbf{x}$, let note the positive constant $A(\mathbf{x})$, then

$$\Pr\left\{\alpha_i^k > A(\mathbf{x})\middle|\mathbf{x}\right\} = \exp\left(-A(\mathbf{x})\right). \tag{D.6}$$

Gathering (D.6) and (D.5) into (D.4) gives:

$$\begin{aligned}
p_i^k(\Gamma_0) &= \exp\left(-\frac{\Gamma_0}{\mathrm{SNR}_i^k}\right)\int_{\mathbf{x}}\exp\left(-\sum_{n\in\mathcal{I}_k}\sum_{j\in\mathcal{M}_n}x_{n,j}\left(\frac{\Gamma_0\ \mathrm{INR}_{j,i_r}^{nk}}{\mathrm{SNR}_i^k}\right)\right)\prod_{n\in\mathcal{I}_k}\prod_{j\in\mathcal{M}_n}\exp\left(-x_{n,j}\right)d\mathbf{x},\\
&= \exp\left(-\frac{\Gamma_0}{\mathrm{SNR}_i^k}\right)\prod_{n\in\mathcal{I}_k}\prod_{j\in\mathcal{M}_n}\int_{x_{n,j}}\exp\left(-x_{n,j}\left(1+\frac{\Gamma_0\ \mathrm{INR}_{j,i_r}^{nk}}{\mathrm{SNR}_i^k}\right)\right)dx_{n,j},
\end{aligned} \tag{D.7}$$

and integrating over each $x_{n,j}$ gives the final result:

$$p_i^k(\Gamma_0) = \frac{\exp\left(-\frac{\Gamma_0}{\mathrm{SNR}_i^k}\right)}{\prod_{n\in\mathcal{I}_k}\prod_{j\in\mathcal{M}_n}\left(1+\frac{\Gamma_0\ \mathrm{INR}_{j,i_r}^{nk}}{\mathrm{SNR}_i^k}\right)}. \tag{D.8}$$

Appendix E

Some statistics when interference are subject to fading

In this appendix, we derive the pdf of the SINR and its transformation $X = e^{-\frac{SINR}{\beta}}$ used in Section 5.1.3.5 in page 112. In this appendix, we consider that the interference in the SINR are subject to Rayleigh fading. For clarity, let recall the SINR of link i in cluster k in such context (5.36):

$$\mathrm{SINR}_i^k(m) = \frac{\alpha_i^k(m)\mathrm{SNR}_i^k}{1 + \sum_{n\in\mathcal{I}_k}\sum_{j\in\mathcal{M}_n}\alpha_{j,i_r}^{n,k}(m)\mathrm{INR}_{j,i_r}^{n,k}}, \tag{E.1}$$

where $\alpha_i^k(m)$ and $\alpha_{j,i_r}^{n,k}(m)$ are exponential rvs with parameter 1. The pdf of previous rvs is hence $p_\alpha(y) = e^{-y}$. In the following steps we compute the pdf, denoted by $p_{\mathrm{SINR}_i^k}(.)$, of the SINR given by (E.1). For clarity we drop the m indices, and we note $Z = \alpha_i^k\mathrm{SNR}_i^k$ and $W = \sum_{n\in\mathcal{I}_k}\sum_{j\in\mathcal{M}_n}\alpha_{j,i_r}^{n,k}\mathrm{INR}_{j,i_r}^{n,k}$. The pdf of Z is obtained after using a change of variable (it is an exponential rv multiplied by a constant) and $p_\alpha(.)$. It is given by

$$p_Z(z) = \frac{1}{\mathrm{SNR}_i^k}e^{-\frac{z}{\mathrm{SNR}_i^k}}. \tag{E.2}$$

