Université / / Paris Seine




ÉCOLE DOCTORALE EM2PSI

ECONOMIE, MANAGEMENT, MATHÉMATIQUES, PHYSIQUE ET SCIENCES INFORMATIQUES

THÈSE

pour obtenir le titre de

Docteur en Sciences

Spécialité Sciences et Technologies
de l'Information et de la Communication

par

MARCASTEL ALEXANDRE

ALLOCATION DE PUISSANCE EN LIGNE DANS UN RÉSEAU IOT DYNAMIQUE ET NON-PRÉDICTIBLE

Impact de l'information reçue

---

21 Fevrier 2019

Devant le jury composé de :

Rapporteure	MYLÈNE PISCHELLA	Maître de conférences au CNAM, CEDRIC
Rapporteur	DAVID GESBERT	Professeur à Eurecom
Examinateur	JEAN-MARIE GORCE	Professeur des Universités à l'INSA Lyon, CITI
Examinateur	MARCEAU COUPECHOUX	Professeur à Télécom ParisTech
Examinateur	ZWI ALTMAN	Ingénieur de Recherche à Orange Labs
Directrice de thèse	INBAR FIJALKOW	Professeure des Universités à l'ENSEA, ETIS
Co-encadrante	E. VERONICA BELMEGA	Maître de conférences à l'ENSEA, ETIS
Co-encadrant	PANAYOTIS MERTIKOPOULOS	Chargé de recherche CNRS au LIG
Invité	JEAN SCHWOERER	Ingénieur de Recherche à Orange Labs

Remerciements

Je tiens dans un premier temps à remercier Inbar FIJALKOW ma directrice de thèse sans qui rien de tout cela n'aurait pu être possible. Merci tout d'abord de m'avoir fait confiance en m'encourageant et en trouvant le financement de cette thèse. Merci également de m'avoir poussé à m'investir dans le fonctionnement du laboratoire et de l'école doctorale, participation qui a amenée de nombreuses discussions intéressantes.

Merci à Panayotis MERTIKOPOULOS d'avoir accepté de participer à l'encadrement de cette thèse et de m'avoir aidé dans les moments de troubles et particulièrement concernant les aspects les plus théoriques que j'ai rencontré.

Un énorme merci à E. Véronica BELMEGA pour son encadrement au quotidien, sa patience, en particulier lors des nombreuses relectures des différentes publications et de ce manuscrit, et ses encouragement permanent lors des moments difficiles inévitables lors d'une thèse. Sa bonne humeur et sa gentillesse ont très fortement contribué au succès de cette thèse.

Mes remerciement vont également à Mylène PISCHELLA et David GESBERT pour avoir accepté de relire cette thèse et d'en être rapporteurs ainsi qu'à Jean-Marie GORCE, Zwi ALTMAN et Marceau COUPECHOUX d'avoir accepté de faire partie de mon jury de thèse.

Je remercie tous les membres du laboratoire pour les nombreuses discussions lors de repas et de pauses, en particulier Marie-Morgane Paumard, Pierre JACOB, Mohamed Amine KHELIF, Habiba LAHDIRI, Javier CEBEIRO et Jérome LORAINE.

Merci à toute ma famille en particulier Maïté MARCASTEL, ma maman, pour les nombreuses relectures parfois (souvent ?) au dernier moment. Merci également à eux pour leurs soutiens et les pauses revigorantes à la campagne pendant ces trois années parfois difficiles.

Finalement un immense merci à Marie-Loup NIVET, ma compagne, qui m'a supporté au jour le jour alors que je n'étais pas toujours supportable. Sans son soutien et sa patience je n'aurais jamais pu terminer cette thèse.

A celles et ceux qui m'ont conseillé et que je n'ai pas cité ici, le cœur y est.

SOMMAIRE

Page

CHAPITRE 1

INTRODUCTION

1.1 L'Internet des Objets

La démocratisation de l'accès à l'Internet via les réseaux cellulaires 3G et 4G [Li et al., 2009; Khan et al., 2009] ainsi que la réduction de la taille des circuits électroniques [Kim et al., 2017; Adegbija et al., 2018] ont permis d'entrevoir la possibilité de connecter de nombreux équipements à l'Internet. Au commencement, ces connections restaient cantonnées aux smartphones mais il est vite apparu un nouveau marché aux contours presque infinis, celui des objets connectés à l'Internet des Objets (IoT).

Une des idées principales derrière l'IoT consiste à connecter ou à interconnecter, massivement, des objets autonomes à un même réseau. Ce dernier peut être soit le réseau Internet, soit un réseau privé. Nous allons voir dans cette introduction que les contours et les défis soulevés par l'IoT sont nombreux et complexes, mais dans un premier temps nous allons nous concentrer sur la définition et les contours de l'IoT.

1.1.1 Définition et contours de l'Internet des Objets

Lorsque nous parlons de connecter des «objets autonomes» il nous faut définir les termes «objet» et «autonome». Donner une définition précise à ces deux mots est l'un des problèmes majeurs concernant les systèmes IoT.

En effet, un «objet» peut être un équipement ou dispositif comme : une montre, un réfrigérateur, une voiture, un capteur, etc. Dans le contexte de cette thèse, nous considérons qu'un objet est un élément matériel disposant des technologies nécessaires aux communications sans fil. Ces objets peuvent avoir des caractéristiques très différentes que ce soit en termes d'applications (transmission de flux vidéo, détection d'événements, mesures, etc.) ou de caractéristiques

physiques (consommation de puissance, contraintes matérielles, etc.)[Sulyman et al., 2017].

La définition du mot «autonome» est encore plus problématique. La notion d'autonomie peut faire référence à l'autonomie énergétique comme des objets sur batterie ou fonctionnant à l'aide de capteurs solaires, etc. Mais peut aussi faire référence à l'autonomie de décision au sein du réseau, dans ce cas l'objet doit être en mesure de déterminer son comportement en fonction de l'évolution du réseau. Dans le cadre de cette thèse, lorsque nous parlons d'un objet autonome nous faisons référence à son autonomie au sein du réseau (bien que l'autonomie énergétique soit prise en compte par le biais de fonction objectif ou des contraintes).

La prochaine section nous permettra de détailler les contours et les possibilités de l'IoT.

L'Internet des Objets : contours et possibilités

Comme nous l'avons déjà précisé, les contours de l'IoT ne font pas consensus. Une définition plus précise d'un réseau IoT pourrait être : un réseau d'objets autonomes possédant la capacité de s'adapter, de s'auto-organiser, de partager des informations, des ressources et s'adaptant aux changements imprévisibles du réseau. Cette notion de réseau autonome implique un grand nombre d'objets hétérogènes partageant le même réseau sans entité centrale pour le gérer. De plus, il existe des applications possibles dans beaucoup de domaines technologiques où sont présentes les communications sans fil.

Le premier endroit pour lequel ont été développés les objets connectés sont les maisons. Alors connu sous le nom de la domotique, l'objectif était de rendre les maisons autonomes [Harmo et al., 2005; Gomez and Paradells, 2010]. Par exemple, la domotique permet la gestion des lumières et de la température par le biais de capteurs de température ou de luminosité. L'IoT permet de généraliser cette idée en connectant les capteurs et les actionneurs au réseau Internet. Ainsi on peut partager des informations avec d'autres utilisateurs mais aussi gérer une maison à distance [Gaikwad et al., 2015; Lee et al., 2014].

En plus de la domotique, nous avons vu apparaître des applications de capteurs personnels comme les montres connectées qui nous permettent de recevoir des messages à notre poignet mais aussi de récupérer des informations (telles que la qualité du sommeil) ou encore le monitoring d'une activité sportive [Bardyn et al., 2016]. Il s'agit ici d'exemples d'applications auxquelles nous pensons quand nous évoquons le développement de l'IoT. Cependant, il y a bien d'autres applications possibles à la fois dans le secteur industriel (production de biens matériels, textiles, constructions, etc.) que dans le secteur tertiaire (commerce, transports, administrations, enseignement, santé, etc.).

Certaines industries, comme les chaînes de productions par exemple, requièrent un grand nombre de capteurs permettant de garantir la sécurité des chaînes de production aussi bien que le suivi de ces dernières [Sadeghi et al., 2015]. Le développement de capteurs sans fil autonomes permet de réduire le nombre de connexions câblées et d'infrastructures tout en augmentant la flexibilité du réseau (en terme de facilité de la mise en place des capteurs et de leurs modifica-

tions). En ce qui concerne le secteur tertiaire nous pouvons imaginer de nombreuses applications dans le domaine du médical et celui de l'aide à la personne par exemple en développant des capteurs permettant le suivi médical des patients à leur domicile, pour ensuite transmettre les informations en temps réel aux médecins.

Enfin, pour faciliter la mobilité des objets, les réseaux cellulaires doivent prendre en compte le développement de l'IoT afin d'optimiser l'utilisation des ressources spectrales (communication radio-cognitive et celui du véhicule connecté) [Goursaud and Gorce, 2015; Chen et al., 2014]. Pour utiliser les réseaux cellulaires, les objets communicants doivent garantir un niveau de qualité de service suffisant pour permettre la communication des utilisateurs originels.

Bien que les applications de l'IoT soient hétérogènes car les contraintes de latence, de débit et de quantité de transmission ne sont pas les mêmes que pour un autre milieu (exemple : le milieu de l'aide à la personne), nous pouvons distinguer certains défis communs à la majorité des applications ci-dessous.

1.1.2 Les défis de l'Internet des Objets

Nous allons différentier deux types de défis : les défis génériques et les défis spécifiques à certaines applications.

Les défis génériques de l'Internet des Objets

Lorsque nous souhaitons connecter un grand nombre d'équipements à un réseau une des premières questions qui se pose est l'**identification de ces objets**. Pour y parvenir, il est nécessaire de développer de nouvelles technologies. Une des pistes étudiées est la technologie *Radio Frequency IDentification* (RFID) qui permet de mettre une étiquette ou un «tag» aux objets [Amendola et al., 2014; Al-Fuqaha et al., 2015]. Ces tags peuvent contenir différentes informations concernant l'objet comme son nom, le protocole de transmission ou toute autre information nécessaire. De plus, les tags RFID ont la particularité de ne pas consommer de puissance puisqu'ils sont passifs ce qui les rend particulièrement attractifs pour l'IoT (car la consommation de puissance est une question essentielle dans ce contexte).

En effet, même si les objets connectés au réseau peuvent être de types différents, une grande partie de ces derniers seront des petits objets fonctionnant sur batterie : capteurs isolés, montres, vêtements, etc. L'**efficacité énergétique** est donc primordiale afin d'augmenter la durée de vie des équipements [Technologies, 2013]. Pour y parvenir, plusieurs axes d'études sont possibles : par exemple réduire la consommation des circuits électriques, augmenter l'efficacité des batteries ou contrôler la puissance nécessaire à la transmission. C'est cette dernière possibilité qui nous intéresse particulièrement car, en améliorant la performance des algorithmes d'allocation des ressources, il est possible de transmettre la même quantité d'informations tout en réduisant la consommation de puissance.

Une augmentation du nombre d'équipements dans un réseau amplifie les **interférences** créées sur le réseau [Al-Fuqaha et al., 2015; Da Xu et al., 2014]. Bien que certains équipements ne transmettent pas en continu (capteurs transmettant une information, une ou deux fois par jour) un état de veille consomme presque autant. De plus, si leur nombre grandit fortement, les interférences augmenteront aussi. Il est donc nécessaire de prendre en compte ces interférences et de développer de nouveaux algorithmes efficaces d'allocation de ressource.

Lors de la communication dans des réseaux cellulaires ou dans des réseaux faiblement dynamiques, il est possible d'**estimer l'état du réseau** en utilisant des trames d'apprentissages [Al-Fuqaha et al., 2015; Da Xu et al., 2014]. Cependant, si le nombre d'objets, donc le nombre de transmissions, augmente, il devient de plus en plus difficile d'estimer l'état du réseau. De plus, il n'est pas possible de prédire le comportement des objets : leur temps de connexions ou le moment de leurs transmissions. Par exemple, nous pouvons imaginer un capteur qui doit détecter un piéton ou encore un capteur de sécurité dans une entreprise qui transmettront uniquement en cas de détection. Les solutions proposées doivent être capable de s'adapter au fait qu'il n'est pas possible de prédire l'évolution du réseau.

Finalement, l'augmentation du nombre d'objets sur un même réseau ainsi que les évolutions dynamiques et non-prévisibles de ces derniers entraînent aussi des **limitations en termes de retour d'information nécessaire par feedback** [Miorandi et al., 2012; Goursaud and Gorce, 2015]. En effet, pour que les transmissions d'un objet soient le plus efficaces possible, il a besoin d'un certain nombre d'informations sur le réseau (gain de canal, interférences, etc.). Ces informations sont transmises en utilisant des bandes de fréquences dédiées à l'objet par le récepteur qui estime ces informations à l'aide de trames d'apprentissages. L'augmentation du nombre d'équipements, propre au réseau IoT densifie le flux d'informations sur ces bandes de fréquences dédiées, ce qui peut créer des problèmes comme : perte de trame par collision entre différentes trames, retransmissions, etc. La réduction de l'information est nécessaire afin de réduire le nombre, ou la taille, des messages envoyés sur la voix de retour. Il est donc impératif de prendre en compte la réduction de feedback pour le développement de solutions efficaces pour les systèmes IoT.

Tous les défis présentés ci-dessus sont communs à la majorité des applications IoT. Cependant nous pouvons aussi distinguer des applications plus spécifiques à certains domaines.

Les défis spécifiques à certaines applications de l'Internet des Objets

Nous avons déjà parlé des capteurs qui, dans la majorité des cas, transmettront d'une manière parcimonieuse (peu de fois par jour et des petites quantités d'informations) [Goursaud and Gorce, 2015]. A l'opposé de ces derniers, il existe des objets qui nécessitent une plus grande quantité d'informations et ont donc besoin d'un **haut-débit**, comme par exemple des caméras ou des casques sans fils qui doivent transmettre un flux d'informations important.

Il existe aussi des applications qui demandent une **très faible latence** [Goursaud and

Gorce, 2015]. Nous pouvons citer en exemple les capteurs médicaux, encore les capteurs de sécurité d'une chaîne de montage ou des futures voitures connectées. Il faut donc que ces équipements transmettent leurs informations avec une latence très faible afin de garantir la sécurité des utilisateurs/patients.

En plus de la latence, il faut aussi prendre en compte la **fiabilité des transmissions** [Goursaud and Gorce, 2015]. Certaines applications requièrent une grande fiabilité en termes de limitation du nombre d'erreurs lors de la transmission comme par exemple la transmission d'informations pour le suivi médical d'un patient. Les objectifs de fiabilité peuvent devenir problématiques lorsque beaucoup d'objets cherchent à transmettre en même temps. Si le nombre de communications augmente (beaucoup d'objets communiquent en même temps) sur un même support le risque d'erreur augmente ce qui entraîne des pertes de message et donc une perte de fiabilité.

Tous ces défis peuvent être appliqués dans une grande partie des technologies utilisées pour les communications sans fils, comme le codage canal pour l'aspect fiabilité [Zhang et al., 2016b], la cryptographie pour la sécurité [Routray et al., 2017], mais aussi en ce qui concerne la gestion des ressources au niveau de la couche physique [Karpuk and Chorti, 2016]. Concernant la couche physique, une optimisation de l'utilisation des ressources (spectrales et énergétiques) permet d'améliorer le comportement général du réseau : augmentation de la durée de vie des équipements, augmentation du nombre d'objets sur le réseau, etc. C'est pourquoi, ma thèse se focalise en particulier sur ce type de problème d'allocation de ressource. L'objectif de la prochaine section est donc d'extraire des défis ci-dessus, les différents objectifs sur lesquels nous souhaitons nous concentrer.

1.1.3 Les objectifs de cette thèse

Le premier des objectifs dont nous allons parler est la **gestion de puissance**. Une grande partie des systèmes de communication utilise des transmissions multi-porteuses (*Orthogonal frequency-division multiplexing* (OFDM)) ou multibandes. La question qui se pose est de savoir comment répartir la puissance sur ces différentes porteuses ou bandes afin d'optimiser les communications en fonction de contraintes propres à chaque problème.

Ensuite viennent le **contrôle de débit** et le **contrôle d'interférences**. Bien qu'étroitement liés les deux objectifs sont opposés : augmenter le débit d'un objet va augmenter la puissance d'émission et donc les interférences sur le réseau. Cette augmentation des interférences va obliger les autres utilisateurs du réseau à augmenter leur puissance, ce qui va nécessairement créer de nouvelles interférences. Il y a un compromis à faire entre les débits des objets et les interférences du réseau.

Ensuite vient la **gestion de la dynamique du réseau**. Dû à la forte mobilité des objets et à la présence de communications sporadiques, il est difficile pour l'objet de connaître ou de prédire l'état du système à l'instant où il doit transmettre. Contrairement, à la majorité des

travaux sur les problèmes d'allocations de ressources, qui se focalisent sur des cas statiques ou stationnaires, nous faisons l'hypothèse que le réseau est dynamique et imprévisible.

Finalement, le dernier objectif concerne la réduction du feedback d'information nécessaire à l'objet. Nous avons vu que plus le nombre augmente dans le réseau plus la taille du feedback devient importante. Ainsi l'objectif est de concevoir des algorithmes efficaces d'allocation de ressource qui requièrent le moins d'informations possibles afin de maximiser l'utilisation du réseau (en libérant des ressources spectrales pour d'autres applications).

Maintenant que nous avons isolé les objectifs relatifs à l'IoT nous allons faire l'état de l'art concernant les problèmes d'allocation de ressources qui relèvent de ces objectifs.

1.2 Allocation de puissance dans les communications sans fil

Les problèmes d'allocation de puissance et plus généralement des ressources ont été largement étudiés dans la littérature portant sur l'allocation de ressource dans les systèmes de communication, en particulier en ce qui concerne les systèmes statiques ou stochastiques.

1.2.1 Allocation de puissance dans des systèmes statiques/stochastiques

Dans cette partie nous allons réaliser un bref résumé des techniques utilisées pour résoudre les problèmes d'allocation de puissance dans le cas des systèmes statiques/stochastiques. Ce résumé sera découpé en deux sous-parties : une première qui se concentrera sur les problèmes classiques de communications sans-fils et une seconde qui se focalisera sur des systèmes de communications IoT.

Allocation de puissance dans des systèmes sans-fils

Les problèmes d'allocation de ressources et en particulier les problèmes d'allocation de puissance dans les réseaux de communication sans fils ont fait l'objet de nombreuses études [Saraydar et al., 2002], [Scutari and Barbarossa, 2003], [Gesbert et al., 2007], [Wang et al., 2007], [Pang et al., 2008], [Scutari et al., 2009], [Pischella and Le Ruyet, 2013]. Ce domaine étant très vaste, nous allons dans cette partie présenter uniquement quelques approches et les algorithmes principaux utilisés pour résoudre ce type de problème.

Un des algorithmes fondateurs de l'allocation de puissance dans les réseaux de communication sans fil est l'algorithme du *water-filling* [Shannon, 1949]. Cet algorithme peut être adapté afin de déterminer l'allocation de puissance pour un grand nombre de systèmes différents : par exemple, dans le cas de d'allocation de puissance sous contrainte de débit dans un système de canal à interférence [Pang et al., 2008], mais aussi dans le cas des systèmes *Multiple Inputs Multiple outputs* (MIMO) [Scutari et al., 2009]. Cet algorithme a aussi été utilisé pour résoudre les problèmes d'allocation de puissances dans des systèmes de communication radio-cognitifs comme par exemple [Wang et al., 2007]. De plus, ils requièrent une connaissance des gains de

canaux qui peuvent être limités, comme dans [Bagayoko et al., 2011] où les auteurs utilisent un estimateur des gains de canaux pour déterminer l'allocation de puissance.

Un autre algorithme classique pour déterminer contrôler l'interférences dans un réseau de communication sans fils et l'algorithme adaptatif proposé par Foschini et Miljanic [Foschini and Miljanic, 1993]. Les auteurs proposent un algorithme distribué de contrôle d'interférence qui requière uniquement des informations locales (de puissances et d'interférences) tout en garantissant de bonne performance en terme de contrôle d'interférences.

Une autre approche classique pour résoudre les problèmes d'allocation de ressource est d'utiliser la théorie des jeux [Von Neumann and Morgenstern, 1944], [Nash, 1951], [Morgenstern and Von Neumann, 1953]. L'idée de cette théorie est de considérer chaque transmetteur comme un preneur de décision avec un objectif et des actions propres. La théorie des jeux permet de résoudre des problèmes multi-agents (multiples preneurs de décision) et ainsi de prendre en compte les actions des autres objets. La théorie des jeux a été appliquée dans de nombreux cas différents. Dans [Saraydar et al., 2002], les auteurs utilisent la théorie des jeux, et en particulier des jeux non-coopératifs, pour résoudre un problème de contrôle de puissance sous contrainte de débit. La théorie des jeux a aussi été appliquée dans le cadre de système *Multiple Inputs Single output* (MISO) [Scutari and Barbarossa, 2003] afin de déterminer le codage et l'allocation de puissance optimale.

Allocation de puissance dans des systèmes IoT

Les problèmes d'allocation de ressource dans un environnent IoT statique ont été largement étudiés [Safdar et al., 2013], [Ali et al., 2016], [Zheng et al., 2016].

Dans [Safdar et al., 2013] les auteurs se focalisent sur un système de communication Machines vers Machines (M2M) dans lequel chaque machine souhaite maximiser son propre débit sous contrainte de puissance. L'objectif des communications M2M est de faire communiquer d'un manière autonome des groupes d'utilisateurs/objets sans perturber le réseau déjà existant, cf les communications de type radio-cognitive. Les auteurs utilisent la théorie des jeux pour résoudre ce problème et ils étudient la différence entre les jeux non-coopératifs et les jeux-coopératifs. Ils montrent que dans les jeux non-coopératifs, où chaque joueur joue de manière égoïste et non-coopérative, la solution correspond à l'équilibre de Nash. Cependant, si les joueurs ont des fonctions différentes alors l'équilibre de Nash n'est pas forcement équitable (certains joueurs risquent de ne jamais communiquer). C'est pourquoi les auteurs étudient aussi le problème sous la forme d'un jeux coopératif et ils montrent que dans ce cas la solution obtenue est plus équitable comparée à une approche non-coopérative.

Dans [Ali et al., 2016] les auteurs s'attaquent à un problème de regroupement d'objets (*clustering*) et de maximisation de débit global dans un réseaux d'accès non-orthogonal (*Non-Orthogonal Multiple Access* (NOMA)). Dans un premier temps les auteurs utilisent les propriétés des réseaux NOMA et la technique de décodage utilisée pour développer un algorithme qui

regroupe les utilisateurs. La seconde étape consiste à déterminer l'allocation de puissance qui maximise le débit du groupe d'utilisateurs. Pour cela, les auteurs se servent des outils classiques d'optimisation (conditions d'optimalité de Karush-Kuhn-Tucker). Pour parvenir à leurs résultats, les gains de canaux sont considérés statiques lors de la phase de regroupement et d'allocation de puissance. De plus, une entité centrale est nécessaire pour déterminer quels utilisateurs regrouper.

En ce qui concerne le contrôle d'interférence, les auteurs de [Zheng et al., 2016] cherchent à limiter l'interférence mutuelle dans un système IoT industriel. L'objectif ici est de permettre aux objets d'utiliser un réseau déjà existant (le réseau primaire) sans trop le perturber. Leur premier résultat consiste à définir les conditions pour limiter les interférences sur le réseau primaire. Une fois ces conditions définies, les auteurs utilisent des techniques d'optimisation convexe classique pour déterminer l'allocation de puissance optimale.

Tous les travaux cités ci-dessus considèrent que le système est statique pendant la durée de la communications et qu'il est nécessaire d'avoir des informations sur l'état du système au moment de la transmission.

1.2.2 Allocation de puissance dans des systèmes dynamiques et imprévisibles

Exemples de solution pour l'allocation de ressource dans des systèmes dynamiques

Des algorithmes d'allocation de ressources adaptatives basés sur des techniques d'optimisation en ligne ont été récemment proposés, mais dans des systèmes de communications et pour des problèmes différents.

Dans [Anandkumar et al., 2011], les auteurs posent le problème de l'allocation de bandes de fréquences pour des utilisateurs secondaires d'un réseau radio-cognitif dynamique et non-prédictible. Les auteurs détaillent des algorithmes permettant la détection de la disponibilité et l'allocation des bandes de fréquences en ligne, dans les cas où le nombre d'utilisateurs est connu et inconnu.

Une approche similaire a été utilisée par les auteurs de [Hashemi et al., 2017] pour un problème d'alignement de faisceaux dans le cas de communication à onde millimétrique où *millimaterWave*. Ici l'objectif n'est plus d'allouer une certaine quantité de puissance, mais plutôt de choisir l'alignement optimal de l'antenne parmi un ensemble fixé. Pour cela, les auteurs proposent d'utiliser une approche à l'aide de l'apprentissage en ligne, notamment inspirée du problème du bandit manchot.

Les travaux les plus proches de cette thèse, sont [Mertikopoulos and Belmega, 2014, 2016]. Dans [Mertikopoulos and Belmega, 2014], les auteurs étudient un problème d'allocation de puissance dans un système radio-cognitif MIMO-OFDM dynamique et imprévisible. Les auteurs proposent un algorithme d'apprentissage basé sur les méthodes d'optimisation en ligne (*matrix exponential learning*) afin de déterminer une allocation de puissance efficace lorsque l'objet dis-

pose d'un feedback vectoriel. Dans [Mertikopoulos and Belmega, 2016], les auteurs utilisent une approche similaire pour étudier le problème de l'efficacité énergétique dans un système MIMO. Les différences majeures avec nos travaux est la quantité nécessaire d'informations de feedback nécessaire au receveur pour déterminer son allocation de puissance.

A notre connaissance, nos travaux sont les premiers à étudier la réduction du feedback à un scalaire dans le cas de systèmes de communication IoT dynamiques et imprévisibles.

1.3 Structure et contributions

Dans cette section, nous allons présenter la structure et les principales contributions de cette thèse.

Dans le **Chapitre 2**, nous allons détailler le modèle du système IoT étudié, le problème que nous souhaitons résoudre ainsi que les notions de base de l'optimisation en ligne et les métriques utilisées. Le problème d'optimisation en ligne présenté se focalise sur la minimisation de puissance sous contraintes de débit.

Dans le **Chapitre 3**, nous allons présenter notre algorithme *Online Exponential Learning* (OXL). Cet algorithme permet de déterminer une allocation de ressource qui minimise le regret dans le cas où l'objet a accès à l'information du gradient ou à une version bruitée de ce dernier. Nous illustrerons ensuite les performances de notre algorithme en termes de regret, mais aussi en comparaison avec un algorithme classique en allocation de puissance : le *water-filling*.

Dans le **Chapitre 4**, nous allons nous intéresser à l'impact de la réduction du feedback. Pour cela, nous allons passer d'un feedback vectoriel (Chapitre 3) à un feedback scalaire. Nous étudierons l'impact de cette réduction de feedback sur l'algorithme OXL et nous présenterons un nouvel algorithme *Online Exponential Learning with zeroth order feedback* (OXL_0). Cette présentation sera suivie par des simulations numériques illustrant l'impact de la réduction d'information.

Le **Chapitre 5**, sera dédié à la généralisation du problème présenté dans le Chapitre 2. Ce chapitre sera l'occasion de généraliser aussi les algorithmes proposés (OXL et OXL_0) afin de les appliquer à d'autres types de problèmes. Pour illustrer cela, nous présenterons un problème de contrôle d'interférence dans un système IoT.

Finalement, le **Chapitre 6** présentera une conclusion des travaux réalisés ainsi que les différentes perspectives possibles.

1.4 Publications de l'auteur

Pour clore ce chapitre, cette thèse a mené aux publications suivantes :

— [C4] **A. Marcastel**, E.V. Belmega, P. Mertikopoulos, and I. Fijalkow, "Gradient-free Online Resource Allocation Algorithms for Dynamic Wireless Networks", papier invité á *IEEE SPAWC*, Nice 2019

[Jsub] **A. Marcastel**, E. V. Belmega, P. Mertikopoulos, and I. Fijalkow, "Online power optimization in feedback-limited, dynamic and unpredictable IoT networks", en révision (*major revision*), *IEEE Trans. on Signal Processing*, Sep. 2018.

[C3] **A. Marcastel**, E.V. Belmega, P. Mertikopoulos, and I. Fijalkow, "Online interference mitigation via learning in dynamic IoT environments", *IOE Worksop in IEEE GLOBECOM 2016*, Washington DC, USA, 4-8 Dec. 2016.

[C2] **A. Marcastel**, E.V. Belmega, P. Mertikopoulos, and I. Fijalkow, "Interference mitigation via pricing in time-varying cognitive radio systems", **invited paper**, *NetGCoop 2016*, Avignon, France, Nov. 2016.

[C1] **A. Marcastel**, E.V. Belmega, P. Mertikopoulos, and I. Fijalkow, "Online power allocation for opportunistic radio access in dynamic OFDM networks", *IEEE VTC-Fall 2016*, Montreal, Canada, 18-21 Sep. 2016.

1.5 Présentations de l'auteur

[P5] **A. Marcastel**, "Online Power Minimization in Dynamic IoT Networks : The Impact of Feedback Scarcity", *Séminaire IMT* , Lille, France Décembre 2018.

[P4] **A. Marcastel**, "Online Power Minimization in Dynamic IoT Networks : The Impact of Feedback Scarcity", *Journée des doctorants*, ETIS, Cergy-Pontoise, France Février 2018.

[P3] **A. Marcastel**, "Resource Allocation via Learning in Higly Dynamic Environment", *DIGICOSME Spring School*, Gif-surYvette, France, Mai 2017 (Poster).

[P2] **A. Marcastel** "Resource Allocation via Learning in Higly Dynamic Environment", *Journée des Doctorants*, ETIS, Cergy-Pontoise, France Février 2017, **Prix de la meilleure présentation**.

[P1] **A. Marcastel**, "Interference Mitigation via Pricing in Time-Varying Multi-User Cognitive Radio System", *Journée GDR ISIS : Learning in Networks and Beyond*, Télécom ParisTech, France, Paris, Mai 2016

CHAPITRE 2

PROBLÈME DE MINIMISATION DE PUISSANCE DANS UN RÉSEAU IOT

Dans ce chapitre,nous allons détailler le modèle du système IoT étudié et définir le problème que nous allons chercher à résoudre. Le problème de minimisation de puissance étudié rentre dans la famille des problèmes d'optimisation en ligne. Problème pour lesquels la fonction objectif à minimiser n'est pas connue à l'instant où l'objet doit déterminer son allocation de puissance. Suite à la présentation du problème, nous allons introduire quelques éléments de base de l'optimisation en ligne et surtout la notion de **non regret**, qui permet d'évaluer l'efficacité d'une politique d'allocation de puissance en ligne.

2.1 Modèle du système

Le système étudié est composé de M dispositifs émetteurs et N récepteurs utilisant des communications multi-porteuses de type OFDM avec S sous-porteuses, comme illustré dans la

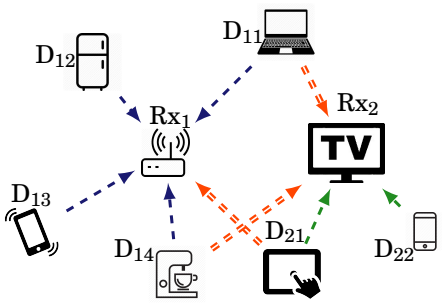


FIGURE 2.1: Système composé de six émetteurs (D_{11}, D_{12}, etc.) et deux récepteurs (Rx_1, Rx_2). Les flèches bleues simples représentent les liens directs tandis que le doubles-flèches représentent les liens interférants.

Fig. 2.1, tous les utilisateurs peuvent utiliser les S sous-porteuses, ou une sous-partie, simultanément. Cependant, chaque émetteur communique avec un receveur unique, qui reçoit plusieurs signaux en même temps.

Nous pouvons écrire le signal reçu par le récepteur dans la bande S comme :

$$r^s(t) = h^s(t)x^s(t) + \sum_j h^s_j(t)x^s_j(t) + z^s(t), \tag{2.1}$$

où :

— $h^s(t)$ et $x^s(t)$ sont respectivement le gain du lien direct et le signal transmis par l'objet focal,

— $h^s_j(t)$ et $x^s_j(t)$ sont le gain de canal interférant et le signal transmis par l'objet j (un objet interférant),

— $z^s(t)$ représente le bruit.

L'indice $s = \{1,\dots,S\}$ dans l'équation (2.1) représente la sous-porteuse OFDM ou la bande.

Nous pouvons maintenant définir les gains de canaux effectifs de l'objet focal $\mathbf{w}(t) = (w^s(t))$ par :

$$w^s(t) = \frac{g^s(t)}{\sigma^2 + \sum_j g^s_j(t)p^s_j(t)}, \tag{2.2}$$

où :

— σ^2 est la variance du bruit $z^s(t)$,

— $p^s_j(t)$ est la puissance allouée par l'objet j dans la sous-porteuse s,

— $g^s(t) = |h^s(t)|^2$,

— $g^s_j(t) = |h^s_j(t)|^2$.

L'équation définissant les gains de canaux effectifs implique que le récepteur utilise un décodage uniquement sur l'utilisateur focal (*Single User Decoding* (SUD)). Cela signifie que lorsque le récepteur décode le signal d'un objet en particulier, il considère les autres signaux entrants comme du bruit. Cette approche est réaliste dans les réseaux IoT car un décodeur n'est pas forcement en mesure de décoder le signal des autres objets interférant (e.g., le type de codage utilisé par les autres objets est inconnu). De plus, dans un système décentralisé il est difficile d'utiliser une méthode de décodage par annulation succesive d'interference (*Succesive Interference Cancelation* (SIC)) par manque de coordination entre les objets.

2.2 Formulation du problème

Un des objectifs majeurs de l'IoT est de connecter un grand nombre d'équipements hétérogènes sur un même réseau. Leur fonctionnement, majoritairement alimenté par des batteries, rend leur consommation de puissance cruciale [Technologies, 2013]. L'objectif de cette thèse est donc de concevoir des algorithmes d'allocation de ressources limitant au maximum la consommation de puissance, tout en garantissant une certaine Qualité de Service (QoS). Les indicateurs

de QoS sont de nature multiple et dépendent fortement de l'application. Il est possible de définir des indicateurs de QoS pour la majorité des défis présentés dans le Chapitre 1, que ce soit la latence, le débit, etc. En ce qui concerne le débit de transmission, il existe plusieurs possibilités : le débit minimal par utilisateur ou encore le débit global du réseau (le débit global est équivalent à la somme des débits de tous les objets du réseau). Le choix de ces indicateurs dépendra de l'application ciblée.

Dans notre contexte d'efficacité énergétique, nous considérerons la minimisation de puissance sous contrainte de débit minimal par objet. Le problème d'optimisation peut donc être formulé de la façon suivante :

$$\begin{array}{lll} \text{minimiser} & \sum_{s=1}^{S} p^s(t) & \\ \text{sur} & \mathbf{p}(t) = (p^1(t), \ldots, p^S(t)) & \\ \text{sous contraintes} & p^s(t) \geq 0, \ \forall s & \text{(C11)} \\ & \sum_{s=1}^{S} p^s(t) \leq P_{\max} & \text{(C12)} \\ & R_t(\mathbf{p}) \geq R_{\min} & \text{(C13)} \end{array} \tag{2.3}$$

où $\mathbf{p} = (p^1, \ldots, p^S)$ est le vecteur des allocations de puissance de l'objet focal pour chacune des sous-porteuses et $R_{\min}$ est la contrainte de débit, $R_t(\mathbf{p})$ est le débit de l'objet focal défini par le débit de Shannon :

$$R_t(\mathbf{p}) = \sum_{s=1}^{S} \log(1 + w^s(t) p^s(t)). \tag{2.4}$$

Du à l'absence d'hypothèse quand à l'évolution du réseau, l'objet n'est pas en mesure de calculer le débit $R_t(\mathbf{p})$ à l'instant t, ce qui implique que l'espace faisable du problème d'optimisation 2.3 est inconnu. Le fait que l'espace faisable ne soit pas connu rend ce problème très difficile à résoudre. Pour y remedier, il faut remarquer que seule la contrainte (C13) est inconnue, nous allons donc introduire cette contrainte dans la fonction objectif. Pour réaliser cela nous allons utiliser une fonction de pénalité définie par :

$$L_t(\mathbf{p}) = \sum_{s=1}^{S} p^s + \lambda \left[R_{\min} - R_t(\mathbf{p}) \right]^+, \tag{2.5}$$

et $[x]^+ = \max\{0, x\}$. Cette fonction de pénalité implique que si l'objet a un débit supérieur à la contrainte ($R_t(\mathbf{p}(t)) \geq R_{\min}$) alors aucune pénalité n'est appliquée, sinon nous appliquons une pénalité égale à la différence multipliée par λ. Le paramètre λ, mesuré en W/(bps/Hz), nous permet donc de contrôler le compromis entre la minimisation de puissance et la contrainte de débit minimal comme nous allons le voir dans le Chapitre 3. Nous avons choisi une pénalité linéaire car ce type de pénalité est souvent utilisé dans les problèmes d'allocation de ressources [Alpcan et al., 2002; Altman and Wynter, 2004; Masmoudi et al., 2014], cependant les solutions peuvent être généralisées à des fonctions de pénalité plus générales, i.e., fonctions concaves, comme les fonctions de pénalité logarithmique [Chiang et al., 2008].

Maintenant que nous avons détaillé la fonction objectif, il est possible de poser formellement le problème d'optimisation étudié comme :

$$\begin{array}{lll} \text{minimiser} & L_t(\mathbf{p}(t)) & \\ \text{sur} & \mathbf{p}(t) = (p^1(t), \ldots, p^S(t)) & \\ \text{sous contraintes} & p^s(t) \geq 0, \ \forall s & \text{(C1)} \\ & \sum_{s=1}^{S} p^s(t) \leq P_{\max} & \text{(C2)} \end{array} \tag{2.6}$$

Notons la présence de deux contraintes :

— (C1) qui impose la positivé des puissances allouées dans chacune des sous-porteuses,

— (C2) qui limite la puissance maximale disponible.

Pour résoudre le problème ci-dessus nous ne pouvons pas utiliser les méthodes classiques d'optimisation [Boyd and Vandenberghe, 2004]. En effet, $L_t(\mathbf{p}(t))$ dépend du débit $R_t(\mathbf{p}(t))$ qui lui dépend de $\mathbf{w}(t)$ et donc des allocations de puissance des autres objets à l'instant t et de l'évolution des gains de canaux $g_j^s(t)$ et $g^s(t)$. Les gains de canaux peuvent être modélisés ou estimés à l'inverse des allocations de puissance des autres objets qui sont non-prévisibles. L'objet doit minimiser une fonction objectif qui lui est inconnue. Ce genre de problème sort du cadre des problème cadre de l'optimisation décentralisé classique et rentre dans celui des problèmes d'optimisation en ligne.

Dans la prochaine section nous allons nous concentrer sur une introduction rapide à l'optimisation en ligne et sur les politiques d'allocation de puissance en ligne [Shalev-Shwartz, 2011; Bubeck et al., 2012].

2.3 Optimisation en ligne et politiques d'allocation de puissance dynamique

Lorsque la fonction objectif $L_t(\mathbf{p}(t))$ est inconnue à l'instant de décision on parle d'un problème d'optimisation en ligne. L'objectif n'est plus de trouver l'allocation optimale, $\mathbf{p}^*(t)$ qui minimise $L_t(\mathbf{p}(t))$ à chaque instant t car cela est impossible. L'objectif est donc de déterminer une allocation de puissance en ligne $\mathbf{p}(t)$ qui donne une valeur la meilleure possible en utilisant uniquement une quantité d'informations limitée sur le réseau. Pour cela, nous supposons que l'objet reçoit un retour d'information ou feedback du récepteur après chaque transmission. L'objet utilisera ce feedback pour déterminer l'allocation de puissance à l'instant suivant $t+1$. La politique d'allocation de puissance en ligne à l'instant t est supposée causale ce qui implique que les informations contenues dans le feedback sont relatives uniquement à l'état du réseau à l'état du passé $t-1$.

Pour proposer des politiques efficaces d'allocation en ligne, nous allons utiliser des méthodes d'apprentissage et d'optimisation en ligne. Puisque l'objet n'a aucune information sur le système à l'instant t, il va chercher à utiliser les informations dont il dispose, c'est à dire le feedback envoyé par le récepteur après la communication à l'instant $t-1$, pour essayer d'«apprendre» et de

Algorithme OLG : Online Learning Generalized

Initialisation : $\mathbf{y}(0) \leftarrow 0$; $t \leftarrow 0$.

Répéter

- **Phase de pré-transmission :** mise à jour de la puissance $\mathbf{p}(t)$ en utilisant le score $\mathbf{y}(t)$
- **Transmission avec** $\mathbf{p}(t)$
- **Phase de post-transmission :** réception du feedback $\mathbf{v}(t)$

Mise à jour du score : $\mathbf{y}(t+1) = \mathbf{y}(t) - \mu(t)\mathbf{v}(t)$

$t \leftarrow t+1$

jusqu'à : fin de transmission

«s'adapter» aux évolutions du système. À chaque nouvelle itération (i.e., après avoir transmis un message) l'objet va recevoir des informations (i.e., un feedback) sur l'état du système à l'itération précédente. Ces informations permettent à l'objet d'apprendre sur le réseau dans lequel il communique à l'instant t par exemple cela peut lui permettre de connaître le niveau d'interférence dans les différentes sous-porteuses ou bandes.

Puisque l'objet n'est pas en mesure de prédire l'évolution du système, ce dernier va se baser sur les informations recueillies après la transmission à l'instant t pour déterminer son allocation de puissance à la prochaine itération, $t+1$. L'objet va répéter ce processus pour toutes les itérations, i.e. transmettre puis recevoir un feedback et ensuite modifier son allocation de puissance en fonction de ce feedback pour l'itération suivante. À chaque nouvelle itération, l'objet dispose d'un peu plus d'information sur le système, ce qui lui permet de s'adapter à ce dernier.

Un exemple général d'algorithme d'allocation de puissance est donné avec l'algorithme 1. Dans cet exemple, les feedbacks de tous les instants précédent sont agrégés dans un score interne $\mathbf{y}(t)$ et l'objet utilise ce score pour déterminer l'allocation de puissance à chaque instant. Le paramètre $\mu(t)$ est le pas de l'algorithme. Ce pas permet de contrôler l'influence des feedbacks dans l'allocation de puissance, nous verrons dans les chapitres 3 et 4 que ce pas joue un grand rôle dans les performances des nos algorithmes.

L'avantage des méthodes d'optimisation et d'apprentissage en ligne est que nous ne faisons aucune hypothèse quant à l'évolution du système (i.e. gains de canaux, évolution des utilisateurs, etc.).

2.3.1 La notion de regret comme métrique de performance

Dans les problèmes d'optimisation classique il est relativement facile de définir une métrique pour quantifier l'efficacité de la solution obtenue. En effet, il suffit de comparer la solution obtenue avec la solution centralisée optimale (la solution qui minimise la fonction objectif). Cette solution optimale peut être calculée en utilisant un algorithme dont on connaît la fiabilité pour résoudre le type de problème étudié. Cependant, dans le cas de l'optimisation en ligne, la comparaison avec la solution optimale est difficile et trop ambitieuse. En effet, nous avons vu que la fonction objectif n'est pas connue à l'instant où l'objet doit déterminer son allocation de puis-

sance. Puisque cette fonction objectif n'est pas connue, et que nous n'avons fait aucune hypothèse quant à la variation des fonctions objectif, l'objet n'est pas en mesure de déterminer l'allocation de puissance qui minimise cette fonction. La comparaison avec la solution optimale est donc désavantageuse pour l'objet dans le sens où nous savons que l'objet ne peut pas atteindre cette solution. Nous pouvons utiliser une nouvelle métrique de performance pour juger de l'efficacité de nos algorithmes en ligne. Pour cela, nous allons utiliser la **notion de regret** [Hannan, 1957; Shalev-Shwartz, 2011; Bubeck et al., 2012].

La notion de regret compare notre allocation de puissance dynamique $\mathbf{p}(t)$ à une référence fixe dans le temps $\mathbf{q}$. L'idée derrière l'allocation de puissance fixe est de déterminer ce que l'objet aurait alloué comme puissance s'il avait eu connaissance du système sur toute la fenêtre de transmission. La fenêtre de transmission T, correspond au nombre d'itérations total de transmissions et donc au nombre total d'allocations de puissance à définir : $\mathbf{p}(t),\ t \in \{1,\dots,T\}$. Autrement dit, l'allocation de puissance fixe est la solution optimale qui minimise la moyenne des fonctions objectifs sur l'horizon T. Le regret est la différence des valeurs de la fonction objectif obtenues à l'aide de l'allocation de puissance dynamique $\mathbf{p}(t)$ et par l'allocation de puissance fixe. Cela peut être formulé comme suit :

$$\mathrm{Reg}(T) \triangleq \sum_{t=1}^{T} L_t(\mathbf{p}(t)) - \min_{\mathbf{q}\in\mathscr{P}} \sum_{t=1}^{T} L_t(\mathbf{q}), \tag{2.7}$$

où :

$$\mathscr{P} \triangleq \left\{ \mathbf{p} \in \mathbb{R}^S | p^s \geq 0\ \forall s, \sum_{s=1}^{S} p^s \leq P_{\max} \right\}, \tag{2.8}$$

correspond à l'ensemble faisable du problème.

L'intuition derrière le regret est que, plus ce dernier est grand, plus l'allocation de puissance dynamique s'éloigne de la meilleure allocation fixe. A l'inverse, si le regret est nul ou négatif, alors l'allocation de puissance dynamique fait au moins aussi bien que la meilleure allocation fixe. Donc, notre objectif est de concevoir des algorithmes qui ont le meilleur regret possible (i.e., le plus faible possible).

Pour quantifier l'efficacité d'une politique d'allocation de puissance dynamique, nous allons utiliser la propriété de non regret [Hannan, 1957; Shalev-Shwartz, 2011]. Une allocation de puissance dynamique a la propriété de non regret si

$$\limsup_{T\to\infty} \frac{1}{T}\mathrm{Reg}(T) \leq 0. \tag{2.9}$$

Une allocation de puissance qui a la propriété de non regret est optimale de manière asymptotique, c'est-à-dire que la solution dynamique garantit d'obtenir des performances au moins aussi bonnes que la meilleure solution fixe et ceci avec une information strictement causale sur le système.

Remarque 2.1. *Même si le regret ne compare pas l'allcation en ligne avec l'allocation de puissance optimale dynamique, $\boldsymbol{p}^*(t)$, la proprieté de non-regret reste non triviale car le calcul de la*

meilleure solution moyenne fixe nécéssite une connaissance non causale de l'évolution des fonctions objectifs. Il faut remarquer que dans le cas statique, i.e. lorsque $L_t(\boldsymbol{p}(t)) = L(\boldsymbol{p}(t))\ \forall t$*, la proprieté de non-regret garantit que l'allocation de puissance en ligne converge vers l'allocation de puissance optimale* $\boldsymbol{p}^*$ *qui minimise* $L(\boldsymbol{p})$*. À l'opposé, lorsque les gains de cannaux sont i.i.d., la propriété de non-regret garantit la convergence vers l'allocation de puissance uniforme qui est optimale allocation optimale dans ce cas [E. V. Belmega and Sanguinetti, 2018].*

Il faut remarquer que le regret dépend des allocations de puissance dynamique $\mathbf{p}(t)$ et ces allocations de puissances dépendent du feedback, $\mathbf{v}(t)$, reçu par l'objet à chaque instant. Si ce feedback est une variable aléatoire, ce qui peut être le cas lorsque ce feedback est un estimateur bruité, alors l'allocation de puissance dynamique $\mathbf{p}(t)$ est aussi aléatoire. Dans ce cas là, les variations aléatoire de l'allocation de puissance rendent la notion de regret difficile à étudier, c'est pourquoi il nous faut introduire la notion de **regret moyen**.

Regret moyen

La notion de **regret moyen** est une extension de la notion de regret dans la situation où l'allocation de puissance dynamique dépend d'une variable aléatoire[Shalev-Shwartz, 2011] . Le regret moyen est noté $\mathrm{EReg}(T)$ et est défini par :

$$\mathrm{EReg}(T) = \mathbb{E}[\mathrm{Reg}(T)], \tag{2.10}$$

où l'espérance est calculée par rapport à la variable aléatoire du système. L'espérance dans la définition du regret moyen dépendra du problème spécifique étudié. Nous verrons deux applications différentes dans lesquelles nous devrons calculer le regret moyen. Dans le Chapitre 3, nous étudierons le cas d'un feedback bruité, i.e., le gradient plus un bruit d'estimation. D'après [Shalev-Shwartz, 2011], ce bruit doit respecter les conditions ci-dessous :

$$\begin{aligned} &\mathbb{E}[\tilde{\mathbf{v}}(t)] = \nabla L_t(\mathbf{p}(t)) \\ &\mathbb{E}[\|\tilde{\mathbf{v}}(t)\|_\infty^2] \leq \tilde{V}^2, \end{aligned} \tag{2.11}$$

où $\tilde{V}$ est une constante. Dans le Chapitre 4, nous étudierons le cas où l'objet a accès uniquement à un feedback scalaire. L'objet devra construire un estimateur du gradient de la fonction objectif à partir de ce feedback scalaire.

Puisque nous avons défini la propriété de regret moyen, il est aussi possible de lui associer la propriété de non regret de la même manière qu'avec le regret classique. Ainsi, une politique d'allocation de puissance en ligne a la propriété de non regret en moyenne si :

$$\limsup_{T\to\infty} \frac{1}{T}\mathrm{EReg}(T) \leq 0. \tag{2.12}$$

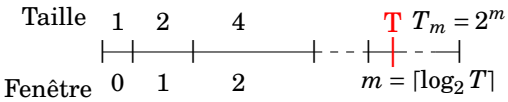


FIGURE 2.2: Illustration du doubling-trick pour une transmission de longueur T découpée en m fenêtres ;

Rôle des paramètres du système

Notre objectif est de concevoir des algorithmes dynamiques qui ont la propriété de non regret comme défini par l'équation (2.9) où (2.12). Pour cela, nous devons trouver une borne supérieure du regret puis calculer la limite de cette dernière lorsque la fenêtre de transmissions T tend vers l'infini.

Cette borne sur le regret dépend des paramètres du système. Ces paramètres vont être différents en fonction des applications. Dans notre cas, les paramètres peuvent être : le nombre total de sous-porteuses S, la puissance maximale $P_{\max}$, le débit minimal $R_{\min}$, la durée de transmission T, mais aussi le pas $\mu(t)$. Le pas $\mu(t) > 0$ est un paramètre qui permet à l'objet de contrôler l'importance qu'il accorde au feedback. Plus $\mu(t)$ est grand plus l'objet accordera d'importance au feedback et inversement lorsque $\mu(t)$ est petit. La question qui se pose est de savoir comment déterminer un pas, $\mu(t)$, qui garantit la propriété de non regret. Puisque la borne du regret peut dépendre de tous les paramètres, y compris T, la durée de transmission, et de $\mu(t)$, le pas de l'algorithme, il faut faire la distinction entre le cas où la fenêtre de transmission T est connue et celui où cette fenêtre est inconnue.

Durée de transmission T connue

Dans le cas où la fenêtre de transmission est connue à l'avance, l'objet connaît toutes les informations sur le système. Il peut utiliser un pas constant $\mu(t) = \mu$ et déterminer la valeur optimale du pas, en optimisant la borne du regret en fonction de μ. De cette manière, l'objet est capable de déterminer le pas optimal qui va minimiser la borne du regret en fonction de tous les paramètres du systèmes.

Dans le cas où la fenêtre de transmission n'est pas connue, l'objet n'est pas en mesure de déterminer le pas optimal. Nous devons donc trouver une autre méthode pour déterminer ce pas.

