# Massive multi-player multi-armed bandits for internet of things networks

Hiba Dakdouk

DOCTORAT BRETAGNE LOIRE / MATHSTIC

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

# THÈSE DE DOCTORAT DE

L'ÉCOLE NATIONALE SUPÉRIEURE MINES-TÉLÉCOM ATLANTIQUE
BRETAGNE PAYS-DE-LA-LOIRE - IMT ATLANTIQUE

ÉCOLE DOCTORALE N° 601
*Mathématiques et Sciences et Technologies de l'Information et de la Communication*
Spécialité : *Informatique*

Par
**Hiba DAKDOUK**

## Bandits Massifs Multi-Bras Multi-Joueurs pour les Réseaux de l'Internet des Objets

***Massive Multi-Player Multi-Armed Bandits for Internet of Things Networks***

**Thèse présentée et soutenue à Grenoble, le *lundi 30 mai 2022***
**Unité de recherche : SRCD/IRISA**
**Thèse N° : 2022IMTA0296**

### Rapporteurs avant soutenance :

Vianney PERCHET — Professeur, ENSAE
Yezekael HAYEL — Maitre de conférence (HDR), Université d'Avignon

### Composition du Jury :

| | | |
|---|---|---|
| Président : | Vianney PERCHET | Professeur, ENSAE |
| Examinateurs : | Yezekael HAYEL | Maitre de conférence (HDR), Université d'Avignon |
| | Michele ZORZI | Professeur, University of Padova (Italie) |
| | Nadège VARSIER | Ingénieure de recherche, Orange Labs |
| | Raphaël FERAUD | Ingénieur de recherche, Orange Labs |
| Dir. de thèse : | Patrick MAILLE | Professeur, IMT Atlantique |

# Acronyms

| | |
|---|---|
| ADR | Adaptive Data Rate |
| CR | Cognitive Radio |
| DOFG | Decreasing Order Fairness Greedy |
| DORG | Decreasing Order Reward Greedy |
| IoT | Internet of Things |
| ISM | Industrial Scientific and Medical |
| LCB | Lower Confidence Bound |
| LoRa | Long Range |
| LoRaWAN | Long-Range Wide Area Network |
| LPWAN | Low Power Wide Area Network |
| MAB | Multi-Armed Bandit |
| MP-MAB | Multi-Player Multi-Armed Bandit |
| NS | Network Server |
| OSA | Opportunistic Spectrum Access |
| PAC | Probably Approximately Correct |
| PDR | Packet Delivery Ratio |
| PER | Packet Error Rate |
| PU | Primary User |
| QoS | Quality of Service |
| RL | Reinforcement Learning |
| RSSI | Received Signal Strength Indicator |
| SA | Slotted ALOHA |
| SF | Spreading Factor |
| SINR | Signal to Interference and Noise Ratio |
| SNR | Signal to Noise Ratio |
| SU | Secondary User |
| TP | Transmitting Power |
| TS | Thompson Sampling |
| UCB | Upper Confidence Bound |

# NOTATIONS

| | |
|---|---|
| $N$ | number of players |
| $[N]$ | set of players |
| $p_n$ | probability that player $n$ sends a packet |
| $K$ | number of arms |
| $[K]$ | set of arms |
| $\theta_k$ | mean reward of arm $k$ |
| $\boldsymbol{\theta}$ | model $\boldsymbol{\theta} = (\theta_1, ..., \theta_K)$ |
| $\hat{\theta}_k$ | estimated mean reward of arm $k$ |
| $\hat{\boldsymbol{\theta}}$ | estimated model $\hat{\theta} = (\hat{\theta}_1, ..., \hat{\theta}_K)$ |
| $\epsilon$ | approximation term |
| $\delta$ | probability of failure |
| $\pi_n^k$ | probability that player $n$ chooses arm $k$ |
| $\pi_n$ | policy of player $n$, $\pi_n = (\pi_n^1, ..., \pi_n^K)$ |
| $\pi$ | policy of players, $\pi = (\pi_1, ..., \pi_n)$ |
| $\pi_u$ | uniform policy |
| $\pi^\dagger$ | decreasing order fair greedy policy generated by Algorithm 2 |
| $\pi^*_{\boldsymbol{\theta}}$ | optimal policy in model $\boldsymbol{\theta}$, which is deterministic, when it is clear in the context, we use $\pi^*$ |
| $\mu_{\boldsymbol{\theta}}(\pi)$ | mean reward in model $\boldsymbol{\theta}$ of the policy $\pi$, when it is clear in the context, we use $\mu(\pi)$<br>For a stochastic policy: $\mu_{\boldsymbol{\theta}}(\pi) = \sum_{k=1}^K \theta^k \sum_{n=1}^N p_n \pi_n^k \prod_{n' \neq n} (1 - p_{n'} \pi_{n'}^k)$<br>For a deterministic policy $\mu_{\boldsymbol{\theta}}(\pi) = \sum_{k=1}^K \theta^k z^k l^k$ |
| $z^k$ | probability that arm $k$ is not used by any other players, $z^k = \prod_{n' \in [N], k_n = k} (1 - p_n)$ |
| $l^k$ | sum of activation odds on arm $k$ of other players, $l^k = \sum_{n' \in [N], k_n = k} \frac{p_n}{1 - p_n}$ |
| $k_n$ | arm assigned to player $n$ |
| $\pi[n]$ | policy $\pi$ when players $n' > n$ do not play |
| $z^k[n]$ | probability that arm $k$ is not used by any of the first $n$ players |
| $l^k[n]$ | sum of activation odds of the $n$ first players for arm $k$ |
| $\rho_n^k(\pi)$ | probability that no other players have chosen arm $k$ using policy $\pi$ |

*Dedicated to the memory of my grandmother*
*Naama Osman*
*23 April 1944 - 31 May 2021*

*Until we meet again.*

# Acknowledgement

Undertaking this PhD has been a truly life-changing experience for me and it would not have been possible to do without the support and guidance that I received from many people.