The pdf $p_W(.)$ is more tricky to obtain but it is given in [71] using some convolution derivations

$$p_W(w) = \sum_{n\in\mathcal{I}_k}\sum_{j\in\mathcal{M}_n}\frac{\pi_j^n}{\mathrm{INR}_{j,i_r}^{n,k}}e^{-\frac{w}{\mathrm{INR}_{j,i_r}^{n,k}}}, \tag{E.3}$$

with

$$\pi_j^n = \prod_{n'\neq n\in\mathcal{I}_k}\prod_{j'\neq j\in\mathcal{M}_n}\frac{\mathrm{INR}_{j,i_r}^{n,k}}{\mathrm{INR}_{j,i_r}^{n,k} - \mathrm{INR}_{j',i_r}^{n',k}}. \tag{E.4}$$

With these notations, the pdf of SINR_i^k is computed as follows

$$\begin{aligned} p_{\mathrm{SINR}_i^k}(\gamma) &= \frac{d\mathrm{Pr}\left\{\mathrm{SINR}_i^k < \gamma\right\}}{d\gamma}, \\ &= \int_0^{\infty} \frac{d\mathrm{Pr}\left\{\frac{z}{w+1} < \gamma \middle| w\right\}}{d\gamma} p_W(w)dw, \\ &= \int_0^{\infty} \frac{d\mathrm{Pr}\left\{z < \gamma(w+1) \middle| w\right\}}{d\gamma} p_W(w)dw, \\ &= \int_0^{\infty} (w+1)p_Z(\gamma(w+1))p_W(w)dw, \end{aligned} \tag{E.5}$$

Replacing (E.2) and (E.3) in (E.5) gives

$$\begin{aligned} p_{\mathrm{SINR}_i^k}(\gamma) &= \int_0^{\infty} (w+1)\frac{1}{\mathrm{SNR}_i^k} e^{-\gamma(w+1)\frac{1}{\mathrm{SNR}_i^k}} \sum_{n\in\mathcal{I}_k}\sum_{j\in\mathcal{M}_n} \frac{\pi_j^n}{\mathrm{INR}_{j,i_r}^{n,k}} e^{-\frac{w}{\mathrm{INR}_{j,i_r}^{n,k}}} dw, \\ &= \sum_{n\in\mathcal{I}_k}\sum_{j\in\mathcal{M}_n} \frac{\pi_j^n}{\mathrm{INR}_{j,i_r}^{n,k}} \int_0^{\infty} (w+1)e^{-\frac{w}{\mathrm{INR}_{j,i_r}^{n,k}}} e^{-\gamma(w+1)\frac{1}{\mathrm{SNR}_i^k}} dw, \\ &= \frac{1}{\mathrm{SNR}_i^k} e^{-\frac{\gamma}{\mathrm{SNR}_i^k}} \sum_{n\in\mathcal{I}_k}\sum_{j\in\mathcal{M}_n} \frac{\pi_j^n}{\mathrm{INR}_{j,i_r}^{n,k}} \int_0^{\infty} (w+1)e^{-w(\frac{\gamma}{\mathrm{SNR}_i^k}+\frac{1}{\mathrm{INR}_{j,i_r}^{n,k}})} dw, \end{aligned} \tag{E.6}$$

For clarity of the derivation, we compute the integral term denoted by A separately as follows

$$\begin{aligned} A &= \int_0^{\infty} (w+1)e^{-w(\frac{\gamma}{\mathrm{SNR}_i^k}+\frac{1}{\mathrm{INR}_{j,i_r}^{n,k}})} dw, \\ &= \int_0^{\infty} we^{-w(\frac{\gamma}{\mathrm{SNR}_i^k}+\frac{1}{\mathrm{INR}_{j,i_r}^{n,k}})} dw + \int_0^{\infty} e^{-w(\frac{\gamma}{\mathrm{SNR}_i^k}+\frac{1}{\mathrm{INR}_{j,i_r}^{n,k}})} dw, \\ &= \frac{1}{(\frac{\gamma}{\mathrm{SNR}_i^k}+\frac{1}{\mathrm{INR}_{j,i_r}^{n,k}})^2} + \frac{1}{\frac{\gamma}{\mathrm{SNR}_i^k}+\frac{1}{\mathrm{INR}_{j,i_r}^{n,k}}}, \\ &= \frac{1}{\frac{\gamma}{\mathrm{SNR}_i^k}+\frac{1}{\mathrm{INR}_{j,i_r}^{n,k}}}\left(1+\frac{1}{\frac{\gamma}{\mathrm{SNR}_i^k}+\frac{1}{\mathrm{INR}_{j,i_r}^{n,k}}}\right). \end{aligned} \tag{E.7}$$