Durée de transmission T inconnue

Dans le cas où la durée de transmission n'est pas connue à l'avance, il existe une méthode générique qui permet d'obtenir la propriété de non regret. Cette méthode s'appelle l'astuce du dédoublement ou *doubling-trick* [Shalev-Shwartz, 2011]. L'idée est d'utiliser des fenêtres de transmission dont la taille double à chaque fin de fenêtre et ce jusqu'à la fin de la transmission, comme illustré en Fig. 2.2. Ainsi, la première fenêtre de transmission sera de taille 1. Si la

transmission n'est pas finie à la fin de cette fenêtre, l'objet considérera une fenêtre de taille 2, puis de taille 4 et ainsi de suite jusqu'à la fin de la transmission. Le fait de définir des fenêtres de taille fixe permet de connaître le pas optimal et donc la borne du regret dans chacune des fenêtres. Et ainsi, le regret total est la somme des regrets de chaque fenêtre.

Un calcul rapide [Shalev-Shwartz, 2011] permet de montrer que, si le regret pour la fenêtre i est borné par $\mathrm{Reg}_i(T_i) = CT_i^\gamma$, alors le regret total pour un horizon T inconnu peut être borné comme suit :

$$\mathrm{Reg}(T) \leq \sum_{i=1} \mathrm{Reg}_i(T_i) \leq C\frac{2^{2\gamma}}{2^\gamma - 1}T^\gamma, \tag{2.13}$$

où $R(T)$ est la borne du regret calculé avec la pas optimal correspondant à une durée de transmission T. Cela signifie que le regret, dans le cas où la durée de transmission T est inconnue, est borné similairement, à une constante près, que la borne du regret dans le cas où T est connue. L'avantage principal de cette méthode est qu'il suffit de concevoir des algorithmes efficaces (en terme de regret) dans le cas où la durée de transmission est connue pour pouvoir le généraliser au cas où la durée de transmission n'est pas connue.

CHAPITRE 3

ALLOCATION DE PUISSANCE À L'AIDE D'UN FEEDBACK DU PREMIER ORBRE

Dans ce chapitre nous allons supposer que l'objet focal reçoit un feedback vectoriel de taille S noté $\mathbf{v} \in \mathbb{R}^S$, où S est le nombre de sous-porteuses OFDM ou le nombre de bandes. Il existe de nombreuses possibilités de feedback vectoriel. Par exemple, le récepteur peut transmettre directement l'allocation de puissance à chaque objet ou le gradient de la fonction objectif ou encore le rapport signal sur bruit dans chaque sous-porteuse. Étant donné que nous étudions le cas d'un système décentralisé dans lequel chaque émetteur doit calculer sa propre politique d'allocation de puissance, nous allons nous intéresser uniquement au cas d'un feedback vectoriel qui est soit le gradient parfait soit un estimateur non biaisé de ce dernier.

3.1 Information parfaite sur le gradient

Pour comprendre pourquoi nous allons utiliser le gradient comme feedback, il faut revenir aux problèmes d'optimisations classiques. Une des méthodes possibles pour résoudre un problème d'optimisation convexe consiste à suivre la direction inverse du gradient [Boyd and Vandenberghe, 2004] pour mettre à jour l'allocation de puissance. Cependant, dans le cas de l'optimisation en ligne l'objet n'a pas accès à la fonction objectif à l'instant t. Il ne peut donc pas calculer le gradient de la fonction objectif à cet instant. Pour pallier ce problème, l'objet utilisera le gradient de la fonction objectif à l'itération précédente $t-1$.

L'information donnée par le gradient de la fonction objectif à l'instant $t-1$ peut-être soit pertinente, s'il y a de faibles variations des fonctions objectifs dans le temps, soit totalement obsolète dans le cas de variations trop importantes à l'instant t. Notre objectif est d'assurer que sur un horizon de temps T, le gradient à l'itération précédente suffit pour déterminer une

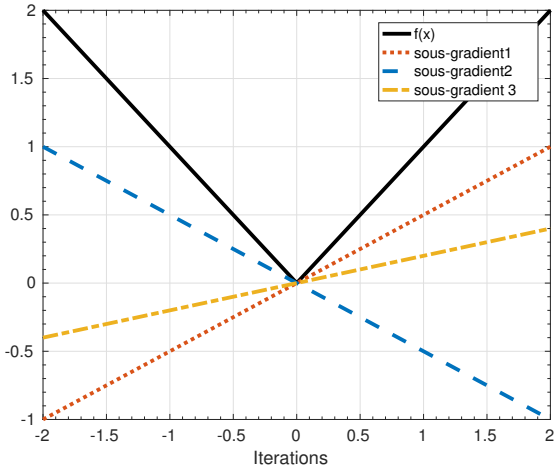


FIGURE 3.1: Illustration de la notion de sous-gradients pour la fonction $f(x) = |x|$. La fonction $f : \mathbb{R} \to \mathbb{R}_+$ est différentiable partout sauf en 0. Toutes les pentes des tangentes qui minorent la fonction f et qui passent par 0 sont dans l'ensemble des sous-gradients de la fonction f au point 0.

politique d'allocation de puissance qui garantit de bonnes performances en terme de regret, c'est à dire comparé à la meilleure solution moyenne.

3.1.1 Analyse du feedback

Avant de détailler l'algorithme que nous allons étudier, la première étape est de définir à quel type d'information l'objet a accès. Comme nous l'avons précisé dans l'introduction de ce chapitre, il existe plusieurs types de feedback vectoriel.

Pour déterminer le type de feedback renvoyé par le récepteur, il faut déterminer l'information connue par le récepteur. Dans le cas d'un système décentralisé, le récepteur connaît peu d'information sur les objets. Ainsi le récepteur ne connaît pas le débit minimal que doit respecter un objet, $R_{\min}$ et les paramètres (s'il y en a) utilisés dans les algorithmes. Cependant, le récepteur est en mesure de connaître les Rapport Signal sur Interférence plus Bruit (RSIB) dans chaque sous-porteuses, via l'utilisation de trame d'apprentissage connue à l'émetteur et au récepteur, mais aussi les interférences de tous les objets. Ces informations permettent au récepteur de calculer le rapport signal à bruit d'un objet particulier.

C'est pourquoi, nous allons faire l'hypothèse que l'objet a accès au rapport signal-à-bruit. Le rapport signal-à-bruit sera utilisé pour calculer le gradient de la fonction objectif définie en (2.5). Bien que convexe sur l'ensemble $\mathscr{P}$, la fonction objectif définie dans (2.5) n'est pas différentiable partout, à cause de la pénalité sur le débit atteignable : $\lambda[R_{\min} - R_t(\mathbf{p})]^+$. Nous ne pouvons donc pas parler du gradient, mais nous devons utiliser la notion de sous-gradient.

La notion de sous-gradient permet d'étendre le gradient aux points non-différentiables. L'en-

semble des sous-gradients regroupe toutes les pentes des tangentes qui passent par le point non différentiables et qui minore la fonction objectif au point en question, mathématiquement cela ce traduit par :

Definition 3.1. L'ensemble des sous-gradients d'une fonction convexe $f : \mathscr{P} \to \mathbb{R}$ au point $\mathbf{p}_0$ est l'ensemble des points $\mathbf{s} \in \mathscr{P}$ tel que :

$$f(\mathbf{y}) \geq f(\mathbf{p}_0) + \langle \mathbf{s} | \mathbf{y} - \mathbf{p}_0 \rangle \,, \ \forall \mathbf{y} \in \mathscr{P}. \tag{3.1}$$

La figure 3.1 permet d'illustrer la notion de sous-gradient en utilisant la fonction simple $f(x) = |x|$. La courbe en noire continue $f(x)$ est différentiable entre $[-2;0)$ et $(0;2]$ cependant elle ne l'est pas en 0. Un sous-gradient de $f(x)$ au point 0 est donc la pente de n'importe quelle droite qui minore (i.e. qui ne dépasse jamais la courbe de $f(\mathbf{x})$) la courbe noire et qui passe par le point en 0, cela correspond par exemple aux pentes des courbes rouge, orange et bleue.

Pour revernir à notre fonction objectif, nous avons aussi deux parties différentiables et la composante s du gradient est :

$$\frac{\partial L_t(\mathbf{p})}{\partial p^s} = \begin{cases} 1 & \text{si } R_t(\mathbf{p}) > R_{\min} \\ 1 - \lambda \frac{g^s}{\sigma^2 + g^s p^s + \sum_j g_j^s p_j^s} & \text{si } R_t(\mathbf{p}) < R_{\min}. \end{cases} \tag{3.2}$$

L'ensemble des points où la fonction $L_t(\mathbf{p})$ n'est pas différentiable est le sous-ensemble de $\mathscr{P}$ des points $\mathbf{p}$ tels que $R_t(\mathbf{p}) = R_{\min}$. Dans ce cas, les deux expressions ci-dessus sont des sous-gradients possibles. Nous pouvons choisir une des deux indifféremment. Nous choisirons $\frac{\partial L_t(\mathbf{p})}{\partial p^s} = 1$ et de ne pas appliquer la pénalité lorsque $R_t(\mathbf{p}) = R_{\min}$. Ce qui nous donne finalement pour la composante s du sous-gradient :

$$\frac{\partial L_t(\mathbf{p})}{\partial p^s} = \begin{cases} 1 & \text{si } R_t(\mathbf{p}) \geq R_{\min} \\ 1 - \lambda \frac{g^s}{\sigma^2 + g^s p^s + \sum_j g_j^s p_j^s} & \text{si } R_t(\mathbf{p}) < R_{\min}. \end{cases} \tag{3.3}$$

Par simplicité dans la présentation et dans les notations par la suite nous parlerons de gradient et garderons la notation $\nabla L_t(\mathbf{p})$ au lieu du sous-gradient.

Maintenant que nous avons défini le gradient, qui représente l'information reçue par l'objet, la question qui se pose est : comment peut-on utiliser cette information pour déterminer une allocation de puissance ? Pour expliquer l'algorithme que nous allons utiliser, nous allons commencer par présenter une approche classique de l'optimisation en ligne : l'algorithme du suivi du leader ou *Follow-the-Leader* (FoL) et une de ces variantes : l'algorithme du *Follow-the-Regularized-Leader* (FoRL).

3.1.2 Présentation de quelques algorithmes classique de l'optimisation en ligne

Rappelons que notre objectif est de concevoir un algorithme qui garantit la propriété de non regret sur une fenêtre de transmission T. Une des manières les plus naturelles de déterminer l'allocation de puissance est d'utiliser l'allocation de puissance qui minimise la somme

des fonctions objectif de toutes les itérations précédentes. Il s'agit de l'algorithme appelé FoL [Shalev-Shwartz, 2011] et il peut être résumé par l'équation suivante, qui donne l'allocation de puissance à l'itération $t+1$ en fonction du passé :

$$\mathbf{p}(t+1) = \underset{\mathbf{q}\in\mathscr{P}}{\arg\min} \left\{ \sum_{i=1}^{t} L_i(\mathbf{q}) \right\}. \tag{3.4}$$

Sachant que nous n'avons fait aucune hypothèse sur les variations des fonctions objectif $L_t(\mathbf{q}), \forall t$ il est possible que les variations de ces dernières puissent entraîner des oscillations sur l'allocation de puissance $\mathbf{p}(t+1)$.

Pour comprendre l'impact de ces oscillations, imaginons une situation simple dans le cas où $S = 1$. Les fonctions objectif sont définies à chaque instant par :

$$L_t(p) = \begin{cases} +p & \text{si } t \text{ est pair,} \\ -p & \text{si } t \text{ est impair,} \end{cases} \tag{3.5}$$

et $p \in [0, P_{\max}]$. Ainsi à chaque instant t l'objet utilise l'Algorithme FoL pour déterminer sa politique d'allocation de puissance. L'algorithme du FoL nécessite de connaître la somme des fonctions objectif jusqu'à l'instant t, dans cet exemple nous pouvons différencier deux cas :

$$\sum_{i=1}^{t} L_i(p) = \begin{cases} 0 & \text{si } t \text{ est pair,} \\ -p & \text{si } t \text{ est impair.} \end{cases} \tag{3.6}$$

Pour déterminer son allocation de puissance, l'objet doit donc résoudre :

$$p(t+1) = \begin{cases} \arg\min_{\mathbf{q}\in[0,P_{\max}]}\{0\} = 0 & \text{si } t \text{ est pair,} \\ \arg\min_{\mathbf{q}\in[0,P_{\max}]}\{-p\} = P_{\max} & \text{si } t \text{ est impair.} \end{cases} \tag{3.7}$$

Cela implique que lorsque l'itération est pair ($L_t(p) = +p$) l'objet allouera $P_{\max}$ et inversement lorsque t est impair ($L_t(p) = -p$) l'objet utilisera 0. Il y a donc la présence d'oscillations dans la solution dynamique, oscillations qui peuvent être plus ou moins fortes en fonction des fonctions objectif et de l'espace faisable.

Pour contrer ce phénomène d'oscillations une solution, est d'utiliser une fonction de régularisation $f(\mathbf{p})$ en plus de l'Algorithme FoL. L'algorithme utilisant cette fonction de régularisation s'appelle : l'algorithme du FoRL et est défini par l'équation suivante :

$$\mathbf{p}(t+1) = \underset{\mathbf{q}\in\mathscr{P}}{\arg\min} \left\{ \sum_{i=1}^{t} L_i(\mathbf{q}) + f(\mathbf{q}) \right\}. \tag{3.8}$$

Le choix de la fonction de régularisation $f(\mathbf{p})$ dépend des applications cependant cette fonction doit être K-fortement convexe, où K est une constante positive, comme défini dans [Shalev-Shwartz, 2011]. La forte convexité d'une fonction $f(\mathbf{p})$ est une contrainte supplémentaire sur les variations d'une fonction convexe. Une fonction K fortement convexe par rapport à la norme $\|\cdot\|_2^2$ si elle respecte l'équation suivante :

$$f(\mathbf{q}) \geq f(\mathbf{p}) + \langle \nabla f(\mathbf{p}) | \mathbf{q} - \mathbf{p} \rangle + \frac{K}{2} ||\mathbf{q} - \mathbf{p}||_2^2, \quad \forall \mathbf{q}, \mathbf{p} \in \mathscr{P}. \tag{3.9}$$

Nous verrons plus tard quelques exemples de fonctions de régularisation ainsi que leurs impactes sur les différents algorithmes.

Il faut noter que l'objet ne peut pas utiliser l'algorithme du FoRL pour déterminer son allocation de puissance car il ne connaît pas les fonctions objectif des instants précédents. Nous savons, par hypothèse, que l'objet connaît uniquement le feedback qu'il reçoit du récepteur : c'est à dire les valeurs des gradients aux instants précédents. Pour contourner ce problème, nous allons utiliser la convexité des fonctions objectif, et plus particulièrement la condition de convexité d'ordre I [Boyd and Vandenberghe, 2004] définie par :

$$L_t(\mathbf{q}) \geq L_t(\mathbf{p}) + \langle \mathbf{v}(t) | \mathbf{q} - \mathbf{p} \rangle, \quad \forall t \quad \forall \mathbf{p}, \mathbf{q} \in \mathscr{P}, \tag{3.10}$$

pour déterminer une borne des fonctions objectif. Cette propriété, vraie pour tout instant t et pour toutes les fonctions objectif $L_t(\mathbf{p})$, nous permet de majorer le regret. En utilisant la définition du regret (2.7) et la propriété de convexité d'ordre I des fonctions objectif (3.10) nous obtenons la borne suivante du regret :

$$\begin{aligned} \mathrm{Reg}(T) &= \sum_{t=1}^{T} L_t(\mathbf{p}(t)) - \min_{\mathbf{q} \in \mathscr{P}} \sum_{t=1}^{T} L_t(\mathbf{q}) && (3.11) \\ &\leq \sum_{t=1}^{T} \langle \mathbf{v}(t) | \mathbf{p} - \mathbf{q}^* \rangle, && (3.12) \end{aligned}$$

où $\mathbf{q}^*$ est l'allocation de puissance fixe qui minimise la somme des fonctions objectif $L_t(\mathbf{p})$ sur tout l'horizon T, mathématiquement cette allocation de puissance $\mathbf{q}^*$ est définie par :

$$\mathbf{q}^* = \arg\min_{\mathbf{q} \in \mathscr{P}} \sum_{t=1}^{T} L_t(\mathbf{q}). \tag{3.13}$$

Le fait de borner le regret nous donne un problème équivalent à résoudre. En effet, si l'objet trouve une allocation de puissance telle que :

$$\limsup_{T \to \infty} \frac{\sum_{t=1}^{T} \langle \mathbf{v}(t) | \mathbf{p} - \mathbf{q}^* \rangle}{T} \leq 0, \tag{3.14}$$

alors la propriété de non regret est garantie. L'avantage, est que ce nouveau problème dépend des gradients et donc l'objet connaît les informations nécessaires pour le résoudre. Finalement, l'objet ne doit pas chercher l'allocation de puissance qui minimise la somme des fonctions objectif mais l'allocation de puissance qui minimise la somme des produits scalaires entre les puissances et les gradients. Pour y arriver, nous allons utiliser l'algorithme du FoRL qui permet de minimiser la somme des gradients aux instants précédents et donc de borner le regret. Il faut donc remplacer les fonctions objectif par la somme des produits scalaires du gradient dans l'équation du FoRL [Shalev-Shwartz, 2011] et nous obtenons l'allocation de puissance suivante :

$$\mathbf{p}(t+1) = \arg\min_{\mathbf{q} \in \mathscr{P}} \left\{ \sum_{i=1}^{t} \langle \mathbf{v}(i) | \mathbf{q} \rangle + f(\mathbf{q}) \right\}, \tag{3.15}$$

Maintenant, que nous avons déterminé le problème d'optimisation que l'objet doit résoudre pour déterminer son allocation de puissance, il reste une difficulté. En effet, pour définir cette allocation de puissance l'objet doit résoudre un nouveau problème d'optimisation à chaque itération ce qui peut prendre du temps et peut être complexe (en termes de ressources de calcul). Un calcul rapide nous permet de réarranger le problème d'optimisation (5.6) sous la forme suivante :

$$\mathbf{p}(t+1) = \arg\min_{\mathbf{q}\in\mathscr{P}} \left\{ \sum_{i=1}^{t} \langle \mathbf{v}(i)|\mathbf{q}\rangle + f(\mathbf{q}) \right\} \tag{3.16}$$

$$= \arg\max_{\mathbf{q}\in\mathscr{P}} \left\{ \langle -\sum_{i=1}^{t} \mathbf{v}(i)|\mathbf{q}\rangle - f(\mathbf{q}) \right\}. \tag{3.17}$$

Nous allons maintenant définir $\mathbf{y}(t)$ un score interne qui contient la somme des gradients $\mathbf{y}(t) = -\sum_{i=1}^{t-1} \mathbf{v}(i)$ du passé. Ce score interne peut être mis à jour comme suit :

$$\mathbf{y}(0) = 0, \tag{3.18}$$

$$\mathbf{y}(t+1) = \mathbf{y}(t) - \mathbf{v}(t). \tag{3.19}$$

On définit maintenant une fonction de projection $\mathbf{Q}(\mathbf{y}) : \mathbb{R}^S \to \mathscr{P}$ telle que :

$$\mathbf{Q}(\mathbf{y}) = \arg\max_{\mathbf{q}\in\mathscr{P}} \{\langle \mathbf{y}|\mathbf{q}\rangle - f(\mathbf{q})\}. \tag{3.20}$$

Cependant nous devons noter que le fait de réécrire l'étape de projection (3.20) sous la forme d'une fonction de projection ne change pas l'étape d'optimisation que l'objet doit effectuer pour déterminer l'allocation de puissance $\mathbf{p}(t+1)$. C'est pourquoi, un des objectifs pour concevoir des algorithmes efficaces d'allocation de puissance dynamique est de déterminer l'étape de projection $\mathbf{Q}(\mathbf{y})$ en fonction du problème d'optimisation (3.17), et plus particulièrement de la fonction de régularisation $f(\mathbf{p})$.

Maintenant que nous avons défini toutes les étapes intermédiaires, nous allons utiliser les équations (3.18) et (3.20) avec la définition de notre problème (3.16), ce qui nous donne l'algorithme suivant afin de définir la nouvelle allocation de puissance :

$$\mathbf{y}(t+1) = \mathbf{y}(t) - \mathbf{v}(t) \tag{3.21}$$

$$\mathbf{p}(t+1) = \mathbf{Q}(\mathbf{y}). \tag{3.22}$$

Grâce à ces calculs, nous nous retrouvons avec un algorithme en deux étapes : 1) mise à jour du score interne ; 2) projection dans l'espace faisable à l'aide de la fonction $\mathbf{Q}(\mathbf{y})$. La forme analytique de la fonction de projection $\mathbf{Q}(\mathbf{y})$ va dépendre en particulier du choix de la fonction de régularisation $f(\mathbf{p})$. Nous allons considérer deux exemples de fonction de régularisation $f(\mathbf{p})$ les plus utilisées.

Par exemple, si la fonction de régularisation $f(\mathbf{p})$ est la norme Euclidienne telle que :

$$f(\mathbf{p}) = \frac{1}{2v}||\mathbf{p}||_2^2, \tag{3.23}$$

où ν est une constante positive, alors l'algorithme obtenu est équivalent à l'algorithme de descente du gradient en ligne (*Online Gradient Descent* (OGD)) [Shalev-Shwartz, 2011]. Pour rappel, l'algorithme OGD est défini par les équations suivantes :

$$\mathbf{p}(t+1) = \underset{\mathbf{q}\in\mathscr{P}}{\arg\min}\left\{\nu\langle\mathbf{y}(t)|\mathbf{q}\rangle + \frac{1}{2}||\mathbf{p}-\mathbf{q}||_2^2\right\} \tag{3.24}$$

$$= \underset{\mathbf{q}\in\mathscr{P}}{\arg\min}\left\{\frac{1}{2}\|\mathbf{p}(t) - \nu\mathbf{v}(t) + \mathbf{q}\|\right\}. \tag{3.25}$$

Cet algorithme nécessite cependant la résolution d'un problème d'optimisation convexe à chaque itération, pour projeter la solution dans l'espace faisable.

Pour contourner ce problème nous pouvons utiliser une autre fonction de régularisation $f(\mathbf{p})$. Nous remarquons que l'espace faisable défini par

$$\mathscr{P} = \left\{\mathbf{p}\in\mathbb{R}^S | p^s \geq 0 \ \forall s, \sum_{s=1}^{S} p^s \leq P_{\max}\right\}, \tag{3.26}$$

est proche du simplexe :

$$\Delta = \left\{\mathbf{p}\in\mathbb{R}^S | p^s \geq 0 \ \forall s, \sum_{s=1}^{S} p^s = 1\right\}. \tag{3.27}$$

Cette similarité des espaces faisable nous permet de faire le lien entre notre problème d'optimisation e ligne et le problème du bandit manchot (*multi-armed bandit*). Dans le problème du bandit manchot à S bras il faut déterminer la distribution de probabilité qui permet choisir le meilleur bras (qui maximise le gain espéré). Une des solutions pour résoudre ce problème est d'utiliser l'algorithme FoRL avec une fonction de régularisation entropique, qui est mieux adaptée au simplexe et définie comme

$$f(\mathbf{q}) = \sum_{s=1}^{S} q_s \log(q_s). \tag{3.28}$$

L'idée est donc d'adapter cette fonction de régularisation entropique à notre esapce faisable, ce qui nous permet d'obtenir la fonction de régularisation suivante :

$$f(\mathbf{q}) = \sum_{s=1}^{S} q_s \log(q_s) + \left(P_{\max} - \sum_{s=1}^{S} q_s\right)\log\left(P_{\max} - \sum_{s=1}^{S} q_s\right). \tag{3.29}$$

Le dernier terme correspond à la puissance non utilisée ce qui nous permet de transformer notre ensemble faisable dans un simplexe de taille $S+1$. Grâce à cette fonction de régularisation, nous pouvons montrer que l'étape de projection dans l'équation (3.24) devient

$$p^s(t) = Q^s(\mathbf{y}(t)), \ s\in\{1,\ldots,S\}, \tag{3.30}$$

$$= P_{\max}\frac{\exp(y^s(t))}{1+\sum_{i=1}^{S}\exp(y^i(t))}, \tag{3.31}$$

Algorithme OXL : *Online Exponentiel Learning*

Initialisation : $\mathbf{y}(0) \leftarrow 0$; $t \leftarrow 0$.

Répéter

- **Phase de pré-transmission :** mise à jour de la puissance

$p^s(t) \leftarrow P_{\max} \frac{\exp(y^s(t))}{1+\sum_{i=1}^{S} \exp(y^i(t))}, \ \forall s \in S$

- **Transmission avec** $\mathbf{p}(t)$
- **Phase de post-transmission :** réception du feedback $\mathbf{v}(t)$

Mise à jour du score $\mathbf{y}(t+1) \leftarrow \mathbf{y}(t) - \mu(t)\ \mathbf{v}(t)$

$t \leftarrow t+1$

jusqu'à : fin de transmission

où $\mathbf{Q}(\mathbf{y}(t)) = \left(Q^1(\mathbf{y}(t)), \ldots, Q^S(\mathbf{y}(t))\right)$. Cette fonction de projection a plusieurs avantages : premièrement, elle garantit que l'allocation de puissance obtenue est dans l'espace faisable, deuxièmement, l'objet n'a pas besoin de résoudre un problème d'optimisation pour déterminer cette allocation de puissance. La projection exponentielle a l'avantage d'être rapide et peu coûteuse en terme de calcul.

Pour illustrer le fonctionnement de notre algorithme, nous allons considérer ce qu'il se passe lorsqu'un objet arrive sur le réseau pour la première fois. Ce dernier n'a aucune information sur le réseau, que ce soit en terme du nombre d'objets, des gains de canaux ou des interférences. Il va donc transmettre avec une politique $\mathbf{p}(1) = \frac{P_{\max}}{S}(1,1,\ldots,1)$ qui est une allocation de puissance uniforme dans chacune des sous-porteuses. Le récepteur va lui transmettre $\nabla L_1(\mathbf{p}(1))$ le feedback qui correspond à l'allocation de puissance utilisé par l'objet. Ce dernier utilise un score interne $\mathbf{y}(.)$ pour suivre l'évolution du gradient. Á la première itération, il n'y a eu qu'un seul retour d'information et donc $\mathbf{y}(1) = \nabla L_1(\mathbf{p}(1))$. L'objet utilise ce score interne pour déterminer l'allocation de puissance $\mathbf{p}(2)$ et transmettre avec cette politique. Une fois la transmission finie, le récepteur envoie $\nabla L_1(\mathbf{p}(2))$ à l'objet. Il ajoutera cette information dans son score interne, et obtient $\mathbf{y}(2) = \nabla L_1(\mathbf{p}(1)) + \nabla L_2(\mathbf{p}(2))$. L'objet va répéter cette opération jusqu'à la fin de la transmission.

Le score interne $\mathbf{y}(t)$ est la somme des gradients des itérations 0 à $t-1$, c'est à dire : $\mathbf{y}(t) = -\sum_{i=1}^{t-1} \mathbf{v}(i)$. Ce score représente donc le cumul d'informations que l'objet a sur l'environnement à l'instant t.

En conclusion, l'Algorithme OXL repose sur trois étapes principales : 1) la mise à jour de l'allocation de puissance en exploitant le score interne et la transmission effective ; 2) l'étape de réception du feedback ; 3) la mise à jour du score interne. Il s'avère que cet algorithme a des garanties théoriques par rapport à ses performances en terme du regret. Nous allons voir dans la section suivante les résultats théoriques en fonction des cas étudiés ainsi qu'un bref résumé des preuves correspondantes.

3.1.3 Résultats théoriques

Dans l'annexe A.1 nous présentons la preuve du théorème suivant concernant la borne du regret de l'Algorithme OXL.

Théorème 1. *Si l'Algorithme OXL est utilisé avec un pas constant μ et un feedback parfait du gradient alors le regret est borné par :*

$$\mathrm{Reg}(T) \leq \frac{P_{\max}\log(1+S)}{\mu} + \frac{\mu P_{\max} T V^2}{2}, \tag{3.32}$$

où $\|\boldsymbol{v}(t)\|_\infty^2 \leq V^2$.

Nous pouvons voir que cette borne dépend des paramètres du système comme la puissance maximale $P_{\max}$, le nombre de sous-porteuses ou encore μ le pas de l'algorithme. Cette borne dépend aussi de V qui est une borne de l'information reçue du gradient $\mathbf{v}(t)$, i.e. $\|\mathbf{v}(t)\|_\infty^2 \leq V^2$.

La preuve de ce théorème repose sur 4 étapes. Tout d'abord comme expliqué précédemment :

1. nous allons utiliser la convexité de la fonction objectif pour borner le regret par :

$$\mathrm{Reg}(T) \leq \sum_{t=1}^{T} \langle \mathbf{v}(t) | \mathbf{p} - \mathbf{q}^* \rangle, \tag{3.33}$$

   où $\mathbf{q}^*$ est défini comme :

$$\mathbf{q}^* = \arg\max_{\mathbf{q}\in\mathscr{P}} \sum_{t=1}^{T} L_t(\mathbf{q}). \tag{3.34}$$

2. Une fois cette étape réalisée nous allons utiliser la fonction f^* qui est la fonction convexe conjuguée de la fonction de régularisation f définie dans l'équation (3.29) et est définie par :

$$f^*(\mathbf{p}) = \sup_{\mathbf{y}\in\mathbb{R}^S} \{\langle \mathbf{y} | \mathbf{p} \rangle - f(\mathbf{p})\}. \tag{3.35}$$

3. L'utilisation de la fonction f^* ainsi que son approximation de Taylor du deuxième ordre nous donne :

$$\mathrm{Reg}(T) \leq \frac{1}{\mu}\left[f^*(0) - f^*(\mathbf{y}(T+1))\right] + \frac{\mu}{2} P_{\max} \sum_{t=1}^{T} \|\mathbf{v}(t)\|_\infty^2 + \frac{1}{\mu}\langle \mathbf{y}(T+1) | \mathbf{q} \rangle. \tag{3.36}$$

4. Il faut ensuite utiliser l'inégalité de Fenchel qui relie les fonctions f et f^* ainsi que l'inégalité de Jensen pour montrer que $f(\mathbf{p}) \geq 0$ pour tous les $\mathbf{p}$ de l'espace faisable. Ce qui nous donne finalement :

$$\mathrm{Reg}(T) \leq \frac{P_{\max}\log(1+S)}{\mu} + \frac{\mu P_{\max} T V^2}{2}. \tag{3.37}$$

Nous pouvons maintenant remarquer l'importance du choix du pas μ dans la borne ci-dessus. En effet, nous avons vu dans le Chapitre 2 que notre objectif est d'avoir un algorithme qui a la

propriété de non regret. Pour vérifier cela, nous devons calculer la limite du regret moyen définie dans l'équation (3.37). Un calcul rapide nous montre que :

$$\limsup_{T\to\infty} \frac{1}{T}\mathrm{Reg}(T) \leq \frac{\mu P_{\max} V^2}{2}. \tag{3.38}$$

Ainsi, nous remarquons que pour un pas μ quelconque la propriété de non regret n'est pas forcement garantie car le regret peut avoir une croissance linéaire en T. Puisque nous ne pouvons pas utiliser n'importe quelle valeur de μ, nous allons voir, dans la suite de cette section, comment déterminer ce pas en fonction des différents cas : horizon de transmission T connu et horizon de transmission T inconnu.

De plus, il faut noter que le pas μ permet de contrôler le compromis entre l'exploitation et l'exploration de l'information par notre algorithme. Intuitivement, plus μ est faible, moins l'objet utilise les informations reçues par feedback, il est en mode exploitation. A l'inverse, lorsque μ est grand, l'objet accorde une grande importance à l'information du feedback. Il va faire évoluer son allocation de puissance beaucoup plus rapidement et exploiter l'information. Le choix de la valeur du pas μ est de ce fait très importante pour le bon fonctionnement de l'algorithme et pour obtenir ou non la propriété de non regret.

Afin de déterminer la valeur optimale du pas μ, nous devons résoudre un problème d'optimisation convexe simple qui permet de minimiser la borne du regret de l'équation (3.32). Ainsi, nous trouvons un pas optimal μ^* :

$$\mu^* = \sqrt{\frac{2\log(1+S)}{TV^2}}. \tag{3.39}$$

Comme pour la borne du regret, le pas optimal va dépendre des paramètres du système, mais plus important encore, de la durée de transmission T. Dans la prochaine partie, nous allons voir le comportement de l'algorithme et de la borne du regret en fonction de la connaissance de la durée de transmission.

Pour étudier la propriété de non regret, nous devons étudier l'évolution du regret asymptotiquement lorsque la durée de transmission tend vers l'infini. Dans un premier temps nous allons nous concentrer au cas où l'objet connaît la durée de transmission à l'avance.

Durée de transmission T connue

Quand l'objet connaît la durée de transmission en avance, il peut calculer le pas optimal μ^*. Il est alors possible de remplacer la valeur de μ dans l'équation (3.32) par la valeur calculée en (3.39). Grâce à cela nous obtenons la borne suivante pour le regret :

Corollaire 1. *Si l'Algorithme OXL est utilisé pour une transmission de durée T, avec un gradient parfait comme feedback et le pas optimal défini dans l'équation* (3.39)*, alors la propriété de non regret est garantie et le regret est borné par :*

$$\mathrm{Reg}(T) \leq P_{\max}\sqrt{2TV^2\log(1+S)}. \tag{3.40}$$

Nous remarquons que $\limsup_{T\to\infty} \frac{1}{T}\text{Reg}(T) = 0$, donc que notre algorithme a bien la propriété de non regret dans le cas où la durée de transmission T est connue.

Durée de transmission T inconnue

Dans le cas où la durée de transmission n'est pas connue, l'objet ne peut pas déterminer le pas optimal μ^*. Cependant, il est possible d'utiliser l'astuce du doublement de la fenêtre de transmission ou le doubling-trick décrit dans la Section 2.3. Ainsi l'objet peut calculer le pas optimal μ^* dans chaque fenêtre et nous pouvons donc borner le regret total.

Corollaire 2. *Si l'algorithme OXL est utilisé pour une transmission de durée inconnue, avec le gradient parfait comme feedback et en utilisant le doubling-trik, alors la propriété de non regret est garantie et le regret est borné par :*

$$\text{Reg}(T) \leq \frac{2}{\sqrt{2}-1} P_{\max} \sqrt{2TV^2 \log(1+S)}. \tag{3.41}$$

Ainsi la borne du regret dans le cas où l'objet ne connaît pas la durée de transmission est similaire (à une constante multiplicative prés) au cas où l'objet connaît en avance la durée de transmission.

3.1.4 Résultats numériques

Après avoir présenté les résultats théoriques qui garantissent la propriété de non regret de l'Algorithme OXL, nous allons dans cette section présenter différents résultats numériques. Pour réaliser ces simulations, nous utilisons MatLab ainsi que le modèle de canal COST-HATA [Pedersen, 1999]. Ce modèle de canal permet de simuler de manière réaliste les différents gains de canaux des objets pour des environnements de types : urbains et suburbains. Ce modèle a aussi l'avantage de prendre en compte la mobilité des objets dans ces environnements. Les détails complets des paramètres utilisés pour ces simulations sont détaillés dans l'Annexe B.

Nous souhaitons illustrer les performances de notre algorithme, c'est pourquoi dans un premier temps, nous allons présenter une comparaison de notre algorithme avec un algorithme classique de la littérature : le *Water-Filling* (WF).

Comparaison avec l'algorithme de *Water-Filling*

L'algorithme classique du WF [Scutari et al., 2009] permet de déterminer l'allocation de puissance optimale lorsque les gains de canaux restent statiques pendant la période de transmission. Afin de réduire le temps de calcul et de simplifier la complexité des algorithmes utilisés par les objets, et ainsi réduire les temps de calcul, nous allons utiliser l'Algorithme du WF pour résoudre le problème suivant :

$$\begin{array}{lll} \text{minimiser} & \sum_{s=1}^{S} p^s(t) & \\ \text{par rapport à} & \mathbf{p}(t) = (p^1(t), \ldots, p^S(t)) & \\ \text{sous contraintes} & p^s(t) \geq 0, \quad \forall s & \text{(C21)} \\ & \sum_{s=1}^{S} p^s(t) \leq P_{\max} & \text{(C22)} \\ & R_{t-1}(\mathbf{p}) \geq R_{\min} & \text{(C23)} \end{array} \tag{3.42}$$

Cela signifie qu'il n'est pas possible de contrôler le compromis débit puissance en utilisant cet algorithme. De plus, il est possible que l'espace faisable défini par les contraintes (C21), (C22) et (C23) ne garantit pas l'existence d'une solution. Par exemple, si $R_{\min}$ est trop grand comparé à $P_{\max}$ l'espace faisable du problème (3.34) est vide. Afin de contourner cet obstacle, nous proposons deux algorithmes alternatifs : 1) *Water-Filling* efficace en terme de consommation de puissance (WF_0), 2) *Water-Filling* efficace en terme de débit ($\text{WFP}_{\max}$).

Dans le cas du WF_0, si l'objet ne peut pas déterminer une politique d'allocation de puissance qui garantit que $R(\mathbf{p}) \geq R_{\min}$, alors il ne transmet pas (i.e., $p^s = 0, \quad \forall s$). Cette solution favorise l'économie de la consommation de puissance de l'objet en acceptant de ne pas toujours pouvoir transmettre d'information.

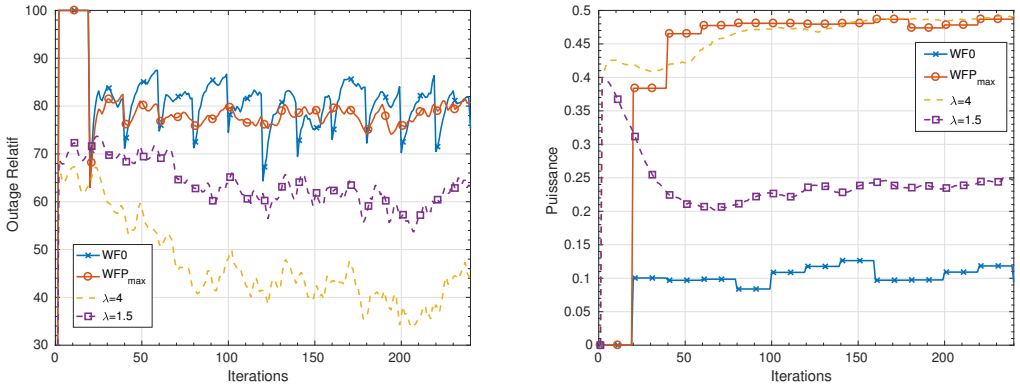
Dans le cas où l'objet favorise le débit ($\text{WFP}_{\max}$), l'objet va allouer uniformément la puissance dans chacune des sous-porteuse : $p^s(t) = \frac{P_{\max}}{S}$ dans le cas où il ne peut pas transmettre à $R_{\min}$. De cette manière, l'objet transmet quand même de l'information au détriment de la consommation de puissance.

Nous avons vu que notre objectif principal est de réduire au maximum la consommation de puissance tout en garantissant un débit minimal $R_{\min}$. Afin de comparer les performances des algorithmes en terme de débit nous allons utiliser la notion d'*outage* relatif. Cet *outage* relatif est défini par l'équation suivante :

$$Out = \left[1 - \frac{R_t(\mathbf{p}(t))}{R_{\min}}\right]^+, \tag{3.43}$$

où $[\cdot]^+ = \max(\cdot, 0)$. Cette métrique nous permet de quantifier, en pourcentage, la perte de débit en comparaison de $R_{\min}$. Ainsi, si l'*outage* relatif est égal à 0%, alors cela signifie que l'objet transmet à un débit supérieur à $R_{\min}$. A l'inverse, si l'outage est égal à 100% alors le débit de l'objet est inférieur à $R_{\min}$. Quant aux cas intermédiaires, si l'*outage* relatif est égal à 20% alors nous savons que le débit de l'objet est 20% plus faible que $R_{\min}$. Cet *outage* relatif, nous permet donc de connaître la qualité de la transmission en fonction de $R_{\min}$.

Sur la figure 3.2a est tracée l'*outage* relatif pour l'Algorithme de *water-filling* (avec les deux options définies ci-dessus) et l'*outage* relatif pour l'Algorithme OXL avec différentes valeurs de λ. Pour rappel, λ est le coefficient de pénalité appliqué dans le cas où l'objet ne respecte pas la contrainte de débit $R_{\min}$. La première chose que nous pouvons remarquer est que l'Algorithme OXL donne des meilleurs résultats que l'Algorithme de *water-filling*.



(a) *Outage* relatif

(b) Consommation de puissance

FIGURE 3.2: Comparaison des performances en terme d'*outage* relatif et consommation de puissance pour les Algorithmes OXL et WF_0. L'Algorithme WF_0 est rigide en terme de compromis débit puissance alors qu'avec l'Algorithme OXL il est possible de le contrôler en utilisant la constante de pénalité λ.

Sur la figure 3.2b est tracée la consommation de puissance totale, $\sum_s p^s(t)$ en fonction des itérations. Sur cette figure, la différence entre la consommation de puissance des Algorithmes WF_0 et WFP_{max} est nettement visible.

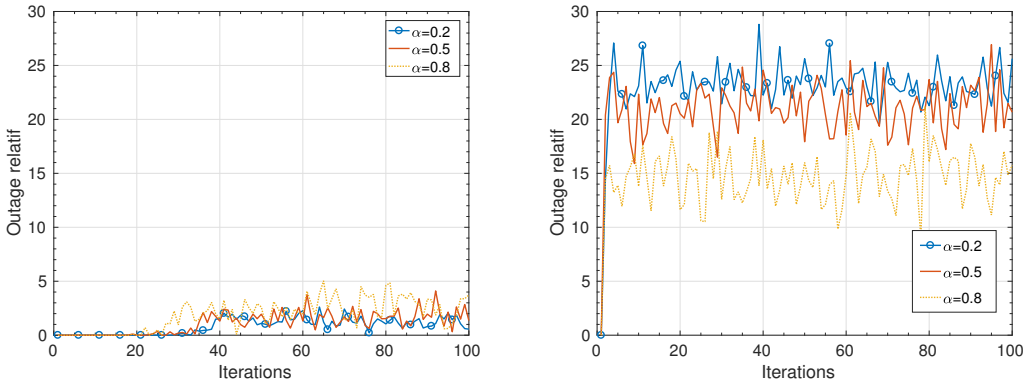
Le paramètre λ permet de contrôler le compromis entre le débit et la puissance de l'Algorithme OXL. En effet, lorsque λ est grand (i.e. $\lambda = 4$), l'*outage* relatif est plus faible et la consommation de puissance est plus grande. A l'inverse, lorsque λ est plus faible, l'*outage* relatif est plus grand mais la consommation de puissance est aussi plus faible.

Maintenant que nous avons comparé les deux algorithmes dans le cas d'un système réaliste (COST-HATA), nous allons comparer l'*outage* relatif pour différentes valeurs de corrélation des gains de canaux. Pour cela, nous utiliserons le modèle de gains de canaux suivant :

$$h^s(t+1) = \alpha h^s(t) + (1-\alpha)\epsilon^s, \tag{3.44}$$

avec $\epsilon^s \sim \mathcal{N}(0,\sigma^2)$ et $\alpha \in [0,1]$. Pour rappel, les gains de canaux sont définis comme $g^s(t) = |h^s(t)|^2$. Si $\alpha = 1$, nous nous retrouvons dans le cas statique où $h(t) = h(0)$ quelque soit t. A l'inverse, lorsque $\alpha = 0$, l'équation du canal devient $h^s(t+1) = \epsilon^s$ et donc l'évolution du canal est i.i.d., i.e., canal de Rayleigh. Ce modèle de canal nous permet de contrôler la corrélation temporelle des gains de canaux entre les instants t et $t-1$. La perte d'*outage* relatif pour l'Algorithme WF_0 est tracé sur la figure 3.3b.

L'objectif étant de limiter la consommation de puissance, nous décidons d'utiliser uniquement l'algorithme du WF_0. Ce dernier est très impacté par l'augmentation de la dynamique du système. Plus la corrélation temporelle diminue (i.e. α diminue), plus l'algorithme a des difficultés pour déterminer une allocation de puissance efficace.



(a) *Outage* relatif OXL (b) *Outage* relatif WF

FIGURE 3.3: Comparaison de la perte d'*outage* relatif pour les Algorithmes WF_0 et OXL. L'Algorithme OXL donne de bien meilleurs résultats en terme de débit mais aussi en terme de robustesse aux variations de gains de canaux.

Sur la figure 3.3a, nous avons tracé la perte d'*outage* relatif pour l'Algorithme OXL. L'algorithme donne des résultats bien meilleurs. Il est important de noter que la diminution de la corrélation temporelle n'a pas beaucoup d'importance sur les performances de l'Algorithme OXL.

Nous pouvons déduire de la comparaison entre notre Algorithme OXL et l'Algorithme du WF_0 : 1) que l'Algorithme OXL donne de meilleures performances en terme d'*outage* relatif ; 2) l'Algorithme OXL permet de contrôler finement le compromis entre la consommation de puissance et le débit en utilisant le coefficient de pénalité λ ; 3) notre Algorithme OXL s'adapte de manière beaucoup plus efficace aux variations du système comme nous le montre les figures 3.3a et 3.3b ; 4) l'Algorithme OXL est plus robuste à la perte d'information sur le réseau à l'instant de transmission comparé à l'Algorithme de WF_0. Maintenant que nous avons comparé les performances de notre Algorithme OXL et l'Algorithme du WF_0, nous allons étudier l'impact de certains paramètres, nombre d'objets M et nombre de sous-porteuses S, sur les performances de notre algorithme.

Sur la figure 3.5, nous avons tracé l'évolution du regret moyenné sur 100 réalisations, d'un objet focal déterminé de manière aléatoire, en fonction des itérations pour un nombre d'objets variable, $M \in \{10, 20, 40\}$ partageant 2 sous-porteuses ($S = 2$). Ces courbes de regret ont étées obtenues en utilisant une méthode de Monte Carlo sur 100 réalisations. Pour chaque réalisation nous avons positionné les objets de manière aléatoire dans le réseau, mais nous avons gardé les paramètres de chaque objet fixes : $P_{\max}, \lambda, \mu$. Dans notre cas les paramètres de l'objet sont les suivants : $P_{\max} = 1.25\ W$, $\lambda = 2.25$ et $\mu = 0.01$. Sur cette figure, nous pouvons remarquer que, comme l'avait prédit nos résultats théoriques, le nombre d'objets n'a pas une grande importance sur l'évolution du regret. Nous pouvons en déduire que l'Algorithme OXL est relativement peu sensible à l'augmentation du nombre d'objets.

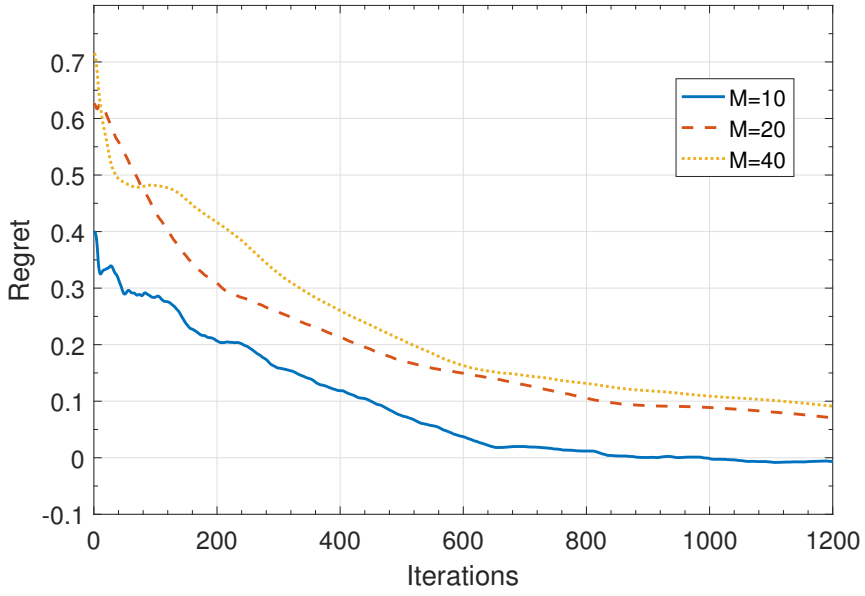


FIGURE 3.4: Évolution du regret moyen en fonction des itérations pour un nombre variable d'objets $M \in \{10, 20, 40\}$. Le regret moyen est calculé sur 100 réalisations. Pour chaque réalisation, nous avons gardé les mêmes paramètres des objets, $P_{\max}, \lambda, \mu$, mais nous avons réparti les objets de manière aléatoire. Cette figure nous permet de voir que l'Algorithme OXL est relativement peu sensible aux variations du nombre d'objets.

Nous avons tracé l'évolution du regret moyen, sur 100 réalisations, en fonction du nombre d'itérations avec nombre de sous-porteuses variable, $S \in \{2, 4, 8\}$ dans la figure 3.5. Ces courbes de regret ont étés obtenues en utilisant une méthode de Monte Carlo sur 100 réalisations. Pour chaque réalisation nous avons augmenté le nombre de sous-porteuses mais nous avons gardé les paramètres de chaque objet : $P_{\max}$, λ et μ fixes ainsi que leurs positions. Dans notre cas les paramètres de l'objet sont les suivants : $P_{\max} = 2\ W$, $\lambda = 1.25$ et $\mu = 0.01$. Comme nous l'indique la borne théorique du regret, ce dernier dépend fortement du nombre de sous-porteuses. Bien que le regret décroîsse dans chacun des cas, nous pouvons noter que plus le nombre de sous-porteuses est grand plus le regret décroît lentement vers 0. En conclusion, l'Algorithme OXL est plus sensible au nombre de sous-porteuses S qu'au nombre d'objets M.

La prochaine étape de notre étude consiste à analyser le comportement de notre Algorithme OXL lorsque l'objet dispose de moins d'informations. La première étape dans la réduction de l'information reçue par l'objet est de relacher l'hypothèse du feedback parfait.

3.2 Information imparfaite sur le gradient

Dans la section précédente, nous avons étudié le cas où l'objet avait accès au gradient de la fonction objectif à l'itération précédente. Le but principal de cette thèse est de chercher à

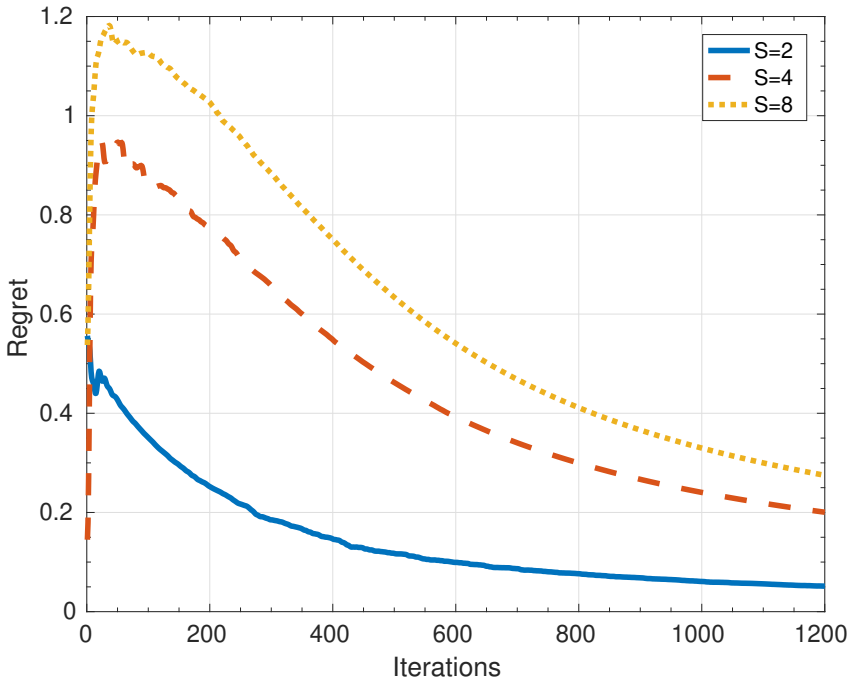


FIGURE 3.5: Évolution du regret moyen sur 100 réalisations en fonction des itérations avec un nombre variable de sous-porteuses, $S \in \{2,4,8\}$. Pour chaque réalisation, nous avons augmenté le nombre de sous-porteuses du système mais nous avons gardé les paramètres de chaque objet ; $P_{\max}, \lambda$ et μ fixes. Nous pouvons remarquer que le regret dépend fortement du nombre de sous-porteuses : plus le nombre de sous-porteuses et grand plus le regret mettra de temps pour décroître vers 0.

réduire l'information reçue par l'objet. Pour cela, une première étape consiste à étudier le cas du feedback bruité.