Introducing A in (E.6) provides

$$p_{\mathrm{SINR}_i^k}(\gamma) = \frac{1}{\mathrm{SNR}_i^k} e^{-\frac{\gamma}{\mathrm{SNR}_i^k}} \sum_{n\in\mathcal{I}_k}\sum_{j\in\mathcal{M}_n} \frac{\pi_j^n}{\mathrm{INR}_{j,i_r}^{n,k}} \frac{1}{\frac{\gamma}{\mathrm{SNR}_i^k}+\frac{1}{\mathrm{INR}_{j,i_r}^{n,k}}}\left(1+\frac{1}{\frac{\gamma}{\mathrm{SNR}_i^k}+\frac{1}{\mathrm{INR}_{j,i_r}^{n,k}}}\right), \tag{E.8}$$

which concludes the proof of (5.37).

To complete this appendix we provide the derivation of (5.39) which is the pdf of $X = C_1 e^{-\frac{SINR}{\beta}}$ with $C_1 > 0$. Hence, we realize the change of variable $x = g(\gamma) = C_1 e^{-\frac{\gamma}{\beta}}$ using (5.20)

$$p_X(x) = \frac{p_{\mathrm{SINR}_i^k}(g^{-1}(x))}{\left|\frac{dg}{d\gamma}\right|_{\gamma=g^{-1}(x)}}, \tag{E.9}$$

where $g^{-1}(x) = -\beta \ln(\frac{x}{C_1})$ and $\frac{dg}{d\gamma}\Big|_{\gamma=g^{-1}(x)} = -\frac{C_1}{\beta} e^{-\frac{g^{-1}(x)}{\beta}} = -x/\beta$. Now it suffices to apply these changes into (E.8) as follows

$$\begin{aligned} p_X(x) &= \frac{\beta}{x}\frac{1}{\mathrm{SNR}_i^k}\frac{x^{\frac{\beta}{\mathrm{SNR}_i^k}}}{C_1^{\frac{\beta}{SNR_i^k}}}\sum_{n\in\mathcal{I}_k}\sum_{j\in\mathcal{M}_n}\frac{\pi_j^n}{\mathrm{INR}_{j,i_r}^{n,k}}\frac{1}{\frac{1}{\mathrm{INR}_{j,ir}^{n,k}}-\frac{\beta\ln(\frac{x}{C_1})}{\mathrm{SNR}_i^k}}\left(1+\frac{1}{\frac{1}{\mathrm{INR}_{j,ir}^{n,k}}-\frac{\beta\ln(\frac{x}{C_1})}{\mathrm{SNR}_i^k}}\right), \\ &= \frac{\nu}{C_1^{\nu}}x^{\nu-1}\sum_{n\in\mathcal{I}_k}\sum_{j\in\mathcal{M}_n}\frac{\pi_j^n}{\mathrm{INR}_{j,i_r}^{n,k}}\frac{1}{\frac{1}{\mathrm{INR}_{j,ir}^{n,k}}-\nu\ln(\frac{x}{C_1})}\left(1+\frac{1}{\frac{1}{\mathrm{INR}_{j,ir}^{n,k}}-\nu\ln(\frac{x}{C_1})}\right), \end{aligned} \tag{E.10}$$

where $\nu = \frac{\beta}{\mathrm{SNR}_i^k}$ which concludes the proof of (5.39).

Bibliography

[1] M. Conti and S. Giordano, "Mobile ad hoc networking: milestones, challenges, and new research directions," **IEEE Communications Magazine**, vol. 52, no. 1, pp. 85–96, January 2014.

[2] C. R. Lin and M. Gerla, "Adaptive clustering for mobile wireless networks," **IEEE Journal on Selected Areas in Communications**, vol. 15, no. 7, pp. 1265–1275, Sep 1997.

[3] A. Asadi, V. Sciancalepore, and V. Mancuso, "On the efficient utilization of radio resources in extremely dense wireless networks," **IEEE Communications Magazine**, vol. 53, no. 1, pp. 126–132, January 2015.

[4] J. Sucec and I. Marsic, "Hierarchical routing overhead in mobile ad hoc networks," **IEEE Transactions on Mobile Computing**, vol. 3, no. 1, pp. 46–56, Jan 2004.

[5] R. Massin, C. J. L. Martret, and P. Ciblat, "A coalition formation game for distributed node clustering in mobile ad hoc networks," **IEEE Transactions on Wireless Communications**, vol. 16, no. 6, pp. 3940 – 3952, June 2017.

[6] Y. Babichenko, "Completely uncoupled dynamics and Nash equilibria," **Games and Economic Behavior**, vol. 76, no. 1, pp. 1–14, 2012.

[7] H. Tembine, **Distributed strategic learning for wireless engineers**. CRC Press, 2012.

[8] S. Lasaulce and H. Tembine, **Game theory and learning for wireless networks: fundamentals and applications**. Academic Press, 2011.

[9] S. Hart and A. Mas-Colell, "Uncoupled dynamics do not lead to Nash equilibrium," **The American Economic Review**, vol. 93, no. 5, pp. 1830–1836, 2003.

[10] B. S. Pradelski and H. P. Young, “Learning efficient Nash equilibria in distributed systems,” **Games and Economic behavior**, vol. 75, no. 2, pp. 882–897, July 2012.

[11] W. Wang, A. Kwasinski, D. Niyato, and Z. Han, “A survey on applications of model-free strategy learning in cognitive wireless networks,” **IEEE Communications Surveys Tutorials**, vol. 18, no. 3, pp. 1717–1757, thirdquarter 2016.

[12] J. R. Marden, H. P. Young, and L. Y. Pao, “Achieving Pareto optimality through distributed learning,” **SIAM Journal on Control and Optimization**, vol. 52, no. 5, pp. 2753–2770, 2014.

[13] H. Young, “Learning by trial and error,” **Games and Economic Behavior**, vol. 65, no. 2, pp. 626–643, 2009.

[14] M. Bennis and D. Niyato, “A Q-learning based approach to interference avoidance in self-organized femtocell networks,” in **2010 IEEE Globecom Workshops**, Dec 2010, pp. 706–710.

[15] A. Galindo-Serrano and L. Giupponi, “Distributed Q-learning for interference control in OFDMA-based femtocell networks,” in **2010 IEEE 71st Vehicular Technology Conference**, May 2010, pp. 1–5.

[16] F. Wilhelmi, B. Bellalta, C. Cano, and A. Jonsson, “Implications of Decentralized Q-learning Resource Allocation in Wireless Networks,” **ArXiv e-prints**, May 2017.

[17] R. S. Sutton and A. G. Barto, **Reinforcement learning: An introduction**. MIT press Cambridge, 1998, vol. 1, no. 1.

[18] M. A. L. Thathachar and P. S. Sastry, “Varieties of learning automata: an overview,” **IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)**, vol. 32, no. 6, pp. 711–722, Dec 2002.

[19] K. S. Narendra and M. A. Thathachar, **Learning automata: an introduction**. Courier Corporation, 2012.