3.2.1 Feedback imparfait et son impact sur l'Algorithme OXL

La première étape est de définir mathématiquement le bruit qui va perturber notre feedback. Pour faire cela, nous allons devoir définir quelques hypothèses sur l'estimation que va recevoir l'objet. La première est l'absence d'erreur systématique qui implique que l'estimateur du gradient ne doit pas être biaisé, mathématiquement cela ce traduit par :

$$\mathbb{E}[\tilde{\mathbf{v}}(t)] = \nabla L_t(\mathbf{p}(t)) \tag{3.45}$$

La seconde hypothèse que nous faisons sur l'estimateur est que sa variance est bornée cela s'exprime mathématiquement par :

$$\mathbb{E}[\|\tilde{\mathbf{v}}(t)\|_\infty^2] \leq \tilde{V}^2 \ \forall s, \tag{3.46}$$

où l'espérance est calculée par rapport à l'aléatoire de l'estimateur. Ces conditions ne sont pas très restrictives car elles impliquent l'absence d'erreur systématique (estimateur non-biaisé) et le fait que la variance de cet estimateur soit bornée. De plus, ces conditions sont respectées dans la majorité des modèles d'erreur utilisés dans la littérature. Par exemple, le modèle commun d'erreur défini comme :

$$\tilde{\mathbf{v}}(t) = \nabla L_t(\mathbf{p}(t)) + \mathbf{z}, \tag{3.47}$$

où $\mathbf{z} \in \mathcal{N}(0, \sigma^2 \mathbf{I})$ entre dans la famille des modèles qui respectent les conditions citées ci-dessus.

Maintenant que nous avons défini le modèle de l'estimation $\tilde{\mathbf{v}}(t)$, nous devons étudier l'impact de ce dernier sur l'Algorithme OXL.

Impact de la réduction d'information reçue

Premièrement, dû à la présence d'aléatoire dans l'information reçue par l'objet, l'allocation de puissance $\mathbf{p}(t)$ va dépendre de l'aléa de l'estimateur $\tilde{\mathbf{v}}(t)$. C'est pourquoi nous ne pouvons plus étudier le regret mais le regret moyen comme défini dans le Chapitre 2 dans l'équation (2.10). Nous devons trouver un algorithme qui a la propriété de non regret en moyenne définie par :

$$\limsup_{T\to\infty} \frac{1}{T} \mathrm{EReg}(T) \leq 0. \tag{3.48}$$

3.2.2 Résultats théoriques

Dans l'annexe A.2 nous montrons qu'en utilisant l'Algorithme OXL nous obtenons le résultat suivant.

Théorème 2. *Si l'Algorithme OXL est utilisé avec un pas constant μ et un feedback imparfait du gradient $\tilde{\boldsymbol{v}}(t)$ alors l'espérance du regret est bornée par :*

$$\mathrm{EReg}(T) \leq \frac{P_{\max} \log(1+S)}{\mu} + \frac{\mu P_{\max} T \tilde{V}^2}{2}. \tag{3.49}$$

Ainsi nous remarquons que la borne est similaire à la borne dans le cas du gradient parfait aux différences suivantes : 1) la borne de la variance de l'estimateur, $\tilde{V}$, remplace la borne du gradient V ; et 2) il s'agit de l'espérance du regret et non pas du regret instantané.

La majorité de la preuve de ce théorème est similaire à la preuve du Théorème 1. En effet, les fonctions objectif étant convexes nous pouvons utiliser cette propriété et obtenir :

$$\mathrm{EReg}(T) \leq \mathbb{E}\left[\sum_{t=1}^{T} \langle \nabla L_t(\mathbf{p}(t)) | \mathbf{p}(t) - \mathbf{q}^* \rangle\right]. \tag{3.50}$$

L'idée ensuite est de lier l'équation (3.50) à l'estimateur du gradient $\mathbf{v}(t)$. Par définition, nous avons $\nabla L_t(\mathbf{p}(t)) = \mathbb{E}[\tilde{\mathbf{v}} | \tilde{\mathbf{v}}(t-1), \ldots, \tilde{\mathbf{v}}(1)]$. Ainsi en utilisant la loi de l'espérance totale nous obtenons :

$$\mathbb{E}\left[\sum_{t=1}^{T} \langle \nabla L_t(\mathbf{p}(t)) | \mathbf{p}(t) - \mathbf{q} \rangle\right] = \mathbb{E}\left[\sum_{t=1}^{T} \langle \tilde{\mathbf{v}} | \mathbf{p}(t) - \mathbf{q} \rangle\right]. \tag{3.51}$$

Le résultat du théorème est obtenu en notant que : $\mathbb{E}[\|\tilde{\mathbf{v}}(t)\|_\infty^2] \leq \tilde{V}^2,\ \forall s$.

La borne du regret moyen est très ressemblante de la borne obtenue dans le cas du regret avec le gradient parfait. Il faut toutefois noter que la perte d'information (du à l'ajout de bruit dans le feedback) implique le passage du regret au regret moyen. Nos résultats ne sont plus garantis pour une réalisation mais uniquement en moyenne. Nous verrons en conclusion qu'une des perspectives possibles est l'étude du regret instantané dans le cas du gradient bruité.

Comme dans le Théorème 1 la borne du regret dépend de la durée de transmission T. Nous allons donc différencier les deux cas comme précédemment : 1) la durée de transmission T est connue ; 2) la durée de transmission est inconnue.

Durée de transmission connue

Nous allons procéder, dans le cas où le temps de transmission est connu, en optimisant la borne du regret définie dans l'équation (3.49) pa rapport au pas μ, ce qui nous permet de trouver le pas optimal μ^* :

$$\mu^* = \sqrt{\frac{2\log(1+S)}{T\tilde{V}^2}}. \tag{3.52}$$

Maintenant, pour trouver la borne du regret, il suffit de remplacer μ par le pas optimal défini dans l'équation (3.52) et nous trouvons la propriété suivante :

Corollaire 3. *Si l'Algorithme OXL est utilisé pour une transmission de durée T, avec un feedback imparfait et le pas optimal défini dans l'équation* (3.52)*, alors la borne du regret moyen est :*

$$\mathrm{EReg}(T) \leq \frac{2}{\sqrt{2}-1} P_{\max} \sqrt{2T\tilde{V}^2 \log(1+S)}, \tag{3.53}$$

ainsi la propriété de non regret moyen est garantie et $\frac{\mathrm{Ereg}(T)}{T}$ décroît en $\mathcal{O}(T^{-\frac{1}{2}})$.

Nous pouvons donc remarquer que la réduction d'information dans le cas où l'objet connaît le temps de transmission T n'a pas une grande influence sur la bonne du regret moyen.

Durée de transmission inconnue

Dans le cas où la durée de transmission T n'est pas connue, nous avons vu dans le chapitre précédent qu'il est possible d'utiliser le doubling-trick. Cependant, pour pouvoir utiliser cette méthode il est nécessaire de connaître quelques informations sur le système, comme la borne de la variance de l'estimateur imparfait $\tilde{V}$. Dans un système qui varie dans le temps, d'une manière arbitraire (non stationnaire) et non-prévisible, il peut être difficile de connaître cette information. Une des solutions possible pour contourner ce problème est d'utiliser un pas variable à la place d'un pas fixe.

Le pas variable $\mu(t)$ va donc dépendre de l'itération t. Ce pas variable doit respecter les conditions suivantes :

$$\mu(t+1) \leq \mu(t) \tag{3.54}$$

$$\frac{\sum_{t=1}^{T} \mu(t)}{T} = \mathcal{O}(T). \tag{3.55}$$

La première contrainte garantit que le pas doit être décroissant et la seconde contrainte impose une vitesse de décroissance suffisamment faible afin de toujours prendre en compte le feedback même lorsque t augmente. Dans cette étude, nous allons nous focaliser sur un pas variable simple défini par $\mu(t) = \frac{\alpha}{\sqrt{t}}$, où α est une constante positive. En utilisant l'Algorithme OXL avec le pas variable, nous montrons en annexe A.2 le résultat suivant :

Corollaire 4. *Si l'Algorithme OXL est utilisé avec un feedback imparfait et un pas variable $\mu = \frac{\alpha}{\sqrt{t}}$, où $\alpha > 0$ alors le regret moyen est borné par :*

$$\mathrm{EReg}(T) \leq \frac{P_{\max} \log(1+S)}{\alpha\sqrt{T}} + \frac{\mu P_{\max} \tilde{V}^2 \alpha (1+\log(T))}{2\sqrt{T}}. \tag{3.56}$$

Par conséquence le regret moyen de l'objet décroît en $\mathcal{O}(\log(T)T^{-1/2})$, et ainsi la propriété de non regret moyen est garantie.

La preuve du théorème dans le cas du pas variable repose sur une comparaison du regret avec le regret pondéré $\mathrm{WReg}(T)$ défini par :

$$\mathrm{WReg}(T) \triangleq \mathbb{E}\left[\sum_{t=1}^{T} \mu(t)\left(L_t(\mathbf{p}(t) - L_t(\mathbf{q}^*)\right)\right], \tag{3.57}$$

En utilisant le même raisonnement que pour le Théorème 3 avec le regret pondéré nous obtenons :

$$\mathrm{WReg}(T) \leq P_{\max}\log(1+S) + \frac{P_{\max}}{2}\tilde{V}^2 \sum_{t=1}^{T} \mu^2(t). \tag{3.58}$$

Finalement, nous utilisons un théorème de [Hardy, 1949] qui permet de comparer la convergence d'une suite avec la convergence de sa suite pondérée. Ainsi le résultat est obtenu en montrant que $\mathrm{WReg}(T)$ converge et donc que le regret $\mathrm{EReg}(T)$ converge aussi.

L'utilisation d'un pas variable par l'objet garantit donc la propriété de non regret moyen, cependant nous pouvons noter le facteur $\log T$ dans la borne du regret. Cela signifie que, lorsque l'objet utilise le pas variable, le regret mettra un peu plus de temps à décroître à cause de la réduction d'information nécessaire. En effet, si l'objet est en mesure d'utiliser le doubling-trick, i.e. l'objet est capable de déterminer $\tilde{V}$ alors le regret décroît en $\mathcal{O}(T^{-\frac{1}{2}})$. À l'inverse, s'il n'est pas en mesure de déterminer $\tilde{V}$ alors il ne peut pas utiliser le doubling-trick et doit donc utiliser le pas variable ce qui implique une évolution du regret en $\mathcal{O}(\log(T)T^{-\frac{1}{2}})$.

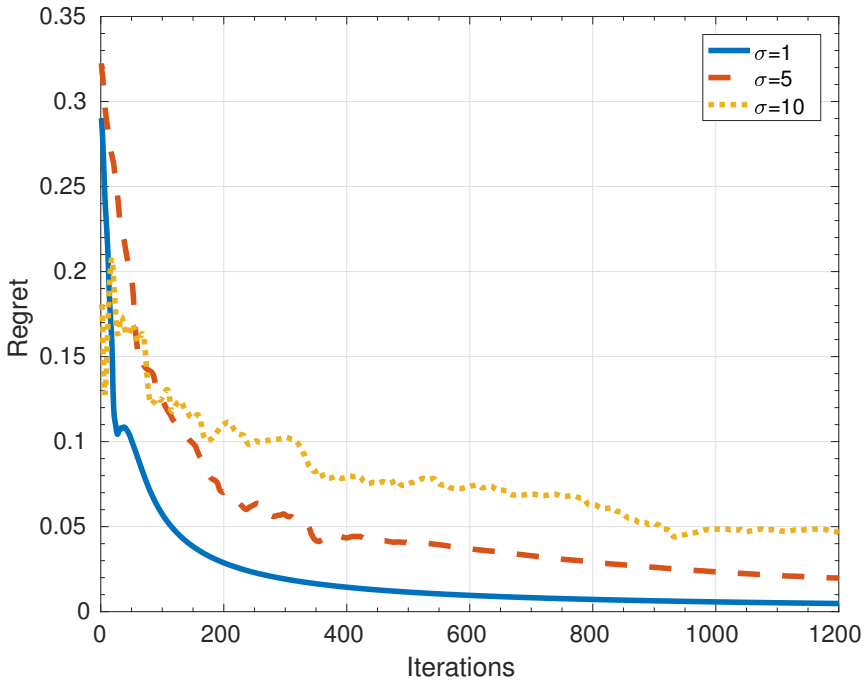
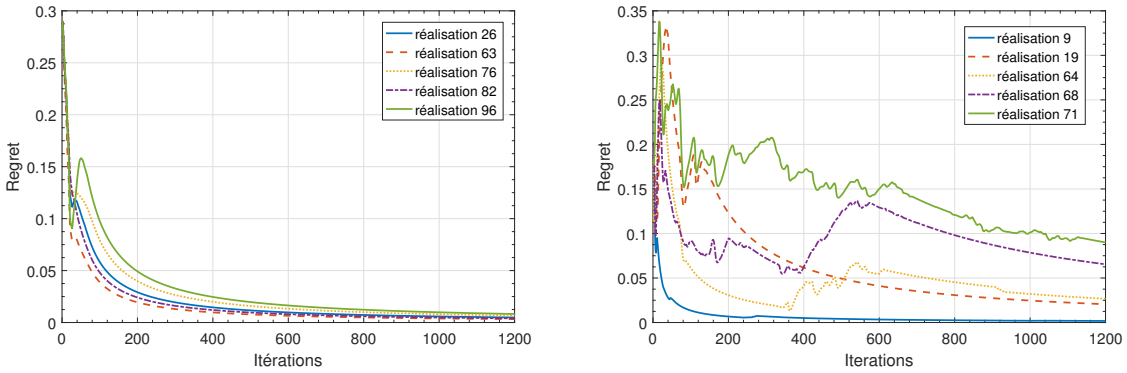


FIGURE 3.6: Cette figure représente l'évolution du regret moyen sur 100 réalisations en fonction des itérations avec un écart type de l'erreur variable $\sigma \in \{1,5,10\}$. L'erreur de l'estimation est générée en utilisant une variable gaussienne centrée et d'écart type σ. Plus l'écart type est grand, et donc plus l'estimation du gradient est mauvaise, plus le regret met du temps pour décroître vers 0. La réduction de l'information reçue par l'objet a donc un impact direct sur l'évolution du regret.

3.2.3 Résultats numériques

Dans un premier temps nous allons étudier l'impact de bruit sur l'évolution du regret. Pour cela, sur la figure 3.6, nous avons tracé l'évolution du regret moyenné sur 100 réalisations en fonction des itérations pour un bruit variable. Pour générer l'erreur de l'estimation, nous avons utilisé une variable gaussienne centrée d'écart type $\sigma \in \{1,5,10\}$. Pour chacune des réalisations, nous avons gardé les paramètres des objets, $P_{\max}, \lambda, \mu$ et S fixes ainsi que la position des objets. De cette façon le seul changement d'une itération à l'autre est l'évolution de l'estimation du gradient. De plus, seul notre objet focal utilise l'Algorithme OXL avec un gradient bruité car les autres objets utilisent l'Algorithme OXL avec le gradient parfait. Comme nous pouvions nous y attendre, plus les erreurs sont grandes (σ est grand) moins le regret décroît vite vers 0. Cela s'explique par la perte d'information, à cause des erreurs d'estimations, qui fait que certaines réalisations prennent plus de temps à décroître vers 0. Maintenant, que nous avons regardé l'évolution du regret moyen sur 100 réalisations, nous allons nous concentrer sur l'évolution du regret instantané dans le cas du gradient bruité.

Dans les figures 3.7a et 3.7b, nous avons tracé l'évolution du regret instantané pour 5 réa-



(a) Évolution du regret instantané pour $\sigma = 1$ (b) Évolution du regret instantané pour $\sigma = 10$

FIGURE 3.7: Évolution du regret instantané en fonction du nombre d'itérations pour différentes valeurs d'écart type de l'erreur de l'estimation du gradient $\sigma \in \{1, 10\}$. L'erreur est générée en utilisant une variable aléatoire centrée et d'écart type σ. Plus les estimations du gradient sont mauvaises (σ grand), plus les réalisations vont s'éloigner de la valeur moyenne du regret. Nous visualisons ainsi pourquoi le regret moyen met plus de temps à décroître dans le cas ou σ est grand. En effet, dans le cas où $\sigma = 10$, le regret varie beaucoup d'une réalisation à l'autre, ce qui impacte directement le regret moyen. Cependant, nous pouvons noter que malgré un bruit important, le regret instantané décroît toujours vers 0.

lisations (choisies arbitrairement parmi toutes les réalisations) en fonction des itérations et de l'écart type de l'erreur. Pour générer l'erreur, nous avons utilisé une variable aléatoire centrée et d'écart type σ. Les paramètres de l'objet focal, qui a été déterminé de manière arbitraire, sont les suivants : $P_{\max} = 0.5\ W$, $\lambda = 2.25$, $\mu = 0.01$. Nous remarquons que plus l'écart type de l'erreur est grand plus les réalisations instantanées s'écartent de la valeur moyenne. Ce phénomène permet d'expliquer pourquoi le regret moyen met plus de temps à décroître lorsque l'erreur augmente. En effet, plus l'erreur est grande plus il y aura de mauvaises réalisations qui vont ralentir le regret moyen.

Dans cette section nous allons aussi nous intéresser à la comparaison des performances de l'algorithme OXL entre les cas où l'objet reçoit le gradient parfait et le celui où il reçoit le gradient imparfait. Pour réaliser ces simulations, nous utilisons le modèle réaliste de canal COST-HATA avec les paramètres définis dans l'annexe B. Si l'objet a accès au gradient imparfait nous avons vu que le regret va dépendre des réalisations. Nous avons réalisé une simulation reposant sur la méthode de Monte-Carlo avec 100 réalisations afin de déterminer le regret moyen. Cette comparaison est illustrée sur la figure 3.4. Sur cette dernière figure, nous pouvons voir que pour les mêmes paramètres de simulations le regret évoluera plus rapidement vers 0 dans le cas où l'objet a accès au gradient parfait. Cependant, il faut aussi noter que malgré la réduction de l'information le regret décroit quand même vers 0.

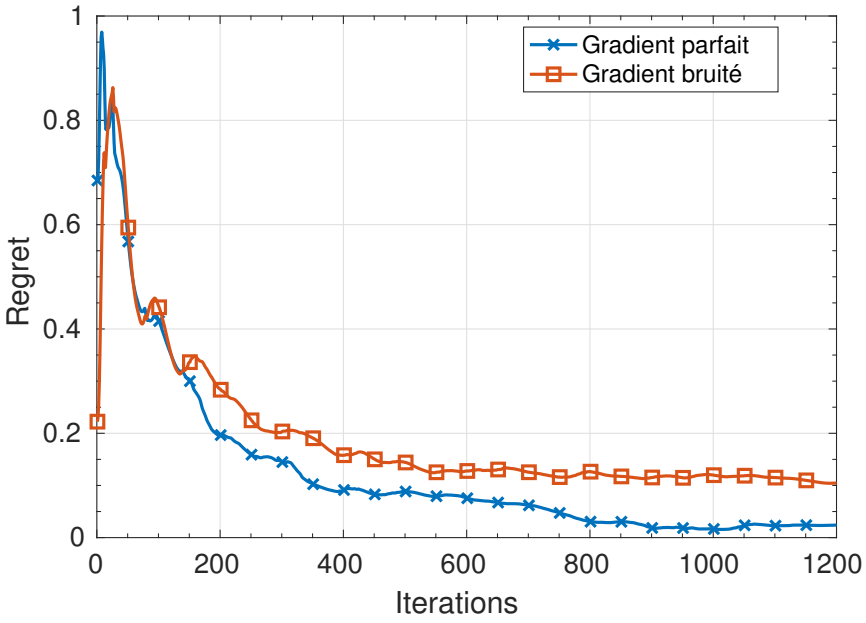


FIGURE 3.8: Comparaison du regret dans le cas où l'objet a accès au gradient parfait et au gradient imparfait. Dans le cas où l'objet a accès au gradient imparfait nous avons réalisé un moyennage sur 100 réalisations de l'erreur de l'estimation du gradient. Nous pouvons observer que dans les deux cas le regret décroît vers 0. Cependant, la vitesse de convergence est plus lente dans le cas où l'information est réduite, c'est à dire le cas où l'objet a accès au gradient imparfait.

3.3 Conclusion

Dans ce chapitre, nous avons proposé un Algorithme OXL qui permet de déterminer l'allocation de puissance des objets dans un réseau IoT dynamique et non prévisible. Nous avons vu que l'Algorithme OXL a la propriété de non regret dans le cas où l'objet dispose du gradient comme feedback (pour une durée de transmission connue ou inconnue). Cet algorithme dispose aussi de la propriété de non regret en moyenne dans la situation où l'objet a accès uniquement à un gradient erroné ou imparfait. Dans ce dernier cas, nous avons vu que les réalisations du regret instantané dépendent fortement de l'intensité de l'erreur sur le feedback. Plus les erreurs sont grandes, plus les différences entres les différentes réalisations se feront ressentir.

Dans ce chapitre, nous avons aussi comparé notre Algorithme OXL à un algorithme classique de la littérature : le WF. Nous avons montré que notre algorithme permet de contrôler plus finement le compromis entre la consommation de puissance et le débit tout en ayant une meilleure résistance aux évolutions imprévisibles du réseau.

Maintenant que nous avons étudié le cas où l'objet a accès à un feedback d'ordre I parfait ou imparfait, l'objectif du prochain chapitre est de réduire encore plus cette information et, si possible, la réduire à un unique scalaire, i.e. le feedback d'odre 0.

CHAPITRE 4

ALLOCATION DE PUISSANCE À L'AIDE D'UN FEEDBACK D'ORDRE ZÉRO

Dans le Chapitre 2, nous avons étudié le scénario où l'objet a accès à un vecteur de dimension S comme feedback, i.e., le gradient (parfait bruité). Dans ce cas, l'objet est en mesure de déterminer une allocation de puissance dynamique qui a la propriété de non regret. Ce type de feedback ne pose pas de problème dans le cas de communication classique (4G, WiFi, etc...) car le nombre d'utilisateurs qui utilisent les mêmes ressources simultanément est plus faible que dans le cas de système IoT. Cependant, lorsque le nombre d'objets est grand, la quantité du feedback envoyé (ici un vecteur de taille S par objet) devient un problème, d'autant plus lorsque certains objets du réseau doivent transmettre une faible quantité d'information. En effet, à ce moment là, le réseau pourrait se retrouver avec plus de ressources allouées aux liens de feedback qu'à la transmission de données. De plus, l'augmentation du nombre d'utilisateurs risque d'augmenter les interférences. Ces interférences risquent de corrompre les vecteurs de feedback et d'entraîner des erreurs ou des pertes de feedback. C'est pourquoi la réduction de la taille du feedback est un paramètre important dans le cadre de communication IoT. Dans ce chapitre, nous allons chercher à réduire le feedback à un seul scalaire.

4.1 Estimateur du gradient basé sur un scalaire

L'objectif de ce chapitre est de proposer un estimateur du gradient en utilisant uniquement une valeur scalaire, plus précisément la valeur de la fonction objectif. En effet, nous avons vu dans le Chapitre 2 que l'objet est en mesure de déterminer une allocation de puissance en utilisant le gradient. Si l'objet est capable d'estimer le gradient en utilisant un scalaire, il sera en mesure de définir une allocation de puissance dynamique qui aura la propriété de non regret.

La première question est de savoir de quelles informations le récepteur dispose. Précédemment, nous avons supposé que le récepteur était en mesure calculer le gradient, notamment en utilisant des trames d'apprentissage et en utilisant la réciprocité du canal. Dans ce chapitre nous allons supposer que le récepteur est en mesure de calculer la valeur de la fonction objectif, qui elle est un scalaire, afin d'utiliser cette valeur pour déterminer l'allocation de puissance. Pour calculer la valeur de la fonction objectif le récepteur doit connaître les informations suivantes : le débit minimal $R_{\min}$, le coefficient de pénalité λ, l'allocation de puissance $\mathbf{p}(t)$ et le RSIB. Le récepteur est en mesure d'estimer $\mathbf{p}(t)$ et le RSIB mais il ne peut pas estimer λ et $R_{\min}$. Cependant λ et $R_{\min}$ étant constantes dans le temps, l'objet peut les transmettre, une seule fois, au récepteur avant la première transmission. Ainsi, nous rajoutons un échange unique entre l'émetteur et le récepteur. Cet échange n'impacte pas significativement le réseau car, en contre partie de ce premier échange, le feedback passe d'un vecteur à un scalaire pour tous les autres envois de message. Cela implique que plus la transmission sera longue, plus cette solution deviendra efficace en termes de feedback. La prochaine question est de savoir comment l'objet peut estimer le gradient en se basant sur cette information.

4.1.1 Estimateur biaisé du gradient

L'estimateur du gradient est basé sur une méthode d'estimation stochastique : *Simulaneous Perturbation Stochastic Approximation* (SPSA) [Spall, 1999; Flaxman et al., 2005]. Cette méthode consiste à tirer un échantillon aléatoire de la fonction objectif $L_t(\mathbf{p})$ autour du point d'intérêt $\mathbf{p}$ afin d'obtenir un estimateur, potentiellement biaisé, du gradient $\nabla L_t(\mathbf{p})$.

Pour illustrer le fonctionnement de cet estimateur du gradient, nous allons nous concentrer sur la dérivée directionnelle de la fonction $L_t(\mathbf{p})$ en fonction d'un vecteur unitaire $\mathbf{x}$. Il est possible d'utiliser cette simplification car le gradient regroupe des dérivées directionnelles suivant les S vecteurs qui composent la base de l'espace vectoriel. La dérivée directionnelle, notée $\nabla_{\mathbf{x}} L_t(\mathbf{p})$, est définie par :

$$\nabla_{\mathbf{x}} L_t(\mathbf{p}) = \lim_{\delta \to 0} \frac{L_t(\mathbf{p}+\delta\mathbf{x}) - L_t(\mathbf{p}-\delta\mathbf{x})}{2\delta}. \tag{4.1}$$

Pour calculer cette dérivée directionnelle, il est nécessaire de connaître deux valeurs de la fonction objectif : $L_t(\mathbf{p}+\delta\mathbf{x})$ et $L_t(\mathbf{p}-\delta\mathbf{x})$ or dans notre cas nous n'avons accès qu'à une seule valeur (valeur renvoyée par le récepteur via le feedback). Pour contourner cet obstacle, nous allons échantillonner la fonction $L_t(\mathbf{p})$ selon la direction de $\mathbf{x}$ autour du point $\mathbf{p}$ en utilisant un scalaire $u \in \{-1,+1\}$ tiré de manière aléatoire suivant un processus de Bernoulli équiprobable. De manière plus concise, ce processus d'échantillonnage nous donne accès à la valeur suivante :

$$L_t(\mathbf{p}+\delta u\mathbf{x}), \quad \forall u \in \{-1,+1\}. \tag{4.2}$$

Ainsi en calculant l'espérance de ces échantillons multipliée par le scalaire u, par rapport à la

variable aléatoire u nous trouvons :

$$\mathbb{E}[u\ L_t(\mathbf{p}+\delta u\mathbf{x})] = \frac{L_t(\mathbf{p}+\delta\mathbf{x}) - L_t(\mathbf{p}-\delta\mathbf{x})}{2}. \tag{4.3}$$

Il suffit ensuite de diviser l'espérance calculée en (4.3) par δ pour trouver une approximation de $\nabla_{\mathbf{x}} L_t(\mathbf{p})$:

$$\mathbb{E}\left[u\ \frac{L_t(\mathbf{p}+\delta u\mathbf{x})}{\delta}\right] \approx \nabla_{\mathbf{x}} L_t(\mathbf{p}) \tag{4.4}$$

Il faut noter que l'équation (4.4) n'est vraie que lorsque δ tend vers 0. En utilisant le fait que les fonctions objectifs $L_t(\mathbf{p})$ soit K-Lipschitz nous montrons dans l'annexe C.1 que l'estimateur défini en (4.2) est un estimateur biaisé de $\nabla_{\mathbf{x}} L_t(\mathbf{p})$ (pour $\delta \neq 0$). Le biais entre l'estimateur et la dérivée directionnelle est donné par K et δ, où K est la constante de Lipschitz de la fonction objectif $L_t(\mathbf{p})$.

Maintenant que nous avons détaillé la manière d'estimer la dérivée directionnelle, nous devons généraliser cette méthode pour obtenir un estimateur du gradient. Comme dit précédemment, le gradient est une collection de dérivées directionnelles. L'idée est de calculer cette dérivée selon la direction $\mathbf{u}$, où le vecteur $\mathbf{u}$ est généré aléatoirement suivant une loi uniforme définie sur la sphère uniforme de dimension S. Ainsi, en calculant l'espérance sur le vecteur aléatoire $\mathbf{u}$ (et non plus sur le scalaire u) nous obtenons :

$$\nabla L_t(\mathbf{p}) \approx \frac{S}{\delta}\mathbb{E}[\mathbf{u}(t)\ L_t(\tilde{\mathbf{p}}(t))], \tag{4.5}$$

où $\tilde{\mathbf{p}}(t)$ est défini par :

$$\tilde{\mathbf{p}}(t) = \mathbf{p}(t) + \delta\mathbf{u}(t),\ \ \mathbf{u}(t) \in \left\{\mathbf{u}' \in \mathbb{R}^S \,\middle|\, ||\mathbf{u}'||_2^2 = 1\right\}. \tag{4.6}$$

Pour plus de détail sur la manière de définir cet estimateur, nous invitons le lecteur à lire l'Annexe C.1. Il faut remarquer que dans l'équation (4.5) il ne s'agit plus de la dérivée directionnelle $\nabla_{\mathbf{x}} L_t(\mathbf{p})$ mais bien du gradient $\nabla L_t(\mathbf{p})$. Pour finir, l'estimateur utilisé par l'objet est donc le suivant :

$$\tilde{\mathbf{v}}(t) = \frac{S}{\delta} L_t(\tilde{\mathbf{p}}(t))\ \mathbf{u}. \tag{4.7}$$

Le récepteur doit donc renvoyer $L_t(\tilde{\mathbf{p}}(t))$ à l'objet afin que ce dernier calcule l'estimateur du gradient à chaque itération de l'algorithme.

4.1.2 Impact de l'estimateur sur l'Algorithme OXL

Nous avons vu dans la section précédente que l'objet doit connaître $L_t(\tilde{\mathbf{p}}(t))$ afin de calculer l'estimateur du gradient. Or l'objet ne transmet plus avec le vecteur d'allocation de puissance $\mathbf{p}(t)$ mais avec son allocation de puissance modifiée : $\tilde{\mathbf{p}}(t) = \mathbf{p}(t) + \delta\mathbf{u}$. La conséquence principale de cette modification du vecteur d'allocation de puissance est sa possible sortie de l'espace faisable. Il est donc possible que l'objet se retrouve à devoir transmettre à des puissances supérieures à $P_{\max}$ ou pire à des puissances négatives. Ceci est impossible ; nous devons donc trouver un moyen de contourner ce problème.

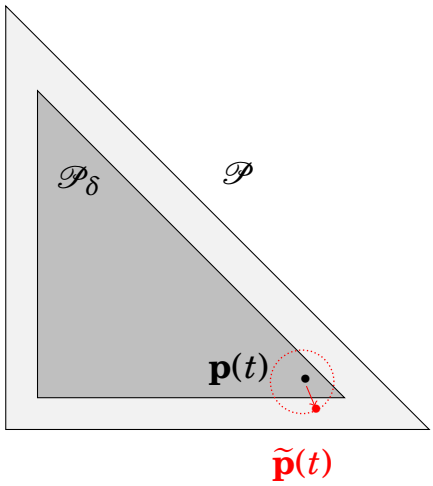


FIGURE 4.1: Cette figure illustre le fonctionnement du nouvel espace faisable, représenté en gris sur la figure. Cet espace est construit de telle manière que chaque point $\mathbf{p}_\delta$ de l'espace gris garantit que $\mathbf{p}_\delta + \delta\mathbf{u}$ est dans l'espace faisable original.

L'approche innovante que nous proposons est de définir un nouvel espace faisable $\mathscr{P}_\delta$ plus petit (inclus dans l'espace original $\mathscr{P}$). Ce nouvel espace est défini de telle manière que pour n'importe quel $\mathbf{p}_\delta \in \mathscr{P}_\delta$ appartement à cet espace alors $\mathbf{p}_\delta + \delta\mathbf{u}$ sera forcement dans $\mathscr{P}$, mathématiquement cela peut s'écrire comme suit :

$$\forall\ \mathbf{p}_\delta \in \mathscr{P}_\delta\ \Rightarrow \mathbf{p}_\delta + \delta\mathbf{u} \in \mathscr{P}. \tag{4.8}$$

Cette idée est illustrée dans la Figure 4.1 où nous pouvons voir un exemple d'espace $\mathscr{P}$ ainsi que l'espace réduit $\mathscr{P}_\delta$. En utilisant les nouvelles contraintes, nous pouvons définir le nouvel espace faisable comme :

$$\mathscr{P}_\delta = \left\{ \mathbf{p}_\delta \in \mathbb{R}^S \,\middle|\, p_\delta^s \geq \delta,\ \sum_{s=1}^{S} p_\delta^s \leq P_{\max} - \sqrt{S}\delta \right\}. \tag{4.9}$$

Le fait de définir un nouvel espace faisable $\mathscr{P}_\delta$ nous oblige à modifier notre Algorithme OXL, plus particulièrement en ce qui concerne l'étape de projection exponentielle. En effet, l'étape de projection exponentielle a été définie pour garantir que l'allocation de puissance soit dans l'espace $\mathscr{P}$.

Afin de modifier l'étape de projection exponentielle il faut commencer par changer la fonction de régularisation entropique f en remplaçant les contraintes de l'espace faisable $\mathscr{P}$ par les contraintes définies en (4.9). Une fois cette étape réalisée, il faut calculer la fonction convexe conjuguée $f^*(\mathbf{y}(t))$ de $f(\mathbf{p}(t))$. Dans l'Annexe A1 nous avons vu que $\mathbf{p}(t) \triangleq \nabla f^*(\mathbf{y}(t))$. Finalement, nous obtenons l'étape de projection suivante :

$$p_\delta^s = Q_\delta^s(\mathbf{y}(t)) = \delta + P_{\max}(1 - C_\delta)\frac{\exp(y^s(t))}{1 + \sum_{i=1}^{S}\exp(y^i(t))},\ \forall s \tag{4.10}$$

Algorithme OXL$_0$: *Online Exponential Learning Algorithm with Zeroth Order Feedback*

Initialisation : $\mathbf{y}(0) \leftarrow 0$; $t \leftarrow 0$.

Répéter

- **Phase de pré-transmission :** mise à jour de la puissance

$\mathbf{p}(t) \leftarrow \mathbf{Q}_\delta(\mathbf{y}(t))$ définie par (3.21)

Tirage aléatoire de $\mathbf{u}(t)$ sur la sphère unitaire de dimension S

- **Transmission avec** $\tilde{\mathbf{p}}(t) = \mathbf{p}_\delta(t) + \delta \mathbf{u}(t)$
- **Phase de post-transmission :** réception du feedback $L_t(\tilde{\mathbf{p}}(t))$

Calcul de l'estimateur du gradient $\tilde{\mathbf{v}}(t) = \frac{S}{\delta} L_t(\tilde{\mathbf{p}}(t))\mathbf{u}$

Mise à jour du score $\mathbf{y}(t+1) \leftarrow \mathbf{y}(t) - \mu(t)\ \tilde{\mathbf{v}}(t)$

$t \leftarrow t+1$

jusqu'à : fin de transmission

où C_δ est défini par :

$$C_\delta = \frac{\delta}{P_{\max}}(S + \sqrt{S}). \tag{4.11}$$

Nous retrouvons une expression similaire à l'étape de projection de notre premier Algorithme OXL (partie droite de l'équation (4.10)) avec des modifications qui dépendent de $P_{\max}$, δ et S. De l'équation (4.10) et de la définition de l'espace faisable (4.9) nous pouvons déduire deux contraintes sur le choix du paramètre δ :

$$0 \leq \delta \leq \frac{P_{\max}}{S + \sqrt{S}} \leq \frac{P_{\max}}{\sqrt{S}}. \tag{4.12}$$

En plus de la modification de l'étape de projection exponentielle, il faut rajouter deux étapes à notre algorithme : 1) la première consiste à générer le vecteur aléatoire $\mathbf{u}(t)$; 2) la seconde consiste à calculer l'estimateur en fonction de la valeur de la fonction objectif $L_t(\mathbf{p}(t))$ et du vecteur $\mathbf{u}(t)$. Maintenant que nous avons toutes les modifications de notre algorithme, nous allons pouvoir le détailler.

4.2 Le nouvel Algorithme OXL$_0$

Dans cette section, nous allons détailler l'Algorithme OXL$_0$ que l'objet utilisera dans le cas où ce dernier a accès uniquement à un scalaire.

Les modifications apportées à l'Algorithme OXL peuvent être résumées par les équations suivantes :

$$\begin{aligned} \tilde{\mathbf{v}}(t) &= \frac{S}{\delta} L_t(\tilde{\mathbf{p}}(t))\mathbf{u}(t), \\ \mathbf{y}(t+1) &= \mathbf{y}(t) - \mu\ \tilde{\mathbf{v}}(t) \\ \mathbf{p}_\delta(t) &= \mathbf{Q}_\delta(\mathbf{y}(t)), \end{aligned} \tag{OXL$_0$}$$

où $\tilde{\mathbf{v}}(t)$ représente l'estimateur du gradient défini dans l'équation (4.7). L'Algorithme OXL$_0$ est détaillé ci-dessous.

Nous allons présenter les différents résultats théoriques concernant les performances de l'Algorithme OXL_0.

4.3 Résultats théoriques

Dans l'Annexe C.2.1 nous présentons la preuve du Théorème 3 énoncé comme suit :

Théorème 3. *Si l'Algorithme OXL_0 est utilisé avec les paramètres fixes δ et μ alors le regret moyen est borné par :*

$$\text{EReg}(T) \leq \frac{P_{\max}\log(1+S)}{2\mu} + \mu T S^2 \left(\frac{B}{\delta}+K\right)^2 + KT\delta \left(3 + P_{\max}\left(S+2\sqrt{S}\right)\right). \tag{4.13}$$

où K est la constante de Lipschitz et B la valeur maximale de toutes les fonctions objectif $L_t(\cdot)$.

Nous retrouvons les deux termes présents dans la borne du Théorème 1, ainsi qu'un troisième du à la réduction de l'information et à l'estimateur du gradient. Nous allons maintenant donner quelques pistes pour comprendre le résultat de ce théorème.

Étapes de la preuve

La différence majeure entre les preuves du Théorème 1 et la preuve du Théorème 3 est que, dans le dernier cas, l'objet transmet avec une politique modifiée $\mathbf{p}(t)+\delta\mathbf{u}$. Ainsi, nous sommes en mesure d'écrire la relation suivante :

$$\mathbb{E}\left[\sum_{t=1}^{T} L_t(\mathbf{p}_\delta(t)) - L_t(\mathbf{q})\right] = \mathbb{E}\left[\sum_{t=1}^{T} L_t(\mathbf{p}(t)+\delta\mathbf{u}(t)) - L_t(\mathbf{q})\right], \tag{4.14}$$

où l'espérance est calculée en fonction de l'aléa des variables $\mathbf{u}(t)$. L'objectif de cette première étape est de relier cette quantité au regret. Pour cela, nous utilisons le fait que les fonctions objectif sont des fonctions K-Lipschitz et nous trouvons :

$$\text{EReg}(T) \leq \mathbb{E}\left[\sum_{t=1}^{T} L_t(\mathbf{p}_\delta(t)) - L_t(\mathbf{q})\right] + KT\delta\left(1 + P_{\max}(S+\sqrt{S})\right). \tag{4.15}$$

La prochaine étape consiste à lier l'espérance de l'équation (4.15) à l'estimateur du gradient calculé par l'objet. L'estimateur $\tilde{\mathbf{v}}(t)$ possède les propriétés suivantes :

$$\mathbb{E}[\tilde{\mathbf{v}}(t)] = \nabla\tilde{L}_t(\mathbf{p}(t)) \tag{4.16}$$

$$\tilde{L}_t(\mathbf{p}(t)) \triangleq \mathbb{E}\left[L_t(\mathbf{p}(t)+\delta\mathbf{u}(t))\right], \tag{4.17}$$

où l'espérance est calculée en fonction du vecteur aléatoire $\mathbf{u}(t)$. De ces propriétés nous pouvons déduire que $\tilde{\mathbf{v}}(t)$ est un estimateur non-biaisé du gradient de la fonction $\tilde{L}_t(\mathbf{p}(t))$ définie en (4.17). Nous montrons dans l'Annexe C.1, que la fonction $\tilde{L}_t(\mathbf{p}(t))$ est une approximation biaisé de la fonction $L_t(\mathbf{p}(t))$. Il faut donc maintenant déterminer le biais qui existe entre les fonctions

objectif $L_t(\mathbf{p})$ et les fonctions $\tilde{L}_t(\mathbf{p})$. En utilisant le fait que les fonctions objectif sont K-Lipschitz nous avons :

$$|L_t(\mathbf{p}) - \tilde{L}_t(\mathbf{p})| \leq K\delta. \tag{4.18}$$

En utilisant cette borne entre les fonctions objectif $L_t(\mathbf{p})$ et les fonctions $\tilde{L}_t(\mathbf{p})$, ainsi que la borne du regret (4.15) nous trouvons :

$$\mathrm{EReg}(T) \leq \mathbb{E}\left[\sum_{t=1}^{T} \tilde{L}_t(\mathbf{p}_\delta(t)) - \tilde{L}_t(\mathbf{q})\right] + KT\delta\left(3 + P_{\max}(S+\sqrt{S})\right). \tag{4.19}$$

Maintenant que nous avons relié le regret à la fonction $\tilde{L}_t(\mathbf{p}(t))$, il est possible de faire apparaître l'estimateur $\tilde{\mathbf{v}}(t)$. Nous remarquons que nous pouvons réécrire $\nabla\tilde{L}_t(\mathbf{p}_\delta(t))$ comme :

$$\nabla\tilde{L}_t(\mathbf{p}_\delta(t)) = \mathbb{E}[\tilde{\mathbf{v}}(t)|\mathbf{u}(1),...,\mathbf{u}(t-1)], \tag{4.20}$$

où l'espérance est calculée en fonction des variables aléatoires $\mathbf{u}(t)$. Par conséquent, en utilisant la convexité de la fonction $\tilde{L}_t(\mathbf{p})$ et le Théorème de l'espérance totale, nous trouvons :

$$\mathbb{E}\left[\sum_{t=1}^{T} \tilde{L}_t(\mathbf{p}_\delta(t)) - \tilde{L}_t(\mathbf{q})\right] \leq \mathbb{E}\left[\sum_{t=1}^{T} \langle\tilde{\mathbf{v}}(t)|\mathbf{p}_\delta(t) - \mathbf{q}\rangle\right]. \tag{4.21}$$

Á partir de cette étape, la preuve devient similaire à la preuve du Théorème 1, à la différence que la fonction entropique de régularisation $f_\delta(\mathbf{p})$ change (du aux changements de l'espace faisable) et devient :

$$f_\delta(\mathbf{p}) = \sum_{s=1}^{S}\left(p_\delta^s - \delta\right)\log\left(p_\delta^s - \delta\right) + \left(C - \sum_{s=1}^{S} p_\delta^s\right)\log\left(C - \sum_{s=1}^{S} p_\delta^s\right), \tag{4.22}$$

où $C = P_{\max} - \delta\sqrt{S}$. En utilisant cette fonction de régularisation et sa fonction convexe conjuguée, nous pouvons borner $\sum_{t=1}^{T}\langle\tilde{\mathbf{v}}(t)|\mathbf{q}\rangle$ par :

$$\mathbb{E}\left[\sum_{t=1}^{T} \langle\tilde{\mathbf{v}}(t)|\mathbf{p}_\delta(t) - \mathbf{q}\rangle\right] \leq \mathbb{E}\left[\sum_{t=1}^{T} \langle\tilde{\mathbf{v}}(t)|\mathbf{p}_\delta(t+1) - \mathbf{p}_\delta(t)\rangle\right] + \frac{P_{\max}\log(1+S)}{2\mu}. \tag{4.23}$$

Pour obtenir le résultat final, il nous reste à borner la somme de droite de l'équation (4.23). Pour cela nous allons encore une fois utiliser la fonction de régularisation et l'inégalité de Cauchy-Schwartz et nous trouvons la borne du regret suivante :

$$\mathrm{EReg}(T) \leq \frac{P_{\max}\log(1+S)}{2\mu} + \mu TS^2\left(\frac{B}{\delta} + K\right)^2 + KT\delta\ \left(3 + P_{\max}\left(S + 2\sqrt{S}\right)\right). \tag{4.24}$$

Maintenant, que nous avons détaillé les étapes de la preuve, nous allons nous concentrer sur les différents paramètres de l'algorithme.

4.3.1 Impact des différents paramètres de l'Algorithme OXL_0

Dans la borne du regret (4.13) et dans l'Algorithme OXL_0 nous retrouvons différents paramètres. Certains sont communs avec les résultats de l'Algorithme OXL, comme μ et d'autres sont nouveaux, comme le paramètre δ.

Le paramètre μ va jouer le même rôle que dans l'Algorithme OXL. Pour rappel, ce paramètre contrôle l'impact du feedback (de l'information reçue) d'une itération sur l'autre. En effet, si μ est grand alors il y aura des grandes variations dans l'allocation de puissance. Cela peut accélérer la vitesse de décroissance du regret vers zéro, cependant il peut y avoir des oscillations. Á l'inverse, si μ est faible, les variations de l'allocation de puissance seront plus faibles d'une itération à l'autre. La vitesse de décroissance du regret sera plus faible, mais le risque d'oscillations sera lui aussi plus faible. Comme nous l'avons déjà dit dans le chapitre précédent, il y a donc un compromis à faire dans le choix de μ entre la vitesse de décroissance et la stabilité dans l'évolution de l'allocation de puissance.

Concernant le paramètre δ, propre à l'Algorithme OXL_0, il représente le rayon d'exploration autour du point d'intérêt $\mathbf{p}(t)$ pour lequel on va estimer $\nabla L_t(\mathbf{p}(t))$. Comme pour le choix de μ, le choix de δ repose sur un compromis, cette fois entre le biais de l'estimateur et sa variance. Si δ est faible, cela signifie que l'allocation de puissance $\mathbf{p}_\delta(t)$ ne sera pas fortement perturbée et éloignée par rapport à $\mathbf{p}(t)$ et implique un plus faible biais de l'estimateur du gradient. Cependant, comme l'objet a accès à un faible nombre d'échantillons (il n'a accès qu'à un seul échantillon par itération) une valeur de δ trop faible augmente la variance de l'estimateur. Á l'inverse, si δ est grand le biais de l'estimateur augmentera mais sa variance diminuera.

Nous allons voir dans la section suivante quelles valeurs nous devons prendre pour les paramètres δ et μ en fonction des deux cas suivants : lorsque la durée de transmission T est connue ou lorsque cette durée est inconnue.

4.3.2 Propriété de non regret

La borne du regret étant convexe par rapport à μ, nous allons dans un premier temps chercher sa valeur optimale, pour cela nous allons calculer et annuler la dérivé d'ordre I de la borne du regret. Une fois ce calcul fait, nous trouvons la valeur suivante du pas optimal μ^* :

$$\mu^* = \sqrt{\frac{P_{\max}\log(1+S)}{2T}}\left[S\left(\frac{B}{\delta}+K\right)\right]^{-1}, \tag{4.25}$$

où B est la valeur maximale des fonctions objectif et K la constante de Lipschitz des fonctions objectif. Cette borne dépend des différents paramètres du système et plus particulièrement de δ, le paramètre propre à l'Algorithme OXL_0. La prochaine étape consiste donc à déterminer la valeur de δ qui minimise la borne du regret. Nous allons remplacer μ par μ^* dans la borne du regret, ce qui nous donne :

$$\text{EReg}(T) \leq \sqrt{TP_{\max}\log(1+S)}\left[S\left(\frac{B}{\delta}+K\right)\right] + KT\delta\ \left(3+P_{\max}\left(S+2\sqrt{S}\right)\right). \tag{4.26}$$

La borne du regret obtenue dans l'équation (4.26) est convexe en fonction de δ.

Cependant, bien que convexe la présence des contraintes suivantes : $\delta \leq \frac{P_{\max}}{S+\sqrt{S}} \leq \frac{P_{\max}}{\sqrt{S}}$, nous empêche de trouver une solution en forme close de δ^*. Il faut toutefois noter que notre objectif premier est d'obtenir la propriété de non regret. Pour y parvenir, il suffit de trouver un δ qui respecte les contraintes tout en limitant la croissance (qui doit être inférieure à $\mathcal{O}(T)$) de la borne du regret définie en (4.26). Il faut donc dans un premier temps déterminer l'ordre de grandeur des variations de δ en fonction de T. Pour cela, nous remarquons que nous pouvons réécrire le regret comme :

$$\mathrm{EReg}(T) \leq \frac{\mathcal{O}(\sqrt{T})}{\delta} + \mathcal{O}(T)\delta. \tag{4.27}$$

De l'équation ci-dessus, nous remarquons que si le pas croît en $\mathcal{O}(T^{-1/4})$ alors le regret va croître en $\mathcal{O}(T^{\frac{3}{4}})$ ce qui est suffisant pour avoir la propriété de non regret. Ainsi, il ne nous reste plus qu'à trouver un pas qui décroît en $\mathcal{O}(T^{-1/4})$ et qui respecte les contraintes sur δ, ce pas peut être défini comme :

$$\delta^* = \frac{P_{\max}}{(S+\sqrt{S})T^{\frac{1}{4}}}. \tag{4.28}$$

Cette valeut de δ respecte les contraintes tout en limitant la croissance de la borne du regret. Pour visualiser cela, il faut remplacer δ par δ^* dans la borne (4.26) ce qui nous donne :

$$\mathrm{EReg}(T) \leq U_1 T^{\frac{3}{4}} + U_2 T^{\frac{1}{2}}, \tag{4.29}$$

où les termes U_1 et U_2 sont définis respectivement par :

$$\begin{aligned} U_1 &= SB\left(S+\sqrt{S}\right)\sqrt{\frac{2\log(1+S)}{P_{\max}}} + K\left(3+P_{\max}(S+2\sqrt{S})\right)\frac{P_{\max}}{S+\sqrt{S}} \\ U_2 &= \sqrt{2P_{\max}\log(1+S)}\,SK. \end{aligned} \tag{4.30}$$

Les paramètres δ^* et μ^* ci-dessus dépendent de la durée de la transmission T. Nous allons donc dans un premier temps nous pencher sur le cas ou cette durée est connue en avance par l'objet.

4.3.3 Durée de transmission connue

Dans le cas où la durée de transmission est connue à l'avance, l'objet est en mesure de déterminer les valeurs des paramètres optimaux δ^* et μ^*. Une fois ces paramètres obtenus, il suffit de remplacer ces valeurs dans la borne du regret définie en (4.13), ce qui nous donne :

$$\mathrm{EReg}(T) \leq U_1 T^{\frac{3}{4}} + U_2 T^{\frac{1}{2}}, \tag{4.31}$$

où les termes U_1 et U_2 sont définis respectivement par :

$$\begin{aligned} U_1 &= SB\left(S+\sqrt{S}\right)\sqrt{\frac{2\log(1+S)}{P_{\max}}} + K\left(3+P_{\max}(S+2\sqrt{S})\right)\frac{P_{\max}}{S+\sqrt{S}} \\ U_2 &= \sqrt{2P_{\max}\log(1+S)}\,SK. \end{aligned} \tag{4.32}$$

Ce qui nous amène au corollaire suivant.

Corollaire 5. *Si l'Algorithme OXL_0 est utilisé pour une transmission de durée T connue, avec un feedback scalaire et en utilisant les paramètres optimaux δ^* et μ^* définis dans les équations* (4.28) *et* (4.25)*, alors la propriété de non regret moyen est garantie et le regret moyen $\frac{\mathrm{EReg}(T)}{T}$ décroît en $\mathcal{O}(T^{-\frac{1}{4}})$.*

La remarque la plus importante concernant le Corollaire 5 est que la vitesse de décroissance du regret diminue en comparaison du cas où l'objet reçoit un feedback vectoriel, cette vitesse de décroissance passe de $\mathcal{O}(T^{-\frac{1}{2}})$ à $\mathcal{O}(T^{-\frac{1}{4}})$. Bien que plus compliquée que la borne du Corollaire 3, cette borne dépend uniquement des paramètres du système comme : le nombre de sous-porteuses S, K ou encore B. Nous remarquons que cette borne ne dépend pas du nombre d'utilisateurs M présents dans le système ce qui est un avantage majeur dans le cas des réseaux IoT où le nombre d'objets peut être grand.

Maintenant que nous avons étudié le cas où la durée de transmission est connue nous allons explorer le cas ou cette durée est inconnue.

4.3.4 Durée de transmission inconnue

Lorsque la durée de transmission n'est pas connue nous avons vu que deux méthodes étaient possibles. La première consiste à utiliser le doubling-trick et la seconde nécessite l'utilisation d'un pas variable. Nous avons vu qu'il y avait une différence en termes de vitesse de décroissance du regret moyen. Lorsque l'objet utilise le pas variable, le regret décroît moins rapidement (la différence de vitesse dépend du cas étudié) cependant il n'est pas nécessaire de calculer les paramètres optimaux pour chaque fenêtre du doubling-trick. Á l'inverse, le doubling-trick garantit une vitesse de décroissance similaire au cas où l'objet connaît en avance la durée de transmission, cependant il doit calculer les paramètres optimaux pour chaque fenêtre de la transmission.

La question que nous devons nous poser, afin de déterminer si l'objet est en mesure d'utiliser le doubling-trick, est : quelles informations l'objet doit calculer pour chacune des fenêtres ? Pour cela il faut regarder les équations (4.28) et (4.25), elles nous indiquent que l'objet doit connaître $P_{\max}$, S, B et K. Les deux premières valeurs sont faciles à connaître, en effet l'objet connaît la puissance maximale $P_{\max}$ qu'il peut allouer ainsi que le nombre de sous-porteuses S. Pour les deux informations suivantes B et K qui sont respectivement : la valeur maximale des fonctions objectifs et la constante de Lipschitz, nous devons trouver leurs expressions analytiques. Un calcul rapide nous permet de trouver les valeurs de B et K dans notre cas :

$$B = SP_{\max} + \lambda R_{\min} \tag{4.33}$$

$$K = 1 + 2\lambda R_{\min}. \tag{4.34}$$

De ces équations, qui ne sont valables que pour le problème défini dans le Chapitre 2, nous pouvons déduire que l'objet doit aussi connaître $R_{\min}$ et λ. Ces informations sont connues par

l'objet et cela implique qu'il est en mesure de déterminer les paramètres optimaux pour chacune des fenêtres. Ainsi, nous pouvons formuler le corollaire suivant :

Corollaire 6. *Si l'Algorithme OXL$_0$ est utilisé pour une transmission de durée inconnue, avec un feedback scalaire et en utilisant le doubling-trick avec les paramètres optimaux δ^* et μ^* définis dans les équations* (4.28) *et* (4.25) *dans chacune des fenêtres, alors le regret moyen est borné par :*

$$\mathrm{EReg}(T) \leq \frac{2}{2^{\frac{3}{4}}-1} U_1 T^{\frac{3}{4}} + \frac{2}{\sqrt{2}-1} U_2 T^{\frac{1}{2}}. \tag{4.35}$$

Ceci implique que le regret moyen $\frac{\mathrm{EReg}(T)}{T}$ décroît en $\mathcal{O}(T^{-\frac{1}{4}})$.

Comme nous l'avons énoncé plus tôt dans cette section, lorsque l'objet utilise le doubling-trick, son regret décroît au même ordre de grandeur (décroissance en $\mathcal{O}(T^{-\frac{1}{4}})$) que lorsque la durée de transmission est connue à l'avance. Nous pouvons en tirer les mêmes conclusions, que lorsque la durée de transmission est connue : malgré la réduction de l'information reçue, l'objet est en mesure de déterminer une politique d'allocation de puissance qui garantit la propriété de non regret moyen.

Afin d'illustrer le fonctionnement de l'Algorithme OXL$_0$, nous allons présenter une sélection des simulations numériques pertinentes.

4.4 Résultats numériques

Toutes les simulations réalisées dans cette partie utilisent le modèle présenté dans l'Annexe B. Pour rappel, il s'agit du modèle COST-HATA, qui est un modèle réaliste pour les gains de canaux dans un environnement urbain/suburbain. Dans ces simulations, nous considérerons $M = 10$ objets utilisant S sous-porteuses (S variant d'un cas à l'autre). En ce qui concerne les paramètres de chaque objet : $P_{\max}$, $R_{\min}$ et λ ils sont générés de manière aléatoire indépendamment pour chaque objet. Pour plus de détails le lecteur est invité à se reporter à l'annexe B.

Nous allons dans un premier temps nous concentrer sur l'étude du regret moyen. Pour cela nous avons réalisé trois simulations différentes. Dans la première, nous allons comparer le regret de nos deux algorithmes et ainsi illustrer l'impact de la réduction du feedback sur la performance en terme du regret. Dans une seconde simulation, nous allons illustrer l'impact du nombre de sous-porteuses S. Et finalement, une dernière simulation qui représente l'évolution du regret instantané pour quelques réalisations de l'Algorithme OXL$_0$ choisies arbitrairement.

4.4.1 Impact de la réduction du feedback à un scalaire

Dans cette simulation, chaque objet partagera les quatre mêmes sous-porteuses $S = 4$. Nous avons désigné un objet focal de manière arbitraire. Tous les autres objets utiliserons l'Algo-

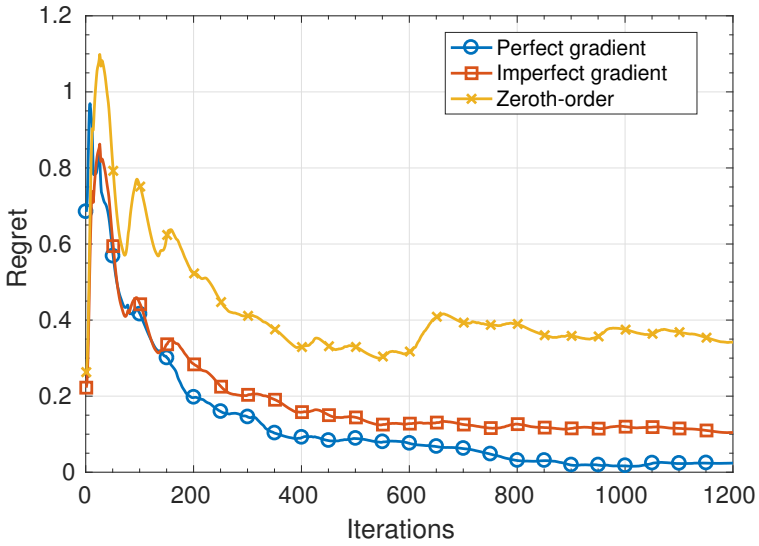


FIGURE 4.2: Cette Figure illustre l'évolution du regret moyen pour les trois cas étudiés : OXL avec gradient parfait, OXL avec gradient imparfait et non-biaisé et OXL_0 avec la valeur de la fonction objectif. L'évolution du regret moyen dépend de l'information reçue par l'objet comme prédit par les résultats théoriques. Nous pouvons noter que le gradient parfait et le gradient bruité donnent des résultats relativement proches, bien qu'un peu plus lent dans le cas imparfait. Par contre l'évolution du regret est plus lente dans le cas où l'objet a accès uniquement à un scalaire. Cependant bien que lent, le regret tend vers zéro dans tous les cas ce qui confirme nos résultats théoriques.

rithme OXL avec un gradient parfait. Notre objet focal utilisera nos deux Algorithmes : OXL avec gradient parfait, OXL avec gradient bruité et OXL_0.

Pour déterminer le regret moyen dans les deux cas où une estimation du gradient est utilisée, nous avons utilisé une méthode de Monte-Carlo avec un moyennage sur 100 réalisations. Le résultat de ces simulations est présenté dans la Figure 4.2

Nous pouvons remarquer que, comme prédit par les résultats théoriques, le regret moyen tend à décroître vers zéro moins vite lorsque le feedback reçu diminue. De plus, il faut noter que lorsque l'objet a accès au gradient bruité non-biaisé, la vitesse de décroissance du regret n'est pas fortement diminuée, ce qui était aussi prédit par nos résultats théoriques. Cependant, dans le cas où l'objet a accès uniquement à un scalaire, le regret décroît plus lentement. La présence d'oscillations plus importantes avec le feedback scalaire vient de l'aléatoire de l'estimateur du gradient construit à partir d'un seul échantillon de la fonction objectif.

4.4.2 Impact du nombre de sous-porteuses

La prochaine étape de notre analyse est d'étudier l'impact du nombre de sous-porteuses S sur l'Algorithme OXL_0. Pour cela nous allons générer un réseau en utilisant les mêmes propriétés que la simulation précédente, à savoir : un nombre d'objets fixes où chaque objet (excepté notre

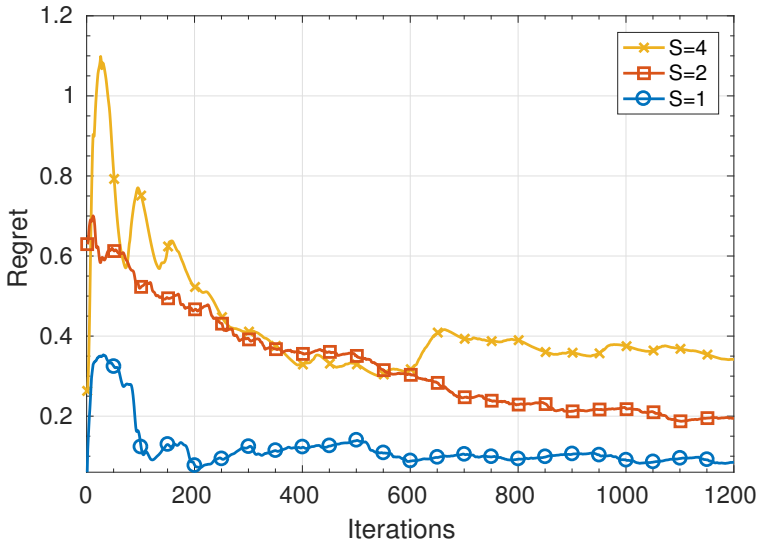


FIGURE 4.3: Cette figure nous montre l'impact du nombre de sous-porteuses S sur l'évolution du regret moyen. Plus le nombre de sous-porteuses augmente, plus il est difficile pour le regret de décroître. Cela vient de la construction de notre estimateur qui détermine la direction du gradient de manière aléatoire. Plus S augmente plus la probabilité de trouver la bonne direction du gradient diminue.

objet focal) utilise l'Algorithme OXL avec un gradient parfait et des paramètres générés de manière aléatoire à l'exception de l'objet focal qui utilisera toujours l'Algorithme OXL_0. Ce qui changera sera le nombre de sous-porteuses $S \in \{1,2,4\}$. Pour calculer le regret moyen, nous allons utiliser une méthode de Monte-Carlo sur 100 réalisations. Les résultats concernant ces simulations sont présentés sur la Figure 4.3.

Nous pouvons conclure que le nombre de sous-porteuses impacte fortement l'évolution du regret moyen. Lorsque $S = 1$ le regret moyen tend très vite vers zéro et lorsque $S = 4$ le regret moyen met met plus de temps pour décroître. Nous pouvons s'expliquer en regardant la construction de notre estimateur. L'objet cherche à estimer un vecteur de dimension S à l'aide d'un scalaire. La direction du vecteur estimé est tirée de manière aléatoire à l'aide du vecteur $\mathbf{u}$. Plus S augmente, plus il est difficile que le vecteur aléatoire $\mathbf{u}$ pointe vers la bonne direction du gradient. Pour l'illustrer, nous allons prendre l'exemple d'une fonction en dimension 1. Dans cette situation, il y a deux directions possibles : positive ou négative. Il y a donc une chance sur deux de se tromper étant donné que le vecteur $\mathbf{u}$ est généré de manière uniforme. Lorsque $S = 2$, il y a un grand nombre de possibilités (le cercle unitaire), déjà à ce stade il devient difficile statistiquement parlant d'obtenir un vecteur $\mathbf{u}$ qui donne parfaitement la bonne direction ou une direction approchée.

Pour conclure au sujet de ces simulations, notre algorithme est sensible en ce qui concerne le nombre de sous-porteuses. Ce dernier aura des difficultés à donner de bons résultats lorsque

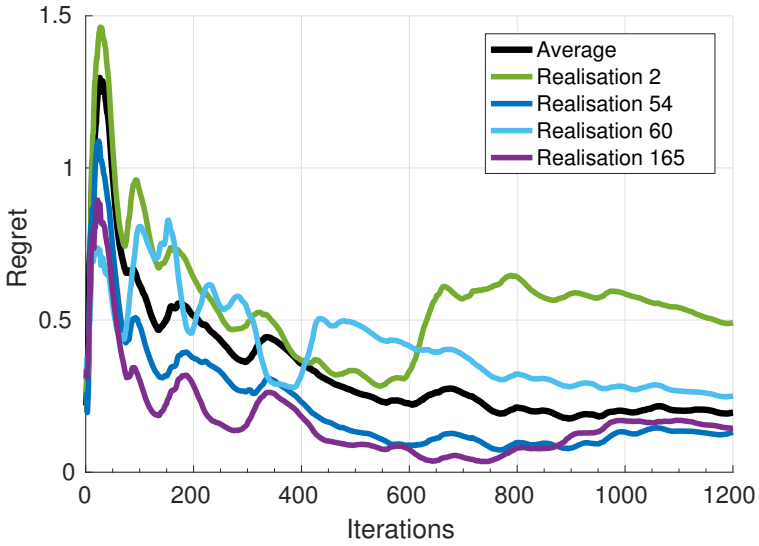


FIGURE 4.4: La figure ci-dessus représente l'évolution du regret moyen et du regret instantané pour des réalisations choisies arbitrairement. Nous pouvons voir que même si le regret moyen tend vers zéro relativement rapidement, certaines réalisations mettent plus de temps que d'autres. Cela est du à l'estimateur du gradient construit à partir d'un unique échantillon de la fonction objectif. Si l'estimateur pointe souvent vers les bonnes directions du gradient alors le regret va décroître rapidement (quasiment aussi rapidement que dans le cas du gradient parfait). Á l'opposé lorsque l'estimation pointe souvent vers les mauvaises directions le regret met plus de temps à décroître.

le nombre de sous-porteuses devient trop grand. Dans le cas des réseaux IoT, cela n'est pas un problème majeur étant donné que la plupart des objets n'ont pas besoin de transmettre beaucoup d'informations et que dans la majorité des cas les objets utiliseront un faible nombre de sous-porteuses.

Jusqu'à présent, nous avons tracé l'évolution du regret moyen. Il peut être pertinent d'étudier le regret instantané.

4.4.3 Évolution du regret instantané

Pour illustrer la variance du regret en fonction des itérations nous avons utilisé le même modèle que précédemment avec $S = 2$. L'objectif étant d'étudier l'évolution du regret instantané, nous avons utilisé une méthode de Monte-Carlo pour calculer le regret moyen et avons choisi d'illustrer quelques réalisations de manières arbitraire.

Dans la Figure 4.4, nous pouvons remarquer que même si le regret moyen diminue de manière relativement rapide, certaines réalisations prennent plus de temps que d'autres. Cela vient de notre estimateur biaisé qui a une direction choisie aléatoirement. Il y a des réalisations ou l'estimateur pointe plus souvent vers les directions du gradient $\nabla L_t(\mathbf{p}(t))$, comme la réalisation 165 où le regret décroît très rapidement vers zéro. Á l'opposé, lorsque l'estimateur pointe plus

souvent vers des mauvaises directions le regret met plus de temps à diminuer et peut même dans certain cas remonter. C'est le cas de la réalisation 2, où le regret décroît en suivant le regret moyen au début, puis soudainement (autour de la réalisation 800) le regret remonte.

4.5 Conclusions

En conclusion dans ce chapitre nous avons étudié le cas ou l'objet a accès uniquement à la valeur de la fonction objectif (à l'itération précédente) comme feedback. Dans un premier temps, nous avons montré qu'il est possible de construire un estimateur (biaisé) du gradient des fonctions objectif en utilisant ce scalaire. Cependant, la construction de cet estimateur du gradient requière la transmission d'une politique d'allocation de puissance modifiée par une variable aléatoire. Cette modification implique qu'il est possible que la politique d'allocation de puissance sorte de l'espace faisable. C'est pourquoi dans un second temps, nous avons modifié l'Algorithme OXL afin de prendre en compte cette variable aléatoire. La principale nouveauté de ce chapitre vient de la modification de l'espace faisable. En effet, nous avons défini un nouvel espace faisable, plus petit et contenu dans l'espace original. Ainsi, n'importe quelle allocation de puissance de cet espace réduit est dans l'espace original et cela même après la modification aléatoire de l'allocation de puissance. De ce nouvel espace faisable, nous avons proposé l'Algorithme OXL_0 qui permet de déterminer la politique d'allocation de puissance et nous avons montré que cet algorithme a la propriété de non regret moyen. Finalement, dans une dernière partie nous avons illustré le fonctionnement de l'Algorithme OXL_0 pour différents scénarios et nous avons aussi comparé cet algorithme à l'Algorithme OXL afin de mesurer l'impact de la réduction de feedback sur les performances en terme de regret.

CHAPITRE 5

GÉNÉRALISATION ET APPLICATIONS DIVERSES

Dans les Chapitres 3 et 4 nous avons proposé des algorithmes pour résoudre le problème de minimisation de puissance sous contraintes de QoS dans un réseau IoT distribué du Chapitre 2. Dans cette section nous allons généraliser la méthodologie derrière ces algorithmes afin de pouvoir résoudre différents problèmes d'optimisation de ressources dans les réseaux dynamiques et imprévisibles. Pour cela, nous allons définir un modèle mathématique général des problèmes d'allocation de ressources. Ensuite, nous détaillerons les algorithmes associés à ce problème générique ainsi que leurs garanties théoriques. Enfin, nous présenterons un exemple d'application différent qui permet d'adresser le problème du contrôle d'interférence dans un réseau IoT.

5.1 Généralisation du problème d'allocation de ressources

La première étape dans la généralisation du problème d'optimisation défini en (2.6) est de généraliser notre fonction objectif définie en (2.5). Nous allons utiliser une fonction de coût $L_t(\mathbf{p})$ avec $\mathbf{p} \in \mathscr{P}$ où $\mathscr{P}$ est l'ensemble faisable des allocations de puissance.

5.1.1 Hypothèses

Dans cette section, nous allons détailler les différentes hypothèses que $\mathscr{P}$ et $L_t(\mathbf{p})$ doivent respecter.

Hypothèse 1 : $\mathscr{P}$ est un ensemble convexe. L'ensemble $\mathscr{P}$ est dit convexe si la propriété suivante est respectée :

$$\forall(\mathbf{p},\mathbf{q}) \in \mathscr{P}^2,\ \forall x \in [0,1],\ x\mathbf{p} + (1-x)\mathbf{q} \in \mathscr{P}. \tag{H1}$$

Autrement dit, si pour chaque point $\mathbf{p}$ et $\mathbf{q}$ de l'ensemble $\mathscr{P}$, tous les points sur le segment qui relie les points $\mathbf{p}$ et $\mathbf{q}$ sont dans l'ensemble $\mathscr{P}$, alors cet ensemble est convexe [Boyd and Vandenberghe, 2004].

Hypothèse 2 : les fonctions $L_t(\mathbf{p})$ sont des fonctions convexes par rapport à $\mathbf{p} \in \mathscr{P}$. Cette seconde hypothèse, implique la propriété suivante :

$$\forall (\mathbf{p},\mathbf{q}) \in \mathscr{P}^2,\ \ L(\mathbf{q}) \geq L(\mathbf{q}) + \langle \nabla L(\mathbf{p}) | \mathbf{q}-\mathbf{p} \rangle\,, \tag{H2}$$

qui représente la condition d'ordre I de convexité [Boyd and Vandenberghe, 2004]. Cette propriété nous permet de borner le regret par le regret d'un problème linéaire qui dépend du gradient (ou du sous gradient) $\nabla L(\mathbf{p})$. De plus, la convexité des fonctions objectif nous permet d'utiliser le gradient comme direction vers la solution optimale. Cependant, la convexité seule n'est pas suffisante pour la formulation de nos algorithmes.

Hypothèse 3 : les fonctions $L_t(\mathbf{p})$ sont bornées. Nous définissons donc la constante B telle que :

$$L_t(\mathbf{q}) \leq B,\ \ \forall t \in \{1,...,T\},\ \forall \mathbf{p} \in \mathscr{P}. \tag{H3}$$

La constante B est ainsi une borne supérieure à toutes les fonctions objectif.

Hypothèse 4.a : les gradients (ou sous-gradients) $\nabla L_t(\mathbf{p})$ sont bornés. La constante V représente la borne supérieure de la norme de tous les gradients à tous les instants t et est définie par :

$$||\nabla L_t(\mathbf{p})||_\infty^2 \leq V^2,\ \ \forall t \in \{1,...,T\},\ \forall \mathbf{p} \in \mathscr{P}. \tag{H4a}$$

Il faut noter que dans les cas où les gradients ne sont pas connus mais estimés, c'est la norme des estimateurs $\tilde{\mathbf{v}}(t)$ qui doit être bornée.

Hypothèse 4.b : les normes des estimateurs $\tilde{\mathbf{v}}(t)$ des gradients $\nabla a L_t(\mathbf{p})$ sont bornées. Pour cela nous définissons la constante $\tilde{V}$ telle que :

$$||\tilde{\mathbf{v}}(t)||_\infty^2 \leq \tilde{V},\ \ \forall t \in \{1,...,T\},\ \forall \mathbf{p} \in \mathscr{P}, \tag{H4.b}$$

où $\tilde{\mathbf{v}}(t)$ est l'estimateur du gradient.

Hypothèse 5 : les fonctions $L_t(\mathbf{p})$ sont des fonctions K-Lipschitz. Une fonction $L_t(\mathbf{p})$ est dite K-Lipschitz si elle respecte la condition suivante :

$$\forall (\mathbf{p},\mathbf{q})\ \in\ \mathscr{P}^2,\ |L_t(\mathbf{p}) - L_t(\mathbf{q})|\ \leq\ K\|\mathbf{p}-\mathbf{q}\|, \tag{H5}$$

où K la constante de Lipschitz. Cette contrainte sur la fonction objectif, qui limite les variations en fonction de $\mathbf{p}$, est nécessaire pour être en mesure de borner la valeur du gradient envoyée par le feedback. Il est important de noter que cette hypothèse ne limite pourtant pas les variations des fonctions objectif d'une itération t à l'autre. Grâce à cette propriété, il est possible de définir un estimateur du gradient qui est à la base de l'algorithme OXL_0 dans le cas où le récepteur renvoie un unique scalaire comme feedback.

Maintenant que nous avons défini les hypothèses du modèle générique, nous devons nous demander si elles sont réalistes pour des systèmes de communication. La première contrainte, la convexité des ensembles faisables, est classique dans la majorité des problèmes d'optimisation de ressources. La seconde hypothèse est un peu plus restrictive mais elle est respectée dans la majorité des problèmes pour lesquels les fonctions objectif dépendent de la consommation totale de puissance, la capacité de Shannon ou encore le RSIB. Les conditions restrictives sur les valeurs des fonctions objectifs ainsi que la norme des gradients sont aisément respectées dans le cas des systèmes de communication, du au fait que l'espace faisable soit fini et que des fonctions objectif relativement simples (somme des puissances, capacité de Shannon, etc.).

5.1.2 Problème d'optimisation de ressources en ligne général

Maintenant, nous pouvons présenter le problème générique que l'objet cherche à résoudre à chaque instant t :

$$\begin{array}{ll} \text{minimiser} & L_t(\mathbf{p}(t)) \\ \text{sur} & \mathbf{p}(t) = (p^1(t), \dots, p^S(t)) \\ \text{sous contraintes} & f_i(\mathbf{p}(t)) \leq 0, \quad \forall i \in \{1, ..., I\}. \end{array} \tag{5.1}$$

où $L_t(\mathbf{p})$ respectent les contraintes $(H2)-(H5)$: fonctions convexes, K-Lipschitz, bornées par B et norme du gradient bornée par V. Les contraintes sont définies à l'aide des fonctions $f_i(\mathbf{p})$, $\forall i$ qui sont des fonctions convexes par rapport à $\mathbf{p}$. Ces conditions sont classiques et facilement respectées dans le cas des problèmes d'allocation de puissance, où les contraintes concernent la puissance totale disponible, les puissances dans chaque ressource disponible (e.g. sous-porteuses) ou encore le débit de Shannon. Les contraintes ainsi définies garantissent la convexité de l'espace faisable $(H1)$. L'espace faisable $\mathscr{P}$ défini par les conditions ci-dessus s'écrit comme :

$$\mathscr{P} = \left\{ \mathbf{q} \in \mathbb{R}^S : f_i(\mathbf{q}) \leq 0, \ \forall i \in \{1, ..., I\} \right\}. \tag{5.2}$$

Maintenant que le problème à résoudre ainsi que l'espace faisable sont définis de manière générique, nous pouvons définir la métrique à minimiser, c'est-à-dire le regret. Le regret est défini de la même manière que dans l'équation (2.7) à la différence que les fonctions utilisées sont les fonctions génériques :

$$\mathrm{Reg}(T) \triangleq \sum_{t=1}^{T} L_t(\mathbf{p}(t)) - \min_{\mathbf{q} \in \mathscr{P}} \sum_{t=1}^{T} L_t(\mathbf{q}). \tag{5.3}$$

L'objectif de chaque objet est donc de chercher une politique d'allocation de puissance en ligne qui garantit la propriété de non-regret telle que définie par l'équation (2.9).

5.2 Mise en forme générale de l'algorithme OXL

Comme nous l'avons fait dans le Chapitres 3, nous allons présenter l'algorithme OXL utilisé par l'objet pour déterminer l'allocation de puissance dans le cas où ce dernier a accès : au gradient ou au gradient bruité non biaisé. Dans chacun des cas cités ci-dessus nous présenterons

les résultats théoriques à la fois dans le cas où la durée de transmission est connue mais aussi lorsque la durée de transmission est inconnue.

Pour illustrer le fonctionnement de l'algorithme nous allons uniquement parler du cas où le feedback est le gradient parfait $\mathbf{v}(t) = \nabla L_t(\mathbf{p}(t))$ puis expliquerons les changements dans le cas du gradient imparfait.

La première étape dans la minimisation du regret consiste à majorer le regret par une somme de termes linéaires. Pour cela, il faut utiliser la propriété de convexité des fonctions objectifs $L_t(\mathbf{p})$. Ainsi, le regret peut être borné par :

$$\mathrm{Reg}(T) \leq \sum_{t=1}^{T} \langle \nabla L_t(\mathbf{p}(t)) | \mathbf{p}(t) - \mathbf{q}^* \rangle \,, \tag{5.4}$$

où $\mathbf{q}^*$ est l'allocation de puissance fixe qui minimise la somme des fonctions objectif.

$$\mathbf{q}^* = \arg\max_{\mathbf{q}\in\mathscr{P}} \left\{ \sum_{t=1}^{T} L_t(\mathbf{q}) \right\}. \tag{5.5}$$

Grâce à l'équation (5.4), l'objet doit maintenant chercher à minimiser la quantité de droite, qui est bien une somme de fonctions linéaires.

Pour minimiser cette somme, nous avons vu dans le Chapitre 3 que l'algorithme utilisé découle d'une variante de l'algorithme FoL qui est l'algorithme du FoRL [Shalev-Shwartz, 2011]. Pour rappel cet algorithme est énoncé de la façon suivante :

$$\mathbf{p}(t+1) = \arg\min_{\mathbf{q}\in\mathscr{P}} \left\{ \sum_{i=1}^{t} \langle \nabla L_i(\mathbf{p}(i)) | \mathbf{q} \rangle + f(\mathbf{q}) \right\}, \tag{5.6}$$

où $f(\cdot)$ est la fonction de régularisation. Le choix de cette fonction de régularisation va déterminer le type d'algorithme obtenu. Dans notre cas, nous allons utiliser la fonction de régularisation entropique basée sur les fonctions de contraintes de notre problème d'optimisation, mathématiquement elle est définie par :

$$f(\mathbf{p}) = \sum_i (-f_i(\mathbf{p})) \log(-f_i(\mathbf{p})). \tag{5.7}$$

Un calcul similaire à celui utilisé dans le Chapitre 3, nous permet de réécrire (5.6) sous la forme :

$$\mathbf{y}(t+1) = \mathbf{y}(t) - \mathbf{v}(t) \tag{5.8}$$

$$\mathbf{p}(t+1) = \mathbf{Q}(\mathbf{y}). \tag{5.9}$$

où la fonction de projection $\mathbf{Q}(\mathbf{y})$ est définie telle que :

$$\mathbf{Q}(\mathbf{y}) = \arg\max_{\mathbf{q}\in\mathscr{P}} \{ \langle \mathbf{y}(t) | \mathbf{q} \rangle + f(\mathbf{q}) \}, \tag{5.10}$$

avec $\mathbf{v}(t) = \nabla L_t(\mathbf{p}(t))$. Afin de trouver une forme analytique à la fonction de projection $\mathbf{Q}(\mathbf{y})$ nous devons résoudre le problème d'optimisation (5.10). Pour cela nous allons utiliser la fonction de convexe conjuguée $f^*(\mathbf{y})$ de $f(\mathbf{p})$ définie par :

$$f^*(\mathbf{y}) = \max_{\mathbf{q}} \langle \mathbf{y}(t) | \mathbf{q} \rangle - f(\mathbf{q}). \tag{5.11}$$

Algorithme GMD : *Generalized Mirror Descent*

Initialisation : $\mathbf{y}(0) \leftarrow 0$; $t \leftarrow 0$.

Répéter

- **Phase de pré-transmission :** mise à jour de la puissance

$\mathbf{p}(t) \leftarrow \nabla f^*(\mathbf{y}(t))$

- **Transmission avec** $\mathbf{p}(t)$
- **Phase de post-transmission :** réception du feedback $\mathbf{v}(t)$

Mise à jour du score $\mathbf{y}(t+1) \leftarrow \mathbf{y}(t) - \mu(t)\ \mathbf{v}(t)$

$t \leftarrow t+1$

jusqu'à : fin de transmission

La fonction convexe conjuguée est M-fortement régulière (*strongly-smooth*) par rapport à la norme $\|\cdot\|_\infty$ [1] et possède la propriété intéressante [Shalev-Shwartz, 2011] définie par :

$$\nabla f^*(\mathbf{y}) = \arg\max_{\mathbf{q}\in\mathscr{P}} \{\langle \mathbf{y}(t)|\mathbf{q}\rangle - f(\mathbf{q})\}. \tag{5.13}$$

Cette propriété nous permet de déterminer facilement l'allocation de puissance à partir du moment où nous sommes en mesure de déterminer la fonction $f^*(\mathbf{y})$ ainsi que son gradient. En utilisant cette propriété et la fonction convexe conjuguée il est possible de réécrire l'algorithme comme :

$$\mathbf{y}(t+1) = \mathbf{y}(t) - \mathbf{v}(t) \tag{5.14}$$

$$\mathbf{p}(t+1) = \nabla f^*(\mathbf{y}). \tag{5.15}$$

Un fois le problème mis sous forme générique, il est possible d'obtenir un algorithme d'allocation en respectant les étapes suivantes. La première étape est de vérifier que nos fonctions objectif respectent bien les hypothèses ($H0$)-($H5$), définies dans la Section 5.1.1, puis de calculer la fonction de régularisation $f(\cdot)$ et sa convexe conjuguée $f^*(\cdot)$. Cette dernière étape nous permet de calculer la mise à jour de l'allocation de puissance $\mathbf{p}(t)$. D'une manière plus concise, l'algorithme d'allocation de puissance peut être résumé de la manière ci-dessus. Nous allons maintenant appliquer cet algorithme au cas où l'objet a accès au gradient parfait puis à celui où il a accès au gradient imparfait.

Cas du gradient parfait

Dans le cas du gradient parfait, nous obtenons le résultat suivant :

1. Une fonction $f^*(\mathbf{y}) : \mathbb{R}^S \to \mathbb{R}$ est M-fortement régulière par rapport à la norme $\|\cdot\|_\infty$ si :

$$f^*(\mathbf{y}_1+\mathbf{y}_2) \leq f^*(\mathbf{y}_1) + \langle \nabla f^*(\mathbf{y}_1)|\mathbf{y}_1\rangle - \mathbf{y}_2\rangle - \frac{M}{2}\|\mathbf{y}_1-\mathbf{y}_2\|_\infty^2. \tag{5.12}$$

Théorème 4. *Si l'algorithme Generalised Mirror Descent (GMD) est utilisé avec un pas constant μ et un feedback parfait alors le regret est borné par :*

$$\mathrm{Reg}(T) \leq \frac{f^*(\boldsymbol{0})}{\mu} + \frac{\mu M T V^2}{2}, \tag{5.16}$$

où $\boldsymbol{0}$ est le vecteur de taille S contenant que des 0 et M est la constante de forte régularité de la fonction $f^*(\mathbf{y})$.

Comme dans le Chapitre 3 nous pouvons faire la distinction du cas où l'objet connaît ou non la durée de transmission. Quand ce dernier connaît la durée de transmission à l'avance, il est possible de calculer le pas optimal μ en fonction des paramètres du système. Ce pas optimal s'obtient après optimisation de la borne du regret (5.16), par rapport à μ et est égal à :

$$\mu^* = \sqrt{\frac{2f^*(\mathbf{0})}{TMV^2}}, \tag{5.17}$$

où M est la constante de forte régularité de la fonction $f^*(\mathbf{y})$. En utilisant ce pas optimal, nous obtenons le corollaire suivant qui permet de trouver la borne optimale, en fonction de μ, du regret.

Corollaire 7. *Si l'algorithme GMD est utilisé pour une transmission de durée T, avec un feedback parfait et le pas optimal défini dans l'équation* (3.39)*, alors la propriété de non-regret est garantie et le regret est borné par :*

$$\mathrm{Reg}(T) \leq \sqrt{2TMV^2 f^*(\boldsymbol{0})}. \tag{5.18}$$

Quand l'objet ne connaît pas la durée de transmission à l'avance, il est possible d'utiliser le doubling-trick ce qui nous donne le corollaire suivant :

Corollaire 8. *Si l'algorithme GMD est utilisé pour une transmission de durée inconnue, avec un feedback parfait et en utilisant le doubling-trik, alors la propriété de non-regret est garantie et le regret est borné par :*

$$\mathrm{Reg}(T) \leq \frac{\sqrt{2}}{\sqrt{2}-1}\sqrt{2TMV^2 f^*(\boldsymbol{0})}. \tag{5.19}$$

Cas du gradient imparfait

Quand l'objet reçoit un estimateur non biaisé du gradient $\tilde{\mathbf{v}}(t)$, il est toujours possible d'utiliser l'algorithme GMD. La différence est que maintenant ce ne sont plus les gradients qui doivent être bornés mais les estimateurs de ces derniers. La première hypothèse concernant l'estimateur est l'absence d'erreur systématique qui ce traduit par :

$$\mathbb{E}[\tilde{\mathbf{v}}(t)] = \nabla L_t(\mathbf{p}(t)) \tag{5.20}$$

La seconde concerne les variations de chaque composant du vecteur. Pour cela on définit $\tilde{V}$ une constante qui majore la norme des estimateurs du gradient définie telle que :

$$\|\tilde{\mathbf{v}}(t)\|_{\infty}^{2} \leq \tilde{V}^{2}, \ \ \forall t \in \{1,...,T\}, \ \forall s. \tag{5.21}$$

Une autre différence dans cette situation est qu'il ne s'agit plus de borner le regret mais l'espérance du regret ou le regret moyen EReg(T) défini dans l'équation (2.10). Malgré ces différences nous obtenons une borne similaire que celle obtenue dans le cas du gradient parfait.

Théorème 5. *Si l'algorithme GMD est utilisé avec un pas constant μ et un feedback du gradient imparfait alors le regret est borné par :*

$$\mathrm{EReg}(T) \leq \frac{f^{*}(\boldsymbol{0})}{\mu} + \frac{\mu}{2} M T \tilde{V}^{2}. \tag{5.22}$$

La borne du regret moyen définie dans le Théorème 5 dépend de μ, il est donc possible de déterminer le pas optimal en utilisant la même méthode que précédemment. De cette manière nous trouvons le pas μ^{*} :

$$\mu^{*} = \sqrt{\frac{2 f^{*}(\mathbf{0})}{T M \tilde{V}^{2}}}. \tag{5.23}$$

Ce pas optimal dépend du temps de transmission T, il faut donc faire la différence entre la situation où l'objet connaît ou ne connaît pas la durée de transmission à l'avance.

Lorsque l'objet connaît la durée de transmission à l'avance, il peut calculer le pas optimal μ^{*}. Il faut donc remplacer le pas optimal défini par l'équation (5.23) dans la borne du Théorème 5 ce qui nous donne le corollaire suivant :

Corollaire 9. *Si l'algorithme GMD est utilisé pour une transmission de durée T, avec un feedback du gradient imparfait et le pas optimal défini dans l'équation* (5.23)*, alors la propriété de non-regret est garantie et le regret est borné par :*

$$\mathrm{EReg}(T) \leq \sqrt{2 M T \tilde{V}^{2} f^{*}(\boldsymbol{0})}. \tag{5.24}$$

Cependant, lorsque la durée de transmission T n'est pas connue à l'avance, l'objet n'est pas en mesure de calculer le pas optimal μ^{*}. De plus, nous avons vu dans le Chapitre 3, qu'il était parfois difficile d'avoir accès aux paramètres $\tilde{V}$, F et B nécessaires au calcul du pas. Si l'objet ne peut pas calculer le pas optimal, alors il ne peut pas utiliser le doubling-trick. Une solution possible est d'utiliser un pas variable $\mu(t)$ qui respecte les conditions suivantes :

$$\mu(t+1) \leq \mu(t) \tag{5.25}$$

$$\frac{\sum_{t=1}^{T} \mu(t)}{T} = \mathcal{O}(T). \tag{5.26}$$

En utilisant cette méthode nous obtenons le résultat suivant.

Corollaire 10. *Si l'algorithme GMD est utilisé pour une transmission de durée inconnue, avec un feedback du gradient imparfait et en utilisant un pas variable* $\mu(t) = \alpha t^{-\frac{1}{2}}$*, alors la propriété de non-regret moyen est garantie et le regret moyen est borné par :*

$$\frac{\mathrm{EReg}(T)}{T} \leq \frac{f^*(\boldsymbol{0})}{\alpha\sqrt{T}} + \frac{\alpha}{\sqrt{T}} M\tilde{V}^2(1+\log(T)). \tag{5.27}$$

Les détails du calcul concernant le doubling-trick sont donnés dans l'Annexe A.1.3.

Maintenant que nous avons généralisé l'utilisation de l'Algorithme OXL, nous allons nous concentrer sur le cas où l'objet reçoit uniquement un scalaire comme feedback.

5.3 Mise en forme générale de l'algorithme OXL_0

Ici le feedback n'est plus un vecteur mais un scalaire, i.e., la valeur de la fonction objectif $U_t(\mathbf{p}(t)$. Nous avons vu dans le Chapitre 4, que le récepteur transmet à une allocation de puissance modifiée $\tilde{\mathbf{p}}(t) = \mathbf{p}(t) + \delta\mathbf{u}(t)$. En effet, le gradient étant une collection de dérivées directionnelles, nous avons vu qu'il était possible de l'estimer par $\tilde{\mathbf{v}}(t)$ défini par :

$$\tilde{\mathbf{v}}(t) = \frac{S}{\delta}\mathbb{E}\left[L_t(\mathbf{p}(t) + \delta\mathbf{u}(t))\ \mathbf{u}(t)\right], \tag{5.28}$$

où δ est une constante positive et $\mathbf{u}(t)$ un vecteur tiré de manière aléatoire uniformément sur la sphère $\mathbb{S}$ définie dans le Chapitre 4. Ainsi, pour être en mesure de construire un estimateur du gradient à partir de la valeur de la fonction objectif, il faut transmettre avec une allocation de puissance définie par $\tilde{\mathbf{p}}(t) = \mathbf{p}(t) + \delta\mathbf{u}(t)$. Le problème de cette allocation de puissance, outre le fait qu'elle soit aléatoire, est qu'elle peut sortir de l'espace faisable $\mathscr{P}$.

Pour contourner ce problème, l'idée est d'utiliser un nouvel espace faisable rétréci $\mathscr{P}_\delta \subset \mathscr{P}$ tel que :

$$\mathscr{P}_\delta = \left\{\mathbf{p} \in \mathbb{R}^S \,|\mathbf{p} + \delta\mathbf{u} \in \mathscr{P}\ ,\ \forall \mathbf{u} \in \mathbb{S}\right\}. \tag{5.29}$$

La détermination de ce nouvel espace faisable est un des points sensibles dans la mise en place de l'algorithme *Generalised Mirror Descent with zeroth order feedback* (GMD_0). Une fois cet espace faisable défini, il est possible de déterminer les contraintes qui le caractérisent. Ces fonctions vont permettre de définir une nouvelle fonction de régularisation $f_\delta(\mathbf{p})$. A partir de cette étape, il faut suivre le raisonnement présenté lors de la construction de l'algorithme OXL_0 du Chapitre 4 ; 1) Il faut calculer la fonction convexe conjuguée $f_\delta^*(\mathbf{y})$ de $f_\delta(\mathbf{p})$; 2) Cette fonction permet de déterminer l'allocation de puissance en utilisant la propriété de la convexe conjuguée (5.11) ; 3) Une fois toutes ces étapes réalisées, il est possible de détailler l'algorithme GMD_0 ci-dessus.

Si l'objet utilise l'algorithme GMD_0 alors le regret est borné comme suit.

Algorithme GMD$_0$: *Generalized Mirror Descent with Zeroth Order Feedback*

Initialisation : $\mathbf{y}(0) \leftarrow 0$; $t \leftarrow 0$.

Répéter

- **Phase de pré-transmission :** mise à jour de la puissance

$\mathbf{p}(t) \leftarrow \nabla f_\delta^*(\mathbf{y}(t))$

Tirage aléatoire de $\mathbf{u}(t)$ uniformément distribué sur la sphère unitaire de dimension S

- **Transmission avec** $\tilde{\mathbf{p}}(t) = \mathbf{p}_\delta(t) + \delta\mathbf{u}(t)$
- **Phase de post-transmission :** réception du feedback $L_t(\tilde{\mathbf{p}}(t))$

Calcul de l'estimateur du gradient $\tilde{\mathbf{v}}(t) = \frac{S}{\delta} L_t(\tilde{\mathbf{p}}(t))\mathbf{u}(t)$

Mise à jour du score $\mathbf{y}(t+1) \leftarrow \mathbf{y}(t) - \mu(t)\ \tilde{\mathbf{v}}(t)$

$t \leftarrow t+1$

jusqu'à : fin de transmission

Théorème 6. *Si l'algorithme GMD$_0$ est utilisé avec un feedback scalaire et avec les paramètres fixes δ et μ, alors le regret moyen est borné par :*

$$\begin{aligned}\mathrm{EReg}(T) \leq \frac{H}{\mu} + \frac{\mu T S^2}{M}\left(\frac{B}{\delta} + K\right)^2 \\ + KT\delta\ (3+A).\end{aligned} \tag{5.30}$$

où K est la constante de Lipschitz, B la valeur maximum des fonctions objectifs $L_t(\cdot)$, H est la valeur minimale de $f(\boldsymbol{p})$, M est la constante de forte régularité de $f^*(\mathbf{y}(t))$ par rapport à la norme $\|\cdot\|_\infty$ et A est défini comme :

$$\|\boldsymbol{p}_\delta - \boldsymbol{p}\|_2^2 \leq A,\ \forall \boldsymbol{p} \in \mathscr{P},\ \forall \boldsymbol{p}_\delta \in \mathscr{P}_\delta. \tag{5.31}$$

Comme dans la situation où l'objet a accès à un feedback vectoriel, la borne du regret dépend des paramètres du système mais aussi des pas μ et δ. L'influence de ces paramètres est similaire quelque soit le problème traité. C'est pourquoi, pour plus de détails sur ces comportements le lecteur est invité à lire le Chapitre 4.

Pour déterminer les paramètres optimaux, il faut résoudre un problème d'optimisation en fonction de δ et μ. Comme expliqué dans le Chapitre 4, il est difficile de trouver une solution analytique pour δ, cependant une valeur sous-optimale bien choisie suffit pour garantir la propriété de non-regret. Une fois que nous avons déterminé la valeur sous-optimale de δ nous pouvons déterminer la valeur optimale de μ en résolvant un problème d'optimisation convexe. Ces paramètres dépendent de la durée de transmission, il faut donc différencier les deux cas : la durée de transmission connue et la durée de transmission inconnue. Dans la mesure où la durée de transmission est connue l'objet peut déterminer les valeurs optimales des pas ce qui nous donne le corollaire suivant :

Corollaire 11. *Si l'algorithme GMD$_0$ est utilisé pour une transmission de durée connue T, avec un feedback scalaire et en utilisant les paramètres optimaux δ^* et μ^*, alors la propriété de non regret moyen est garantie et le regret moyen $\frac{\mathrm{EReg}(T)}{T}$ décroît en $\mathscr{O}(T^{-\frac{1}{4}})$.*

Dans le cas où l'objet ne connaît pas la durée de transmission à l'avance, ce dernier ne peut pas calculer les paramètres optimaux. Encore une fois il est possible d'utiliser le doubling-trick. Il faudra cependant faire attention et vérifier que l'objet est bien en mesure de déterminer les paramètres μ^* et δ^* dans chaque fenêtre. Pour cela, il faut calculer la valeur des paramètres, K et F pour chaque problème spécifique. Si le calcul de ces paramètres est possible, alors nous obtenons le corollaire suivant.

Corollaire 12. *Si l'algorithme GMD_0 est utilisé pour une transmission de durée inconnue, avec un feedback scalaire et en utilisant le doubling-trick et les paramètres optimaux δ^* et μ^* dans chacune des fenêtres, alors le regret moyen $\frac{\mathrm{EReg}(T)}{T}$ décroît en $\mathcal{O}(T^{-\frac{1}{4}})$.*

La généralisation de nos algorithmes permet leurs exploitations pour résoudre d'autres problèmes d'allocation de puissances. Comme nous l'avons dit en introduction, pour pouvoir utiliser cette méthode de résolution de problème d'optimisation en ligne, il faut d'abord vérifier que les fonctions objectif et les espaces faisables respectent les hypothèses ($H1$)-($H5$).

Dans la section suivante nous allons présenter une autre application de notre algorithme GMD dans le cas du contrôle d'interférence dans un réseau IoT.

5.4 Exemple : contrôle de l'interférence dans un réseau IoT

Comme exemple illustratif, nous allons nous concentrer sur un problème de maximisation de débit sous contraintes d'interférence maximale dans un réseau IoT. Pour cela, nous allons illustrer uniquement le cas du feedback du gradient parfait.

5.4.1 Présentation du problème

Nous supposons maintenant que l'objet focal souhaite maximiser son débit de communication tout en limitant les interférences créées dans le réseau. De plus nous souhaitons développer des algorithmes décentralisés, nous allons donc nous concentrer sur un objet focal.

Pour la maximisation du débit nous allons utiliser la capacité de Shannon définie par l'équation (2.4). Il faut ensuite définir à quel niveau il faut limiter les interférences du réseau : 1) limiter les interférences créées par chaque objet ; 2) limiter la somme des interférences créées par tous les objets. Limiter les interférences créées par chaque objet peut poser problème car cette limitation ne prend pas en compte le nombre total d'objets dans le réseau. Le second cas permet permet de limiter le niveau d'interférence global dans le réseau. Cependant, cette seconde approche implique de connaître les interférences qui vont être créées par les autres objets à l'instant t. Nous allons donc prendre en compte la limitation totale d'interférence.

Pour cela, nous allons nous concentrer sur un réseau composé d'un récepteur et de M émetteurs (les objets). Comme illustré dans la figure 5.1, chaque objet communique avec le même récepteur et utilise une modulation OFDM avec S sous-porteuses. L'objectif de cet objet focal

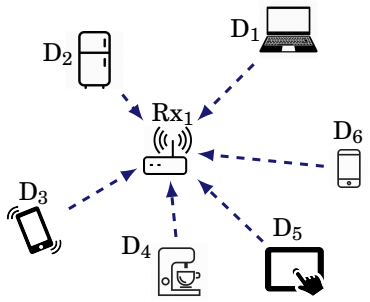


FIGURE 5.1: Système composé de six émetteurs (D_1, D_2, etc.) et d'un récepteur (Rx_1).

est de maximiser son débit, donné par la capacité de Shannon $R_t(\mathbf{p}(t))$ définie pour l'objet focal comme :

$$R_t(\mathbf{p}(t)) = \sum_{s=1}^{S} \log(1 + w^s(t)p^s(t)), \tag{5.32}$$

où $w^s(t)$ est le gain de canal effectif dans la bande s. Ces gains de canaux sont définis par :

$$w^s(t) = \frac{g^s(t)}{\sigma^2 + \sum_j g_j^s(t)p_j^s(t)}, \tag{5.33}$$

où :

— σ^2 est la variance du bruit $z^s(t)$,
— $p_j^s(t)$ est la puissance allouée par l'objet j dans la sous-porteuse s,
— $g^s(t) = |h^s(t)|^2$,
— $g_j^s(t) = |h_j^s(t)|^2$.

Comme dit précédemment, nous allons limiter les interférences totales, pour cela nous définissons une contrainte d'interférence dans chaque sous-porteuse :

$$\sum_{m=1}^{M} g_m^s(t)p_m^s(t) \leq I_{\max}^s, \ \forall \ s \ \in \{1,\ldots,S\}, \tag{5.34}$$

où $g_m^s(t)$ et $p_m^s(t)$ sont respectivement le gain et la puissance de l'objet m dans la sous-porteuse s et $I_{\max}^s$ est la contrainte d'interférence dans la sous-porteuse s. Nous allons introduire ces contraintes d'interférence dans la fonction objectif en utilisant une fonction de pénalité :

$$C(\mathbf{p}) = \lambda \sum_{s=1}^{S} \max\left[\sum_{m=1}^{M} g_m^s p_m^s - I_{\max}^s, 0\right], \tag{5.35}$$

où $\lambda > 0$ est le coefficient de pénalité. Ainsi, la fonction objectif de l'objet focal est :

$$U_t(\mathbf{p}(t)) = R_t(\mathbf{p}(t)) - \lambda \sum_{s=1}^{S} \max\left[\sum_{m=1}^{M} g_m^s(t)p_m^s(t) - I_{\max}^s, 0\right], \tag{5.36}$$

Si les interférences dans le réseau sont plus faibles que $I_{\max}^s$, alors l'objet focal va maximiser uniquement son débit. Cependant, si les interférences sont plus grandes que $I_{\max}^s$ dans au moins

une des sous-porteuses, alors des pénalités seront appliquées, ce qui impliquera une diminution des puissances de transmission et donc du débit.

L'espace faisable quant à lui est le même que dans le Chapitre 2 c'est-à-dire :

$$\mathscr{P} = \left\{ \mathbf{p} \in \mathbb{R}^S | p_s \geq 0, \ \sum_{s=1}^{S} p_s \leq P_{\max} \right\}. \tag{5.37}$$

L'espace faisable étant le même que précédemment, nous savons qu'il respecte déjà les conditions nécessaires pour pouvoir appliquer nos algorithmes ou ($H1$).

Le problème d'optimisation en ligne peut donc être mis sous la forme suivante :

$$\begin{array}{ll} \text{minimiser} & -U_t(\mathbf{p}(t)) \\ \text{sur} & p^s(t) \geq 0, \ \forall s \\ \text{sous contraintes} & \sum_{s=1}^{S} p^s(t) \leq P_{\max}. \end{array} \tag{5.38}$$

Pour pouvoir appliquer l'algorithme GMD il faut étudier la fonction objectif. Nous devons donc vérifier que notre fonction objectif respecte bien les contraintes ($H2$)-($H5$) définies dans la section précédente.

Hypothèse 2 : $-U_t(\mathbf{p})$ est-elle une fonction convexe ? La fonction $U_t(\mathbf{p})$ est la somme d'une fonction concave et d'une fonction linéaire, elle est donc concave et donc $-U_t(\mathbf{p})$ est convexe. L'hypothèse ($H2$) est vérifiée.