[20] D. J. Leith and P. Clifford, “A self-managed distributed channel selection algorithm for WLANs,” in **2006 4th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks**, April 2006, pp. 1–9.

[21] Y. Xu, J. Wang, Q. Wu, A. Anpalagan, and Y. D. Yao, "Opportunistic spectrum access in unknown dynamic environment: A game-theoretic stochastic learning solution," **IEEE Transactions on Wireless Communications**, vol. 11, no. 4, pp. 1380–1391, April 2012.

[22] Q. Wu, Y. Xu, J. Wang, L. Shen, J. Zheng, and A. Anpalagan, "Distributed channel selection in time-varying radio environment: Interference mitigation game with uncoupled stochastic learning," **IEEE Transactions on Vehicular Technology**, vol. 62, no. 9, pp. 4524–4538, Nov 2013.

[23] P. S. Sastry, V. V. Phansalkar, and M. A. L. Thathachar, "Decentralized learning of Nash equilibria in multi-person stochastic games with incomplete information," **IEEE Transactions on Systems, Man, and Cybernetics**, vol. 24, no. 5, pp. 769–777, May 1994.

[24] M. Benaïm, "Dynamics of stochastic approximation algorithms," **Séminaire de Probabilités XXXIII**, pp. 1–68, 1999.

[25] M. Bennis, S. M. Perlaza, P. Blasco, Z. Han, and H. V. Poor, "Self-organization in small cell networks: A reinforcement learning approach," **IEEE Transactions on Wireless Communications**, vol. 12, no. 7, pp. 3202–3212, July 2013.

[26] S. M. Perlaza, H. Tembine, and S. Lasaulce, "How can ignorant but patient cognitive terminals learn their strategy and utility?" in **2010 IEEE 11th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)**, June 2010, pp. 1–5.

[27] K. A. Chaitanya, V. Sharma, and U. Mukherji, "Distributed learning of equilibria for a stochastic game on interference channels," in **2015 IEEE 16th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)**, June 2015, pp. 650–654.

[28] J. Zheng, Y. Cai, Y. Xu, and A. Anpalagan, "Distributed channel selection for interference mitigation in dynamic environment: A game-theoretic stochastic learning solution," **IEEE Transactions on Vehicular Technology**, vol. 63, no. 9, pp. 4757–4762, Nov 2014.

[29] S. Hart and A. Mas-Colell, "A simple adaptive procedure leading to correlated equilibrium," **Econometrica**, vol. 68, no. 5, pp. 1127–1150, 2000.

[30] L. Rose, S. Lasaulce, S. M. Perlaza, and M. Debbah, "Learning equilibria with partial information in decentralized wireless networks," **IEEE Communications Magazine**, vol. 49, no. 8, pp. 136–142, August 2011.

[31] L. Rose, S. M. Perlaza, C. J. Le Martret, and M. Debbah, "Self-organization in decentralized networks: A trial and error learning approach," **IEEE Transactions on Wireless Communications**, vol. 13, no. 1, pp. 268–279, January 2014.

[32] L. Rose, S. M. Perlaza, C. J. L. Martret, and M. Debbah, "Achieving Pareto optimal equilibria in energy efficient clustered ad hoc networks," in **2013 IEEE International Conference on Communications (ICC)**, June 2013, pp. 1491–1495.

[33] M. Sheng, C. Xu, X. Wang, Y. Zhang, W. Han, and J. Li, "Utility-based resource allocation for multi-channel decentralized networks," **IEEE Transactions on Communications**, vol. 62, no. 10, pp. 3610–3620, Oct 2014.

[34] C. Ramesh, M. Schmitt, and J. Lygeros, "Distributed learning in the presence of disturbances," in **2016 European Control Conference (ECC)**, June 2016, pp. 257–262.

[35] L. Rose, "Decisional process for ad hoc networks," Ph.D. dissertation, Supélec, ED STIC, 2014.