Hypothèse 3 : toutes les valeurs de $L_t(\mathbf{p})$ sont-elles bornées ? Nous pouvons facilement montrer que $U_t(\mathbf{p}(t))$ est bornée :

$$U_t(\mathbf{p}(t)) \leq R_t(\mathbf{p}(t)) \leq B = S \log(1 + WP_{\max}), \tag{5.39}$$

dans la mesure où W est la borne supérieure des gains de canaux effectifs, ($H3$) est donc vraie.

Hypothèse 4.a : les variations du gradient de $L_t(\mathbf{p})$ sont-elles bornées. Pour vérifier cette hypothèse nous pouvons montrer que :

$$\|\nabla U_t(\mathbf{p}(t))\|_\infty^2 \leq V^2 = SW\left(\lambda^2 + \frac{1}{\sigma^4}\right), \tag{5.40}$$

où σ^2 est la variance du bruit. Pour parvenir à appliquer nos algorithmes la seule condition est que les gains de canaux, de l'objet focal, soient bornés.

L'hypothèse ($H5$) n'est utile que dans le cas où l'objet a accès au gradient imparfait ou au feedback scalaire. Dans cet exemple nous nous concentrons uniquement sur le cas où l'objet reçoit le gradient parfait comme feedback.

Nous avons vu que l'étape de projection $\mathbf{Q}(\mathbf{y})$ dépend uniquement de l'espace faisable. Puisque l'espace faisable est $\mathscr{P}$, nous pouvons en déduire que la politique d'allocation de puissance dans le cas de la maximisation de débit sous contraintes d'interférences est donnée par :

$$p^s(t+1) = \mathbf{Q}(\mathbf{y}(t)) = P_{\max} \frac{\exp(y^s(t))}{1 + \sum_{j=1}^{S} \exp(y^j(t))}. \tag{5.41}$$

En effet, tout problème d'optimisation en ligne qui a le même espace faisable de l'équation (2.8) aura la même fonction de projection exponentielle.

En utilisant la propriété (5.14) et l'algorithme GMD nous obtenons l'algorithme suivant :

$$\mathbf{y}(t+1) = \mathbf{y}(t) - \mu(t)\mathbf{v}(t) \tag{5.42}$$

$$\mathbf{p}(t+1) = \mathbf{Q}(\mathbf{y}(t)). \tag{5.43}$$

où $\mathbf{Q}(\mathbf{y}(t))$ est défini en (5.41). La différence majeure entre l'algorithme OXL utilisé pour résoudre le problème présenté dans le Chapitre 3 et l'algorithme GMD utilisé pour résoudre le problème de maximisation de débit vient du gradient qui ne sera pas le même dans ces cas.

5.4.2 Propriété de non-regret

Puisque nous appliquons l'algorithme GMD à ce problème nous pouvons en déduire que son regret est borné par la borne du Théorème 5 :

$$\mathrm{Reg}(T) \leq \frac{P_{\max}\log(1+S)}{\mu} + \frac{\mu P_{\max} T V^2}{2}, \tag{5.44}$$

où B est défini par (5.39), V par l'équation (5.40) (pour plus de détails sur les valeurs de $f^*(\mathbf{0})$ et F, voir l'annexe A1). En effet, la fonction de régularisation $f(\cdot)$ dépend uniquement de l'espace faisable. Ce dernier étant $\mathscr{P}$, le même espace que dans le problème du Chapitre 2, nous pouvons ainsi réutiliser la fonction de régularisation $f(\cdot)$ de l'équation (3.29) et donc nous obtenons la même borne du regret.

De cette borne et en utilisant les Corollaires 7 et 8 nous pouvons déduire que la propriété de non-regret est garantie dans les cas où l'objet connaît ou non la durée de transmission à l'avance.

5.4.3 Résultats numériques

Pour illustrer le comportement de l'algorithme GMD dans ce problème de contrôle d'interférences, nous avons réalisé un ensemble de simulations. Nous avons utilisé le modèle de canal COST-HATA avec des paramètres de simulation différents des Chapitres 3 et 4. Ainsi le système utilise une bande de fréquence de 10 Mhz centrée autour d'une porteuse de 2 GHz. Cette bande de fréquence est divisée en 512 sous-porteuses de 19.5 kHz. Dans cet environnement nous considérons un ensemble d'objets, de taille comprise entre 50 et 200, positionnés de manière aléatoire dans une cellule carré de 2km de côté (les objets sont positionnés en utilisant un processus de Poisson). Chaque utilisateur utilise le même algorithme GMD avec des pas μ et λ différents. Les contraintes d'interférences sont les mêmes dans chacune des sous-porteuses et valent -110 dBm. Les objets ont une puissance maximale variable dans un intervalle de 0.5 W à 2 W.

Sur la figure 5.2, nous avons représenté l'évolution de la capacité de Shannon en fonction du temps (une itération vaut 5ms) pour différents objets. La présence de variations rapides et d'écroulements de la capacité de Shannon est due aux contraintes d'interférence. En effet, dès

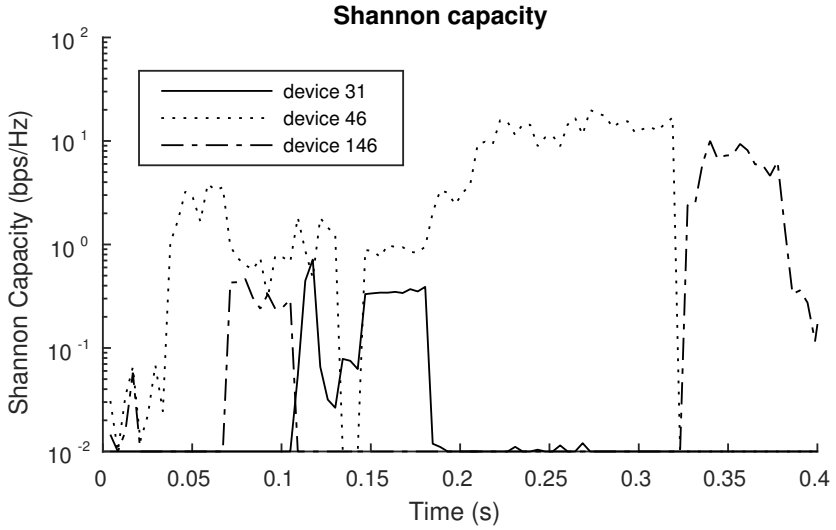


FIGURE 5.2: Évolution de la capacité de Shannon en fonction du temps. Les effondrements de la capacité de Shannon viennent des contraintes d'interférences qui ne sont pas respectées. La capacité de Shannon n'évolue pas de la même façon en fonction des objets car les coefficients de pénalité λ et les gains de canaux ne sont pas les mêmes. Nous pouvons aussi voir que certains équipements ne transmettent pas à certains instants à cause des contraintes d'interférences (c'est le cas de l'objet 31 par exemple).

qu'une contrainte d'interférence n'est pas respectée les objets sont pénalisés et leurs puissances diminuent. Cette diminution des puissances implique une diminution des interférences. Les variations de la capacité de Shannon ne sont pas les mêmes en fonction des objets car ces derniers n'ont pas les mêmes coefficients de pénalité λ et les mêmes gains de canaux.

La figure 5.3, représente l'évolution de l'interférence totale dans différentes sous-porteuses en fonction du temps. La ligne droite pointillée, à -110dBM, représente la valeur d'interférence maximale. Dès que l'interférence dans le réseau a dépassée la contrainte, cette dernière diminue. Cette violation de la contrainte implique une pénalité sur les objets, qui diminueront leurs puissances en conséquence.

La figure 5.4 trace l'évolution du regret moyen pour différents nombres d'objets connectés en fonction du temps. Le regret moyen est calculé en moyennant les regrets de tous les objets. Peut importe le nombre d'objets $M \in \{50, 100, 200\}$, le regret décroît rapidement vers 0. Cela implique qu'indépendamment du nombre d'objets, les politiques d'allocation de puissance dynamique ont une performance au moins aussi bonne que leur meilleure solution fixe calculée à posteriori. Cependant, nous pouvons noter une différence de rapidité dans la convergence du regret moyen en fonction du nombre d'objets.

Finalement, la figure 5.5 nous montre l'évolution (en pourcentage) du nombre de fois où l'objet ne respecte pas au moins une contrainte en fonction du paramètre λ. Si ce pourcentage est égal à 100% cela signifie qu'à chaque itération, il y a des interférences dans au moins une des sous-porteuses. Naturellement, lorsque λ augmente cela implique que nous pénalisons d'avan-

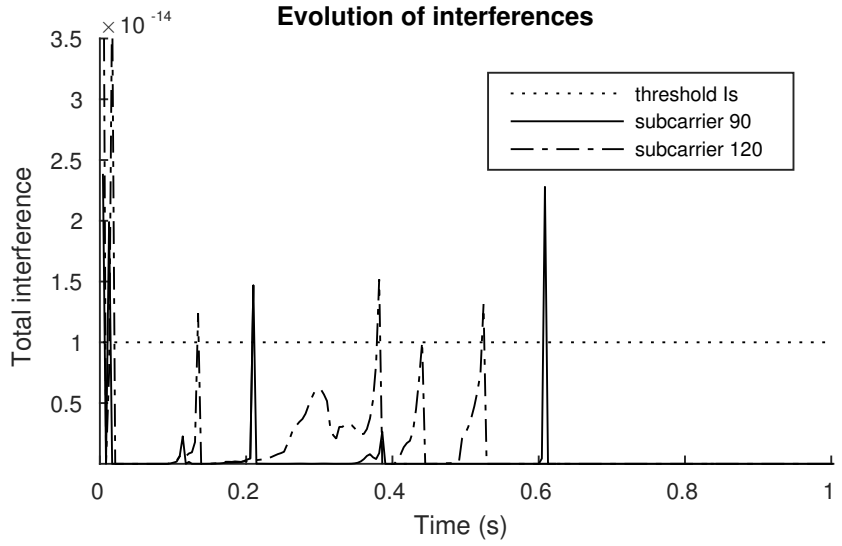


FIGURE 5.3: Évolution des interférences en fonction du temps pour différentes sous-porteuses. La ligne droite en pointillée, à -110dBm, correspond à la contrainte d'interférence maximale. Nous pouvons remarquer que, dès que l'interférence dans une sous-porteuse dépasse la contrainte, des pénalités sont appliquées aux objets ce qui implique la diminution de leur puissance de transmission. Notre algorithme est donc en mesure de contrôler les interférences dans le réseau.

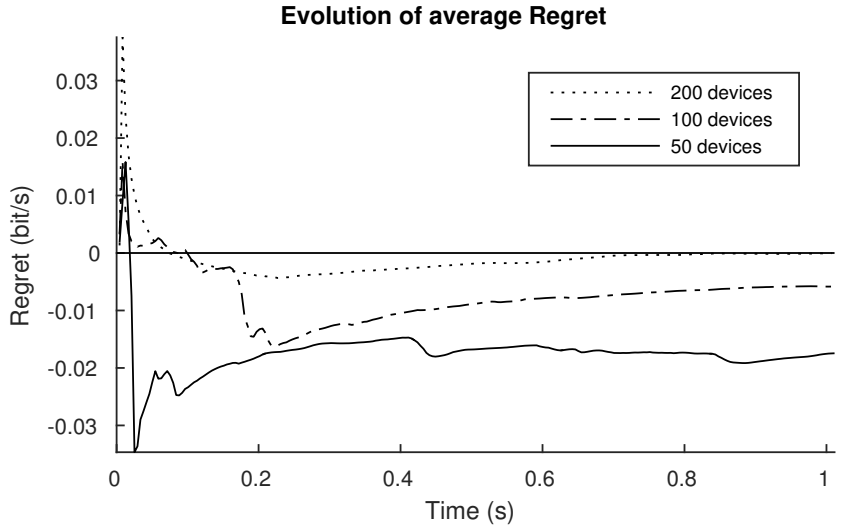


FIGURE 5.4: Cette figure représente l'évolution du regret moyen en fonction du temps pour différents nombres d'objets connectés sur le réseau. Peu importe le nombre d'objets dans le réseau, le regret moyen décroît rapidement. Cela signifie que l'allocation de puissance dynamique des objets donne de meilleurs résultats que leur meilleur solution fixe et ceci de manière relativement rapide (moins de 0.2s).

tage l'objet en cas d'interférences trop grande. Ainsi si λ est grand, le pourcentage des violations diminue et l'objet respecte de plus en plus les contraintes d'interférence. Il faut aussi noter que

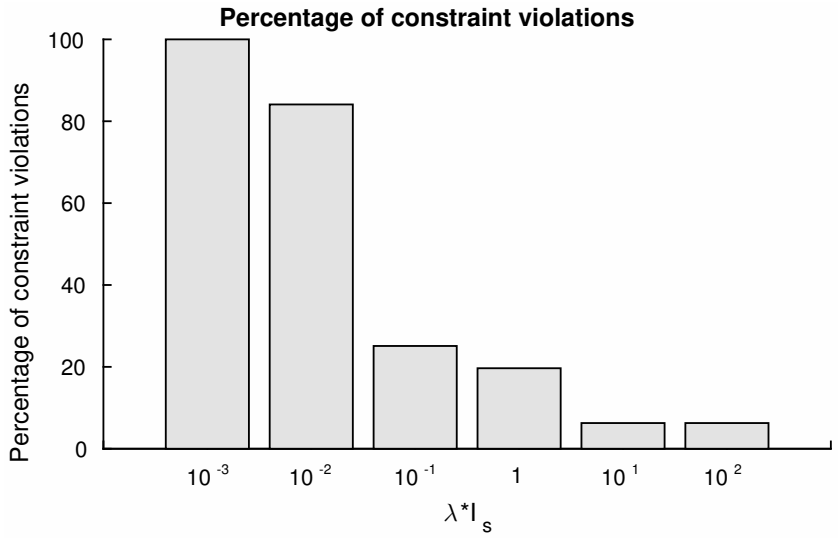


FIGURE 5.5: Pourcentage du temps pendant lequel un objet ne respecte pas une contrainte d'interférence en fonction du paramètre λ. Plus λ est grand plus on pénalise l'objet en cas de violation de la contrainte d'interférence. Naturellement, plus λ est grand moins l'objet crée d'interférence et donc il dépassera moins la contrainte d'interférence.

si l'objet respecte les contraintes d'interférence, sa capacité de Shannon diminue forcément (à cause de la réduction de la puissance). Il y a donc un compromis à faire dans le choix du paramètre λ entre les interférences dans le réseau et le débit de communication.

5.5 Conclusions

Dans ce chapitre, nous avons présenté un problème général d'allocation de ressource pour lequel il est possible de généraliser nos différents algorithmes. Ce modèle général nous a permis de définir clairement les hypothèses que doivent respecter les fonctions objectif ainsi que l'espace faisable. De plus, nous avons présenté les résultats théoriques pour différents types de feedback : 1) gradient parfait ; 2) estimateur non-biaisé du gradient ; 3) estimateur du gradient basé sur un scalaire. Finalement, nous avons illustré l'application de cette méthodologie générale à un nouveau problème : celui du contrôle de l'interférence dans un réseau IoT dynamique et non-prévisible.

CHAPITRE 6

CONCLUSIONS ET PERSPECTIVES

Nous allons conclure ce manuscrit et proposer différentes perspectives et pistes possibles pour des travaux futurs.

6.1 Conclusion

L'objet principal de cette thèse était l'étude et le développement d'algorithmes distribués et adaptatifs pour l'allocation de puissance dans les réseaux IoT qui varient arbitrairement dans le temps. En plus d'être hétérogènes, les réseaux IoT se définissent par leurs dynamiques imprévisibles dues à la mobilité et à la connectivité intermittente des objets. Nous avons vu dans le Chapitre 2 qu'il était difficile d'utiliser les méthodes classiques d'optimisation pour résoudre ce type de problèmes. C'est pourquoi nous avons utilisé des algorithmes d'allocation de puissance dynamiques et adaptatifs exploitant des outils d'optimisation en ligne et des techniques d'apprentissage. Un problème d'optimisation est dit «en ligne» lorsque la fonction objectif varie au cours du temps et n'est pas connue de l'objet à l'instant de décision ; donc, l'objet doit déterminer une allocation de puissance qui minimise une fonction inconnue.

Nous avons vu que la majorité des objets des réseaux IoT sont des objets fonctionnemant sur batterie. L'efficacité énergétique est un critère capital dans la conception d'algorithme d'allocation de puissance. C'est pourquoi, dans cette thèse, nous avons étudié le problème de minimisation de puissance sous contraintes de qualité de service. Ce type de problème permet de minimiser la puissance tout en garantissant un certain débit à l'objet.

De plus, l'augmentation du nombre d'objets connectés au réseau implique nécessairement une augmentation des communications. Afin d'optimiser l'utilisation du réseau il est important de développer des algorithmes qui limitent au maximum le feedback d'informations nécessaires pour déterminer l'allocation de puissance.

Dans le Chapitre 3, nous avons proposé un algorithme d'allocation de puissance dynamique dans le cas où l'objet reçoit le gradient de la fonction objectif par feedback. Nous avons montré que cet algorithme a la propriété de non regret, c'est-à-dire que notre algorithme donne des résultats au moins aussi bons que la meilleure allocation fixe en moyenne. Cette propriété est garantie dans le cas où l'objet connaît en avance la durée de transmission mais aussi dans le cas où cette durée n'est pas connue. Dans ce dernier cas, l'objet doit utiliser le *doubling-trick*, technique qui permet de découper la transmission en une succession de fenêtres de taille connue et qui double à chaque fois.

Comme la réduction du feedback est importante dans le contexte des réseaux IoT, la première étape est d'étudier la situation où l'objet reçoit non pas le gradient parfait mais une estimation erronée de ce dernier. Nous avons montré dans le Chapitre 3 que dans ce cas l'algorithme que nous avons proposé a la probabilité de non regret. Cette propriété est garantie indépendamment de la connaissance ou non de la durée de transmission. Cependant, dans ce cas particulier l'objet ne peut pas appliquer le *doubling-trick* car il nécessite de connaître des informations sur les variations maximales de l'estimateur du gradient. C'est pourquoi, nous proposons l'utilisation d'un pas variable dans le cas où la durée de transmission n'est pas connue en avance. L'utilisation de ce pas variable ralentit un peu la vitesse de décroissance du regret mais ne requière aucune connaissance sur l'évolution de l'estimateur du gradient.

Bien que la quantité d'information soit réduite dans le cas du gradient imparfait, il est quand même nécessaire d'envoyer à l'émetteur un vecteur de taille S (la dimension du problème, i.e., le nombre de sous-porteuses disponibles) comme feedback. Afin de réduire ce feedback, nous avons étudié le cas où l'émetteur reçoit uniquement la valeur de la fonction objectif comme feedback et de se servir de cette valeur pour construire un estimateur du gradient. Cette idée a été présentée dans [Flaxman et al., 2005; Shalev-Shwartz, 2011; Bubeck et al., 2012]. Afin de pouvoir construire cet estimateur il est nécessaire d'échantillonner la fonction objectif non pas dans le point d'intérêt mais dans un nouveau point aléatoirement choisi dans le voisinage du point d'intérêt. Cet ajout d'aléatoire implique la possibilité que l'allocation de puissance utilisée pour transmettre sorte de l'espace faisable, ce qui est impossible dans notre cas. Pour pallier ce problème, nous avons défini un nouvel espace faisable réduit qui garantit que n'importe quelle allocation de puissance de cet espace reste dans l'espace faisable d'origine après l'ajout de l'aléatoire. Grâce à ce nouvel espace faisable, nous avons pu définir un nouvel algorithme qui prend en compte l'estimateur du gradient. Dans le Chapitre 4, nous avons montré que l'algorithme proposé possède la propriété de non regret lorsque la durée de transmission est connue. Dans le cas où la durée de transmission n'est pas connue en avance, nous avons montré que le *doubling-trick* était applicable car les pas optimaux δ et μ ne dépendent que des paramètres du système (la puissance maximale, le nombre de sous-porteuses, le coefficient de pénalité et du débit minimum) connus de l'objet.

Finalement, dans le Chapitre 5 nous proposons une méthodologie de construction des al-

gorithmes d'allocation de puissance afin de résoudre des problèmes plus généraux. Pour cela, nous allons nous appuyer sur les travaux présentés dans [Shalev-Shwartz, 2011]. Une fois le problème d'optimisation générique posé, l'objectif de ce chapitre est de présenter les hypothèses générales que doivent respecter les fonctions objectif et l'espace faisable. Ces contraintes permettent de proposer les algorithmes qui peuvent être utilisés dans les différents cas en fonction du feedback disponible (gradient parfait, gradient imparfait et feedback scalaire). Nous pouvons aussi généraliser les différents résultats théoriques qui offrent des garanties de performance des algorithmes génériques en terme du regret.

6.2 Perspectives

Dans cette section, nous allons présenter différentes perspectives possibles à nos travaux. Nous allons discuter dans un premier temps quelques perspectives à court terme puis dans un second temps quelques perspectives à long terme.

Perspectives à court terme

Comme nous l'avons vu dans les chapitres précédents et plus particulièrement dans le Chapitre 5, les choix du pas de l'algorithme μ et du rayon échantillonnage δ sont très importants pour le bon fonctionnement de nos algorithmes. Les solutions analytiques des paramètres, μ et δ, que nous proposons dans ce manuscrit reposent sur les paramètres du système comme le nombre de sous-porteuses S, la puissance maximale $P_{\max}$ ou encore la constante de Lipschitz des fonctions objectifs. Une petite variation dans l'estimation de cette constante de Lipschitz, entraîne des variations dans la valeur des pas μ et δ ce qui implique une perte d'efficacité de nos algorithmes (c'est à dire une décroissance du regret plus lente). Une évolution possible pour nos travaux consisterait à essayer d'utiliser des algorithmes d'apprentissage, ou d'optimisation en ligne, pour déterminer les pas d'une manière adaptative.

Dans cette thèse, nous avons montré qu'il était possible de définir des algorithmes qui ont la propriété de non regret dans les cas où l'objet a accès à un feedback : vectoriel et scalaire. Il faut cependant noter que dans le cas d'un feedback sous la forme d'un estimateur (le cas de l'estimateur non-biaisé du gradient et le cas du feedback scalaire à partir duquel un estimateur, potentiellement biaisé, est construit) la propriété de non regret n'est pas garantie sur le regret instantané mais sur le regret moyen, calculé par rapport à l'aléatoire des estimateurs. Une des perspectives possibles est de chercher des garanties en probabilité et non des garanties sur le regret moyen. Dit autrement, l'objectif serait de chercher des algorithmes qui garantissent que le regret instantané décroît vers 0 presque sûrement où avec une probabilité 1.

Perspectives à long terme

La notion de regret compare l'allocation de puissance dynamique à la meilleure solution fixe, i.e. la solution fixe qui minimise la somme des fonctions objectif sur l'horizon de transmission. La question qui se pose est de savoir s'il est possible de comparer les performances de nos allocations dynamiques aux meilleures allocations instantanées (i.e. les allocations de puissance qui minimisent les fonctions objectif à chaque instant t) et d'obtenir des résultats en terme du regret dit dynamique. Lorsque nous ne faisons aucune hypothèse quant à l'évolution du réseau, ceci semble trop ambitieux, surtout dans le cas ou la dynamique du réseau est complètement indépendante d'un instant à l'autre. Dans l'autre cas extrême, si le réseau est statique dans le temps, alors il est possible de se comparer à la meilleure allocation instantané car ceci devient équivalent à la notion du regret initiale. Une piste possible est donc de chercher les conditions intermédiaires (entre une variation du réseau complètement aléatoire et indépendante et un réseau statique) sur les variations du réseau afin de garantir des bonnes performances en terme du regret dynamique.

Nous avons vu que la réduction du feedback était une notion importante pour les systèmes IoT. Afin de limiter encore plus l'information transmise par le récepteur, une possibilité est de proposer des algorithmes dynamiques qui ne nécessitent qu'un seul bit de feedback. Ainsi le récepteur ne transmettra qu'une information d'acquittement du message de l'émetteur. Il peut être difficile d'approximer le gradient de la fonction objectif $L_t(\mathbf{p})$, c'est pourquoi une des alternative serait de changer la fonction objectif et de se concentrer par exemple sur la probabilité de coupure (la probabilité que le débit soit plus petit qu'un débit minimum) [Zhang et al., 2016a].

Finalement, la dernière perspective que nous pouvons envisager est l'extension de nos algorithmes à des systèmes de communication dans lesquels les dispositifs sont équipés de multiples antennes ou systèmes MIMO. En effet, les systèmes MIMO sont une technologie prometteuse pour augmenter le débit et la robustesse en profitant de la diversité spatiale offerte par les multiples antennes à l'émission et à la réception [Mertikopoulos and Belmega, 2014]. Bien que le débit ne soit pas une contrainte générale dans les réseaux IoT, il peut être pertinent dans le cas des réseaux IoT cellulaires où les objets connectés doivent cohabiter avec les systèmes cellulaires classiques.

ANNEXE A

GRADIENT PARFAIT

Pour gagner en lisibilité, pour le reste de la preuve nous allons utiliser la notion de regret par rapport à une allocation de puissance fixe $\mathbf{q}$. Ce regret relatif, noté $\mathrm{Reg}_{\mathbf{q}}(T)$, est défini comme :

$$\mathrm{Reg}_{\mathbf{q}} = \sum_{t=1}^{T} L_t(\mathbf{p}(t)) - L_t(\mathbf{q}), \tag{A.1}$$

pour $\mathbf{q} \in \mathscr{P}$ fixe. Il faut noter que le regret relatif à l'allocation fixe $\mathrm{Reg}_{\mathbf{q}}(T)$ est égal au regret $\mathrm{Reg}(T)$ lorsque $\mathbf{q} = \mathbf{q}^*$ où $\mathbf{q}^*$ est la meilleure stratégie fixe qui minimise le coût total sur l'horizon T :

$$\mathbf{q}^* = \arg\min_{\mathbf{q}\in\mathscr{P}} \sum_{t=1}^{T} L_t(\mathbf{q}). \tag{A.2}$$

Maintenant que nous avons défini la notion de regret relatif, nous allons détailler la preuve du Théorème 1.

A.1 Cas du gradient parfait

A.1.1 Preuve du Théorème 1

La première étape dans la preuve du Théorème 1 est d'utiliser la convexité des fonctions objectif $L_t(\mathbf{p})$ pour borner le regret relatif, ce qui nous donne :

$$\mathrm{Reg}_{\mathbf{q}}(T) \leq \langle \nabla L_t(\mathbf{p}(t)) | \mathbf{p}(t) - \mathbf{q} \rangle, \quad \forall \mathbf{q}. \tag{A.3}$$

Nous étudions le cas où le feedback est le gradient parfait, ce qui signifie que : $\mathbf{v}(t) = \nabla L_t(\mathbf{p}(t))$. Nous pouvons donc remplacer le gradient par $\mathbf{v}(t)$ dans l'équation (A.3) et nous obtenons :

$$\mathrm{Reg}_{\mathbf{q}}(T) \leq \sum_{t=1}^{T} \langle \mathbf{v}(t) | \mathbf{p}(t) - \mathbf{q} \rangle, \quad \forall \mathbf{q}. \tag{A.4}$$

Nous utilisons maintenant le score interne et sa définition (3.21), $\mathbf{y}(t+1) = \mathbf{y} - \mu\mathbf{v}(t)$ et $\mathbf{y}(1) = 0$, dans la borne (D.4) pour simplifier la somme des produits scalaires qui concerne l'allocation fixe $\mathbf{q}$:

$$\mathrm{Reg}_{\mathbf{q}}(T) \leq \sum_{t=1}^{T} \langle \mathbf{v}(t)|\mathbf{p}(t)\rangle + \frac{1}{\mu}\langle \mathbf{y}(T+1)|\mathbf{q}\rangle, \quad \forall \mathbf{q}. \tag{A.5}$$

Maintenant que nous avons borné la partie qui concerne l'allocation de puissance fixe $\mathbf{q}$, nous devons nous concentrer sur la somme qui reste (et donc sur l'allocation de puissance dynamique, $\mathbf{p}(t)$). Pour cela nous allons utiliser la fonction de régularisation $f(\mathbf{p})$ et en particulier sa fonction convexe conjuguée $f^*(\mathbf{y})$. Dans notre cas, la fonction de régularisation est définie par :

$$f(\mathbf{p}) = \left(P_{\max} - \sum_{s=1}^{S} p^s\right)\log\left(P_{\max} - \sum_{s=1}^{S} p^s\right) + \sum_{s=1}^{S} p^s \log(p^s). \tag{A.6}$$

Un calcul rapide nous permet de trouver la fonction convexe conjuguée $f^*(\mathbf{y})$ de $f(\mathbf{p})$:

$$f^*(\mathbf{y}) = P_{\max}\log\left(1 + \sum_{s=1}^{S} y^s\right). \tag{A.7}$$

En utilisant le fait que $\mathbf{p}(t) = \nabla f^*(\mathbf{y}(t))$ on définit l'approximation de Taylor d'ordre 2 de la fonction $f^*(\mathbf{y})$ comme :

$$f^*(\mathbf{y}(t+1)) \leq f^*(\mathbf{y}(t)) - \mu\langle \mathbf{v}(t)|\nabla f^*(\mathbf{y}(t))\rangle + \frac{\mu^2}{2}P_{\max}||\mathbf{v}(t)||_\infty^2. \tag{A.8}$$

Nous allons utiliser cette approximation pour borner le produit scalaire entre $\langle \mathbf{p}(t)|\mathbf{v}(t)\rangle$ dans l'équation (A.5), ce qui nous donne :

$$\mathrm{Reg}_{\mathbf{q}} \leq \frac{1}{\mu}\left[f^*(0) - f^*(\mathbf{y}(T+1))\right] + \frac{\mu}{2}P_{\max}\sum_{t=1}^{T}||\mathbf{v}(t)||_\infty^2 + \frac{1}{\mu}\langle \mathbf{y}(T+1)|\mathbf{q}\rangle, \quad \forall \mathbf{q}. \tag{A.9}$$

Nous allons maintenant utiliser l'inégalité de Fenchel [Rockafellar, 2015] qui borne la somme entre $f(\mathbf{p})$ et $f^*(\mathbf{y})$, plus précisément cette inégalité nous donne :

$$f^*(\mathbf{y}) + f(\mathbf{p}) \geq \langle \mathbf{y}|\mathbf{p}\rangle, \ \forall \mathbf{y}, \mathbf{p}. \tag{A.10}$$

Cette inégalité nous permet de remplacer $\langle \mathbf{y}(T+1)|\mathbf{q}\rangle - f^*(\mathbf{y}(T+1))$ par $f(\mathbf{q})$ dans l'équation (A.9), nous utilisons aussi le fait que la norme gradient est borné, $||\mathbf{v}(t)||_\infty^2 \leq V^2$ et nous trouvons :

$$\mathrm{Reg}_{\mathbf{q}} \leq \frac{1}{\mu}\left[f(\mathbf{q}) + P_{\max}\log(1+S)\right] + \frac{\mu}{2}P_{\max}V^2T, \quad \forall \mathbf{q}. \tag{A.11}$$

Il nous reste maintenant à nous occuper de $f(\mathbf{q})$. Nous pouvons montrer que $f(\mathbf{q}) \leq 0$ en utilisant l'inégalité de Jensen ainsi que le changement de variable suivant : $\mathbf{x} = \frac{\mathbf{q}}{P_{\max}}$ dans la définition de la fonction de régularisation (A.6). En utilisant cette propriété de la fonction $f(\mathbf{p})$ et la borne du regret relatif (A.11) nous obtenons la borne du Théorème 1 :

$$\mathrm{Ref}(T) \leq \frac{P_{\max}\log(1+S)}{\mu} + \frac{\mu P_{\max}TV^2}{2}. \tag{A.12}$$

A.1.2 Preuve du Corollaire 1

La borne du regret définie dans l'équation (A.12) dépend des différents paramètres du système comme $P_{\max}$, T ou encore μ. De plus, nous remarquons que lorsque nous calculons la limite du regret moyen pour un paramètre μ quelconque on obtient :

$$\limsup_{T\to\infty} \frac{1}{T}\mathrm{Reg}(T) = \frac{\mu P_{\max} V^2}{2}. \tag{A.13}$$

Cela signifie que la propriété de non regret n'est pas forcement garantie pour un pas μ quelconque.

Pour pallier à ce problème, nous devons déterminer le pas μ optimal, c'est à dire le pas qui minimise la borne du regret. Nous remarquons que la borne du regret est convexe en fonction de μ, il donc possible de calculer la dérivée et de chercher la valeur de μ qui annule cette dérivée. Ce calcul nous donne un pas optimal μ^* :

$$\mu^* = \sqrt{\frac{2\log(1+S)}{TV^2}}. \tag{A.14}$$

Maintenant que nous avons déterminé le pas optimal, nous devons vérifier que ce dernier peut garantir la propriété de non regret. Pour cela, il faut remplacer la valeur de μ par la valeur optimale μ^* dans la borne du regret, ce qui nous donne :

$$\mathrm{Reg}(T) \leq P_{\max}\sqrt{2TV^2\log(1+S)}. \tag{A.15}$$

Nous pouvons déduire de l'équation ci-dessus que la propriété de non regret est bien garantie.

A.1.3 Preuve du Corollaire 2

Dans le cas où la durée de transmission n'est pas connue à l'avance nous allons utiliser l'astuce du *doubling-trick*. Pour cela, nous allons utiliser des fenêtres, $k \in \{0,\dots,\lceil\log_2 T\rceil\}$, de transmission dont la taille, $T_k = 2^k$ double à chaque fois. Puisque l'objet connaît la taille de chaque fenêtre, il peut calculer le pas optimal μ_k^* de chaque fenêtre en utilisant la formule (A.14). De plus, nous pouvons borner le regret dans chaque fenêtre comme :

$$\mathrm{Ref}_k(T) \leq P_{\max}\sqrt{2V^2\log(1+S)2^k}. \tag{A.16}$$

Ainsi on trouve facilement que :

$$\sum_{k=1}^{\lceil\log_2 T\rceil} \mathrm{Reg}_k(T) \leq P_{\max}\sqrt{2V^2\log(1+S)}\sum_{k=1}^{\lceil\log_2 T\rceil} 2^k. \tag{A.17}$$

Il s'agit d'un suite géométrique, nous pouvons donc facilement en calculer la somme :

$$\sum_{k=1}^{\lceil\log_2 T\rceil} \mathrm{Reg}_k(T) \leq P_{\max}\sqrt{2V^2\log(1+S)}\frac{1-\sqrt{2}^{\lceil\log_2 T\rceil+1}}{1-\sqrt{2}}. \tag{A.18}$$

Un calcul rapide nous montre que nous pouvons borner le terme de droite et nous obtenons :

$$P_{\max}\sqrt{2V^2\log(1+S)}\frac{1-\sqrt{2}^{\lceil\log_2 T\rceil+1}}{1-\sqrt{2T}} \leq P_{\max}\sqrt{2V^2\log(1+S)}\frac{1-\sqrt{2}}{1-\sqrt{2}}. \tag{A.19}$$

Et donc en regroupant les équations (A.18) et (A.20) nous trouvons :

$$\sum_{k=1}^{\lceil\log_2 T\rceil} \mathrm{Reg}_k(T) \leq P_{\max}\sqrt{2V^2\log(1+S)}\frac{1-\sqrt{2T}}{1-\sqrt{2}}. \tag{A.20}$$

Ce qui nous donne la borne finale suivante :

$$\mathrm{Reg}(T) \leq P_{\max}\sqrt{2V^2\log(1+S)}\frac{\sqrt{2T}}{\sqrt{2}-1}. \tag{A.21}$$

A.2 Cas du gradient imparfait

Pour facilité la lecture de cette preuve, nous allons utiliser la notion de regret moyen par rapport à l'allocation de puissance fixe $\mathbf{q}$, $\mathrm{EReg}_{\mathbf{q}}$, définie comme :

$$\mathrm{EReg}_{\mathbf{q}}(T) = \mathbb{E}[\mathrm{Reg}_{\mathbf{q}}(T)]. \tag{A.22}$$

Maintenant que nous avons définit le regret moyen relatif à l'allocation de puissance $\mathbf{q}$, nous pouvons détailler la preuve du Théorème 2.

A.2.1 Preuve du Théorème 2

La première étape de la preuve dans le cas du gradient imparfait est la même que dans le cas du gradient parfait. C'est à dire que nous allons utiliser la propriété de convexité des fonctions objectif pour borner le regret moyen relatif à l'allocation de puissance fixe $\mathbf{q}$:

$$\mathrm{EReg}_{\mathbf{q}}(T) \leq \mathbb{E}[\langle \nabla L_t(\mathbf{p}(t)) | \mathbf{p}(t) - \mathbf{q}\rangle], \quad \forall \mathbf{q}. \tag{A.23}$$

L'idée pour la prochaine étape est de lier le gradient des fonctions objectif, $\nabla L_t(\mathbf{p}(t))$, à l'estimateur non-biaisé que l'objet reçoit par feedback, $\tilde{\mathbf{v}}(t)$. Pour cela, nous allons utiliser le fait que le gradient peut s'écrire comme :

$$\nabla L_t(\mathbf{p}(t)) = \mathbb{E}[\tilde{\mathbf{v}}(t) | \tilde{\mathbf{v}}(t-1), \ldots, \tilde{\mathbf{v}}(1)]. \tag{A.24}$$

Nous pouvons donc remplacer le gradient par l'expression définie par l'équation (A.24) dans l'équation (A.23), ce qui nous donne :

$$\mathbb{E}[\langle \nabla L_t(\mathbf{p}(t)) | \mathbf{p}(t) - \mathbf{q}\rangle] = \mathbb{E}[\langle \mathbb{E}[\tilde{\mathbf{v}}(t) | \tilde{\mathbf{v}}(t-1), \ldots, \tilde{\mathbf{v}}(1)] | \mathbf{p}(t) - \mathbf{q}\rangle], \quad \forall \mathbf{q}. \tag{A.25}$$

Á une itération t donnée l'allocation de puissance $\mathbf{p}(t)$ dépend uniquement de la séquence de feedback $\tilde{\mathbf{v}}(t-1), \ldots, \tilde{\mathbf{v}}(1)$ et l'allocation de puissance $\mathbf{q}$ constante, c'est pourquoi nous pouvons écrire :

$$\mathbb{E}[\langle \mathbb{E}[\tilde{\mathbf{v}}(t) | \tilde{\mathbf{v}}(t-1), \ldots, \tilde{\mathbf{v}}(1)] | \mathbf{p}(t) - \mathbf{q}\rangle] = \mathbb{E}[\mathbb{E}[\langle \tilde{\mathbf{v}}(t) | \tilde{\mathbf{v}}(t-1), \ldots, \tilde{\mathbf{v}}(1) | \mathbf{p}(t) - \mathbf{q}\rangle]], \quad \forall \mathbf{q}. \tag{A.26}$$

En utilisant la loi de l'espérance totale nous trouvons :

$$\mathbb{E}[\mathbb{E}[\langle\tilde{\mathbf{v}}(t)|\tilde{\mathbf{v}}(t-1),\ldots,\tilde{\mathbf{v}}(1)|\mathbf{p}(t)-\mathbf{q}\rangle]] = \mathbb{E}[\langle\tilde{\mathbf{v}}(t)|\mathbf{p}(t)-\mathbf{q}\rangle], \quad \forall\mathbf{q}. \tag{A.27}$$

Á partir de cette étape la preuve est la même que dans le cas du gradient parfait. Nous allons utiliser l'approximation de Taylor d'ordre 2 de la fonction $f^*(\mathbf{y})$ ainsi que l'inégalité de Fenchel ce qui nous donne :

$$\mathrm{EReg}_{\mathbf{q}} \leq \mathbb{E}\left[\frac{1}{\mu}\left[f(\mathbf{q})+P_{\max}\log(1+S)\right]+\frac{\mu}{2}P_{\max}\sum_{t=1}^{T}||\tilde{\mathbf{v}}(t)||_{\infty}^{2}\right], \quad \forall\mathbf{q}. \tag{A.28}$$

Les deux termes de l'addition étant indépendants nous pouvons séparer les espérances. Nous profitons de cette étape pour utiliser le fait que la norme de l'estimateur soit bornée et ainsi :

$$\mathrm{EReg}_{\mathbf{q}} \leq \mathbb{E}\left[\frac{1}{\mu}\left[f(\mathbf{q})+P_{\max}\log(1+S)\right]\right]+\mathbb{E}\left[\frac{\mu}{2}P_{\max}T\tilde{V}^{2}\right], \quad \forall\mathbf{q}. \tag{A.29}$$

Les deux termes à l'intérieur des espérances sont des constantes nous pouvons donc les enlever, de plus comme dans la preuve précédente, $f(\mathbf{q}) \leq 0$ ce qui nous donne au final la borne du Théorème 1 :

$$\mathrm{EReg} \leq \frac{1}{\mu}P_{\max}\log(1+S)+\frac{\mu}{2}P_{\max}T\tilde{V}^{2}. \tag{A.30}$$

A.2.2 Preuve du Corollaire 3

Dans le cas où l'objet connaît la durée de transmission à l'avance, nous allons utiliser la même preuve que dans le cas du gradient parfait. La borne du regret de (A.30) est convexe par rapport à μ nous allons donc déterminer la valeur optimale de ce pas. Après avoir calculé et annulé la dérivée (en fonction de μ) de la borne du regret nous trouvons le pas optimal suivant

$$\mu^{*} = \sqrt{\frac{2\log(1+S)}{T\tilde{V}^{2}}}. \tag{A.31}$$

En remplaçant la valeur de μ par la valeur optimale μ^* dans la borne du regret moyen, nous obtenons :

$$\mathrm{EReg}(T) \leq P_{\max}\sqrt{2T\tilde{V}^{2}\log(1+S)}. \tag{A.32}$$

Nous pouvons donc déduire que la propriété de non regret est bien garantie.

A.2.3 Preuve du Corollaire 4

Dans le cas où l'objet ne connaît pas la durée de transmission à l'avance nous allons, dans le cas du gradient bruité, utiliser un pas variable. Ce pas variable est défini comme : $\mu(t) = \frac{\alpha}{\sqrt{t}}$ où α est une constante positive. L'idée de cette preuve est d'étudier un regret pondéré $\mathrm{WEReg}(T)$ défini comme :

$$\mathrm{WEReg}(T) = \mathbb{E}\left[\mu(t)(L_t(\mathbf{p}(t))-L_t(\mathbf{q}))\right]. \tag{A.33}$$

Pour ceci nous allons utiliser la même approche que pour le Théorème 1 ce qui nous donne la borne suivante :

$$\mathrm{WEReg}(T) \leq \frac{P_{\max}\log(1+S)}{\mu} + \frac{\tilde{V}^2 P_{\max} T}{2}\sum_{t=1}^{T}\mu^2(t). \tag{A.34}$$

Pour borner le regret nous allons utiliser le critère de Hardy [Hardy, 1949] qui compare l'évolution d'une suite à sa suite pondérée par un pas variable respectant les conditions suivantes : $\mu(t) \leq \mu(t+1)$ et $\frac{\sum_{t=1}^{T}\mu(t)}{\mu(T)} = \mathcal{O}(T)$. Ainsi si nous sommes en mesure de montrer que $\mathrm{WEReg}(T)$ a la propriété de non regret alors $\mathrm{EReg}(T)$ aura la propriété de non regret. En utilisant le Théorème 14 de [Hardy, 1949] nous trouvons :

$$\frac{\mathbb{E}[\mathrm{Reg}_{\mathbf{q}}(T)]}{T} \sim \frac{\mathrm{WEReg}_{\mathbf{q}}(T)}{\sum_{t=1}^{T}\mu(t)} \tag{A.35}$$

$$\leq \frac{P_{\max}}{\sqrt{T}}\left[\frac{\log(1+S)}{\alpha} + \alpha\tilde{V}^2(1+\log T)\right]. \tag{A.36}$$

$$\mathrm{EReg}(T) \leq \frac{P_{\max}}{\sqrt{T}}\left[\frac{\log(1+S)}{\alpha} + \alpha\tilde{V}^2(1+\log T)\right]. \tag{A.37}$$

ANNEXE B

PARAMÈTRES DES SIMULATIONS

Dans cette annexe nous allons détailler les différents paramètres que nous avons utilisé pour réaliser les simulations (MatLab).

B.1 Paramètres des simulations

Nous avons utilisé le modèle COST-HATA [Pedersen, 1999] pour générer les gains de canaux des objets. Ce modèle permet de générer des gains de canaux réalistes pour des systèmes de communications suburbains et urbains pour des échelles de distance entre 1 km et 3 km. De plus, il prend aussi en compte la mobilité des objets (variant de 0 km/h à 110 km/h). Cependant, la vitesse de chaque objet est considérée constante pour la durée T de la simulation.

Pour toutes les simulations de cette thèse nous avons gardé les mêmes propriétés du réseau : bande de fréquence, fréquence centrale, nombre de sous-porteuses. Les valeurs des ces paramètres sont résumés dans le tableau ci-dessous :

Nombre de sous-porteuses	$S = 4$
Fréquence centrale	$f_c = 2$ GHz
Bande passante	10 MHz
Cellule carré	2 Km de côté

TABLE B.1: Paramètres du réseau.

Remarque B.1. *Par simplicité, nous considérons une cellule carrée de 2 km de côté. Bien que cette taille de cellule semble grande pour un réseau IoT, cela nous permet de prendre en compte la mobilité des objets (véhicules, vélos, etc...) sans avoir à modéliser les changements de cellule d'un objet. De plus ce type de cellule rentre dans le cadre des réseaux cellulaires IoT.*

Les objets sont positionnés de mani ère aléatoire dans la cellule. Pour cela, nous utilisons un tirrage uniforme (valeurs comprisent dans l'intervalle défini par la taille de notre cellule) pour déterminer la position de l'objet selon les axes x et y. Une fois la position déterminée nous atribuons une vitesse à chaque objet. La vitesse de l'objet est comprise entre 0km/h et 110 km/h. Une tirage aléatoire suivant une loi uniforme est utilisé pour déterminer la vitesse de l'objet (un scalaire compris entre 0 et 110 km/h) puis nous utilisons à nouveau une loi uniforme pour déterminer la vitesse selon l'axe x. Une fois la vitesse selon l'axe x determinée, nous pouvons, en utilisant des propriétés géométriques simples, en déduire la vitesse selon l'axe y. Les vitesses selon les axes x et y, nous permettent de définir le vecteur de direction selon lequel l'objet va se déplacer.

Il faut ensuite déterminer la puissance maximale, le débit maximal ainsi que la constante de pénalitée. Pour déterminer ces trois paramètres, nous utilisons encore une fois une une loi uniforme, les ensembles à partir des quels nous tirons les valeurs de ces paramètres sont donnés dans le tableau ci-dessous.

Vitesse des objets	$[0,110]$ Km/h
Puissance maximum	$[0.5,2]$ W
Débit maximum	$[0.5,3]$ bps/Hz
λ	$[0.5,10]$

TABLE B.2: Paramètres des objets.

Finalement, le dernier paramètre à prendre en compte pour la simulation du système est l'activité des objets. Pour paramétrer ces activités nous allons associer un vecteur de taille T à chaque objet autre que l'objet focal. En effet, nous faisons l'hypothèse que l'objet focal transmet pendant toute la durée de la simulation afin de simplifier l'analyse des résultats. Ce vecteur contient une suite de 1 et de 0 tirés en utilisant une loi normale centrée réduite. Ce vecteur binaire indique les moments où l'objet doit transmettre (les 1) et où il ne doit pas transmettre (les 0) et modélise donc le caractères internmittant et imprédicitible de l'activité des objets.

Maintenant que nous avons vu comment le système est généré nous allons détailler les paramètres utilisés par l'objet focal pour l'ensemble des figures de ce manuscrit.

Figure 3.2 et Figure 3.3 : nous utilisons un modèle de gain de canal qui nous permet de contrôler la correlation temporelle entre deux instants de temps. Ce modèle est défini par l'équation suivante :

$$g(t+1) = \alpha g(t) + (1-\alpha)z(t), \tag{B.1}$$

où $z(t) \sim \mathcal{N}(0,\sigma_z^S)$ et avec $\sigma_z^S = 10$, $g(0) = 0$ et $\alpha \geq 0$. Plus de détails sur l'algorithme du water-filling sont donnés dans le Chapitre 3. L'objectif de ces simulations étant de comparer notre algorithme à l'algorithme du water-filling, nous ne placerons qu'un seul objet qui utilise les paramètres suivants : $P_{max} = 2$, $R_{min} = 3$, $\lambda = 4$ et $\mu = 10^{-3}$.

Figure 3.4 et Figure 3.5 : pour ces figures nous avons utilisé le modèle de canal COST-HATA. Les paramètres de l'objet focal sont les suivants : $P_{max} = 2$, $R_{min} = 3$, $\lambda = 1$, $V = 20$ km/h

et $\mu = 10^{-2}$. Le nombre de bandes de fréquence et le nombre d'objets et appartienent respectivement à $\{2,4,8\}$ et $\{10,20,40\}$. Il faut noter que le nombre de bande et fixé à 4 pour la Figure 3.4 et que le nombre d'objets est fixé à 10 pour la Figure 3.5.

Figure 3.6 et Figure 3.7 : pour ces figures nous avons utilisé le modèle de canal COST-HATA. Le nombre d'objets est fixé à 10 et le nombre de bandes à 4. Les paramètres de l'objet focal sont les suivants : $P_{max} = 2,\ R_{min} = 3,\ \lambda = 1,\ V = 20$ km/h et $\mu = 10^{-2}$. L'estimateur du gradient est défini par :

$$\tilde{\mathbf{v}}(t) = \nabla L_t(\mathbf{p}(t)) + z, \tag{B.2}$$

où $z \sim \mathcal{N}(0,\sigma^2)$ avec $\sigma^2 \in \{1,5,10\}$. Les valeurs moyennes du regret sont calculées sur 100 réalisations.

Figure 3.8 : pour cette figure nous avons utilisé le modèle de canal COST-HATA. Le nombre d'objets est fixé à 10 et le nombre de bandes à 4. Les paramètres de l'objet focal sont les suivant : $P_{max} = 1.5,\ R_{min} = 2,\ \lambda = 5,\ V = 30$ km/h et $\mu = 10^{-3}$. Nous utilisons le même modèle de bruit pour l'estimateur que dans les Figures 3.4 et 3.5 avec un $\sigma^2 = 5$. Les valeurs moyennes sont calculées pour 200 réalisations.

Figure 4.2 : pour cette figure nous avons utilisé le modèle de canal COST-HATA. Le nombre d'objets est fixé à 10 et le nombre de bandes à 2. Les paramètres de l'objet focal sont les suivant : $P_{max} = 0.5,\ R_{min} = 1.5,\ \lambda = 7.5,\ V = 70$ km/h et $\mu = 10^{-3}$. Dans le cas du gradient imparfait, nous utilisons le même modèle de bruit pour l'estimateur que dans les Figures 3.4 et 3.5 avec un $\sigma = 5$. Dans le cas de l'estimateur scalaire nous utilisons un rayon d'exploration $\delta = 10^{-2}$. Les valeurs moyennes sont calculées pour 200 réalisations.

Figure 4.3 et Figure 4.4 : pour ces figures nous avons utilisé le modèle de canal COST-HATA. Le nombre d'objets est fixé à 10 et le nombre de bandes est variable : $S \in \{1,2,4\}$. Les paramètres de l'objet focal sont les suivant : $P_{max} = 0.5,\ R_{min} = 1.5,\ \lambda = 7.5,\ V = 70$ km/h et $\mu = 10^{-3}$. Dans le cas de l'estimateur scalaire nous utilisons un rayon d'exploration $\delta = 10^{-2}$. Les valeurs moyennes sont calculées sur 200 réalisations. Pour la Figure 4.4 le nombre de bandes est fixé à 2.

Figure 5.2 et Figure 5.3 : pour ces figures nous avons utilisé le modèle de canal COST-HATA. Le nombre d'objets est fixé à 150 et le nombre de bandes à 200. Les paramètres de l'objet focal sont les suivant : $P_{max} = 1,\ I_s = -110$ dBm, $\forall s,\ \lambda = 100/I_s,\ V = 20$ km/h et $\mu = 10^{-3}$.

Figure 5.4 : pour cette figure nous avons utilisé le modèle de canal COST-HATA. Le nombre d'objets est variable $M \in \{50,100,200\}$ et le nombre de bandes à 200. Les paramètres de l'objet focal sont les suivant : $P_{max} = 1,\ I_s = -110$ dBm, $\forall s,\ \lambda = 100/I_s,\ V = 20$ km/h et $\mu = 10^{-3}$.

Figure 5.5 : pour cette figure nous avons utilisé le modèle de canal COST-HATA. Le nombre d'objets est fixé à 200 et le nombre de bandes à 200. Les paramètres de l'objet focal sont les suivant : $P_{max} = 1,\ I_s = -110$ dBm, $\forall s,\ V = 20$ km/h et $\mu = 10^{-3}$. La constante de pénalité λ est variable : $\lambda * I_s \in \{10^{-3},10^{-2},10^{-1},1,10,10^2\}$.

ANNEXE C

PREUVES RELATIVES AU CHAPITRE 4

C.1 Propriétés de l'estimateur du gradient

L'objectif de l'annexe est de déterminer un estimateur du gradient $\nabla L_t(\mathbf{p}(t))$ en utilisant uniquement la valeur de la fonction objectif $L_t(\mathbf{p}(t))$. Pour cela, nous allons dans un premier temps définir la fonction $\widetilde{L}_t(\mathbf{p}(t))$ qui est une fonction d'approximation de $L_t(\mathbf{p}(t))$. L'avantage de cette fonction $\widetilde{L}_t(\mathbf{p}(t))$ est que nous pouvons calculer un estimateur non biaisé de son gradient en fonction de $L_t(\mathbf{p}(t))$. Nous allons dans cette section détailler les propriétés liées à cette fonction d'approximation.

Commençons par définir la fonction $\widetilde{L}_t(\mathbf{p}(t))$:

Definition C.1. La fonction d'approximation $\widetilde{L}_t(\mathbf{p}(t))$ de $L_t(\mathbf{p}(t))$ est définie pour tout t comme :

$$\widetilde{L}_t(\mathbf{p}(t)) \triangleq \int_{\mathbb{B}} \frac{L_t(\mathbf{p}(t)+\delta\mathbf{u})}{\text{vol}(\mathbb{B})} dV(\mathbf{u}), \tag{C.1}$$

où $\mathbb{B} = \{\mathbf{x} \in \mathbb{R}^S, \|\mathbf{x}\| \leq 1\}$ est la boule unitaire Euclidienne et $dV(\mathbf{u}) = du_1, \ldots, du_S$ est le vecteur différentiel de dimension S.

La fonction $\widetilde{L}_t(\mathbf{p}(t))$ étant construite en utilisant $L_t(\mathbf{p}(t))$, nous pouvons montrer facilement que si $L_t(\mathbf{p}(t))$ est convexe alors $\widetilde{L}_t(\mathbf{p}(t))$ est aussi convexe. Pour montrer cela il suffit d'écrire l'expression de $L_t(\mathbf{p}(t))$ au point $\alpha\mathbf{p}_1 + (1-\alpha)\mathbf{p}_2$ et utiliser la propriété de convexité de la fonction $L_t(\mathbf{p}(t))$.

La prochaine étape consiste à déterminer le biais qui existe entre les fonctions $L_t(\mathbf{p}(t))$ et $\widetilde{L}_t(\mathbf{p}(t))$. Pour cela, nous allons utiliser le fait que les fonctions objectif sont K-Lipschitz ce qui se traduit par :

$$|L_t(\mathbf{p}) - L_t(\mathbf{p}+\delta\mathbf{u})| \leq K\delta \|\mathbf{u}\|_2, \quad \forall \mathbf{p}, \ \forall \mathbf{u}, \tag{C.2}$$

où $\mathbf{u} \in \mathbb{B}$. Puisque $\mathbf{u}$ appartient à la boule unité, nous savons que $\|\mathbf{u}\|_2 \leq 1$, en divisant par vol($\mathbb{B}$) et en prenant l'intégrale sur $\mathbb{B}$ de chaque côté de l'équation (C.2) nous obtenons la propriété suivante :

Proposition C.1. *Si la fonction $\widetilde{L}_t(\boldsymbol{p})$ est une approximation de la fonction K-Lipschitz $L_t(\boldsymbol{p})$ comme définie en* (C.1)*, alors le biais entre ces deux fonctions est borné par :*

$$|L_t(\boldsymbol{p}) - \widetilde{L}_t(\boldsymbol{p})| \leq K\delta, \ \forall \boldsymbol{p}. \tag{C.3}$$

La prochaine étape consiste à montrer qu'il est possible de créer un estimateur non biaisé du gradient de la fonction $\widetilde{L}_t(\mathbf{p})$ en fonction de la valeur de $L_t(\mathbf{p})$. La Proposition C.2 permet de lier le gradient de la fonction d'approximation $\nabla\widetilde{L}_t(\mathbf{p})$ à la valeur de la fonction objectif $L_t(\mathbf{p}+\delta\mathbf{u})$, où $\mathbf{u} \in \mathbb{S} \triangleq \{\mathbf{x} \in \mathbb{R}^S, \|\mathbf{x}\|_2 = 1\}$.

Proposition C.2. *Si la fonction $\widetilde{L}_t(\boldsymbol{p})$ est une approximation de la fonction K-Lipschitz $L_t(\boldsymbol{p})$ comme définie en* (C.1)*, alors $\frac{S}{\delta}L_t(\boldsymbol{p}+\delta\boldsymbol{u})\boldsymbol{u}$ où $\mathbb{S} = \{v \in \mathbb{R}^S, \|v\|_2 = 1\}$ est un estimateur non biaisé du gradient $\nabla\widetilde{L}_t(\boldsymbol{p})$:*

$$\nabla\widetilde{L}_t(\boldsymbol{p}) = \frac{S}{\delta}\int_{\mathbb{S}} \frac{L_t(\boldsymbol{p}+\delta\boldsymbol{u})}{\text{surf}(\mathbb{S})}\boldsymbol{n}(\boldsymbol{u})dA(\boldsymbol{u}), \tag{C.4}$$

où $\boldsymbol{n}(\boldsymbol{u})$ est le vecteur normal au point $\boldsymbol{u}$ sur la sphère unitaire $\mathbb{S}$, surf($\mathbb{S}$) *est la surface de la sphère unitaire de dimension S et $dA(\boldsymbol{u})$ est la dérivé infinitésimale de la surface.*

Pour simplifier la compréhension de cette preuve, nous allons dans un premier temps nous concentrer sur le cas particulier où la dimension du problème est $S = 1$. En faisant le changement de variable suivant : $\tau = \delta u$ dans la Définition C.1, on obtient :

$$\widetilde{L}_t(p) = \frac{1}{2\delta}\int_{-\delta}^{\delta} L_t(p+\tau)d\tau. \tag{C.5}$$

Nous prenons ensuite la dérivée en fonction de p :

$$\frac{d\widetilde{L}_t(p)}{dp} = \frac{1}{2\delta}\frac{d}{dp}\int_{-\delta}^{\delta} L_t(p+\tau)d\tau. \tag{C.6}$$

L'intégrale est calculée par rapport à τ et la dérivée est prise par rapport à p, c'est pourquoi nous pouvons déplacer la dérivé à l'intérieur de l'intégrale

$$\frac{d\widetilde{L}_t(p)}{dp} = \frac{1}{2\delta}\int_{-\delta}^{\delta} \frac{dL_t(p+t)}{dp}d\tau = \frac{L_t(p+\delta) - L_t(p-\delta)}{2\delta}. \tag{C.7}$$

Maintenant, nous observons que :

$$\sum_{v\in\{-1,+1\}} L_t(p+\delta v)v = L_t(p+\delta) - L_t(p-\delta). \tag{C.8}$$

Des équations (C.7) et (C.8), nous pouvons déduire que :

$$\frac{d\widetilde{L}_t(p)}{dp} = \frac{\sum_{v\in\{-1,+1\}} L_t(p+\delta v)v}{\delta}. \tag{C.9}$$

Nous avons donc montré que la proposition (C.2) est vraie pour $S = 1$.

Maintenant, nous allons nous concentrer sur le cas général $S \geq 1$. Premièrement, nous allons rappeler l'équation du volume d'une boule de rayon δ, noté $\text{vol}(\delta\mathbb{B})$:

$$\text{vol}(\delta\mathbb{B}) \triangleq \frac{\pi^{S/2}\delta^S}{\Gamma(\frac{S}{2}+1)}, \tag{C.10}$$

où $\Gamma(\cdot)$ est la fonction gamma. En faisant le changement de variable suivant $\mathbf{w} = \delta\mathbf{u}$ dans la Définition C.1, nous obtenons :

$$\widetilde{L}_t(\mathbf{p}) = \frac{\int_{\delta\mathbb{B}} L_t(\mathbf{p}+\mathbf{w})}{\delta^S \,\text{vol}(\mathbb{B})} dV(\mathbf{w}) \tag{C.11}$$

dû au fait que $dV(\mathbf{u}) = \frac{dV(\mathbf{w})}{\delta^S}$. Nous allons maintenant calculer le gradient par rapport à $\mathbf{p}$ et on remarque, de l'équation (C.10), que $\delta^S \,\text{vol}(\mathbb{B}) = \text{vol}(\delta\mathbb{B})$:

$$\nabla\widetilde{L}_t(\mathbf{p}) = \nabla \frac{\int_{\delta\mathbb{B}} L_t(\mathbf{p}+\mathbf{w})}{\text{vol}(\delta\mathbb{B})} dV(\mathbf{w}). \tag{C.12}$$

L'intégrale est calculée par rapport à $\mathbf{w}$ et le gradient est calculé par rapport à $\mathbf{p}$, nous pouvons donc déplacer le gradient à l'intérieur de l'intégrale :

$$\nabla\widetilde{L}_t(\mathbf{p}) = \frac{\int_{\delta\mathbb{B}} \nabla L_t(\mathbf{p}+\mathbf{w})}{\text{vol}(\delta\mathbb{B})} dV(\mathbf{w}). \tag{C.13}$$

Le Théorème de Stokes nous dit que :

$$\int_{\delta\mathbb{B}} \nabla L_t(\mathbf{p}+\mathbf{w}) dV(\mathbf{w}) = \int_{\delta\mathbb{S}} L_t(\mathbf{p}+\mathbf{w})\mathbf{n}(\mathbf{w})\, dA(\mathbf{w}) \tag{C.14}$$

où le vecteur $\mathbf{n}(\mathbf{w})$ est le vecteur normale au point $\mathbf{w}$ de la sphère $\delta\mathbb{S}$ de rayon δ, et $dA(\mathbf{w})$ est la dérivé infinitésimale de la surface. Si nous appliquons ce Théorème dans notre cas, alors nous obtenons :

$$\nabla\widetilde{L}_t(\mathbf{p}) = \frac{\int_{\delta\mathbb{S}} L_t(\mathbf{p}+\mathbf{w})\mathbf{n}(\mathbf{w})}{\text{vol}(\delta\mathbb{B})}\, dA(\mathbf{w}). \tag{C.15}$$

Vu que $\mathbf{w} = \delta\mathbf{u}$ et $\mathbf{u}$ appartient à la sphère Euclidienne $\mathbb{S}$, nous trouvons :

$$\mathbf{n}(\mathbf{w}) = \frac{\mathbf{w}}{\|\mathbf{w}\|} = \frac{\mathbf{w}}{\delta}. \tag{C.16}$$

Si on substitue $\mathbf{n}(\mathbf{w})$ par la définition précédente et que nous utilisons la propriété du volume suivante :

$$\text{vol}(\delta\mathbb{B}) \triangleq \frac{\delta}{S}\,\text{surf}(\delta\mathbb{S}), \tag{C.17}$$

qui vient de la définition de l'aire d'une sphère de dimension S et de rayon δ :

$$\text{surf}(\delta\mathbb{S}) \triangleq \frac{\pi^{S/2} S\delta^{S-1}}{\Gamma(\frac{S}{2}+1)}, \tag{C.18}$$

alors nous obtenons :

$$\nabla\widetilde{L}_t(\mathbf{p}) = \frac{S}{\delta}\frac{\int_{\delta\mathbb{S}} L_t(\mathbf{p}+\mathbf{w})\mathbf{w}}{\text{surf}(\delta\mathbb{S})}\, dA(\mathbf{w}). \tag{C.19}$$

Maintenant, nous allons utiliser le changement de variable suivant $\mathbf{w} = \delta\mathbf{u}$ et le fait que $\mathrm{dA}(\mathbf{w}) = \delta^{S-1}\mathrm{dA}(\mathbf{u})$, ce qui nous permet d'obtenir :

$$\nabla\widetilde{L}_t(\mathbf{p}) = \frac{S\delta^{S-1}}{\delta}\frac{\int_{\mathbb{S}} L_t(\mathbf{p}+\delta\mathbf{u})\mathbf{u}}{\mathrm{surf}(\delta\mathbb{S})}\, dA(\mathbf{u}). \tag{C.20}$$

De plus, nous observons que :

$$\frac{\delta^{S-1}}{\mathrm{surf}(\delta\mathbb{S})} = \frac{1}{\mathrm{surf}(\mathbb{S})}, \tag{C.21}$$

ce qui nous permet d'écrire :

$$\nabla\widetilde{L}_t(\mathbf{p}) = \frac{S}{\delta}\frac{\int_{\mathbb{S}} L_t(\mathbf{p}+\delta\mathbf{u})\mathbf{u}}{\mathrm{surf}(\mathbb{S})}\, dA(\mathbf{u}). \tag{C.22}$$

Et ainsi la Propriété C.2 est vraie pour tout $S \geq 1$.

Maintenant que nous avons défini notre estimateur du gradient ainsi que ses propriétés, nous allons présenter les preuves des différents résultats théoriques du Chapitre 4.

C.2 Preuves des résultats théoriques de l'Algorithme GMD

Dans cette section nous allons présenter les preuves des résultats du Chapitre 4. C'est à dire les preuves du Théorème 3 et des Corollaires 5 et 6.

C.2.1 Preuve du Théorème 3

La preuve du Théorème 3 repose sur les étapes suivantes :

1. $\mathrm{EReg}_{\mathbf{q}}(T) \leq \mathbb{E}\left[\sum_{t=1}^{T} L_t(\mathbf{p}_\delta(t)) - L_t(\mathbf{q}_\delta)\right] 3KT\delta + KT\delta(2\sqrt{S}+S)$ dans le Lemme C.1

3. $\mathbb{E}\left[\sum_{t=1}^{T} \widetilde{L}_t(\mathbf{p}_\delta(t)) - \widetilde{L}_t(\mathbf{q}_\delta)\right] \leq \mathbb{E}\left[\sum_{t=1}^{T} \langle\tilde{\mathbf{v}}(t)|\mathbf{p}_\delta(t) - \mathbf{q}_\delta\rangle\right]$ dans le Lemme C.2

4. $\sum_{t=1}^{T} \langle\tilde{\mathbf{v}}(t)|\mathbf{p}_\delta(t) - \mathbf{q}_\delta\rangle \leq \sum_{t=1}^{T} \langle\tilde{\mathbf{v}}(t)|\mathbf{p}_\delta(t) - \mathbf{p}_\delta(t+1)\rangle + \frac{P_{\max}\log(1+S)}{\mu}$ dans le Lemme C.3

5. $\sum_{t=1}^{T} \langle\tilde{\mathbf{v}}(t)|\mathbf{p}_\delta(t) - \mathbf{p}_\delta(t+1)\rangle \leq \mu P_{\max}\sum_{t=1}^{T} \|\tilde{\mathbf{v}}(t)\|^2$ dans le Lemme C.4

6. $\mathbb{E}\left[\sum_{t=1}^{T} \|\tilde{\mathbf{v}}(t)\|^2\right] \leq TS^2\left(\frac{S}{\delta}+L\right)^2$ dans le Lemme C.5

7. En combinant tous les lemmes précédents nous trouvons :
   $\mathrm{EReg}_{\mathbf{q}}(T) \leq \frac{P_{\max}\log(1+S)}{\mu} + \mu P_{\max} TS^2\left(\frac{B}{\delta}+L\right)^2 + 3KT\delta + KT\delta(2\sqrt{S}+S)$

Vu que l'allocation de puissance est aléatoire, l'objet connaît uniquement $\tilde{\mathbf{p}}(t) = \mathbf{p}_\delta(t) + \delta\mathbf{u}(t)$. La première étape consiste donc à relier la valeur de la fonction objectif aux points $\tilde{\mathbf{p}}(t)$, $\mathbf{q} \in \mathscr{P}$ et les valeurs prisent aux points $\mathbf{p}_\delta$, $\mathbf{q}_\delta \in \mathscr{P}_\delta$ où $\mathbf{q}_\delta$ est calculée en utilisant la fonction $\Delta(\mathbf{q}) : \mathscr{P}_\delta \rightarrow \mathscr{P}$ définie comme :

$$\Delta(\mathbf{q}) = \delta + \left(1 - \frac{\delta}{P_{\max}}(S + \sqrt{S})\right)\mathbf{q}. \tag{C.23}$$

Lemme C.1. *Si l'algorithme OXL$_0$ est utilisé avec des fonctions K-Lipschitz et avec $\boldsymbol{p}(t) = \boldsymbol{p}_\delta(t) + \delta\boldsymbol{u}(t)$ alors le regret moyen est borné par :*

$$\mathrm{EReg}_{\boldsymbol{q}}(T) \leq \mathbb{E}\left[\sum_{t=1}^{T} L_t(\boldsymbol{p}_\delta(t)) - L_t(\boldsymbol{q}_\delta)\right] + 3KT\delta + KT\delta(2\sqrt{S} + S). \tag{C.24}$$

Démonstration. Le regret moyen est calculé en fonction de l'allocation de puissance $\tilde{\mathbf{p}}(t) = \mathbf{p}_\delta(t) + \delta\mathbf{u}(t)$ et il peut s'écrire :

$$\mathrm{EReg}_{\mathbf{q}}(T) = \mathbb{E}\left[\sum_{t=1}^{T} L_t(\tilde{\mathbf{p}}) - L_t(\mathbf{q}).\right] \tag{C.25}$$

Nous pouvons réécrit le regret comme :

$$\mathrm{EReg}_{\mathbf{q}}(T) = \mathbb{E}\left[\sum_{t=1}^{T} L_t(\tilde{\mathbf{p}}(t)) - L_t(\mathbf{p}_\delta(t)) + L_t(\mathbf{p}_\delta(t)) - L_t(\mathbf{q})\right]. \tag{C.26}$$

La première étape est de comparer $L_t(\mathbf{p}_\delta(t))$ à $L_t(\mathbf{p}_\delta(t))$. Pour faire cela rappelons tout d'abord que $\tilde{\mathbf{p}}(t) = \mathbf{p}_\delta(t) + \delta\mathbf{u}(t)$ où $\mathbf{u}(t) \in \mathbb{S}$. En utilisant le fait que L_t est K-Lipschitz alors nous obtenons :

$$|L_t(\tilde{\mathbf{p}}(t)) - L_t(\mathbf{p}_\delta(t))| = |L_t(\mathbf{p}_\delta(t) + \delta\mathbf{u}(t)) - L_t(\mathbf{p}_\delta(t)| \leq K\delta \left\|\mathbf{u}(t)\right\|_2, \tag{C.27}$$

car les fonctions objectif sont toujours positives et que $\|\mathbf{u}(t)\|_2 = 1$ du au fait que $\mathbf{u}(t) \in \mathbb{S}$. En combinant (C.26) et (C.27) nous trouvons :

$$\mathrm{EReg}_{\mathbf{q}}(t) \leq \mathbb{E}\left[\sum_{t=1}^{T} L_t(\mathbf{p}_\delta(t)) - L_t(\mathbf{q})\right] + KT\delta. \tag{C.28}$$

Maintenant nous avons à comparer $L_t(\mathbf{q})$ à $L_t(\mathbf{q}_\delta)$. Nous remarquons que :

$$\mathbb{E}\left[\sum_{t=1}^{T} L_t(\mathbf{p}_\delta(t)) - L_t(\mathbf{q})\right] = \mathbb{E}\left[\sum_{t=1}^{T} L_t(\mathbf{p}_\delta(t)) + L_t(\mathbf{q}_\delta(t)) - L_t(\mathbf{q}) - L_t(\mathbf{q}_\delta(t))\right]. \tag{C.29}$$

où $\mathbf{q}_\delta$ est calculée en utilisant la fonction $\Delta(\mathbf{q}) : \mathscr{P}_\delta \rightarrow \mathscr{P}$ définie comme :

$$\Delta(\mathbf{q}) = \delta + \left(1 - \frac{\delta}{P_{\max}}(S + \sqrt{S})\right)\mathbf{q}. \tag{C.30}$$

En utilisant le fait que $\mathbf{q}_\delta = \Delta(\mathbf{q})$ et que $L_t(\mathbf{p})$ est K- Lipschitz nous obtenons :

$$|L_t(\mathbf{q}_\delta) - L_t(\mathbf{q})| = |L_t(\Delta(\mathbf{q})) - L_t(\mathbf{q})| \leq K \left\|\Delta(\mathbf{q}) - \mathbf{q}\right\|_2, \quad \forall \mathbf{q} \in \mathscr{P}. \tag{C.31}$$

Nous pouvons borner le terme $\|\Delta(\mathbf{q})-\mathbf{q}\|_2$ comme suit

$$\|\Delta(\mathbf{q})-\mathbf{q}\|_2 = \left\|\mathbf{q}(1-\frac{\delta}{P_{\max}}(\sqrt{S}+S))+\delta\mathbb{1}_S-\mathbf{q}\right\|_2 \leq \frac{\delta}{P_{\max}}(\sqrt{S}+S)\|\mathbf{q}\|_2+\delta\|\mathbb{1}_S\|_2, \tag{C.32}$$

où $\mathbb{1}_S$ est le vecteur de taille S composé uniquement de 1. Nous savons que $\|\mathbf{q}\|_2 \leq P_{\max}$ et que $\|\mathbb{1}_S\|_2 = \sqrt{S}$, en remplaçant tout cela par (D.42) nous trouvons

$$|L_t(\mathbf{q}_\delta)-L_t(\mathbf{q})| \leq K\delta(2\sqrt{S}+S). \tag{C.33}$$

En combinant (C.29) et (C.33) nous obtenons :

$$\mathbb{E}\left[\sum_{t=1}^{T} L_t(\mathbf{p}_\delta(t))-L_t(\mathbf{q})\right] \leq \mathbb{E}\left[\sum_{t=1}^{T} L_t(\mathbf{p}_\delta(t))-L_t(\mathbf{q}_\delta)\right]+KT\delta(2\sqrt{S}+S). \tag{C.34}$$

Maintenant nous devons comparer $L_t(\mathbf{p}_\delta(t))$ et $L_t(\mathbf{q}_\delta)$ à $\widetilde{L}_t(\mathbf{p}_\delta(t))$ et $\widetilde{L}_t(\mathbf{q}_\delta)$ respectivement. En effet, nous ne connaissons pas le gradient de $L_t(\mathbf{p}(t))$ mais de la Proposition C.2 nous connaissons un estimateur de $\nabla\widetilde{L}_t(\mathbf{p}(t)) = \frac{S}{\delta}L_t(\mathbf{p}_\delta(t))\mathbf{u}(t)$.

L'idée est de comparer $L_t(\mathbf{p}_\delta(t))$ à $\widetilde{L}_t(\mathbf{p}_\delta(t))$:

$$\mathbb{E}\left[\sum_{t=1}^{T} L_t(\mathbf{p}_\delta(t))-L_t(\mathbf{q}_\delta)\right] = \mathbb{E}\left[\sum_{t=1}^{T} L_t(\mathbf{p}_\delta(t))-\widetilde{L}_t(\mathbf{p}_\delta(t))+\tilde{L}_t(\mathbf{p}_\delta(t))-L_t(\mathbf{q}_\delta)\right] \tag{C.35}$$

La Proposition C.1 implique que :

$$|L_t(\mathbf{p}_\delta(t))-\widetilde{L}_t(\mathbf{p}_\delta(t))| \leq K\delta. \tag{C.36}$$

Vu que L_t sont toujours positives nous pouvons combiner (C.35) et (C.36) et nous obtenons :

$$\mathbb{E}\left[\sum_{t=1}^{T} L_t(\mathbf{p}_\delta(t))-L_t(\mathbf{q}_\delta)\right] \leq \mathbb{E}\left[\sum_{t=1}^{T} \widetilde{L}_t(\mathbf{p}_\delta(t))-L_t(\mathbf{q}_\delta)\right]+KT\delta. \tag{C.37}$$

Nous devons aussi comparer $L_t(\mathbf{q}_\delta)$ à $\widetilde{L}_t(\mathbf{q}_\delta)$:

$$\mathbb{E}\left[\sum_{t=1}^{T} \widetilde{L}_t(\mathbf{p}_\delta(t))-L_t(\mathbf{q}_\delta))\right] = \mathbb{E}\left[\sum_{t=1}^{T} \widetilde{L}_t(\mathbf{p}_\delta(t))+\widetilde{L}_t(\mathbf{q}_\delta)-L_t(\mathbf{q}_\delta))-\widetilde{L}_t(\mathbf{q}_\delta)\right] \tag{C.38}$$

En utilisant la Proposition C.1 nous trouvons :

$$|\widetilde{L}_t(\mathbf{q}_\delta)-L_t(\mathbf{p}_\delta)| \leq K\delta. \tag{C.39}$$

En combinant (C.38) et (C.39) nous trouvons :

$$\mathbb{E}\left[\sum_{t=1}^{T} \widetilde{L}_t(\mathbf{p}_\delta(t))-L_t(\mathbf{q}_\delta)\right] \leq \mathbb{E}\left[\sum_{t=1}^{T} \widetilde{L}_t(\mathbf{p}_\delta(t))-\widetilde{L}_t(\mathbf{q}_\delta)\right]+KT\delta. \tag{C.40}$$

Le Lemme C.1 suit en combinant (C.28), (C.33), (C.37) et (C.40). □

Le Lemme C.1 relie le regret moyen à l'espérance de la différence cumulée de $\widetilde{L}_t(\mathbf{p}_\delta(t))$ et $\widetilde{L}_t(\mathbf{q}_\delta)$. Nous avons montré que si $L_t(\mathbf{p}(t))$ est convexe par rapport à $\mathbf{p}(t)$ alors la fonction $\widetilde{L}_t(\mathbf{p}(t))$ est convexe par rapport à $\mathbf{p}(t)$ ainsi nous pouvons linéariser la borne du Lemme C.1 en utilisant le gradient de $\widetilde{L}_t(\mathbf{p}(t))$. Nous utilisons ainsi la Proposition C.2 pour remplacer le gradient de $\widetilde{L}_t(\mathbf{p}(t))$ par son estimateur $\tilde{\mathbf{v}}(t) = \frac{S}{\delta} L_t(\mathbf{p}(t) + \delta \mathbf{u}(t))\mathbf{u}(t)$.

Lemme C.2. *Si la fonction $\widetilde{L}_t(\boldsymbol{p}_\delta(t))$ est convexe par rapport à $\boldsymbol{p}_\delta(t)$ et si l'estimateur $\tilde{\boldsymbol{v}}(t)$ est défini par C.2 alors nous obtenons :*

$$\mathbb{E}\left[\widetilde{L}_t(\boldsymbol{p}_\delta(t)) - \widetilde{L}_t(\boldsymbol{q}_\delta)\right] \leq \mathbb{E}\left[\sum_{t=1}^{T} \langle \tilde{\boldsymbol{v}}(t) | \boldsymbol{p}_\delta(t) - \boldsymbol{q}_\delta \rangle\right]. \tag{C.41}$$

Démonstration. Vu que la fonction $\widetilde{L}_t(\mathbf{p}_\delta(t))$ est convexe nous pouvons écrire :

$$\mathbb{E}\left[\sum_{t=1}^{T} \widetilde{L}_t(\mathbf{p}_\delta(t)) - \widetilde{L}_t(\mathbf{q}_\delta)\right] \leq \mathbb{E}\left[\sum_{t=1}^{T} \langle \nabla \widetilde{L}_t(\mathbf{p}_\delta(t)) | \mathbf{p}_\delta(t) - \mathbf{q}_\delta \rangle\right] \quad \forall \mathbf{p}_\delta(t), \mathbf{q}_\delta \in \mathscr{P}_\delta. \tag{C.42}$$

Nous utilisons la Propriété C.2 du gradient de $\widetilde{L}_t(\mathbf{p}_\delta(t))$ en (C.42) et nous obtenons :

$$\mathbb{E}\left[\sum_{t=1}^{T} \langle \nabla \widetilde{L}_t(\mathbf{p}_\delta(t)) | \mathbf{p}_\delta(t) - \mathbf{q}_\delta \rangle\right] = \mathbb{E}\left[\sum_{t=1}^{T} \langle \mathbb{E}[\tilde{\mathbf{v}}(t) | \mathbf{u}(1), ..., \mathbf{u}(t-1)] | \mathbf{p}_\delta(t) - \mathbf{q}_\delta \rangle\right] \quad \forall \mathbf{p}_\delta(t), \mathbf{q}_\delta \in \mathscr{P}_\delta. \tag{C.43}$$

Pour chaque instant t, l'allocation de puissance $\mathbf{p}_\delta(t)$ est totalement déterminée par $\{\mathbf{u}(1), ..., \mathbf{u}(t-1)\}$ et $\mathbf{q}_\delta$ est constante. Ainsi, la propriété suivante est vraie :

$$\mathbb{E}\left[\sum_{t=1}^{T} \langle \mathbb{E}[\tilde{\mathbf{v}}(t) | \mathbf{u}(1), ..., \mathbf{u}(t-1)] | \mathbf{p}_\delta(t) - \mathbf{q}_\delta \rangle\right] = \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{E}[\langle \tilde{\mathbf{v}}(t) | \mathbf{p}_\delta(t) - \mathbf{q}_\delta \rangle | \mathbf{u}(1), ..., \mathbf{u}(t-1)]\right] \quad \forall \mathbf{p}_\delta(t), \mathbf{q}_\delta \in \mathscr{P}_\delta. \tag{C.44}$$

Nous pouvons maintenant déplacer l'espérance à l'intérieur de la somme car tous les $\mathbf{u}(t)$ sont indépendants :

$$\mathbb{E}\left[\sum_{t=1}^{T} \mathbb{E}\langle \tilde{\mathbf{v}}(t) | \mathbf{p}_\delta(t) - \mathbf{q}_\delta \rangle | \mathbf{u}(1), ..., \mathbf{u}(t-1)]\right] = \sum_{t=1}^{T} \mathbb{E}[\left[\mathbb{E}[\langle \tilde{\mathbf{v}}(t) | \mathbf{p}_\delta(t) - \mathbf{q}_\delta \rangle | \mathbf{u}(1), ..., \mathbf{u}(t-1)]\right] \quad \forall \mathbf{p}_\delta(t), \mathbf{q}_\delta \in \mathscr{P}_\delta. \tag{C.45}$$

En utilisant le Théorème de l'espérance totale nous obtenons :

$$\mathbb{E}\left[\mathbb{E}[\langle \tilde{\mathbf{v}}(t) | \mathbf{p}_\delta(t) - \mathbf{q}_\delta \rangle | \mathbf{u}(1), ..., \mathbf{u}(t-1)]\right] = \mathbb{E}\left[\langle \tilde{\mathbf{v}}(t) | \mathbf{p}_\delta(t) - \mathbf{q}_\delta \rangle\right] \quad \forall \mathbf{p}_\delta(t), \mathbf{q}_\delta \in \mathscr{P}_\delta. \tag{C.46}$$

Finalement, en faisant la somme sur tous les instants $t \leq T$ de (C.46) et en déplaçant l'espérance à l'extérieur de la somme car les $\mathbf{u}(t)$ sont indépendants :

$$\sum_{t=1}^{T} \mathbb{E}\left[\langle \tilde{\mathbf{v}}(t) | \mathbf{p}_\delta(t) - \mathbf{q}_\delta \rangle\right] = \mathbb{E}\left[\sum_{t=1}^{T} \langle \tilde{\mathbf{v}}(t) | \mathbf{p}_\delta(t) - \mathbf{q}_\delta \rangle\right] \quad \forall \mathbf{p}_\delta(t), \mathbf{q}_\delta \in \mathscr{P}_\delta. \tag{C.47}$$

En utilisant le même procédé dans les équations (C.42) et (C.47) nous obtenons :

$$\mathbb{E}\left[\sum_{t=1}^{T} \widetilde{L}_t(\mathbf{p}_\delta(t)) - \widetilde{L}_t(\mathbf{q}_\delta)\right] \leq \mathbb{E}\left[\sum_{t=1}^{T} \langle \tilde{\mathbf{v}}(t) | \mathbf{p}_\delta(t) - \mathbf{q}_\delta \rangle\right]. \tag{C.48}$$

□

Á partir de cette étape, nous allons nous concentrer sur le terme de droite à l'intérieur de l'espérance du C.2. Pour borner ce terme nous devons introduire une propriété importante concernant l'étape de projection exponentielle.

Proposition C.3. *L'étape de projection est définie comme :*

$$p^s(t) = \delta + P_{\max}\left(1 - \frac{\delta}{P_{\max}}(S+\sqrt{S})\right)\frac{\exp(y^s(t))}{1+\sum_{s'=1}^{S}\exp(\ y^{s'}(t))}, \quad \forall t \tag{C.49}$$

est équivalente à :

$$\boldsymbol{p}_\delta(t) = \underset{\hat{\boldsymbol{p}}\in\mathscr{P}_\delta}{\arg\max}\{\langle \boldsymbol{y}(t)|\hat{\boldsymbol{p}}\rangle - f(\hat{\boldsymbol{p}})\} \tag{C.50}$$

où $f(\boldsymbol{p}_\delta)$ est définie comme :

$$f(\boldsymbol{p}_\delta) = \sum_{s=1}^{S}\left((p_\delta^s-\delta)\log(p_\delta^s-\delta)\right) + \left(P_{\max}-\delta\sqrt{S}-\sum_{s=1}^{S}p_\delta^s\right)\log\left(P_{\max}-\delta\sqrt{S}-\sum_{s=1}^{S}p_\delta^s\right). \tag{C.51}$$

De l'équation (C.51) nous pouvons en déduire une condition sur δ :

$$P_{\max}-\delta\sqrt{S} \geq 0, \tag{C.52}$$

$$\delta \leq \frac{\sqrt{S}}{P_{\max}}. \tag{C.53}$$

Démonstration. Pour prouver cette proposition nous devons résoudre le problème d'optimisation suivant :

$$\text{maximiser}_{\hat{\mathbf{p}}\in\mathscr{P}_\delta}\{F_t(\hat{\mathbf{p}})\}, \tag{C.54}$$

où $F_t(\mathbf{p}_\delta)$ est une fonction définie comme :

$$F_t(\mathbf{p}_\delta) = \langle \mathbf{y}(t)|\mathbf{p}_\delta\rangle - f(\mathbf{p}_\delta), \tag{C.55}$$

qui est concave par rapport à $\mathbf{p}_\delta$. Pour résoudre ce problème d'optimisation nous devons calculer les dérivées de $F_t(\mathbf{p}_\delta)$ par rapport à $p_\delta^s, \forall s$:

$$\frac{\partial F_t(\mathbf{p}_\delta)}{\partial p_\delta^s} = y_s(t) - \log(p_\delta^s-\delta) + \log\left(P_{\max}-\delta\sqrt{S}-\sum_{s=1}^{S}p_\delta^s\right). \tag{C.56}$$

En annulant ces dérivées et en prenant l'exponentielle de chaque côté nous trouvons :

$$\exp(y^s) = \frac{p_\delta^s-\delta}{P_{\max}-\delta\sqrt{S}-\sum_{s=1}^{S}p_\delta^s} \tag{C.57}$$

En prenant la somme sur toutes les porteuses et en ré-arrengeant les termes nous trouvons :

$$\sum_{s=1}^{S}p_\delta^s = \frac{[P_{\max}-\delta\sqrt{S}]\sum_{s=1}^{S}\exp(y^s)+S\delta}{1+\sum_{j=1}^{S}\exp(y^j)} \tag{C.58}$$

La proposition découle de la substitution de (C.58) dans (C.57) et en faisant ressortir $\mathbf{p}_\delta$. □

Pour borner la somme dans l'espérance du terme de droite du Lemme C.2, nous allons procéder en deux étapes. Premièrement, nous allons utiliser la Proposition C.3 pour borner la somme des $\langle \tilde{\mathbf{v}}(t)|\mathbf{q}_\delta\rangle$ par la somme des $\langle \tilde{\mathbf{v}}(t)|\mathbf{p}_\delta(t+1)\rangle$. Deuxièmement, nous allons borner la différence cumulée entre $\langle \tilde{\mathbf{v}}(t)|\mathbf{p}_\delta(t)\rangle$ et $\langle \tilde{\mathbf{v}}(t)|\mathbf{p}_\delta(t+1)\rangle$.

Proposition C.4. *Si l'allocation de puissance $\boldsymbol{p}_\delta(t)$ est définie par :*

$$\boldsymbol{p}_\delta(t) = \underset{\hat{\boldsymbol{p}}\in\mathscr{P}_\delta}{\arg\max}\{\langle \boldsymbol{y}(t)|\hat{\boldsymbol{p}}\rangle - f(\hat{\boldsymbol{p}})\}, \quad \forall t \geq 1 \tag{C.59}$$

alors nous avons :

$$-\sum_{t=1}^{T}\langle \tilde{\boldsymbol{v}}(t)|\boldsymbol{q}_\delta\rangle \;\leq\; -\sum_{t=1}^{T}\langle \tilde{\boldsymbol{v}}(t)|\boldsymbol{p}_\delta(t+1)\rangle - \frac{f(\boldsymbol{p}_\delta(2))}{\mu} + \frac{f(\boldsymbol{q}_\delta)}{\mu}, \quad \forall \boldsymbol{q}_\delta \in \mathscr{P}_\delta. \tag{C.60}$$

Démonstration. La première étape est de noter que (C.59) est équivalent à :

$$\mathbf{p}_\delta(t+1) = \underset{\hat{\mathbf{p}}\in\mathscr{P}_\delta}{\arg\max}\left\{-\mu\sum_{i=1}^{t}\langle \tilde{\mathbf{v}}(i)|\hat{\mathbf{p}}\rangle - f(\hat{\mathbf{p}})\right\}, \quad \forall t \geq 1. \tag{C.61}$$

Cela vient de la définition de $\mathbf{y}(t)$:

$$\mathbf{y}(t) = \begin{cases} 0, & \text{t=1} \\ \mathbf{y}(t-1) - \mu\tilde{\mathbf{v}}(t-1), & \text{t >1,} \end{cases} \tag{C.62}$$

ce qui nous donne $\mathbf{y}(t+1) = -\mu\sum_{t=1}^{t}\tilde{\mathbf{v}}(i)$ pour tout $t \geq 1$.

Nous allons procéder par récurrence en commençant par $T = 1$. En utilisant la définition de l'étape de projection nous avons :

$$-\mu\langle \tilde{\mathbf{v}}(1)|\mathbf{q}_\delta\rangle - f(\mathbf{q}_\delta) \;\leq\; -\mu\langle \tilde{\mathbf{v}}(1)|\mathbf{p}_\delta(2)\rangle - f(\mathbf{p}_\delta(2)), \quad \forall \mathbf{q}_\delta \in \mathscr{P}_\delta \tag{C.63}$$

$$-\langle \tilde{\mathbf{v}}(1)|\mathbf{q}_\delta\rangle \leq\; -\langle \tilde{\mathbf{v}}(1)|\mathbf{p}_\delta(2)\rangle - \frac{f(\mathbf{p}_\delta(2))}{\mu} + \frac{f(\mathbf{q}_\delta)}{\mu}, \quad \forall \mathbf{q}_\delta \in \mathscr{P}_\delta. \tag{C.64}$$

Alors, la propriété (C.60) est vraie pour $T = 1$.

Maintenant, nous faisons l'hypothèse que la propriété est vraie pour $T-1$, et nous allons vérifier que la propriété est vraie à l'instant T :

$$-\sum_{t=1}^{T-1}\langle \tilde{\mathbf{v}}(t)|\mathbf{q}_\delta\rangle \;\leq\; -\sum_{t=1}^{T-1}\langle \tilde{\mathbf{v}}(t)|\mathbf{p}_\delta(t+1)\rangle - \frac{f(\mathbf{p}_\delta(2))}{\mu} + \frac{f(\mathbf{q}_\delta)}{\mu}, \quad \forall \mathbf{q}_\delta \in \mathscr{P}_\delta. \tag{C.65}$$

En additionnant $-\langle \tilde{\mathbf{v}}(T)|\mathbf{p}_\delta(T+1)\rangle$ de chaque côté nous obtenons :

$$-\sum_{t=1}^{T-1}\langle \tilde{\mathbf{v}}(t)|\mathbf{q}_\delta\rangle - \langle \tilde{\mathbf{v}}(T)|\mathbf{p}_\delta(T+1)\rangle \;\leq\; -\sum_{t=1}^{T}\langle \tilde{\mathbf{v}}(t)|\mathbf{p}_\delta(t+1)\rangle - \frac{f(\mathbf{p}_\delta(2))}{\mu} + \frac{f(\mathbf{q}_\delta)}{\mu}, \quad \forall \mathbf{q}_\delta \in \mathscr{P}_\delta. \tag{C.66}$$

L'équation précédente est vraie pour tous $\mathbf{q}_\delta \in \mathscr{P}_\delta$ et donc elle est vraie pour $\mathbf{q}_\delta = \mathbf{p}_\delta(T+1)$. Après avoir remis en ordre les termes nous trouvons :

$$-\mu\sum_{t=1}^{T}\langle \tilde{\mathbf{v}}(t)|\mathbf{p}_\delta(T+1)\rangle - f(\mathbf{p}_\delta(T+1)) \;\leq\; -\mu\sum_{t=1}^{T}\langle \tilde{\mathbf{v}}(t)|\mathbf{p}_\delta(t+1)\rangle - f(\mathbf{p}_\delta(2)). \tag{C.67}$$

Nous remarquons, de (C.66), que :

$$-\mu\sum_{t=1}^{T}\langle\tilde{\mathbf{v}}(t)|\mathbf{q}_\delta\rangle - f(\mathbf{q}_\delta) \leq -\mu\sum_{t=1}^{T}\langle\tilde{\mathbf{v}}(t)|\mathbf{p}_\delta(T+1)\rangle - f(\mathbf{p}_\delta(T+1)),\ \ \forall\mathbf{q}_\delta\in\mathscr{P}_\delta. \tag{C.68}$$

En utilisant l'équation (C.67) et (C.68), la propriété suivante est vraie :

$$-\mu\sum_{t=1}^{T}\langle\tilde{\mathbf{v}}(t)|\mathbf{q}_\delta\rangle - f(\mathbf{q}_\delta) \ \ \leq\ \ -\mu\sum_{t=1}^{T}\langle\tilde{\mathbf{v}}(t)|\mathbf{p}_\delta(t+1)\rangle - f(\mathbf{p}_\delta(2)),\quad \forall\mathbf{q}_\delta\in\mathscr{P}_\delta. \tag{C.69}$$

$$-\sum_{t=1}^{T}\langle\tilde{\mathbf{v}}(t)|\mathbf{q}_\delta\rangle \ \ \leq\ \ -\mu\sum_{t=1}^{T}\langle\tilde{\mathbf{v}}(t)|\mathbf{p}_\delta(t+1)\rangle - \frac{f(\mathbf{p}_\delta(2))}{\mu} + \frac{f(\mathbf{q}_\delta)}{\mu},\quad \forall\mathbf{q}_\delta\in\mathscr{P}_\delta. \tag{C.70}$$

En conclusion, (C.60) est vraie pour tous les $T \geq 1$. □

Lemme C.3. *Si l'étape de projection est définie comme :*

$$p^s(t) = \delta + \left(1 - \frac{\delta}{P_{\max}}(S+\sqrt{S})\right)P_{\max}\frac{\exp(y^s(t))}{1+\sum_{s'=1}^{S}\exp(\ y^{s'}(t))},\quad \forall t \tag{C.71}$$

alors :

$$\sum_{t=1}^{T}\langle\tilde{\boldsymbol{v}}(t)|\boldsymbol{p}_\delta(t)-\boldsymbol{q}_\delta\rangle \leq \sum_{t=1}^{T}\langle\tilde{\boldsymbol{v}}(t)|\boldsymbol{p}_\delta(t)-\boldsymbol{p}_\delta(t+1)\rangle + \frac{(P_{\max}-\delta(S+\sqrt{S}))\log(1+S)}{\mu}. \tag{C.72}$$

Démonstration. Premièrement, nous remarquons que $f(\mathbf{p}_\delta)$ définie en (C.51) peut être réécrite comme :

$$f(\mathbf{p}_\delta) = \sum_{s=1}^{S}(p_\delta^s-\delta)\log(p_\delta^s-\delta) + \left(P_{\max}-\delta(S+\sqrt{S})-\sum_{s=1}^{S}(p_\delta^s-\delta)\right)\log\left(P_{\max}-\delta(S+\sqrt{S})-\sum_{s=1}^{S}(p_\delta^s-\delta)\right). \tag{C.73}$$

Nous utilisons le changement de variable suivant $\mathbf{x} = (\mathbf{p}_\delta - \mathbb{1}_S\delta)/(P_{\max}-\delta(\sqrt{S}+S))$, ce qui signifie que $\mathbf{p}_\delta = \mathbf{x}(P_{\max}-\delta(S+\sqrt{S})) + \mathbb{1}_S\delta$ et $f(\mathbf{p}_\delta) = f(\mathbf{p}_\delta(\mathbf{x})) = \tilde{f}(\mathbf{x})$ devient :

$$\begin{aligned}\tilde{f}(\mathbf{x}) = (P_{\max}-\delta(S+\sqrt{S}))&\left[\sum_{s=1}^{S}(x_s\log(x_s)) + \left(1-\sum_{s=1}^{S}x_s\right)\log\left(1-\sum_{s=1}^{S}x_s\right)\right]\\ &+(P_{\max}-\delta(S+\sqrt{S}))\log(P_{\max}-\delta(S+\sqrt{S})),\end{aligned} \tag{C.74}$$

où $\mathbf{x}\in\mathscr{X} = \{\mathbf{x}\in\mathbb{R}^S, x_s\geq 0, \sum_{s=1}^{S}x_s \leq 1\}$. Nous appelons $\hat{f}(\mathbf{x})$ la fonction :

$$\hat{f}(\mathbf{x}) = \sum_{s=1}^{S}x_s\log(x_s) + (1-\sum_{s=1}^{S}x_s)\log(1-\sum_{s=1}^{S}x_s). \tag{C.75}$$

La fonction $\hat{f}(\mathbf{x})$ est toujours négative. De la Proposition C.4 nous devons borner :

$$-f(\mathbf{p}_\delta(2)) + f(\mathbf{q}_\delta) \leq -\min_{\mathbf{p}_\delta\in\mathscr{P}_\delta} f(\mathbf{p}_\delta) + \max_{\mathbf{p}_\delta\in\mathscr{P}_\delta} f(\mathbf{p}_\delta). \tag{C.76}$$

Vu que le changement de variable est défini par $\mathbf{x} = (\mathbf{p}_\delta - \mathbb{1}_S\delta)/(P_{\max} - \delta(\sqrt{S}+S))$, nous pouvons calculer le maximum et le minimum de la fonction $f(\mathbf{p})$ en utilisant $\tilde{f}(\mathbf{x})$:

$$\min_{\mathbf{p}_\delta \in \mathscr{P}_\delta} f(\mathbf{p}_\delta) = \min_{\tilde{\mathbf{x}} \in \mathscr{X}} \tilde{f}(\tilde{\mathbf{x}}), \tag{C.77}$$

$$\max_{\mathbf{p}_\delta \in \mathscr{P}_\delta} f(\mathbf{p}_\delta) = \max_{\tilde{\mathbf{x}} \in \mathscr{X}} \tilde{f}(\tilde{\mathbf{x}}). \tag{C.78}$$

Nous allons commencer par le calcul de $\max_{\tilde{\mathbf{x}} \in \mathscr{X}} \tilde{f}(\tilde{\mathbf{x}})$. Vu que $\hat{f}(\mathbf{x})$ est toujours négative nous montrons que :

$$\begin{aligned}\tilde{f}(\mathbf{x}) &= \hat{f}(\mathbf{x}) + (P_{\max} - \delta(S+\sqrt{S}))\log\Big(P_{\max} - \delta(S+\sqrt{S})\Big) \\ &\leq (P_{\max} - \delta(S+\sqrt{S}))\log(P_{\max} - \delta(S+\sqrt{S})), \quad \forall \mathbf{x} \in \mathscr{X}.\end{aligned} \tag{C.79}$$

Maintenant, nous devons calculer $\min_{\tilde{\mathbf{x}} \in \mathscr{X}} \tilde{f}(\tilde{\mathbf{x}})$. Nous observons que :

$$\arg\min_{\tilde{\mathbf{x}} \in \mathscr{X}} \tilde{f}(\tilde{\mathbf{x}}) = \arg\min_{\hat{\mathbf{x}} \in \mathscr{X}} \hat{f}(\hat{\mathbf{x}}), \tag{C.80}$$

de (C.74) et vu que la fonction $\hat{f}(\mathbf{x})$ est convexe par rapport à $\mathbf{x}$, alors pour trouver le minimum nous devons annuler la dérivé par rapport à $\mathbf{x}$ du Lagrangien $\mathscr{L}$ défini comme :

$$\mathscr{L}(\mathbf{x}, \beta) = f(\mathbf{x}) + \beta\left(\sum_{s=1}^{S} x_s - 1\right). \tag{C.81}$$

Nous trouvons :

$$\frac{\partial \mathscr{L}(\mathbf{x}, \beta)}{\partial \mathbf{x}} = \log(\mathbf{x}) - \log\left(1 - \sum_{i=1}^{S} x_i\right) + \beta. \tag{C.82}$$

De (C.82) et en annulant ces dérivées nous notons que :

$$x_s^* = \frac{\exp(-\beta)}{1 + S\exp(-\beta)}, \quad \forall s, \tag{C.83}$$

maintenant nous devons trouver la valeur optimale de β. Pour cela, nous allons utiliser les conditions de Karush–Kuhn–Tucker [Boyd and Vandenberghe, 2004]. Ces conditions nous disent que à l'optimum nous avons :

$$\beta(\sum_{i=1}^{S} x_i - 1) = 0 \rightarrow \begin{cases} \beta^* = 0 \text{ et } \sum_{i=1}^{S} x_i^* \leq 1, & \text{(C1)} \\ \sum_{i=1}^{S} x_i^* = 1 \text{ et }, \beta^* > 0, & \text{(C2).} \end{cases} \tag{C.84}$$

De (C.83) nous remarquons que :

$$\sum_{i=1}^{S} x_s^* = \frac{S\exp(-\beta)}{1 + S\exp(-\beta)} \neq 1, \quad \forall \beta > 0, \tag{C.85}$$

cela implique que la contrainte (C2) est impossible, nous pouvons en conclure que $\beta^* = 0$. Sachant que $\beta^* = 0$ de l'équation (C.83) nous trouvons :

$$\mathbf{x}^* = \frac{1}{1+S} \tag{C.86}$$

En remplaçant $\mathbf{x}$ par (C.86) dans (C.74) nous trouvons :

$$\min_{\hat{\mathbf{p}}\in\mathscr{P}_\delta} f(\hat{\mathbf{p}}) = (P_{\max} - \delta(S+\sqrt{S}))\log\left(\frac{P_{\max}-\delta(S+\sqrt{S})}{1+S}\right). \tag{C.87}$$

Finalement, nous utilisons (C.79) et (C.87) dans (C.76) et nous obtenons :

$$-f(\mathbf{p}_\delta(2)) + f(\mathbf{q}_\delta) \leq P_{\max}\log(1+S). \tag{C.88}$$

Le Lemme C.3 est obtenu en utilisant (C.88) dans l'équation (C.60). □

La dernière étape pour borner le regret moyen est de borner la somme du côté droit de l'équation (C.72) du Lemme C.3. Pour borner cette somme, nous allons utiliser la Proposition C.3, quelques résultats de la théorie de l'optimisation et l'inégalité de Cauchy-Schwartz.