[36] J. Nash, "Non-cooperative games," **Annals of mathematics**, pp. 286–295, 1951.

[37] D. Fudenberg and D. K. Levine, **The theory of learning in games**. MIT press, 1998, vol. 2.

[38] H. P. Young, "The evolution of conventions," **Econometrica: Journal of the Econometric Society**, pp. 57–84, March 1993.

[39] L. Rose, S. M. Perlaza, M. Debbah, and C. J. Le Martret, "Distributed power allocation with SINR constraints using trial and error learning," in **2012 IEEE Wireless Communications and Networking Conference (WCNC)**, April 2012, pp. 1835–1840.

[40] R. Menon, A. B. MacKenzie, R. M. Buehrer, and J. H. Reed, "A game-theoretic framework for interference avoidance in ad hoc networks," in **IEEE Globecom 2006**, Nov 2006, pp. 1–6.

[41] A. Menon and J. S. Baras, "Convergence guarantees for a decentralized algorithm achieving Pareto optimality," in **American Control Conference (ACC), 2013**. IEEE, 2013, pp. 1932–1937.

[42] J. Gaveau, C. J. Le Martret, and M. Assaad, "Performance analysis of trial and error algorithms," **CoRR**, vol. abs/1711.01788, 2017. [Online]. Available: http://arxiv.org/abs/1711.01788

[43] M. Simsek, M. Bennis, and A. Czylwik, "Dynamic inter-cell interference coordination in HetNets: A reinforcement learning approach," in **2012 IEEE Global Communications Conference (GLOBECOM)**, Dec 2012, pp. 5446–5450.

[44] J. G. Kemeny, "Generalization of a fundamental matrix," **Linear Algebra and its Applications**, vol. 38, pp. 193 – 206, 1981. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0024379581900203

[45] J. G. Kemeny, J. L. Snell **et al.**, **Finite Markov chains**. van Nostrand Princeton, NJ, 1960, vol. 356.

[46] L. Comtet, "Partitions of integers," in **Advanced Combinatorics**. Springer, 1974, pp. 94–126.

[47] D. E. Knuth, "Generating all combinations and partitions," in **The Art of Computer Programming, Volume 4, Fascicle 3**. Addison-Wesley, July, 2005.

[48] A. Agresti and B. A. Coull, "Approximate is better than "exact" for interval estimation of binomial proportions," **The American Statistician**, vol. 52, no. 2, pp. 119–126, 1998. [Online]. Available: http://dx.doi.org/10.1080/00031305.1998.10480550

[49] E. Cameron, "On the estimation of confidence intervals for binomial population proportions in astronomy: The simplicity and superiority of the Bayesian approach," **Publications of the Astronomical Society of Australia**, vol. 28, no. 2, p. 128–139, 2011.

[50] M. Abramowitz and I. A. Stegun, **Handbook of mathematical functions: with formulas, graphs, and mathematical tables**. Courier Corporation, 1964, vol. 55.

[51] A. Gelman, J. B. Carlin, H. S. Stern, D. B. Dunson, A. Vehtari, and D. B. Rubin, **Bayesian data analysis**. CRC press Boca Raton, FL, 2014, vol. 2.

[52] K. Butler and M. Stephens, “The distribution of a sum of binomial random variables,” DTIC Document, Tech. Rep., 1993.

[53] S. Ross and B. Chaib-draa, “Satisfaction equilibrium: Achieving cooperation in incomplete information games,” in **Conference of the Canadian Society for Computational Studies of Intelligence**. Springer, 2006, pp. 61–72.

[54] S. M. Perlaza, H. Tembine, S. Lasaulce, and M. Debbah, “Quality-of-service provisioning in decentralized networks: A satisfaction equilibrium approach,” **IEEE Journal of Selected Topics in Signal Processing**, vol. 6, no. 2, pp. 104–116, April 2012.