Lemme C.4. *Si l'étape de projection est définie par :*

$$p^s(t) = \delta + P_{\max}\left(1 - \frac{\delta}{P_{\max}}(S+\sqrt{S})\right)\frac{\exp(y^s(t))}{1+\sum_{s'=1}^{S}\exp(\ y^{s'}(t))}, \quad \forall t, \quad \forall s \tag{C.89}$$

alors :

$$\sum_{t=1}^{T}\langle\tilde{\boldsymbol{v}}(t)|\boldsymbol{p}_\delta(t)\rangle - \boldsymbol{p}_\delta(t+1) \leq \mu P_{\max}\sum_{t=1}^{T}\|\tilde{\boldsymbol{v}}(t)\|_\infty^2. \tag{C.90}$$

Démonstration. Pour borner la différence entre $\langle\tilde{\mathbf{v}}(t)|\mathbf{p}_\delta(t)\rangle$ et $\langle\tilde{\mathbf{v}}(t)|\mathbf{p}_\delta(t+1)\rangle$ nous allons commencer par rappeler quelques notations :

$$F_t(\mathbf{p}_\delta) = -\mu\sum_{i=1}^{t-1}\langle\tilde{\mathbf{v}}(i)|\mathbf{p}_\delta\rangle - f(\mathbf{p}_\delta). \tag{C.91}$$

La fonction $F_t(\mathbf{p}_\delta)$ est une somme de fonctions linéaires en $\mathbf{p}_\delta$ et de la fonction $-f(\mathbf{p}_\delta)$ qui est une fonction $\frac{1}{P_{\max}}$-fortement régulière par rapport à la norme $\|\cdot\|_\infty$. [1] L'addition de fonctions linéaires et de fonctions fortement régulière est aussi fortement régulière.

En utilisant la propriété de forte régularité de la fonction $F_t(\mathbf{p}_\delta)$ nous pouvons écrire :

$$F_t(\mathbf{p}_\delta(t+1)) \leq F_t(\mathbf{p}_\delta(t)) + \langle\nabla F_t(\mathbf{p}_\delta(t))|\mathbf{p}_\delta(t+1)-\mathbf{p}_\delta(t)\rangle - \frac{1}{2P_{\max}}\left\|\mathbf{p}_\delta(t+1)-\mathbf{p}_\delta(t)\right\|_\infty^2 \tag{C.93}$$

nous observons que $\mathbf{p}_\delta(t) = \arg\max_{\mathbf{p}_\delta\in\mathscr{P}_\delta}\left(F_t(\mathbf{p}_\delta)\right)$ alors la théorie de l'optimisation convexe nous dit que [Boyd and Vandenberghe, 2004] :

$$\langle\nabla F_t\mathbf{p}_\delta(t)|\mathbf{p}_\delta(t+1)-\mathbf{p}_\delta(t)\rangle \leq 0. \tag{C.94}$$

1. Une fonction $f(\mathbf{p}):\mathbb{R}^S\to\mathbb{R}$ est K-fortement régulière par rapport à la norme $\|\cdot\|_\infty$ ci :

$$f(\mathbf{p}+\mathbf{q}) \leq f(\mathbf{p}) + \langle\nabla f(\mathbf{p})|\mathbf{p})-\mathbf{p}\rangle - \frac{K}{2}\|\mathbf{p}-\mathbf{p}\|_\infty^2. \tag{C.92}$$

En utilisant l'équation (C.94) dans l'équation (C.93) et en substituant $F_t(\mathbf{p}_\delta)$ par sa définition nous trouvons :

$$-\mu\sum_{i=1}^{t-1}\langle\tilde{\mathbf{v}}(i)|\mathbf{p}_\delta(t+1)\rangle - f(\mathbf{p}_\delta(t+1) \leq -\mu\sum_{i=1}^{t-1}\langle\tilde{\mathbf{v}}(i)|\mathbf{p}_\delta(t)\rangle - \frac{\left\|\mathbf{p}_\delta(t+1)-\mathbf{p}_\delta(t)\right\|_\infty^2}{2P_{\max}} - f(\mathbf{p}_\delta(t)). \quad \text{(C.95)}$$

Nous pouvons faire la même chose en observant que $\mathbf{p}_\delta(t+1) = \arg\max_{\mathbf{p}_\delta\in\mathscr{P}_\delta}\left(F_t(\mathbf{p}_\delta)\right)$:

$$F_{t+1}(\mathbf{p}_\delta(t)) \leq F_{t+1}(\mathbf{p}_\delta(t+1)) + \langle\nabla F_{t+1}(\mathbf{p}_\delta(t+1))|\mathbf{p}_\delta(t)-\mathbf{p}_\delta(t+1)\rangle - \frac{1}{2P_{\max}}\left\|\mathbf{p}_\delta(t+1)-\mathbf{p}_\delta(t)\right\|_\infty^2. \quad \text{(C.96)}$$

Nous pouvons aussi substituer $F_{t+1}(\mathbf{p}_\delta)$ par sa définition et en utilisant le fait que $\mathbf{p}_\delta(t+1)$ minimise, par définition, $F_{t+1}(\mathbf{p}_\delta)$ pour obtenir :

$$-\mu\sum_{i=1}^{t}\langle\tilde{\mathbf{v}}(i)|\mathbf{p}_\delta(t)\rangle - f(\mathbf{p}_\delta(t)) \leq -\mu\sum_{i=1}^{t}\langle\tilde{\mathbf{v}}(i)|\mathbf{p}_\delta(t+1)\rangle - f(\mathbf{p}_\delta(t+1) - \frac{1}{2P_{\max}}\left\|\mathbf{p}_\delta(t+1)-\mathbf{p}_\delta(t)\right\|_\infty^2 \quad \text{(C.97)}$$

Ainsi, en calculant la somme des équations (C.95) et (C.97), nous remarquons que $f(\mathbf{p}_\delta(t+1)$ et $f(\mathbf{p}_\delta(t))$ s'annulent :

$$-\mu\sum_{i=1}^{t-1}\langle\tilde{\mathbf{v}}(i)|\mathbf{p}_\delta(t+1)\rangle - \mu\sum_{i=1}^{t}\langle\tilde{\mathbf{v}}(i)|\mathbf{p}_\delta(t)\rangle \leq -\mu\sum_{i=1}^{t-1}\langle\tilde{\mathbf{v}}(i)|\mathbf{p}_\delta(t)\rangle - \mu\sum_{i=1}^{t}\langle\tilde{\mathbf{v}}(i)|\mathbf{p}_\delta(t+1)\rangle - \frac{1}{P_{\max}}\left\|\mathbf{p}_\delta(t+1)-\mathbf{p}_\delta(t)\right\|_\infty^2. \quad \text{(C.98)}$$

En réarrangeant les termes nous obtenons :

$$-\mu\sum_{i=1}^{t}\langle\tilde{\mathbf{v}}(i)|\mathbf{p}_\delta(t)\rangle + \mu\sum_{i=1}^{t-1}\langle\tilde{\mathbf{v}}(i)|\mathbf{p}_\delta(t)\rangle \leq -\mu\sum_{i=1}^{t}\langle\tilde{\mathbf{v}}(i)|\mathbf{p}_\delta(t+1)\rangle + \mu\sum_{i=1}^{t-1}\langle\tilde{\mathbf{v}}(i)|\mathbf{p}_\delta(t+1)\rangle - \frac{1}{P_{\max}}\left\|\mathbf{p}_\delta(t+1)-\mathbf{p}_\delta(t)\right\|_\infty^2, \quad \text{(C.99)}$$

$$-\mu\langle\tilde{\mathbf{v}}(t)|\mathbf{p}_\delta(t)\rangle \leq -\mu\langle\tilde{\mathbf{v}}(t)|\mathbf{p}_\delta(t+1)\rangle - \frac{1}{P_{\max}}\left\|\mathbf{p}_\delta(t+1)-\mathbf{p}_\delta(t)\right\|_\infty^2. \quad \text{(C.100)}$$

De l'équation (C.100) nous avons une borne inférieure de $\langle\tilde{\mathbf{v}}(t)|\mathbf{p}_\delta(t+1)-\mathbf{p}_\delta(t)\rangle$, qui est définie comme suit :

$$\frac{1}{\mu P_{\max}}\left\|\mathbf{p}_\delta(t+1)-\mathbf{p}_\delta(t)\right\|_\infty^2 \leq \langle\tilde{\mathbf{v}}(t)|\mathbf{p}_\delta(t+1)-\mathbf{p}_\delta(t)\rangle. \quad \text{(C.101)}$$

Nous utilisons inégalité de Cauchy-Schwartz, pour trouver une borne du terme de droite de l'équation (C.101)

$$|\langle\tilde{\mathbf{v}}(t)|\mathbf{p}_\delta(t+1)-\mathbf{p}_\delta(t)\rangle| \leq \left\|\mathbf{p}_\delta(t+1)-\mathbf{p}_\delta(t)\right\|_2 \|\tilde{\mathbf{v}}(t)\|_\infty. \quad \text{(C.102)}$$

De l'équation, (C.101) nous pouvons déduire que le terme $\langle\tilde{\mathbf{v}}(t)|\mathbf{p}_\delta(t+1)-\mathbf{p}_\delta(t)\rangle$ est positif, car μ est strictement positif et la norme $\left\|\mathbf{p}_\delta(t+1)-\mathbf{p}_\delta(t)\right\|_\infty^2$ est aussi positive. C'est pourquoi nous pouvons retirer la valeur absolue de l'équation (C.102) et nous obtenons :

$$\langle\tilde{\mathbf{v}}(t)|\mathbf{p}_\delta(t+1)-\mathbf{p}_\delta(t)\rangle \leq \left\|\mathbf{p}_\delta(t+1)-\mathbf{p}_\delta(t)\right\|_\infty \|\tilde{\mathbf{v}}(t)\|_\infty. \quad \text{(C.103)}$$

Nous avons une borne supérieure de $\langle \tilde{\mathbf{v}}(t) | \mathbf{p}_\delta(t+1) - \mathbf{p}_\delta(t) \rangle$ dans (C.101) et une borne inférieure dans l'équation (C.103). En regroupant ces bornes nous obtenons :

$$\frac{1}{\mu P_{\max}} \left\| \mathbf{p}_\delta(t+1) - \mathbf{p}_\delta(t) \right\|_\infty^2 \leq \langle \tilde{\mathbf{v}}(t) | \mathbf{p}_\delta(t+1) - \mathbf{p}_\delta(t) \rangle \leq \left\| \mathbf{p}_\delta(t) - \mathbf{p}_\delta(t+1) \right\|_\infty \| \tilde{\mathbf{v}}(t) \|_\infty . \quad \text{(C.104)}$$

Nous déduisons de cette équation que la norme de la différence entre les vecteurs $\mathbf{p}_\delta(t)$ et $\mathbf{p}_\delta(t+1)$ est bornée par :

$$\left\| \mathbf{p}_\delta(t) - \mathbf{p}_\delta(t+1) \right\|_\infty \leq \mu P_{\max} \| \tilde{\mathbf{v}}(t) \|_\infty . \quad \text{(C.105)}$$

Finalement, nous utilisons les équations (C.105) et (C.102) pour trouver la borne finale :

$$\langle \tilde{\mathbf{v}}(t) | \mathbf{p}_\delta(t+1) - \mathbf{p}_\delta(t) \rangle \leq \mu P_{\max} \| \tilde{\mathbf{v}}(t) \|_\infty^2 . \quad \text{(C.106)}$$

Nous pouvons utiliser cette équation pour borner la somme sur t :

$$\sum_{t=1}^{T} \langle \tilde{\mathbf{v}}(t) | \mathbf{p}_\delta(t+1) - \mathbf{p}_\delta(t) \rangle \leq \mu P_{\max} \sum_{t=1}^{T} \| \tilde{\mathbf{v}}(t) \|_\infty^2 . \quad \text{(C.107)}$$

□

En combinant les Lemmes 1-4 nous trouvons la bornne suivante :

$$\mathrm{EReg}_{\mathbf{q}}(T) \leq \mathbb{E} \left[\frac{P_{\max} \log(1+S)}{\mu} + P_{\max} \mu \sum_{t=1}^{S} \| \tilde{\mathbf{v}}(t) \|_\infty^2 + TK\delta(3 + (2\sqrt{S} + S)) \right] . \quad \text{(C.108)}$$

Dans la borne ci-dessus tous les termes ne dépendent pas de la variable de l'estimateur à l'execption de $\sum_{t=1}^{S} \| \tilde{\mathbf{v}}(t) \|_\infty^2$. Nous pouvons donc sortir les termes de l'espérance ce que nous donne :

$$\mathrm{EReg}_{\mathbf{q}}(T) \leq \frac{P_{\max} \log(1+S)}{\mu} + P_{\max} \mu \mathbb{E} \left[\sum_{t=1}^{S} \| \tilde{\mathbf{v}}(t) \|_\infty^2 \right] + TK\delta(3 + (2\sqrt{S} + S). \quad \text{(C.109)}$$

Maintenant, nous devons trouver une borne de l'espérance de la norme de l'estimateur de $\tilde{\mathbf{v}}(t)$.

Lemme C.5. *Si la fonction $L_t(\boldsymbol{p})$ est K-Lipschitz et que la fonction est bornée, i.e.*

$$B = \max_{t \in \{1,\dots,T\}, \boldsymbol{p} \in \mathscr{P}} L_t(\boldsymbol{p}) \quad \text{(C.110)}$$

alors l'espérance de la somme des estimateurs $\tilde{\boldsymbol{v}}(t) = \frac{S}{\delta} L_t(\boldsymbol{p}_\delta(t) + \delta \boldsymbol{u}(t)) \boldsymbol{u}$ est bornée par :

$$\mathbb{E} \left[\sum_{t=1}^{T} \| \tilde{\boldsymbol{v}}(t) \|_\infty^2 \right] \leq TS^2 \left(\frac{S}{\delta} + K \right)^2 . \quad \text{(C.111)}$$

Démonstration. Nous substituons l'estimateur du gradient $\tilde{\mathbf{v}}(t)$ par sa définition dans l'équation de la Proposition C.2 et nous remarquons que $\mathbf{u}(t)$ est tiré de manière uniforme sur la sphère Euclidienne unitaire :

$$\mathbb{E} \left[\sum_{t=1}^{T} \| \tilde{\mathbf{v}}(t) \|_\infty^2 \right] = \mathbb{E} \left[\sum_{t=1}^{T} \left\| \frac{S}{\delta} L_t(\mathbf{p}_\delta(t) + \delta \mathbf{u}(t)) \mathbf{u}(t) \right\|_\infty^2 \right] . \quad \text{(C.112)}$$

Du au fait que les fonctions $L_t(\mathbf{p}_\delta)$ sont K-Lipschitz, nous obtenons :

$$L_t(\mathbf{p}_\delta(t)+\delta\mathbf{u}(t)) \leq L_t(\mathbf{p}_\delta(t))+K\delta \quad \forall \mathbf{u}(t) \tag{C.113}$$

et en utilisant la borne B nous obtenons :

$$L_t(\mathbf{p}_\delta(t)+\delta\mathbf{u}(t)) \leq B+K\delta. \tag{C.114}$$

En substituant la borne de $L_t(\mathbf{p}_\delta(t)+\delta\mathbf{u}(t))$ dans (C.112) nous obtenons :

$$\mathbb{E}\left[\sum_{t=1}^{T}\frac{S^2}{\delta^2}||L_t(\mathbf{p}_\delta(t)+\delta\mathbf{u}(t))||_\infty^2\right] \leq \mathbb{E}\left[\sum_{t=1}^{T}\frac{S^2}{\delta^2}(B+K\delta)^2\right] \tag{C.115}$$

$$= TS^2\left(\frac{B}{\delta}+L\right)^2. \tag{C.116}$$

En combinant les équations (C.116) et (C.112) le Lemme C.5 est prouvé. □

La dernière étape consiste à rassembler les Lemmes C.1-C.5 pour trouver la bornne finale du regret :

$$\mathrm{EReg}(T) \leq Z(\mu,\delta) \tag{C.117}$$

où $Z(\mu,\delta)$ est défini comme :

$$Z(\mu,\delta) = \frac{P_{\max}\log(1+S)}{\mu}+\mu TS^2\left(\frac{B}{\delta}+K\right)^2+KT\delta\left(3+\left(S+2\sqrt{S}\right)\right). \tag{C.118}$$

C.2.2 Preuve du Corollaire 5

Il faut dans un premier temps remarquer que la borne du regret définie par (C.118) est linéaire en T. Nous devons donc déterminer des paramètres optimaux δ et μ tel que la croissance du regret est plus faible que $\mathcal{O}(T)$.

Pour faire cela, nous allons commencer par optimiser la borne du regret en fonction de μ. En effet, la borne du regret est convexe en fonction de μ, nous pouvons donc déterminer le pas μ optimal en calculant et annulant la dérivé de la borne. Nous devons maintenant calculer et annuler la dérivé partielle de la fonction Z par rapport à μ :

$$\frac{\partial Z(\mu,\delta)}{\partial\mu} = -\frac{P_{\max}\log(1+S)}{2\mu^2}+TS^2\left(\frac{B}{\delta}+K\right)^2 \tag{C.119}$$

Nous pouvons en déduire que :

$$\mu^* = \sqrt{\frac{P_{\max}\log(1+S)}{2T}}\frac{1}{S}\left(\frac{B}{\delta}+K\right)^{-1} \tag{C.120}$$

Maintenant que nous avons déterminé le pas μ^*, nous allons remplacer μ par μ^* dans la borne (C.118) ce qui nous donne :

$$\mathrm{EReg}_{\mathbf{q}}(T) \leq \frac{3}{2}\sqrt{P_{\max}\log(1+S)T}S\left(\frac{B}{\delta}+K\right)+TK\delta(3+(2\sqrt{S}+S)). \tag{C.121}$$

La borne du regret ci-dessus, après optimisation de μ, est encore linéaire en T. Nous devons donc déterminer un pas δ tel que la croissance du regret est plus faible que $\mathcal{O}(T)$. Cependant, bien que convexe la présence des contraintes suivantes : $\delta \leq \frac{P_{\max}}{S+\sqrt{S}} \leq \frac{P_{\max}}{\sqrt{S}}$, nous empêche de trouver une solution en forme close de δ^*. Il faut toutefois noter que notre objectif premier est d'obtenir la propriété de non regret. Pour y parvenir, il suffit de trouver un δ qui respecte les contraintes tout en limitant la croissance (qui doit être inférieure à $\mathcal{O}(T)$) de la borne du regret ci-dessus. Il faut donc dans un premier temps déterminer l'ordre de grandeur des variations de δ en fonction de T. Pour cela, nous remarquons que nous pouvons réécrire le regret comme :

$$\mathrm{EReg}(T) \leq \frac{\mathcal{O}(\sqrt{T})}{\delta} + \mathcal{O}(T)\delta. \tag{C.122}$$

De l'équation ci-dessus, nous remarquons que si le pas croît en $\mathcal{O}(T^{-1/4})$ alors le regret va croître en $\mathcal{O}(T^{\frac{3}{4}})$ ce qui est suffisant pour avoir la propriété de non regret. Ainsi, il ne nous reste plus qu'à trouver un pas qui croît en $\mathcal{O}(T^{-1/4})$ et qui respecte les contraintes sur δ, ce pas peut être défini comme :

$$\delta^* = \frac{P_{\max}}{(S+\sqrt{S})T^{\frac{1}{4}}}. \tag{C.123}$$

Cette valeut de δ respecte les contraintes tout en limitant la croissance de la borne du regret. Pour visualiser cela, il faut remplacer δ par δ^* dans la borne (C.121) ce qui nous donne :

$$\mathrm{EReg}(T) \leq U_1 T^{\frac{3}{4}} + U_2 T^{\frac{1}{2}}, \tag{C.124}$$

où les termes U_1 et U_2 sont définis respectivement par :

$$\begin{aligned} U_1 &= SB\left(S+\sqrt{S}\right)\sqrt{\frac{2\log(1+S)}{P_{\max}}} + K\left(3+(S+2\sqrt{S})\right)\frac{P_{\max}}{S+\sqrt{S}} \\ U_2 &= \sqrt{2P_{\max}\log(1+S)}\, SK. \end{aligned} \tag{C.125}$$

C.2.3 Preuve du Corollaire 6

Dans le cas où la durée de transmission T n'est pas connue en avance, l'objet n'est pas en mesure de calculer les paramètres optimaux δ^* et μ^*. Pour y pallier, nous allons utiliser l'astuce du *doubling-trick* dérit dans la Section 2.3.1. En utilisant cette astuce nous trouvons la borne suivante pour le regret :

$$\mathrm{EReg}(T) \leq \frac{2\sqrt{2}}{2^{\frac{3}{4}}-1} U_1 T^{\frac{3}{4}} + \frac{2}{\sqrt{2}-1} U_2 T^{\frac{1}{2}}, \tag{C.126}$$

où les termes U_1 et U_2 sont définis respectivement par :

$$\begin{aligned} U_1 &= SB\left(S+\sqrt{S}\right)\sqrt{\frac{2\log(1+S)}{P_{\max}}} + K\left(3+(S+2\sqrt{S})\right)\frac{P_{\max}}{S+\sqrt{S}} \\ U_2 &= \sqrt{2P_{\max}\log(1+S)}\, SK. \end{aligned} \tag{C.127}$$

Ceci implique que le regret moyen décroît en $\mathcal{O}(T^{-\frac{1}{4}})$.

ANNEXE D

PREUVES RELATIVES AU CHAPITRE 5

Pour gagner en lisibilité, pour le reste de la preuve nous allons utiliser la notion de regret par rapport à une allocation de puissance fixe $\mathbf{q}$. Ce regret relatif, noté $\mathrm{Reg}_{\mathbf{q}}(T)$, est défini comme :

$$\mathrm{Reg}_{\mathbf{q}} = \sum_{t=1}^{T} L_t(\mathbf{p}(t)) - L_t(\mathbf{q}). \tag{D.1}$$

Il faut noter que le regret relatif à l'allocation fixe $\mathrm{Reg}_{\mathbf{q}}(T)$ est égal au regret $\mathrm{Reg}(T)$ lorsque $\mathbf{q}$ est définie comme :

$$\mathbf{q}^* = \arg\min_{\mathbf{q}\in\mathscr{P}} \sum_{t=1}^{T} L_t(\mathbf{q}). \tag{D.2}$$

Maintenant que nous avons défini la notion de regret relatif, nous allons détailler la preuve du Théorème 4.

D.1 Preuves des résultats théoriques de l'Algorithme GMD dans le cas parfait

D.1.1 Preuve du Théorème 4

La première étape dans la preuve du Théorème 4 est d'utiliser la convexité des fonctions objectif $L_t(\mathbf{p})$ pour borner le regret relatif, ce qui nous donne :

$$\mathrm{Reg}_{\mathbf{q}}(T) \leq \langle \nabla L_t(\mathbf{p}(t)) | \mathbf{p}(t) - \mathbf{q} \rangle, \quad \forall \mathbf{q}. \tag{D.3}$$

Nous étudions le cas où le feedback est le gradient parfait, ce qui signifie que : $\mathbf{v}(t) = \nabla L_t(\mathbf{p}(t))$. Nous pouvons donc remplacer le gradient par $\mathbf{v}(t)$ dans l'équation (D.3) et nous obtenons :

$$\mathrm{Reg}_{\mathbf{q}}(T) \leq \sum_{t=1}^{T} \langle \mathbf{v}(t) | \mathbf{p}(t) - \mathbf{q} \rangle, \quad \forall \mathbf{q}. \tag{D.4}$$

Nous utilisons maintenant le score interne et sa définition (3.21), $\mathbf{y}(t+1) = \mathbf{y} - \mu\mathbf{v}(t)$ et $\mathbf{y}(1) = 0$, dans la borne (D.4) pour simplifier la somme des produits scalaires qui concerne l'allocation fixe $\mathbf{q}$:

$$\mathrm{Reg}_{\mathbf{q}}(T) \leq \langle \mathbf{v}(t)|\mathbf{p}(t)\rangle + \frac{1}{\mu}\langle \mathbf{y}(t+1)|\mathbf{q}\rangle, \quad \forall \mathbf{q}. \tag{D.5}$$

Maintenant que nous avons borné la partie qui concerne l'allocation de puissance fixe, $\mathbf{q}$, nous devons nous concentrer sur la somme qui reste (et donc sur l'allocation de puissance dynamique, $\mathbf{p}(t)$). Pour cela nous allons utiliser la fonction de régularisation $f(\mathbf{p})$ et en particulier sa fonction convexe conjuguée $f^*(\mathbf{y})$. En utilisant le fait que $\mathbf{p}(t) = \nabla f^*(\mathbf{y}(t))$ et l'approximation de Taylor d'ordre 2 de la fonction $f^*(\mathbf{y})$, qui est F-fortement convexe, approximation définie par :

$$f^*(\mathbf{y}(t+1)) \leq f^*(\mathbf{y}(t)) - \mu\langle \mathbf{v}(t)|\nabla^*(\mathbf{y}(t))\rangle + \frac{\mu^2}{2}F\|\mathbf{v}(t)\|_\infty^2. \tag{D.6}$$

Nous allons utiliser cette approximation pour borner le produit scalaire entre $\mathbf{p}(t)$ et $\mathbf{v}(t)$ dans l'équation (D.4), ce qui nous donne :

$$\mathrm{Reg}_{\mathbf{q}} \leq \frac{1}{\mu}\left[f^*(0) - f^*(\mathbf{y}(T+1))\right] + \frac{\mu}{2}F\sum_{t=1}^{T}||\mathbf{v}(t)||_\infty^2 + \frac{1}{\mu}\langle \mathbf{y}(T+1)|\mathbf{q}\rangle, \quad \forall \mathbf{q}. \tag{D.7}$$

Nous allons maintenant utiliser l'inégalité de Fenchel [Rockafellar, 2015] qui borne la somme entre $f(\mathbf{p})$ et $f^*(\mathbf{y})$, plus précisément cette inégalité nous donne :

$$f^*(\mathbf{y}) + f(\mathbf{p}) \geq \langle \mathbf{y}|\mathbf{p}\rangle, \ \forall \mathbf{y}, \mathbf{p}. \tag{D.8}$$

Cette inégalité nous permet de remplacer $\langle \mathbf{y}(T+1)|\mathbf{q}\rangle - f^*(\mathbf{y}(T+1))$ par $f(\mathbf{q})$ dans l'équation (D.7), nous utilisons aussi le fait que la norme gradient est bornée, $||\mathbf{v}(t)||_\infty^2 \leq V^2$ et nous trouvons :

$$\mathrm{Reg}_{\mathbf{q}} \leq \frac{1}{\mu}\left[f(\mathbf{q}) + f^*(0)\right] + \frac{\mu}{2}FV^2T, \quad \forall \mathbf{q}. \tag{D.9}$$

Il nous reste maintenant à nous occuper de $f(\mathbf{q})$. Nous pouvons montrer que $f(\mathbf{q}) \leq 0$ en utilisant l'inégalité de Jensen ainsi que le changement de variable suivant ($\mathbf{x} = \frac{\mathbf{q}}{F}$) dans la définition de la fonction de régularisation (A.6). En utilisant cette propriété de la fonction $f(\mathbf{p})$ et la borne du regret relatif (D.27) nous obtenons la borne du Théorème 4 :

$$\mathrm{Reg}(T) \leq \frac{f^*(\mathbf{0})}{\mu} + \frac{\mu FTV^2}{2}. \tag{D.10}$$

D.1.2 Preuve du Corollaire 7

La borne du regret définie dans l'équation (D.31) dépend des différents paramètres du système comme F, T ou encore μ. De plus, nous remarquons que lorsque nous calculons la limite du regret moyen pour un paramètre μ quelconque on obtient :

$$\limsup_{T\to\infty}\frac{1}{T}\mathrm{Reg}(T) = \frac{\mu FV^2}{2}. \tag{D.11}$$

Cela signifie que la propriété de non regret n'est pas garantie pour un pas μ quelconque.

Pour pallier à ce problème, nous devons déterminer le pas μ optimal, c'est à dire le pas qui minimise la borne du regret. Nous remarquons que la borne du regret est convexe en fonction de μ, il donc possible de calculer la dérivée et de chercher la valeur de μ qui annule cette dérivée. Ce calcul nous donne un pas optimal μ^* :

$$\mu^* = \sqrt{\frac{2f^*(\mathbf{0})}{TFV^2}}. \tag{D.12}$$

Maintenant que nous avons déterminé le pas optimal, nous devons vérifier que ce dernier garantit la propriété de non regret. Pour cela, il faut remplacer la valeur de μ par la valeur optimale μ^* dans la borne du regret, ce qui nous donne :

$$\mathrm{Reg}(T) \leq \sqrt{2TV^2Ff^*(\mathbf{0})}. \tag{D.13}$$

Nous pouvons déduire de l'équation ci-dessus que la propriété de non regret est bien garantie.

D.1.3 Preuve du Corollaire 8

Dans le cas où le temps de transmission n'est pas connu en avance nous allons utiliser l'astuce du *doubling-trick*. Pour cela, nous allons utiliser des fenêtres, $k = \{0, \ldots, \lceil \log_2 T \rceil\}$ de transmission dont la taille, $T_k = 2^k$ double à chaque fois. Puisque l'objet connaît la taille de chaque fenêtre, il peut calculer le pas optimal μ_k^* de chaque fenêtre en utilisant la formule (D.12). De plus, nous pouvons définir le regret dans chaque fenêtre comme :

$$\mathrm{Reg}_k(T) \leq \sqrt{2TV^2 2^k f^*(\mathbf{0})}. \tag{D.14}$$

Ainsi on trouve facilement que :

$$\sum_{k=1}^{\lceil \log_2 T \rceil} \mathrm{Reg}_k(T) \leq \sqrt{2TV^2 2^k f^*(\mathbf{0})} \sum_{k=1}^{\lceil \log_2 T \rceil} 2^k. \tag{D.15}$$

Il s'agit d'un suite géométrique, nous pouvons donc facilement en calculer la somme :

$$\sum_{k=1}^{\lceil \log_2 T \rceil} \mathrm{Reg}_k(T) \leq \sqrt{2TV^2 2^k f^*(\mathbf{0})} \frac{1-\sqrt{2}^{\lceil \log_2 T \rceil + 1}}{1-\sqrt{2}}. \tag{D.16}$$

Un rapide calcul nous montre que nous pouvons borner le terme de droite et nous obtenons :

$$\frac{1-\sqrt{2}^{\lceil \log_2 T \rceil + 1}}{1-\sqrt{2}} \leq \frac{1-\sqrt{2}}{1-\sqrt{2}}. \tag{D.17}$$

Et donc en regroupant les équations (D.16) et (D.17) nous trouvons :

$$\sum_{k=1}^{\lceil \log_2 T \rceil} \mathrm{Reg}_k(T) \leq \sqrt{2TV^2 2^k f^*(\mathbf{0})} \frac{\sqrt{2}}{\sqrt{2}-1}. \tag{D.18}$$

D.2 Preuves des résultats théoriques de l'Algorithme GMD dans le cas imparfait

Pour facilité la lecture de cette preuve, nous allons utiliser la notion de regret moyen par rapport à l'allocation de puissance fixe $\mathbf{q}$, $\mathrm{EReg}_{\mathbf{q}}$, définie comme :

$$\mathrm{EReg}_{\mathbf{q}}(T) = \mathbb{E}[\mathrm{Reg}_{\mathbf{q}}(T)]. \tag{D.19}$$

Maintenant que nous avons défini le regret moyen relatif à l'allocation de puissance $\mathbf{q}$, nous allons pouvoir détailler la preuve du Théorème 4.

D.2.1 Preuve du Théorème 5

La première étape de la preuve dans le cas du gradient imparfait est la même que dans le cas du gradient parfait. C'est à dire que nous allons utiliser la propriété de convexités des fonctions objectif pour borner le regret moyen relatif à l'allocation de puissance fixe $\mathbf{q}$:

$$\mathrm{EReg}_{\mathbf{q}}(T) \leq \mathbb{E}[\langle \nabla L_t(\mathbf{p}(t)) | \mathbf{p}(t) - \mathbf{q} \rangle], \quad \forall \mathbf{q}. \tag{D.20}$$

L'idée pour la prochaine étape est de lier le gradient des fonctions objectif, $\nabla L_t(\mathbf{p}(t))$, à l'estimateur non-biaisé que l'objet reçoit par feedback, $\tilde{\mathbf{v}}(t)$. Pour cela, nous allons utiliser le fait que le gradient peut s'écrire comme :

$$\nabla L_t(\mathbf{p}(t)) = \mathbb{E}[\tilde{\mathbf{v}}(t) | \tilde{\mathbf{v}}(t-1), \ldots, \tilde{\mathbf{v}}(1)], \quad \forall \mathbf{q}. \tag{D.21}$$

Nous pouvons donc remplacer le gradient par l'expression définie par l'équation (D.21) dans l'équation (D.20), ce qui nous donne :

$$\mathbb{E}[\langle \nabla L_t(\mathbf{p}(t)) | \mathbf{p}(t) - \mathbf{q} \rangle] = \mathbb{E}[\langle \mathbb{E}[\tilde{\mathbf{v}}(t) | \tilde{\mathbf{v}}(t-1), \ldots, \tilde{\mathbf{v}}(1)] | \mathbf{p}(t) - \mathbf{q} \rangle], \quad \forall \mathbf{q}. \tag{D.22}$$

Á une itération t donnée l'allocation de puissance $\mathbf{p}(t)$ dépend uniquement de la séquence de feedback $\tilde{\mathbf{v}}(t-1), \ldots, \tilde{\mathbf{v}}(1)$ et l'allocation de puissance $\mathbf{q}$ est constant, c'est pourquoi nous pouvons écrire :

$$\mathbb{E}[\langle \mathbb{E}[\tilde{\mathbf{v}}(t) | \tilde{\mathbf{v}}(t-1), \ldots, \tilde{\mathbf{v}}(1)] | \mathbf{p}(t) - \mathbf{q} \rangle] = \mathbb{E}[\mathbb{E}[\langle \tilde{\mathbf{v}}(t) | \tilde{\mathbf{v}}(t-1), \ldots, \tilde{\mathbf{v}}(1) | \mathbf{p}(t) - \mathbf{q} \rangle]], \quad \forall \mathbf{q}. \tag{D.23}$$

En utilisant la loi de l'espérance totale nous trouvons :

$$\mathbb{E}[\mathbb{E}[\langle \tilde{\mathbf{v}}(t) | \tilde{\mathbf{v}}(t-1), \ldots, \tilde{\mathbf{v}}(1) | \mathbf{p}(t) - \mathbf{q} \rangle]] = \mathbb{E}[\langle \tilde{\mathbf{v}}(t) | \mathbf{p}(t) - \mathbf{q} \rangle], \quad \forall \mathbf{q}. \tag{D.24}$$

Á partir de cette étape la preuve est la même que dans le cas du gradient parfait. Nous allons utiliser l'approximation de Taylor d'ordre 2 de la fonction $f^*(\mathbf{y})$ ainsi que l'inégalité de Fenchel ce qui nous donne :

$$\mathrm{EReg}_{\mathbf{q}} \leq \mathbb{E}\left[\frac{f^*(\mathbf{0})}{\mu} + \frac{\mu}{2} F \sum_{t=1}^{T} ||\tilde{\mathbf{v}}(t)||_{\infty}^2 \right], \quad \forall \mathbf{q}. \tag{D.25}$$

Les deux termes de l'addition étant indépendants nous pouvons séparer les espérances. Nous profitons de cette étape pour utiliser le fait que la norme de l'estimateur soit bornée et ainsi :

$$\mathrm{EReg}_{\mathbf{q}} \leq \mathbb{E}\left[\frac{f^*(\mathbf{0})}{\mu}\right] + \mathbb{E}\left[\frac{\mu}{2} F T \tilde{V}^2\right], \quad \forall \mathbf{q}. \tag{D.26}$$

Les deux termes à l'intérieur des espérances sont des constantes nous pouvons donc les enlever, de plus comme dans la preuve précédente, $f(\mathbf{q}) \leq 0$ ce qui nous donne au final la borne du Théorème 4 :

$$\mathrm{EReg} \leq \frac{f^*(\mathbf{0})}{\mu} + \frac{\mu}{2} F T \tilde{V}^2. \tag{D.27}$$

D.2.2 Preuve du Corollaire 9

Dans le cas où l'objet connaît le temps de transmission en avance, nous allons utiliser la même preuve que dans le cas du gradient parfait. La fonction objectif est convexe en fonction de μ nous allons donc déterminer la valeur optimale de ce pas. Après avoir calculé et annulé la dérivée (en fonction de μ) de la borne du regret nous trouvons la pas optimal suivant

$$\mu^* = \sqrt{\frac{2 f^*(\mathbf{0})}{T F \tilde{V}^2}}. \tag{D.28}$$

Maintenant que nous avons déterminé le pas optimal, nous devons vérifier que ce dernier garantit la propriété de non regret. Pour cela, il faut remplacer la valeur de μ par la valeur optimale μ^* dans la borne du regret moyen, ce qui nous donne :

$$\mathrm{EReg}(T) \leq \sqrt{2 T F \tilde{V}^2 f^*(\mathbf{0})}. \tag{D.29}$$

Nous pouvons donc déduire de l'équation ci-dessus que la propriété de non regret est bien garantie.

D.2.3 Preuve du Corollaire 10

Dans le cas où l'objet ne connaît pas le temps de transmission en avance nous allons, dans le cas du gradient bruité, utiliser un pas variable. Ce pas variable est défini comme : $\mu(t) = \frac{\alpha}{\sqrt{t}}$ où α est une constante positive. Toute l'idée de cette preuve est d'étudier un regret pondéré $\mathrm{WReg}(T)$ défini comme :

$$\mathrm{WReg}(T) = \mathbb{E}\left[\mu(t)(L_t(\mathbf{p}(t)) - L_t(\mathbf{q}))\right]. \tag{D.30}$$

Pour étudier le comportement du regret pondéré nous allons utiliser la même approche que pour le Théorème 1 ce qui nous donne la borne suivant :

$$\mathrm{WReg}(T) \leq f^*(\mathbf{0}) + \tilde{V}^2 F \sum_{t=1}^{T} \mu^2(t). \tag{D.31}$$

Pour borner le regret nous allons utiliser le critère de Hardy [Hardy, 1949] qui compare l'évolution d'une suite à sa suite pondérée par un pas variable respectant les conditions suivantes :

$\mu(t) \geq \mu(t+1)$ et $\frac{\sum_{t=1}^{T}\mu(t)}{\mu(T)} = \mathcal{O}(T)$. Ainsi si nous sommes en mesure de montrer que $\mathrm{WReg}(T)$ a la propriété de non regret alors $\mathrm{EReg}(T)$ aura la propriété de non regret. Ainsi en utilisant le Théorème 14 de [Hardy, 1949] nous trouvons :

$$\frac{\mathbb{E}[\mathrm{Reg}_{\mathbf{q}}(T)]}{T} \sim \frac{\mathrm{WReg}_{\mathbf{q}}(T)}{\sum_{t=1}^{T}\mu(t)} \leq \frac{f^*(\mathbf{0}) + \tilde{V}^2 F \sum_{t=1}^{T}\mu^2(t)}{\sum_{t=1}^{T}\mu(t)} \tag{D.32}$$

$$\leq \frac{f^*(\mathbf{0})}{\alpha\sqrt{T}} + \frac{\alpha\tilde{V}^2 F(1+\log T)}{\sqrt{T}}. \tag{D.33}$$

Nous pouvons en déduire que :

$$\mathrm{EReg} \leq \frac{1}{\sqrt{T}}\left[\frac{f^*(\mathbf{0})}{\alpha} + \alpha\tilde{V}^2 F\right] + \frac{\alpha\tilde{V}^2 F\log(T)}{\sqrt{T}}. \tag{D.34}$$

D.3 Preuves des résultats théoriques de l'Algorithme GMD_0

Dans cette section, nous allons présenter les résultats théoriques concernant l'algorithme GMD_0. Il faut tant un premier temps noté que toutes les propriétés présentées dans l'annexe C.1 sont encore valable dans le cas général.

D.3.1 Preuve du Théorème 6

Nous allons commencer par présenter la preuve du Théorème 6. Pour cela nous allons réutiliser les mêmes étapes que dans l'Annexe C.2.1 :

1. $\mathrm{EReg}_{\mathbf{q}}(T) \leq \mathbb{E}\left[\sum_{t=1}^{T} L_t(\mathbf{p}_\delta(t)) - L_t(\mathbf{q}_\delta)\right] + 3TK\delta + TK\delta A$ dans le Lemme D.1

3. $\mathbb{E}\left[\sum_{t=1}^{T} \widetilde{L}_t(\mathbf{p}_\delta(t)) - \widetilde{L}_t(\mathbf{q}_\delta)\right] \leq \mathbb{E}\left[\sum_{t=1}^{T} \langle \tilde{\mathbf{v}}(t) | \mathbf{p}_\delta(t) - \mathbf{q}_\delta\rangle\right]$ dans le Lemme D.2

4. $\sum_{t=1}^{T} \langle \tilde{\mathbf{v}}(t) | \mathbf{p}_\delta(t) - \mathbf{q}_\delta\rangle \leq \sum_{t=1}^{T} \langle \tilde{\mathbf{v}}(t) | \mathbf{p}_\delta(t) - \mathbf{p}_\delta(t+1)\rangle + \frac{H}{\mu}$ dans le Lemme D.3

5. $\sum_{t=1}^{T} \langle \tilde{\mathbf{v}}(t) | \mathbf{p}_\delta(t) - \mathbf{p}_\delta(t+1)\rangle \leq \frac{\mu}{M}\sum_{t=1}^{T} \|\tilde{\mathbf{v}}(t)\|_\infty^2$ dans le Lemme D.4

6. $\mathbb{E}\left[\sum_{t=1}^{T} \|\tilde{\mathbf{v}}(t)\|^2\right] \leq TS^2\left(\frac{S}{\delta} + L\right)^2$ dans le Lemme D.5

7. En combinant tous les lemmes précédents nous trouvons :
   $\mathrm{EReg}_{\mathbf{q}}(T) \leq \frac{H}{\mu} + \frac{\mu}{M}TS^2\left(\frac{B}{\delta} + K\right)^2 + TK\delta(3+A)$

où A est défini comme :

$$\|\mathbf{p}-\mathbf{p}_\delta\|_2 \leq A,\ \forall \mathbf{p}\in\mathscr{P},\ \forall \mathbf{p}_\delta\in\mathscr{P}_\delta. \tag{D.35}$$

Cette constante permet de borner l'écart qu'il peut y avoir entre l'espace faisable $\mathscr{P}$ et l'espace faisable réduit $\mathscr{P}_\delta$. Dans notre cas il est toujours possible de trouver cette constante car $\mathscr{P}_\delta \subset \mathscr{P}$.

Vu que l'allocation de puissance est aléatoire, l'objet connaît uniquement $\tilde{\mathbf{p}}(t)=\mathbf{p}_\delta(t)+\delta\mathbf{u}(t)$. La première étape consiste donc à relier la valeur de la fonction objectif aux points $\tilde{\mathbf{p}}(t)$, $\mathbf{q}\in\mathscr{P}$ et les valeurs prisent aux points $\mathbf{p}_\delta,\ \mathbf{q}_\delta\in\mathscr{P}_\delta$ où $\mathbf{q}_\delta$ est calculée en utilisant la fonction $\Delta(\mathbf{q}):\mathscr{P}_\delta\rightarrow\mathscr{P}$.

Lemme D.1. *Si l'algorithme GMD$_0$ est utilisé avec des fonctions objcetif K-Lipschitz et avec $\boldsymbol{p}(t)=\boldsymbol{p}_\delta(t)+\delta\boldsymbol{u}(t)$ alors le regret moyen est borné par :*

$$\mathrm{EReg}_{\boldsymbol{q}}(T)\leq\mathbb{E}\left[\sum_{t=1}^{T}L_t(\boldsymbol{p}_\delta(t))-L_t(\boldsymbol{q}_\delta)\right]+3KT\delta+KT\delta A. \tag{D.36}$$

Démonstration. Le regret moyen est calculé en fonction de l'allocation de puissance $\tilde{\mathbf{p}}(t)=\mathbf{p}_\delta(t)+\delta\mathbf{u}(t)$ et il peut s'écrire :

$$\mathrm{EReg}_{\mathbf{q}}(T)=\mathbb{E}\left[\sum_{t=1}^{T}L_t(\tilde{\mathbf{p}})-L_t(\mathbf{q}).\right] \tag{D.37}$$

Nous pouvons réécrire le regret comme :

$$\mathrm{EReg}_{\mathbf{q}}(T)=\mathbb{E}\left[\sum_{t=1}^{T}L_t(\tilde{\mathbf{p}}(t))-L_t(\mathbf{p}_\delta(t))+L_t(\mathbf{p}_\delta(t))-L_t(\mathbf{q})\right]. \tag{D.38}$$

La première étape est de comparer $L_t(\tilde{\mathbf{p}}(t))$ à $L_t(\mathbf{p}_\delta(t))$. Pour faire cela rappelons tout d'abord que $\tilde{\mathbf{p}}(t)=\mathbf{p}_\delta(t)+\delta\mathbf{u}(t)$ où $\mathbf{u}(t)\in\mathbb{S}$. En utilisant le fait que L_t est K-Lipschitz alors nous obtenons :

$$|L_t(\tilde{\mathbf{p}}(t))-L_t(\mathbf{p}_\delta(t))|=|L_t(\mathbf{p}_\delta(t)+\delta\mathbf{u}(t))-L_t(\mathbf{p}_\delta(t)|\leq K\delta\,\|\mathbf{u}(t)\|_2\,, \tag{D.39}$$

car les fonctions objectif sont toujours positives et que $\|\mathbf{u}(t)\|_2=1$ du au fait que $\mathbf{u}(t)\in\mathbb{S}$. En combinant (D.38) et (D.39) nous trouvons :

$$\mathrm{EReg}_{\mathbf{q}}(t)\leq\mathbb{E}\left[\sum_{t=1}^{T}L_t(\mathbf{p}_\delta(t))-L_t(\mathbf{q})\right]+KT\delta. \tag{D.40}$$

Maintenant nous devons comparer $L_t(\mathbf{q})$ à $L_t(\mathbf{q}_\delta)$. Nous remarquons que :

$$\mathbb{E}\left[\sum_{t=1}^{T}L_t(\mathbf{p}_\delta(t))-L_t(\mathbf{q})\right]=\mathbb{E}\left[\sum_{t=1}^{T}L_t(\mathbf{p}_\delta(t))+L_t(\mathbf{q}_\delta(t))-L_t(\mathbf{q})-L_t(\mathbf{q}_\delta(t))\right]. \tag{D.41}$$

où $\mathbf{q}_\delta$ est calculée en utilisant la fonction $\Delta(\mathbf{q}):\mathscr{P}_\delta\rightarrow\mathscr{P}$. En utilisant le fait que $\mathbf{q}_\delta=\Delta(\mathbf{q})$ et que $L_t(\mathbf{p})$ est K- Lipschitz nous obtenons :

$$|L_t(\mathbf{q}_\delta)-L_t(\mathbf{q})|=|L_t(\Delta(\mathbf{q}))-L_t(\mathbf{q})|\leq K\,\|\Delta(\mathbf{q})-\mathbf{q}\|_2\,,\quad\forall\mathbf{q}\in\mathscr{P}. \tag{D.42}$$

Pour rappel, nous avons A qui est défini comme :

$$\|\Delta(\mathbf{q}) - \mathbf{q}\|_2 \leq A, \tag{D.43}$$

où A est une constante positive. En combinant (D.41) et (D.43) nous obtenons :

$$\mathbb{E}\left[\sum_{t=1}^{T} L_t(\mathbf{p}_\delta(t)) - L_t(\mathbf{q})\right] \leq \mathbb{E}\left[\sum_{t=1}^{T} L_t(\mathbf{p}_\delta(t)) - L_t(\mathbf{q}_\delta)\right] + KT\delta A. \tag{D.44}$$

Maintenant nous devons comparer $L_t(\mathbf{p}_\delta(t))$ et $L_t(\mathbf{q}_\delta)$ à $\widetilde{L}_t(\mathbf{p}_\delta(t))$ et $\widetilde{L}_t(\mathbf{q}_\delta)$ respectivement. En effet, nous ne connaissons pas le gradient de $L_t(\mathbf{p}(t))$ mais de la Proposition C.2 nous connaissons un estimateur de $\nabla\widetilde{L}_t(\mathbf{p}(t)) = \frac{S}{\delta} L_t(\mathbf{p}_\delta(t))\mathbf{u}(t)$.

L'idée est de comparer $L_t(\mathbf{p}_\delta(t))$ à $\widetilde{L}_t(\mathbf{p}_\delta(t))$:

$$\mathbb{E}\left[\sum_{t=1}^{T} L_t(\mathbf{p}_\delta(t)) - L_t(\mathbf{q}_\delta)\right] = \mathbb{E}\left[\sum_{t=1}^{T} L_t(\mathbf{p}_\delta(t)) - \widetilde{L}_t(\mathbf{p}_\delta(t)) + \tilde{L}_t(\mathbf{p}_\delta(t)) - L_t(\mathbf{q}_\delta)\right] \tag{D.45}$$

La Proposition C.1 implique que :

$$\left|L_t(\mathbf{p}_\delta(t)) - \widetilde{L}_t(\mathbf{p}_\delta(t))\right| \leq K\delta. \tag{D.46}$$

Vu que L_t sont toujours positives nous pouvons combiner (D.45) et (D.46) et nous obtenons :

$$\mathbb{E}\left[\sum_{t=1}^{T} L_t(\mathbf{p}_\delta(t)) - L_t(\mathbf{q}_\delta)\right] \leq \mathbb{E}\left[\sum_{t=1}^{T} \widetilde{L}_t(\mathbf{p}_\delta(t)) - L_t(\mathbf{q}_\delta)\right] + KT\delta. \tag{D.47}$$

Nous devons aussi comparer $L_t(\mathbf{q}_\delta)$ à $\widetilde{L}_t(\mathbf{q}_\delta)$:

$$\mathbb{E}\left[\sum_{t=1}^{T} \widetilde{L}_t(\mathbf{p}_\delta(t)) - L_t(\mathbf{q}_\delta))\right] = \mathbb{E}\left[\sum_{t=1}^{T} \widetilde{L}_t(\mathbf{p}_\delta(t)) + \widetilde{L}_t(\mathbf{q}_\delta) - L_t(\mathbf{q}_\delta)) - \widetilde{L}_t(\mathbf{q}_\delta)\right] \tag{D.48}$$

En utilisant la Proposition C.1 nous trouvons :

$$|\widetilde{L}_t(\mathbf{q}_\delta) - L_t(\mathbf{p}_\delta)| \leq K\delta. \tag{D.49}$$

En combinant (D.48) et (D.49) nous trouvons :

$$\mathbb{E}\left[\sum_{t=1}^{T} \widetilde{L}_t(\mathbf{p}_\delta(t)) - L_t(\mathbf{q}_\delta)\right] \leq \mathbb{E}\left[\sum_{t=1}^{T} \widetilde{L}_t(\mathbf{p}_\delta(t)) - \widetilde{L}_t(\mathbf{q}_\delta)\right] + KT\delta. \tag{D.50}$$

Le Lemme D.1 suit en combinant (D.40), (D.44), (D.47) et (D.50). □

Le Lemme D.1 relie le regret moyen à l'espérance de la différence cumulée de $\widetilde{L}_t(\mathbf{p}_\delta(t))$ et $\widetilde{L}_t(\mathbf{q}_\delta)$. Nous avons montré que si $L_t(\mathbf{p}(t))$ est convexe par rapport à $\mathbf{p}(t)$ alors le fonction $\widetilde{L}_t(\mathbf{p}(t))$ est convexe par rapport à $\mathbf{p}(t)$ ainsi nous pouvons linéariser la borne du Lemme C.1 en utilisant le gradient de $\widetilde{L}_t(\mathbf{p}(t))$. Nous utilisons ainsi la Proposition C.2 pour remplacer le gradient de $\widetilde{L}_t(\mathbf{p}(t))$ par son estimateur $\tilde{\mathbf{v}}(t) = \frac{S}{\delta} L_t(\mathbf{p}(t) + \delta\mathbf{u}(t))\mathbf{u}(t)$.

Lemme D.2. *Si la fonction $\widetilde{L}_t(\boldsymbol{p}_\delta(t))$ est convexe par rapport à $\boldsymbol{p}_\delta(t)$ et si l'estimateur $\tilde{\boldsymbol{v}}(t)$ est défini dans la Proposition C.2 alors nous obtenons :*

$$\mathbb{E}\left[\widetilde{L}_t(\boldsymbol{p}_\delta(t)) - \widetilde{L}_t(\boldsymbol{q}_\delta)\right] \leq \mathbb{E}\left[\sum_{t=1}^{T} \langle \tilde{\boldsymbol{v}}(t) | \boldsymbol{p}_\delta(t) - \boldsymbol{q}_\delta \rangle\right]. \tag{D.51}$$

Démonstration. Vu que la fonction $\widetilde{L}_t(\mathbf{p}_\delta(t))$ est convexe nous pouvons écrire :

$$\mathbb{E}\left[\sum_{t=1}^{T} \widetilde{L}_t(\mathbf{p}_\delta(t)) - \widetilde{L}_t(\mathbf{q}_\delta)\right] \leq \mathbb{E}\left[\sum_{t=1}^{T} \langle \nabla \widetilde{L}_t(\mathbf{p}_\delta(t)) | \mathbf{p}_\delta(t) - \mathbf{q}_\delta \rangle\right] \quad \forall \mathbf{p}_\delta(t), \mathbf{q}_\delta \in \mathscr{P}_\delta. \tag{D.52}$$

Nous utilisons la Proposition C.2 du gradient de $\widetilde{L}_t(\mathbf{p}_\delta(t))$ en (D.52) et nous obtenons :

$$\mathbb{E}\left[\sum_{t=1}^{T} \langle \nabla \widetilde{L}_t(\mathbf{p}_\delta(t)) | \mathbf{p}_\delta(t) - \mathbf{q}_\delta \rangle\right] = \mathbb{E}\left[\sum_{t=1}^{T} \langle \mathbb{E}[\tilde{\mathbf{v}}(t) | \mathbf{u}(1), ..., \mathbf{u}(t-1)] | \mathbf{p}_\delta(t) - \mathbf{q}_\delta \rangle\right] \quad \forall \mathbf{p}_\delta(t), \mathbf{q}_\delta \in \mathscr{P}_\delta. \tag{D.53}$$

Pour chaque instant t, l'allocation de puissance $\mathbf{p}_\delta(t)$ est totalement déterminée par $\{\mathbf{u}(1), \ldots, \mathbf{u}(t-1)\}$ et $\mathbf{q}_\delta$ est constante. Ainsi, la propriété suivante est vraie :

$$\mathbb{E}\left[\sum_{t=1}^{T} \langle \mathbb{E}[\tilde{\mathbf{v}}(t) | \mathbf{u}(1), ..., \mathbf{u}(t-1)] | \mathbf{p}_\delta(t) - \mathbf{q}_\delta \rangle\right] = \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{E}[\langle \tilde{\mathbf{v}}(t) | \mathbf{p}_\delta(t) - \mathbf{q}_\delta \rangle | \mathbf{u}(1), ..., \mathbf{u}(t-1)]\right] \quad \forall \mathbf{p}_\delta(t), \mathbf{q}_\delta \in \mathscr{P}_\delta. \tag{D.54}$$

Nous pouvons maintenant déplacer l'espérance de droite à l'intérieur de la somme car tous les $\mathbf{u}(t)$ sont indépendants :

$$\mathbb{E}\left[\sum_{t=1}^{T} \mathbb{E}\langle \tilde{\mathbf{v}}(t) | \mathbf{p}_\delta(t) - \mathbf{q}_\delta \rangle | \mathbf{u}(1), ..., \mathbf{u}(t-1)]\right] = \sum_{t=1}^{T} \mathbb{E}[\left[\mathbb{E}[\langle \tilde{\mathbf{v}}(t) | \mathbf{p}_\delta(t) - \mathbf{q}_\delta \rangle | \mathbf{u}(1), ..., \mathbf{u}(t-1)]\right] \quad \forall \mathbf{p}_\delta(t), \mathbf{q}_\delta \in \mathscr{P}_\delta. \tag{D.55}$$

En utilisant le Théorème de l'espérance totale nous obtenons :

$$\mathbb{E}\left[\mathbb{E}[\langle \tilde{\mathbf{v}}(t) | \mathbf{p}_\delta(t) - \mathbf{q}_\delta \rangle | \mathbf{u}(1), ..., \mathbf{u}(t-1)]\right] = \mathbb{E}\left[\langle \tilde{\mathbf{v}}(t) | \mathbf{p}_\delta(t) - \mathbf{q}_\delta \rangle\right] \quad \forall \mathbf{p}_\delta(t), \mathbf{q}_\delta \in \mathscr{P}_\delta. \tag{D.56}$$

Finalement, en faisant la somme sur tous les instants $t \leq T$ de (D.56) et en déplaçant à droite l'espérance à l'extérieur de la somme car les $\mathbf{u}(t)$ sont indépendants :

$$\sum_{t=1}^{T} \mathbb{E}\left[\langle \tilde{\mathbf{v}}(t) | \mathbf{p}_\delta(t) - \mathbf{q}_\delta \rangle\right] = \mathbb{E}\left[\sum_{t=1}^{T} \langle \tilde{\mathbf{v}}(t) | \mathbf{p}_\delta(t) - \mathbf{q}_\delta \rangle\right] \quad \forall \mathbf{p}_\delta(t), \mathbf{q}_\delta \in \mathscr{P}_\delta. \tag{D.57}$$

En utilisant le même procédé dans les équations (D.52) et (D.57) nous obtenons :

$$\mathbb{E}\left[\sum_{t=1}^{T} \widetilde{L}_t(\mathbf{p}_\delta(t)) - \widetilde{L}_t(\mathbf{q}_\delta)\right] \leq \mathbb{E}\left[\sum_{t=1}^{T} \langle \tilde{\mathbf{v}}(t) | \mathbf{p}_\delta(t) - \mathbf{q}_\delta \rangle\right]. \tag{D.58}$$

□

Á partir de cette étape, nous allons nous concentrer sur le terme de droite à l'intérieur de l'espérance du D.2. Pour borner ce terme nous devons introduire une définition importante concernant l'étape de projection exponentielle [Shalev-Shwartz, 2011].

Definition D.1. L'étape de projection est définie comme :

$$p^s(t) = \nabla f^*(\mathbf{y})(t), \quad \forall s \tag{D.59}$$

est équivalente à :

$$\mathbf{p}_\delta(t) = \underset{\hat{\mathbf{p}} \in \mathscr{P}_\delta}{\arg\max} \{\langle \mathbf{y}(t) | \hat{\mathbf{p}} \rangle - f(\hat{\mathbf{p}})\} \tag{D.60}$$

où $f(\mathbf{p}_\delta)$ est le fonction de régularisation entropique associée à l'espace réduit définie en (4.22).

Pour borner la somme dans l'espérance du terme de droite du Lemme D.2, nous allons procéder en deux étapes. Premièrement, nous allons utiliser la Proposition D.1 pour borner la somme des $\langle \tilde{\mathbf{v}}(t) | \mathbf{q}_\delta \rangle$ par la somme des $\langle \tilde{\mathbf{v}}(t) | \mathbf{p}_\delta(t+1) \rangle$. Deuxièmement, nous allons borner la différence cumulée entre $\langle \tilde{\mathbf{v}}(t) | \mathbf{p}_\delta(t) \rangle$ et $\langle \tilde{\mathbf{v}}(t) | \mathbf{p}_\delta(t+1) \rangle$.

Proposition D.1. *Si l'allocation de puissance $\boldsymbol{p}_\delta(t)$ est définie par :*

$$\boldsymbol{p}_\delta(t) = \underset{\hat{\boldsymbol{p}} \in \mathscr{P}_\delta}{\arg\max} \{\langle \mathbf{y}(t) | \hat{\boldsymbol{p}} \rangle - f(\hat{\boldsymbol{p}})\}, \quad \forall t \geq 1 \tag{D.61}$$

alors nous avons :

$$-\sum_{t=1}^{T} \langle \tilde{\boldsymbol{v}}(t) | \boldsymbol{q}_\delta \rangle \;\leq\; -\sum_{t=1}^{T} \langle \tilde{\boldsymbol{v}}(t) | \boldsymbol{p}_\delta(t+1) \rangle - \frac{f(\boldsymbol{p}_\delta(2))}{\mu} + \frac{f(\boldsymbol{q}_\delta)}{\mu}, \quad \forall \boldsymbol{q}_\delta \in \mathscr{P}_\delta. \tag{D.62}$$

Démonstration. La première étape est de noter que (D.61) est équivalente à :

$$\mathbf{p}_\delta(t+1) = \underset{\hat{\mathbf{p}} \in \mathscr{P}_\delta}{\arg\max} \left\{ -\mu \sum_{i=1}^{t} \langle \tilde{\mathbf{v}}(i) | \hat{\mathbf{p}} \rangle - f(\hat{\mathbf{p}}) \right\}, \quad \forall t \geq 1. \tag{D.63}$$

Cela vient de la définition de $\mathbf{y}(t)$:

$$\mathbf{y}(t) = \begin{cases} 0, & \text{t=1} \\ \mathbf{y}(t-1) - \mu \tilde{\mathbf{v}}(t-1), & \text{t >1,} \end{cases} \tag{D.64}$$

ce qui nous donne $\mathbf{y}(t+1) = -\mu \sum_{t=1}^{t} \tilde{\mathbf{v}}(i)$ pour tout $t \geq 1$.

Nous allons procéder par récurrence en commençant par $T = 1$. En utilisant la définition de l'étape de projection nous avons :

$$-\mu \langle \tilde{\mathbf{v}}(1) | \mathbf{q}_\delta \rangle - f(\mathbf{q}_\delta) \;\leq\; -\mu \langle \tilde{\mathbf{v}}(1) | \mathbf{p}_\delta(2) \rangle - f(\mathbf{p}_\delta(2)), \quad \forall \mathbf{q}_\delta \in \mathscr{P}_\delta \tag{D.65}$$

$$-\langle \tilde{\mathbf{v}}(1) | \mathbf{q}_\delta \rangle \leq \; -\langle \tilde{\mathbf{v}}(1) | \mathbf{p}_\delta(2) \rangle - \frac{f(\mathbf{p}_\delta(2))}{\mu} + \frac{f(\mathbf{q}_\delta)}{\mu}, \quad \forall \mathbf{q}_\delta \in \mathscr{P}_\delta. \tag{D.66}$$

Alors, la propriété (D.62) est vraie pour $T = 1$.

Maintenant, nous faisons l'hypothèse que la propriété est vraie pour $T-1$, et nous allons vérifier que la propriété est vraie à l'instant T :

$$-\sum_{t=1}^{T-1} \langle \tilde{\mathbf{v}}(t) | \mathbf{q}_\delta \rangle \;\leq\; -\sum_{t=1}^{T-1} \langle \tilde{\mathbf{v}}(t) | \mathbf{p}_\delta(t+1) \rangle - \frac{f(\mathbf{p}_\delta(2))}{\mu} + \frac{f(\mathbf{q}_\delta)}{\mu}, \quad \forall \mathbf{q}_\delta \in \mathscr{P}_\delta. \tag{D.67}$$

En additionnant $-\langle\tilde{\mathbf{v}}(T)|\mathbf{p}_\delta(T+1)\rangle$ de chaque côté nous obtenons :

$$-\sum_{t=1}^{T-1}\langle\tilde{\mathbf{v}}(t)|\mathbf{q}_\delta\rangle-\langle\tilde{\mathbf{v}}(T)|\mathbf{p}_\delta(T+1)\rangle \;\leq\; -\sum_{t=1}^{T}\langle\tilde{\mathbf{v}}(t)|\mathbf{p}_\delta(t+1)\rangle-\frac{f(\mathbf{p}_\delta(2))}{\mu}+\frac{f(\mathbf{q}_\delta)}{\mu},\quad\forall\mathbf{q}_\delta\in\mathscr{P}_\delta. \tag{D.68}$$

L'équation précédente est vraie pour tous $\mathbf{q}_\delta\in\mathscr{P}_\delta$ et donc elle est vraie pour $\mathbf{q}_\delta=\mathbf{p}_\delta(T+1)$. Après avoir remis en ordre les termes nous trouvons :

$$-\mu\sum_{t=1}^{T}\langle\tilde{\mathbf{v}}(t)|\mathbf{p}_\delta(T+1)\rangle-f(\mathbf{p}_\delta(T+1)) \;\leq\; -\mu\sum_{t=1}^{T}\langle\tilde{\mathbf{v}}(t)|\mathbf{p}_\delta(t+1)\rangle-f(\mathbf{p}_\delta(2)). \tag{D.69}$$

Nous remarquons, de (D.68), que :

$$-\mu\sum_{t=1}^{T}\langle\tilde{\mathbf{v}}(t)|\mathbf{q}_\delta\rangle-f(\mathbf{q}_\delta)\leq-\mu\sum_{t=1}^{T}\langle\tilde{\mathbf{v}}(t)|\mathbf{p}_\delta(T+1)\rangle-f(\mathbf{p}_\delta(T+1)),\quad\forall\mathbf{q}_\delta\in\mathscr{P}_\delta. \tag{D.70}$$

En utilisant l'équation (D.69) et (D.70), la propriété suivante est vraie :

$$-\mu\sum_{t=1}^{T}\langle\tilde{\mathbf{v}}(t)|\mathbf{q}_\delta\rangle-f(\mathbf{q}_\delta) \;\leq\; -\mu\sum_{t=1}^{T}\langle\tilde{\mathbf{v}}(t)|\mathbf{p}_\delta(t+1)\rangle-f(\mathbf{p}_\delta(2)),\quad\forall\mathbf{q}_\delta\in\mathscr{P}_\delta. \tag{D.71}$$

$$-\sum_{t=1}^{T}\langle\tilde{\mathbf{v}}(t)|\mathbf{q}_\delta\rangle \;\leq\; -\mu\sum_{t=1}^{T}\langle\tilde{\mathbf{v}}(t)|\mathbf{p}_\delta(t+1)\rangle-\frac{f(\mathbf{p}_\delta(2))}{\mu}+\frac{f(\mathbf{q}_\delta)}{\mu},\quad\forall\mathbf{q}_\delta\in\mathscr{P}_\delta. \tag{D.72}$$

En conclusion, (D.62) est vraie pour tous les $T\geq 1$. □

Lemme D.3. *Si l'étape de projection est définie comme :*

$$p^s(t)=\nabla^s f^*(\mathbf{y})(t),\quad\forall t \tag{D.73}$$

alors :

$$\sum_{t=1}^{T}\langle\tilde{\boldsymbol{v}}(t)|\boldsymbol{p}_\delta(t)-\boldsymbol{q}_\delta\rangle\leq\sum_{t=1}^{T}\langle\tilde{\boldsymbol{v}}(t)|\boldsymbol{p}_\delta(t)\rangle-H, \tag{D.74}$$

où $H=\min_{\boldsymbol{p}_\delta\in\mathscr{P}_\delta}f(\boldsymbol{p}_\delta)$.

Démonstration. Premièrement, nous remarquons que $f(\mathbf{p}_\delta)$ définie en (C.51) est toujours négative nous pouvons donc borner(D.62) par :

$$-\sum_{t=1}^{T}\langle\tilde{\mathbf{v}}(t)|\mathbf{q}_\delta\rangle \;\leq\; -\sum_{t=1}^{T}\langle\tilde{\mathbf{v}}(t)|\mathbf{p}_\delta(t+1)\rangle-\frac{f(\mathbf{p}_\delta(2))}{\mu},\quad\forall\mathbf{q}_\delta\in\mathscr{P}_\delta. \tag{D.75}$$

Nous supposons maintenant que la fonction f est bornée et nous notons :

$$H=\min_{p_\delta\in\mathscr{P}_\delta}f(\mathbf{p}_\delta). \tag{D.76}$$

Nous pouvons maintenant combiner (D.75) et (D.76) et nous trouvons :

$$-\sum_{t=1}^{T}\langle\tilde{\mathbf{v}}(t)|\mathbf{q}_\delta\rangle \;\leq\; -\sum_{t=1}^{T}\langle\tilde{\mathbf{v}}(t)|\mathbf{p}_\delta(t+1)\rangle-\frac{H}{\mu},\quad\forall\mathbf{q}_\delta\in\mathscr{P}_\delta. \tag{D.77}$$

□

La dernière étape pour borner le regret moyen est de borner la somme du côté droit de l'équation (D.74) du Lemme C.3. Pour borner cette somme, nous allons utiliser la Proposition C.3, quelques résultats de l'optimisation et l'inégalité de Cauchy-Schwartz.

Lemme D.4. *Si l'étape de projection est définie par :*

$$p^s(t) = \nabla^s f^*(\mathbf{y}(t)), \quad \forall s \tag{D.78}$$

alors :

$$\sum_{t=1}^{T} \langle \tilde{\boldsymbol{v}}(t) | \boldsymbol{p}_\delta(t) \rangle - \boldsymbol{p}_\delta(t+1) \leq \frac{\mu}{M} \sum_{t=1}^{T} \|\tilde{\boldsymbol{v}}(t)\|_\infty^2. \tag{D.79}$$

Démonstration. Pour borner la différence entre $\langle \tilde{\mathbf{v}}(t) | \mathbf{p}_\delta(t) \rangle$ et $\langle \tilde{\mathbf{v}}(t) | \mathbf{p}_\delta(t+1) \rangle$ nous allons commencer par rappeler quelques notations :

$$F_t(\mathbf{p}_\delta) = -\mu \sum_{i=1}^{t-1} \langle \tilde{\mathbf{v}}(i) | \mathbf{p}_\delta \rangle - f(\mathbf{p}_\delta). \tag{D.80}$$

La fonction $F_t(\mathbf{p}_\delta)$ est une somme de fonctions linéaires en $\mathbf{p}_\delta$ et de la fonction $-f(\mathbf{p}_\delta)$ qui est une fonction M-fortement régulière par rapport à la norme $\|\cdot\|_\infty$ [Shalev-Shwartz, 2011]. [1] L'addition de fonctions linéaires et d'une fonction fortement régulière est aussi fortement régulière.

En utilisant la propriété de forte régularité de la fonction $F_t(\mathbf{p}_\delta)$ nous pouvons écrire :

$$F_t(\mathbf{p}_\delta(t+1)) \leq F_t(\mathbf{p}_\delta(t)) + \langle \nabla F_t(\mathbf{p}_\delta(t)) | \mathbf{p}_\delta(t+1) - \mathbf{p}_\delta(t) \rangle - \frac{M}{2} \left\| \mathbf{p}_\delta(t+1) - \mathbf{p}_\delta(t) \right\|_\infty^2 \tag{D.82}$$

nous observons que $\mathbf{p}_\delta(t) = \arg\max_{\mathbf{p}_\delta \in \mathscr{P}_\delta} \left(F_t(\mathbf{p}_\delta) \right)$ alors la théorie de l'optimisation convexe nous dit que [Boyd and Vandenberghe, 2004] :

$$\langle \nabla F_t \mathbf{p}_\delta(t) | \mathbf{p}_\delta(t+1) - \mathbf{p}_\delta(t) \rangle \leq 0. \tag{D.83}$$

En utilisant l'équation (D.83) dans l'équation (D.82) et en substituant $F_t(\mathbf{p}_\delta)$ par sa définition nous trouvons :

$$-\mu \sum_{i=1}^{t-1} \langle \tilde{\mathbf{v}}(i) | \mathbf{p}_\delta(t+1) \rangle - f(\mathbf{p}_\delta(t+1) \leq -\mu \sum_{i=1}^{t-1} \langle \tilde{\mathbf{v}}(i) | \mathbf{p}_\delta(t) \rangle - \frac{M \left\| \mathbf{p}_\delta(t+1) - \mathbf{p}_\delta(t) \right\|_\infty^2}{2} - f(\mathbf{p}_\delta(t)). \tag{D.84}$$

Nous pouvons faire la même chose en observant que $\mathbf{p}_\delta(t+1) = \arg\max_{\mathbf{p}_\delta \in \mathscr{P}_\delta} \left(F_t(\mathbf{p}_\delta) \right)$:

$$F_{t+1}(\mathbf{p}_\delta(t)) \leq F_{t+1}(\mathbf{p}_\delta(t+1)) + \langle \nabla F_{t+1}(\mathbf{p}_\delta(t+1)) | \mathbf{p}_\delta(t) - \mathbf{p}_\delta(t+1) \rangle - \frac{M}{2} \left\| \mathbf{p}_\delta(t+1) - \mathbf{p}_\delta(t) \right\|_\infty^2. \tag{D.85}$$

1. Une fonction $f(\mathbf{p}) : \mathbb{R}^S \to \mathbb{R}$ est A-fortement régulière par rapport à la norme $\|\cdot\|_\infty$ ssi :

$$f(\mathbf{p}+\mathbf{q}) \leq f(\mathbf{p}) + \langle \nabla f(\mathbf{p}) | \mathbf{p} \rangle - \mathbf{p} \rangle - \frac{A}{2} \|\mathbf{p} - \mathbf{p}\|_\infty^2. \tag{D.81}$$

Intuitivement le notion de forte régularité implique que la fonction f est majorée par un plan tangent.

Nous pouvons aussi substituer $F_{t+1}(\mathbf{p}_\delta)$ par sa définition et en utilisant le fait que $\mathbf{p}_\delta(t+1)$ minimise, par définition, $F_{t+1}(\mathbf{p}_\delta)$ pour obtenir :

$$-\mu\sum_{i=1}^{t}\langle\tilde{\mathbf{v}}(i)|\mathbf{p}_\delta(t)\rangle - f(\mathbf{p}_\delta(t)) \leq -\mu\sum_{i=1}^{t}\langle\tilde{\mathbf{v}}(i)|\mathbf{p}_\delta(t+1)\rangle - f(\mathbf{p}_\delta(t+1) - \frac{M}{2}\left\|\mathbf{p}_\delta(t+1)-\mathbf{p}_\delta(t)\right\|_\infty^2 \quad \text{(D.86)}$$

Ainsi, en calculant la somme des équations (D.84) et (D.86), nous remarquons que $f(\mathbf{p}_\delta(t+1)$ et $f(\mathbf{p}_\delta(t))$ s'annulent :

$$-\mu\sum_{i=1}^{t-1}\langle\tilde{\mathbf{v}}(i)|\mathbf{p}_\delta(t+1)\rangle - \mu\sum_{i=1}^{t}\langle\tilde{\mathbf{v}}(i)|\mathbf{p}_\delta(t)\rangle \leq -\mu\sum_{i=1}^{t-1}\langle\tilde{\mathbf{v}}(i)|\mathbf{p}_\delta(t)\rangle - \mu\sum_{i=1}^{t}\langle\tilde{\mathbf{v}}(i)|\mathbf{p}_\delta(t+1)\rangle - \frac{M}{2}\left\|\mathbf{p}_\delta(t+1)-\mathbf{p}_\delta(t)\right\|_\infty^2. \quad \text{(D.87)}$$

En réarrangement les termes nous obtenons :

$$-\mu\sum_{i=1}^{t}\langle\tilde{\mathbf{v}}(i)|\mathbf{p}_\delta(t)\rangle + \mu\sum_{i=1}^{t-1}\langle\tilde{\mathbf{v}}(i)|\mathbf{p}_\delta(t)\rangle \leq -\mu\sum_{i=1}^{t}\langle\tilde{\mathbf{v}}(i)|\mathbf{p}_\delta(t+1)\rangle + \mu\sum_{i=1}^{t-1}\langle\tilde{\mathbf{v}}(i)|\mathbf{p}_\delta(t+1)\rangle - M\left\|\mathbf{p}_\delta(t+1)-\mathbf{p}_\delta(t)\right\|_\infty^2, \quad \text{(D.88)}$$

$$-\mu\langle\tilde{\mathbf{v}}(t)|\mathbf{p}_\delta(t)\rangle \leq -\mu\langle\tilde{\mathbf{v}}(t)|\mathbf{p}_\delta(t+1)\rangle - M\left\|\mathbf{p}_\delta(t+1)-\mathbf{p}_\delta(t)\right\|_\infty^2. \quad \text{(D.89)}$$

De l'équation (D.89) nous avons une borne inférieure de $\langle\tilde{\mathbf{v}}(t)|\mathbf{p}_\delta(t+1)-\mathbf{p}_\delta(t)\rangle$, qui est définie comme suit :

$$\frac{M}{\mu}\left\|\mathbf{p}_\delta(t+1)-\mathbf{p}_\delta(t)\right\|_\infty^2 \leq \langle\tilde{\mathbf{v}}(t)|\mathbf{p}_\delta(t+1)-\mathbf{p}_\delta(t)\rangle. \quad \text{(D.90)}$$

Nous utilisons inégalité de Cauchy-Schwartz, pour trouver une borne du terme de droite de l'équation (D.90)

$$|\langle\tilde{\mathbf{v}}(t)|\mathbf{p}_\delta(t+1)-\mathbf{p}_\delta(t)\rangle| \leq \left\|\mathbf{p}_\delta(t+1)-\mathbf{p}_\delta(t)\right\|_2 \|\tilde{\mathbf{v}}(t)\|_\infty. \quad \text{(D.91)}$$

De l'équation, (D.90) nous pouvons déduire que le terme $\langle\tilde{\mathbf{v}}(t)|\mathbf{p}_\delta(t+1)-\mathbf{p}_\delta(t)\rangle$ est positif, car μ est strictement positif et la norme $\left\|\mathbf{p}_\delta(t+1)-\mathbf{p}_\delta(t)\right\|_\infty^2$ est aussi positive. C'est pourquoi nous pouvons retirer la valeur absolue de l'équation (D.91) et nous obtenons :

$$\langle\tilde{\mathbf{v}}(t)|\mathbf{p}_\delta(t+1)-\mathbf{p}_\delta(t)\rangle \leq \left\|\mathbf{p}_\delta(t+1)-\mathbf{p}_\delta(t)\right\|_\infty \|\tilde{\mathbf{v}}(t)\|_\infty. \quad \text{(D.92)}$$

Nous avons une borne du supérieure du terme $\langle\tilde{\mathbf{v}}(t)|\mathbf{p}_\delta(t+1)-\mathbf{p}_\delta(t)\rangle$ dans (D.90) et une borne inférieure dans l'équation (D.92). En regroupant ces bornes nous obtenons :

$$M\left\|\mathbf{p}_\delta(t+1)-\mathbf{p}_\delta(t)\right\|_\infty^2 \leq \langle\tilde{\mathbf{v}}(t)|\mathbf{p}_\delta(t+1)-\mathbf{p}_\delta(t)\rangle \leq \left\|\mathbf{p}_\delta(t)-\mathbf{p}_\delta(t+1)\right\|_\infty \|\tilde{\mathbf{v}}(t)\|_\infty. \quad \text{(D.93)}$$

Nous déduisons de cette équation que la norme de la différence entre les vecteurs $\mathbf{p}_\delta(t)$ et $\mathbf{p}_\delta(t+1)$ est bornée par :

$$\left\|\mathbf{p}_\delta(t)-\mathbf{p}_\delta(t+1)\right\|_\infty \leq \mu\frac{1}{M}\|\tilde{\mathbf{v}}(t)\|_\infty. \quad \text{(D.94)}$$

Finalement, nous utilisons les équations (D.94) et (D.91) pour trouver la borne finale :

$$\langle\tilde{\mathbf{v}}(t)|\mathbf{p}_\delta(t+1)-\mathbf{p}_\delta(t)\rangle \leq \mu\frac{1}{M}\|\tilde{\mathbf{v}}(t)\|_\infty^2. \quad \text{(D.95)}$$

Nous pouvons utiliser cette équation pour borner la somme sur t :

$$\sum_{t=1}^{T} \langle \tilde{\mathbf{v}}(t) | \mathbf{p}_\delta(t+1) - \mathbf{p}_\delta(t) \rangle \leq \mu \frac{1}{M} \sum_{t=1}^{T} \|\tilde{\mathbf{v}}(t)\|_\infty^2. \tag{D.96}$$

□

En conclusion les Lemmes D.1-D.4 nous trouvons la borne suivante du regret moyen :

$$\mathrm{EReg}_{\mathbf{q}}(T) \leq \mathbb{E}\left[\frac{H}{\mu} + \frac{\mu}{M} \sum_{t=1}^{S} \|\tilde{\mathbf{v}}(t)\|_\infty^2 + TK\delta(3+A) \right]. \tag{D.97}$$

Dans la borne ci-dessus tous les termes ne dépendent pas de la variable de l'estimateur à l'exception de $\sum_{t=1}^{S} \|\tilde{\mathbf{v}}(t)\|_\infty^2$. Nous pouvons donc sortir les termes de l'espérance ce que nous donne :

$$\mathrm{EReg}_{\mathbf{q}}(T) \leq \frac{H}{\mu} + \frac{\mu}{M} \mathbb{E}\left[\sum_{t=1}^{S} \|\tilde{\mathbf{v}}(t)\|_\infty^2 \right] + TK\delta(3+A). \tag{D.98}$$

Maintenant, nous devons trouver une borne de la norme de l'espérance l'estimateur de $\tilde{\mathbf{v}}(t)$.

Lemme D.5. *Si la fonction $L_t(\boldsymbol{p})$ est K-Lipschitz et que la fonction est bornée, i.e.*

$$B = \max_{t \in \{1,\dots,T\}, \boldsymbol{p} \in \mathscr{P}} L_t(\boldsymbol{p}) \tag{D.99}$$

alors l'espérance de la somme des estimateurs $\tilde{\boldsymbol{v}}(t) = \frac{S}{\delta} L_t(\boldsymbol{p}_\delta(t) + \delta \boldsymbol{u}(t)) \boldsymbol{u}(t)$ est bornée par :

$$\mathbb{E}\left[\sum_{t=1}^{T} \|\tilde{\boldsymbol{v}}(t)\|_\infty^2 \right] \leq TS^2 \left(\frac{S}{\delta} + K \right)^2. \tag{D.100}$$

Démonstration. Nous substituons l'estimateur du gradient $\tilde{\mathbf{v}}(t)$ par sa définition dans l'équation de la Proposition C.2 et nous remarquons que $\mathbf{u}(t)$ est tiré de manière uniforme sur la sphère Euclidienne unitaire :

$$\mathbb{E}\left[\sum_{t=1}^{T} \|\tilde{\mathbf{v}}(t)\|_\infty^2 \right] = \mathbb{E}\left[\sum_{t=1}^{T} \left\| \frac{S}{\delta} L_t(\mathbf{p}_\delta(t) + \delta \mathbf{u}(t)) \mathbf{u}(t) \right\|_\infty^2 \right]. \tag{D.101}$$

Du au fait que les fonctions $L_t(\mathbf{p}_\delta)$ sont K-Lipschitz, nous obtenons :

$$L_t(\mathbf{p}_\delta(t) + \delta \mathbf{u}(t)) \leq L_t(\mathbf{p}_\delta(t)) + K\delta \quad \forall \mathbf{u}(t) \tag{D.102}$$

et en utilisant la borne B nous obtenons :

$$L_t(\mathbf{p}_\delta(t) + \delta \mathbf{u}(t)) \leq B + K\delta. \tag{D.103}$$

En substituant la borne de $L_t(\mathbf{p}_\delta(t) + \delta \mathbf{u}(t))$ dans (D.101) nous obtenons :

$$\mathbb{E}\left[\sum_{t=1}^{T} \frac{S^2}{\delta^2} ||L_t(\mathbf{p}_\delta(t) + \delta \mathbf{u}(t))||_\infty^2 \right] \leq \mathbb{E}\left[\sum_{t=1}^{T} \frac{S^2}{\delta^2} (B + K\delta)^2 \right] \tag{D.104}$$

$$= TS^2 \left(\frac{B}{\delta} + K \right)^2. \tag{D.105}$$

En combinant les équations (D.105) et (D.101) le Lemme D.5 est prouvé. □

La dernière étape consiste à rassembler les Lemmes 6-10 pour trouver la bornne finale du regret :

$$\mathrm{EReg}(T) \leq Z(\mu,\delta) \tag{D.106}$$

où $Z(\mu,\delta)$ est défini comme :

$$Z(\mu,\delta) = \frac{H}{\mu} + \frac{\mu T S^2}{M}\left(\frac{B}{\delta}+K\right)^2 + KT\delta\ (3+A). \tag{D.107}$$

D.3.2 Preuve du Corollaire 11

Il faut dans un premier temps remarquer que la borne du regret définie par (D.107) est linéaire en T. Nous devons donc déterminer des paramètres optimaux δ et μ tel que la croissance du regret est plus faible que $\mathcal{O}(T)$.

Pour faire cela, nous allons commencer par optimiser la borne du regret en fonction de μ. En effet, la borne du regret est convexe en fonction de μ, nous pouvons donc déterminer le pas μ optimal en calculant et annulant la dérivé de la borne. Nous devons maintenant calculer et annuler la dérivé partielle de la fonction Z par rapport à μ :

$$\frac{\partial Z(\mu,\delta)}{\partial \mu} = -\frac{H}{2\mu^2} + \frac{TS^2}{M}\left(\frac{B}{\delta}+K\right)^2 \tag{D.108}$$

Nous pouvons en déduire que :

$$\mu^* = \sqrt{\frac{HM}{2T}} S^{-1}\left(\frac{B}{\delta}+K\right)^{-1} \tag{D.109}$$

Maintenant que nous avons déterminer le pas μ^*, nous allons remplacer μ par μ^* dans la borne (D.107) ce qui nous donne :

$$\mathrm{EReg}_{\mathbf{q}}(T) \leq \frac{3}{2}\sqrt{HMT}S\left(\frac{B}{\delta}+K\right) + TK\delta(3+A). \tag{D.110}$$

La borne du regret ci-dessus, après optimisation de μ, est encore linéaire en T. Nous devons donc déterminer un pas δ tel que la croissance du regret est plus faible que $\mathcal{O}(T)$. Cependant, bien que convexe la présence éventuelles des contraintes sur δ (à par $\delta > 0$) nous empêche de trouver une solution en forme close de δ^*. Il faut toutefois noter que notre objectif premier est d'obtenir la propriété de non regret. Pour y parvenir, il suffit de trouver un δ qui respecte les contraintes tout en limitant la croissance (qui doit être inférieure à $\mathcal{O}(T)$) de la borne du regret ci-dessus. Il faut donc dans un premier temps déterminer l'ordre de grandeur des variations de δ en fonction de T. Pour cela, nous remarquons que nous pouvons réécrire le regret comme :

$$\mathrm{EReg}(T) \leq \frac{\mathcal{O}(\sqrt{T})}{\delta} + \mathcal{O}(T)\delta. \tag{D.111}$$

De l'équation ci-dessus, nous remarquons que si le pas croît en $\mathcal{O}(T^{-1/4})$ alors le regret va croître en $\mathcal{O}(T^{\frac{3}{4}})$ ce qui est suffisant pour avoir la propriété de non regret. Ainsi, il ne nous reste plus

qu'à trouver un pas qui croît en $\mathcal{O}(T^{-1/4})$ et qui respecte les contraintes sur δ, ce pas peut être défini comme :

$$\delta^* = OT^{\frac{1}{4}}, \tag{D.112}$$

où O est une constante qui dépend de l'espace faisable réduit $\mathscr{P}_\delta$. Cette valeur de δ respecte les contraintes tout en limitant la croissance de la borne du regret. Pour visualiser cela, il faut remplacer δ par δ^* dans la borne (C.121) ce qui nous donne :

$$\mathrm{EReg}(T) \leq U_1 T^{\frac{3}{4}} + U_2 T^{\frac{1}{2}}, \tag{D.113}$$

où les termes U_1 et U_2 dépendent des paramètres spécifiques au problème d'optimisation à résoudre K, A, B, M, et H.

D.3.3 Preuve du Corollaire 12

Dans le cas où la durée de transmission T n'est pas connue en avance, l'objet n'est pas en mesure de calculer les paramètres optimaux δ^* et μ^*. Pour pallier à cela, nous allons utiliser l'astuce du *doubling-trick* dérit dans la Section 2.3.1. En utilisant cette astuce nous trouvons la borne suivante pour le regret :

$$\mathrm{EReg}(T) \leq \frac{\sqrt{2}}{2^{\frac{3}{4}}-1} U_1 T^{\frac{3}{4}} + \frac{2}{\sqrt{2}-1} U_2 T^{\frac{1}{2}}, \tag{D.114}$$

où les termes U_1 et U_2 dépendent des paramètres spécifiques au problème d'optimisation à résoudre K, A, B, M, et H.

BIBLIOGRAPHIE

Tosiron Adegbija, Anita Rogacs, Chandrakant Patel, and Ann Gordon-Ross.
Microprocessor optimizations for the internet of things : a survey.
IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 37(1) :7–20, 2018.

Ala Al-Fuqaha, Mohsen Guizani, Mehdi Mohammadi, Mohammed Aledhari, and Moussa Ayyash.
Internet of things : A survey on enabling technologies, protocols, and applications.
IEEE Communications Surveys & Tutorials, 17(4) :2347–2376, 2015.

Md Shipon Ali, Hina Tabassum, and Ekram Hossain.
Dynamic user clustering and power allocation for uplink and downlink non-orthogonal multiple access (NOMA) systems.
IEEE Access, 4 :6325–6343, Aug. 2016.

Tansu Alpcan, Tamer Başar, Rayadurgam Srikant, and Eitan Altman.
CDMA uplink power control as a noncooperative game.
Wireless Networks, 8(6) :659–670, Nov. 2002.

Eitan Altman and Laura Wynter.
Equilibrium, games, and pricing in transportation and telecommunication networks.
Networks and Spatial Economics, 4(1) :7–21, Mar. 2004.

Sara Amendola, Rossella Lodato, Sabina Manzari, Cecilia Occhiuzzi, and Gaetano Marrocco.
Rfid technology for iot-based personal healthcare in smart spaces.
IEEE Internet of things journal, 1(2) :144–152, 2014.

Animashree Anandkumar, Nithin Michael, Ao Kevin Tang, and Ananthram Swami.
Distributed algorithms for learning and cognitive medium access with logarithmic regret.
29(4) :731–745, Mar. 2011.

Abdoulaye Bagayoko, Inbar Fijalkow, and Patrick Tortelier.
Power control of spectrum-sharing in fading environment with partial channel state information.

IEEE Transactions on Signal Processing, 59(5) :2244–2256, 2011.

Flavien Bardyn, Martin Savary, Sara Grassi Pauletti, Pierre-André Farine, Benedikt Fasel, and Kamiar Aminian.
Mems inertial motion sensing watch for measuring walking and running activities.
In *Proceedings of the 2016 IEEE Workshop on Signal Processing Systems (SiPS)*, number EPFL-CONF-221013. Ieee, 2016.

Stephen Boyd and Lieven Vandenberghe.
Convex optimization.
Cambridge university press, 2004.

Sébastien Bubeck, Nicolo Cesa-Bianchi, et al.
Regret analysis of stochastic and nonstochastic multi-armed bandit problems.
Foundations and Trends in Machine Learning, 5(1) :1–122, 2012.

Yan Chen, Feng Han, Yu-Han Yang, Hang Ma, Yi Han, Chunxiao Jiang, Hung-Quoc Lai, David Claffey, Zoltan Safar, and KJ Ray Liu.
Time-reversal wireless paradigm for green internet of things : An overview.
1(1) :81–98, 2014.

Mung Chiang, Prashanth Hande, Tian Lan, Chee Wei Tan, et al.
Power control in wireless cellular networks.
Foundations and Trends in Networking, 2(4) :381–533, 2008.

Li Da Xu, Wu He, and Shancang Li.
Internet of things in industries : A survey.
IEEE Transactions on industrial informatics, 10(4) :2233–2243, 2014.

R. Negrel E. V. Belmega, P. Mertikopoulos and L. Sanguinetti.
Online convex optimization and no-regret learning : Algorithms, guarantees and applications.
in preparation for resubmission to IEEE Signal Processing Magazine, 2018.

A. D. Flaxman, A. T. Kalai, and H. B. McMahan.
Online convex optimization in the bandit setting : gradient descent without a gradient.
In *SODA'05 : Proceedings of the 16th annual ACM-SIAM symposium on discrete algorithms*, pages 385–394, Jan. 2005.

Gerard J Foschini and Zoran Miljanic.
A simple distributed autonomous power control algorithm and its convergence.
IEEE transactions on vehicular Technology, 42(4) :641–646, 1993.

Pranay P Gaikwad, Jyotsna P Gabhane, and Snehal S Golait.

A survey based on smart homes system using internet-of-things.
In *Computation of Power, Energy Information and Commuincation (ICCPEIC), 2015 International Conference on*, pages 0330–0335. IEEE, 2015.

David Gesbert, Saad Ghazanfar Kiani, Anders Gjendemsjo, and Geir Egil Oien.
Adaptation, coordination, and distributed resource allocation in interference-limited wireless networks.
Proceedings of the IEEE, 95(12) :2393–2409, 2007.

Carles Gomez and Josep Paradells.
Wireless home automation networks : A survey of architectures and technologies.
IEEE Communications Magazine, 48(6), 2010.

Claire Goursaud and Jean-Marie Gorce.
Dedicated networks for IoT : PHY/MAC state of the art and challenges.
EAI Endorsed Transactions on Internet of Things, 1(1) :1–11, Oct. 2015.

James Hannan.
Approximation to bayes risk in repeated play.
Contributions to the Theory of Games, 3 :97–139, 1957.

Godfrey H. Hardy.
Divergent Series.
Oxford University Press, 1949.

Panu Harmo, Tapio Taipalus, Jere Knuuttila, José Vallet, and Aarne Halme.
Needs and solutions-home automation and service robots for the elderly and disabled.
In *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*, pages 3201–3206. IEEE, 2005.

Morteza Hashemi, Ashu Sabharwal, C Emre Koksal, and Ness B Shroff.
Efficient beam alignment in millimeter wave systems using contextual bandits.
arXiv preprint arXiv :1712.00702, 2017.

David A Karpuk and Arsenia Chorti.
Perfect secrecy in physical-layer network coding systems from structured interference.
IEEE Transactions on Information Forensics and Security, 11(8) :1875–1887, 2016.

Afaq H Khan, Mohammed A Qadeer, Juned A Ansari, and Sariya Waheed.
4g as a next generation wireless network.
In *Future Computer and Communication, 2009. ICFCC 2009. International Conference on*, pages 334–338. IEEE, 2009.

ChoongHoe Kim, Yanggyoo Jung, MinHo Kim, TaeHo Yoon, YunSeok Song, SeokHo Na, DongJoo Park, ByoungWoo Cho, DaeByoung Kang, KwangMo Lim, et al.
Development of extremely thin profile flip chip csp using laser assisted bonding technology.
In *CPMT Symposium Japan (ICSJ), 2017 IEEE*, pages 45–49. IEEE, 2017.

Changmin Lee, Luca Zappaterra, Kwanghee Choi, and Hyeong-Ah Choi.
Securing smart home : Technologies, security challenges, and security requirements.
In *Communications and Network Security (CNS), 2014 IEEE Conference on*, pages 67–72. IEEE, 2014.

Xichun Li, Abudulla Gani, Rosli Salleh, and Omar Zakaria.
The future of mobile wireless communication networks.
In *2009 International Conference on Communication Software and Networks*, pages 554–557. IEEE, 2009.

Raouia Masmoudi, E Veronica Belmega, Inbar Fijalkow, and Noura Sellami.
A unifying view on energy-efficiency metrics in cognitive radio channels.
In *Signal Processing Conference (EUSIPCO), 2014 Proc. 22nd European*, pages 171–175, Sep. 2014.

Panayotis Mertikopoulos and E Veronica Belmega.
Transmit without regrets : Online optimization in MIMO-OFDM cognitive radio systems.
32(11) :1987–1999, Dec. 2014.

Panayotis Mertikopoulos and E. Veronica Belmega.
Learning to be green : Robust energy efficiency maximization in dynamic MIMO-OFDM systems.
34(4) :743–757, Apr. 2016.

Daniele Miorandi, Sabrina Sicari, Francesco De Pellegrini, and Imrich Chlamtac.
Internet of things : Vision, applications and research challenges.
Ad Hoc Networks, 10(7) :1497–1516, Sep. 2012.

Oskar Morgenstern and John Von Neumann.
Theory of games and economic behavior.
Princeton university press, 1953.

John Nash.
Non-cooperative games.
Annals of mathematics, pages 286–295, 1951.

Jong-Shi Pang, Gesualdo Scutari, Francisco Facchinei, and Chaoxiong Wang.
Distributed power allocation with rate constraints in Gaussian parallel interference channels.

54(8) :3471–3489, Jul. 2008.

Gert Frølund Pedersen.
COST 231-Digital mobile radio towards future generation systems.
EU, 1999.

Mylene Pischella and Didier Le Ruyet.
Adaptive resource allocation and decoding strategy for underlay multi-carrier cooperative cognitive radio systems.
Transactions on Emerging Telecommunications Technologies, 24(7-8) :748–761, 2013.

Ralph Tyrell Rockafellar.
Convex analysis.
Princeton University Press, 2015.

Sudhir K Routray, Mahesh K Jha, Laxmi Sharma, Rahul Nyamangoudar, Abhishek Javali, and Sutapa Sarkar.
Quantum cryptography for iot : Aperspective.
In *IoT and Application (ICIOT), 2017 International Conference on*, pages 1–4. IEEE, 2017.

Ahmad-Reza Sadeghi, Christian Wachsmann, and Michael Waidner.
Security and privacy challenges in industrial internet of things.
In *Design Automation Conference (DAC), 2015 52nd ACM/EDAC/IEEE*, pages 1–6. IEEE, 2015.

Hashim Safdar, Norsheila Fisal, Rahat Ullah, Wajahat Maqbool, Faiz Asraf, Zubair Khalid, and AS Khan.
Resource allocation for uplink M2M communication : A game theory approach.
In *Wireless Technology and Applications (ISWTA), 2013 IEEE Symp.*, pages 48–52, Sep. 2013.

Cem U Saraydar, Narayan B Mandayam, and David J Goodman.
Efficient power control via pricing in wireless data networks.
IEEE transactions on Communications, 50(2) :291–303, 2002.

Gesualdo Scutari and Sergio Barbarossa.
Generalized water-filling for multiple transmit antenna systems.
In *Communications, 2003. ICC'03. IEEE International Conference on*, volume 4, pages 2668–2672. IEEE, 2003.

Gesualdo Scutari, Daniel P Palomar, and Sergio Barbarossa.
The MIMO iterative waterfilling algorithm.
57(5) :1917–1935, Jan. 2009.

Shai Shalev-Shwartz.
Online learning and online convex optimization.
Foundations and Trends in Machine Learning, 4(2) :107–194, 2011.

Claude Elwood Shannon.
Communication in the presence of noise.
Proceedings of the IRE, 37(1) :10–21, 1949.

James C Spall.
Stochastic optimization and the simultaneous perturbation method.
In *Proceedings of the 31st conference on Winter simulation : Simulation—a bridge to the future-Volume 1*, pages 101–109. ACM, 1999.

Ahmed Iyanda Sulyman, Sharief MA Oteafy, and Hossam S Hassanein.
Expanding the cellular-IoT umbrella : An architectural approach.
24(3) :66–71, Jun. 2017.

Huawei Technologies.
5G : A technology vision.
White paper, 2013.

John Von Neumann and Oskar Morgenstern.
Theory of games and economic behavior.
1944.

Peng Wang, Ming Zhao, Limin Xiao, Shidong Zhou, and Jing Wang.
Power allocation in ofdm-based cognitive radio systems.
In *Global Telecommunications Conference, 2007. GLOBECOM'07. IEEE*, pages 4061–4065. IEEE, 2007.

Lijun Zhang, Tianbao Yang, Rong Jin, Yichi Xiao, and Zhi-hua Zhou.
Online stochastic linear optimization under one-bit feedback.
In *International Conference on Machine Learning*, pages 392–401, 2016a.

Yuanyu Zhang, Yulong Shen, Hua Wang, Jianming Yong, and Xiaohong Jiang.
On secure wireless communications for iot under eavesdropper collusion.
IEEE Transactions on Automation Science and Engineering, 13(3) :1281–1293, 2016b.

Tao Zheng, Yajuan Qin, Hongke Zhang, and Syyen Kuo.
Adaptive power control for mutual interference avoidance in industrial Internet-of-Things.
China Communications, 13(Supplement 1) :124–131, Sep. 2016.

Résumé

L'Internet des Objets (IoT) est envisagé pour interconnecter des objets communicants et autonomes au sein du même réseau, qui peut être le réseau Internet ou un réseau de communication sans fil. Les objets autonomes qui composent les réseaux IoT possèdent des caractéristiques très différentes, que ce soit en terme d'application, de connectivité, de puissance de calcul, de mobilité ou encore de consommation de puissance. Le fait que tant d'objets hétérogènes partagent un même réseau soulève de nombreux défis tels que : l'identification des objets, l'efficacité énergétique, le contrôle des interférences du réseau, la latence ou encore la fiabilité des communications. La densification du réseau couplée à la limitation des ressources spectrales (partagées entre les objets) et à l'efficacité énergétique obligent les objets à optimiser l'utilisation des ressources fréquentielles et de puissance de transmission. De plus, la mobilité des objets au sein du réseau ainsi que la grande variabilité de leur comportement changent la dynamique du réseau qui devient imprévisible. Dans ce contexte, il devient difficile pour les objets d'utiliser des algorithmes d'allocation de ressources classiques, qui se basent sur une connaissance parfaite ou statistique du réseau. Afin de transmettre de manière efficace, il est impératif de développer de nouveaux algorithmes d'allocation de ressources qui sont en mesure de s'adapter aux évolutions du réseau. Pour cela, nous allons utiliser des outils d'optimisation en ligne et des techniques d'apprentissage. Dans ce cadre nous allons exploiter la notion du *regret* qui permet de comparer l'efficacité d'une allocation de puissance dynamique à la meilleure allocation de puissance fixe calculée à posteriori. Nous allons aussi utiliser la notion de *non-regret* qui garantit que l'allocation de puissance dynamique donne des résultats asymptotiquement optimaux . Dans cette thèse, nous nous sommes concentrés sur le problème de minimisation de puissance sous contrainte de débit. Ce type de problème permet de garantir une certaine efficacité énergétique tout en assurant une qualité de service minimale des communications. De plus, nous considérons des réseaux de type IoT et ne faisons donc aucune hypothèse quant aux évolutions du réseau. Un des objectifs majeurs de cette thèse est la réduction de la quantité d'information nécessaire à la détermination de l'allocation de puissance dynamique. Pour résoudre ce problème, nous avons proposé des algorithmes inspirés du problème du bandit manchot, problème classique de l'apprentissage statistique. Nous avons montré que ces algorithmes sont efficaces en terme du *regret* lorsque l'objet a accès à un vecteur, le gradient ou l'estimateur non-biaisé du gradient, comme feedback d'information. Afin de réduire d'avantage la quantité d'information reçue par l'objet, nous avons proposé une méthode de construction d'un estimateur du gradient basé uniquement sur une information scalaire. En utilisant cet estimateur nous avons présenté un algorithme efficace d'allocation de puissance.

Abstract

One of the key challenges in Internet of Things (IoT) networks is to connect numerous, heterogeneous and autonomous devices. These devices have different types of characteristics in terms of : application, computational power, connectivity, mobility or power consumption. These characteristics give rise to challenges concerning resource allocation such as : a) these devices operate in a highly dynamic and unpredictable environments ; b) the lack of sufficient information at the device end ; c) the interference control due to the large number of devices in the network. The fact that the network is highly dynamic and unpredictable implies that existing solutions for resource allocation are no longer relevant because classical solutions require a perfect or statistical knowledge of the network. To address these issues, we use tools from online optimization and machine learning. In the online optimization framework, the device only needs to have strictly causal information to define its online policy. In order to evaluate the performance of a given online policy, the most commonly used notion is that of the *regret*, which compares its performance in terms of loss with a benchmark policy, i.e., the best fixed strategy computed in hindsight. Otherwise stated, the *regret* measures the performance gap between an online policy and the best mean optimal solution over a fixed horizon. In this thesis, we focus on an online power minimization problem under rate constraints in a dynamic IoT network. To address this issue, we propose a *regret*-based formulation that accounts for arbitrary network dynamics, using techniques used to solve the multi-armed bandit problem. This allows us to derive an online power allocation policy which is provably capable of adapting to such changes, while relying solely on strictly causal feedback. In so doing, we identify an important tradeoff between the amount of feedback available at the transmitter side and the resulting system performance. We first study the case in which the device has access to a vector, either the gradient or an unbiased estimated of the gradient, as information feedback. To limit the feedback exchange in the network our goal is to reduce it as mush as possible. Therefore, we study the case in which the device has access to only a loss-based information (scalar feedback). In this case, we propose a second online algorithm to determine an efficient and adaptative power allocation policy.