[55] J. Proakis, **Digital Communications**, ser. McGraw-Hill series in electrical and computer engineering: Communications and signal processing. McGraw-Hill, 2001. [Online]. Available: https://books.google.fr/books?id=aUp2QgAACAAJ

[56] Y. W. Blankenship, P. J. Sartori, B. K. Classon, V. Desai, and K. L. Baum, “Link error prediction methods for multicarrier systems,” in **IEEE 60th Vehicular Technology Conference, 2004. VTC2004-Fall. 2004**, vol. 6, Sept 2004, pp. 4175–4179 Vol. 6.

[57] M. Pauli, U. Wachsmann, and S.-H. S. Tsai, “Quality determination for a wireless communications link,” June 2007, uS Patent 7,231,183.

[58] S. N. Donthi and N. B. Mehta, “An accurate model for EESM and its application to analysis of CQI feedback schemes and scheduling in LTE,” **IEEE Transactions on Wireless Communications**, vol. 10, no. 10, pp. 3436–3448, October 2011.

[59] M. B. Hcine and R. Bouallegue, “Analytical downlink effective SINR evaluation in LTE networks,” in **2015 IEEE 29th International Conference on Advanced Information Networking and Applications Workshops**, March 2015, pp. 376–381.

[60] J. Fan, Q. Yin, G. Y. Li, B. Peng, and X. Zhu, “MCS selection for throughput improvement in downlink LTE systems,” in **2011 Proceedings of 20th International Conference on Computer Communications and Networks (ICCCN)**, July 2011, pp. 1–5.

[61] H. Song, R. Kwan, and J. Zhang, "Approximations of EESM effective SNR distribution," **IEEE Transactions on Communications**, vol. 59, no. 2, pp. 603–612, February 2011.

[62] J. Francis and N. B. Mehta, "EESM-based link adaptation in point-to-point and multi-cell OFDM systems: Modeling and analysis," **IEEE Transactions on Wireless Communications**, vol. 13, no. 1, pp. 407–417, January 2014.

[63] J. Gaveau, C. J. Le Martret, and M. Assaad, "Grouping of subcarriers and effective SNR statistics in wideband OFDM systems using EESM," in **2017 IEEE 13th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)**, Oct 2017, pp. 1–7.

[64] T. Rappaport, **Wireless Communications: Principles and Practice**. Prentice Hall PTR, 1996, vol. 2.

[65] A. M. Cipriano, R. Visoz, and T. Salzer, "Calibration issues of PHY layer abstractions for wireless broadband systems," in **2008 IEEE 68th Vehicular Technology Conference**, Sept 2008, pp. 1–5.

[66] 3GPP, "Radio transmission and reception," 3rd Generation Partnership Project (3GPP), TS 45.005, Sept 2008. [Online]. Available: http://www.3gpp.org/ftp/Specs/html-info/45005.htm

[67] Z. Wang and G. B. Giannakis, "A simple and general parameterization quantifying performance in fading channels," **IEEE Transactions on Communications**, vol. 51, no. 8, pp. 1389–1398, Aug 2003.

[68] T. W. Anderson and D. A. Darling, "A test of goodness of fit," **Journal of the American statistical association**, vol. 49, no. 268, pp. 765–769, 1954.

[69] 3GPP, "3GPP TSG RAN multiplexing and channel coding (FDD) (release 1999)." TR 25.212, 2002-09.

[70] P. Brémaud, **Markov chains: Gibbs fields, Monte Carlo simulation, and queues**. Springer New York, 1999, vol. 31.

[71] V. Aalo and C. Chayawan, "Outage probability of cellular radio systems using maximal ratio combining in Rayleigh fading channel with multiple interferers," **Electronics Letters**, vol. 36, no. 15, pp. 1314–1315, 2000.

université PARIS-SACLAY

ÉCOLE DOCTORALE
Sciences et technologies
de l'information
et de la communication (STIC)

Titre : Allocation des Ressources pour la Gestion Dynamique du Spectre dans les Réseaux Ad hoc Clustérisés

Mots clés : allocation des ressources, réseaux ad hoc, gestion du spectre, apprentissage sans modèle, OFDM

Résumé : L'objectif de cette thèse concerne l'allocation des canaux fréquentiels dans les réseaux ad hoc organisés en clusters. Les terminaux du réseau sont assemblés localement en clusters afin de garder les avantages des réseaux ad hoc tout en réduisant la quantité de signalisation nécessaire à son fonctionnement. Dans chaque cluster, un chef de cluster (CH en anglais) est désigné parmi les terminaux dont le rôle est de gérer localement les ressources ainsi que les communications. Un des problèmes concerne l'allocation des bandes de fréquence de manière distribuée à chaque cluster pour leur permettre d'opérer correctement. Les fréquences sont une ressource rare ce qui implique que plusieurs clusters sont amenés à utiliser les mêmes et donc à interférer entre eux. Le CH base ses décisions sur une fonction d'utilité qui prend en compte des mesures de performance des communications. Ces dernières peuvent être perturbées à cause des diverses variations dynamiques auxquels sont soumis les réseaux ad hoc. Parmi les algorithmes d'apprentissage distribués, nous avons identifié des méthodes basées sur le paradigme ``d'essais erreur" (TE en anglais) comme des solutions potentielles. Ces algorithmes ont la particularité d'avoir des propriétés de convergence globale intéressantes bien que le problème soit non coopératif.

Dans un premier temps, nous avons étudié la convergence théorique de ces algorithmes en réalisant des approximations de chaînes de Markov dans des cas particuliers. Ensuite, nous avons montré théoriquement et numériquement que le principal défaut des approches TE est leur sensibilité aux variations aléatoires de la mesure d'utilité. Nous avons donc proposé des solutions, avec des preuves théoriques à l'appui, pour adapter ces algorithmes aux cas où l'utilité serait perturbée par des phénomènes aléatoires. Enfin, nous avons analysé de manière plus approfondie l'influence des évanouissements de Rayleigh sur les statistiques de l'utilité.

Title : Resources Allocation for Dynamic Spectrum Management in Clustered Ad hoc Networks

Keywords : resources allocation, ad hoc networks, spectrum management, model-free learning, OFDM

Abstract: This thesis deals with the fully distributed allocation of channels in clustered ad hoc networks. Nodes are gathered locally into clusters in order to keep the advantage of the no infrastructure of ad hoc networks, and to reduce the amount of signalling. In each cluster a node is elected as the Clustered Head (CH) whose role is to manage the resources and the transmissions locally. One of the major problem is to allocate in a distributed way spectrum bands to the clusters in order to make them able to operate. Bandwidth is a scarce resource which implies that several clusters may use the same frequency and hence interfere among each other. The CH realizes the frequency allocation based on a utility function that uses measurements as inputs. These measurements are possibly disturbed due to the wide variety of dynamic changes that face ad hoc networks. Among the distributed learning algorithms, we have identified approaches based on ``trial and error" (TE) paradigm that could solve the channel allocation problem. These approaches possess very attractive global convergence behavior despite the non-cooperativeness of the problem and thus in a broad class of games.

First, based on a specific utility model, we analyse the performance of these algorithms using Markov chains approximations in order to reduce numerical computations complexity. Then, we assess theoretically and numerically that a drawback of TE algorithms is their sensitivity to disturbances. We propose modifications with supporting theoretical proofs in order to adapt the TE algorithms to disturbances of the utility. Furthermore, we study the impact of Rayleigh fading on the utility by deriving its probability density function (pdf) in various contexts.

Université Paris-Saclay
Espace Technologique / Immeuble Discovery
Route de l'Orme aux Merisiers RD 128 / 91190 Saint-Aubin, France