I would like to first say a very big thank you to my supervisors: Nadège Varsier and Raphaël Féraud, and my thesis director Patrick Maillé. I am extremely grateful for the support and well-guidance I had from you throughout this journey. Thanks for your great efforts and determination to bring out the best for this thesis. I am honored to have pursued my PhD under your supervision and to have worked with you.

I would like also to thank Romain Laroche from Microsoft Research Lab, for his significant collaboration with the work achieved in this thesis.

I specially thank my Comité de Suivi (CSI): Christophe Moy and Xavier Lagrange, for their follow-up and precious advice during our meetings.

I would like also to express my special gratitude for my colleagues in Orange Labs for the wonderful time we spent during the three years. Special thanks to my office-mate Stéphane Coutant for his help in reviewing this manuscript.

On a personal level, my warm thanks and gratitude are sent to my family. No words or actions of gratitude can outweigh your favor. My parents, my idol, thank you for being that supportive and patient since ever. I would not have been here without you. My brothers: Mohammed and Ali, and sisters: Zeinab, Marwa, Kawther and Safa, thanks for being always there for me during my tough seasons.

My small family, my soulmates Mahdi Srour and Lamar, I'm greatly blessed to have you next to me during this journey. Thanks to your endless help, support, encouragement and love.

Finally, I express my precious thanks to my friends for always being there for me in weal and woe, especially: Hanan, Rana, Abeer, Batoul and Nour.

# Résumé

Cette thèse de doctorat étudie le problème d'optimisation de la performance des réseaux de l'Internet des objets (IoT). L'objectif est de maximiser le succès des communications dans les réseaux de l'IoT, en proposant des algorithmes de prise de décision dynamiques efficaces pouvant être intégrés dans les futurs équipements IoT, tout en respectant leurs contraintes de faible complexité et de faible consommation d'énergie. Pour cela, l'apprentissage par renforcement (RL) est utilisé et le problème d'optimisation est modélisé comme un problème de bandit multi-joueurs multi-bras (MP-MAB), adapté aux réseaux IoT et permettant de surmonter de nombreuses hypothèses irréalistes dans le cadre des réseaux IoT précédemment effectuées dans la littérature. Dans cette thèse, deux approches différentes sont proposées pour traiter le problème d'optimisation. La première approche permet de blacklister les mauvais canaux de propagation d'un réseau en utilisant un algorithme collaboratif d'identification des meilleurs bras. La seconde approche consiste en deux politiques différentes qui attribuent de manière récursive chaque équipement IoT à un canal ; la première politique se concentre sur le nombre de communications réussies tandis que l'autre garantit un niveau d'équité entre les équipements. Dans un premier temps, nous avons effectué l'étude numérique et expérimentale des différents algorithmes développés pendant cette thèse afin de montrer qu'ils étaient capables de surclasser les autres algorithmes de la littérature. Dans un second temps, une partie importante du travail a consisté en l'application des algorithmes développés au problème concret de choix de la puissance d'émission et du facteur d'étalement dans un réseau LoRa, en analysant les performances en termes de qualité de service et de consommation d'énergie à l'aide d'un simulateur de réseau LoRa réaliste entièrement redéveloppé en C durant la thèse.

# Abstract

This PhD thesis studies the optimization problem of Internet of Things (IoT) networks performance. We aim to maximize the successful communications in IoT networks, by proposing efficient dynamic decision-making algorithms that can be embedded in future IoT devices, while respecting the low complexity and low energy consumption constraints in IoT devices. For this sake, we use Reinforcement Learning (RL), and we model the optimization problem as a massive multi-player multi-armed bandit (MP-MAB) problem to best suit IoT networks, while overcoming many unrealistic assumptions previously made in the literature. In this manuscript, we propose two different approaches to handle the optimization problem. The first blacklists bad channels after a collaborative best-arms identification algorithm. The second consists of two different policies that recursively assign each device to one channel; where one policy focuses on the number of successful communications while the other guarantees a level of fairness between the devices. We provide both numerical and experimental studies of our developed algorithms, and show their out-performance over other algorithms proposed in the literature. Furthermore, we test our algorithms using a realistic LoRa network simulator entirely redeveloped in C during the thesis, and show the gain they achieve in terms of both successful communications and energy consumption compared to other already implemented algorithms.

# Résumé des Travaux de Thèse

Ce manuscrit conclut ma thèse de doctorat, qui a débuté en janvier 2019 et s'est achevée en mai 2022. Mes recherches se sont déroulées à Orange Labs à Grenoble (France) en collaboration avec IMT Atlantique à Rennes (France). Ces travaux ont été réalisés sous la supervision du Docteur Nadège Varsier d'Orange Labs à Grenoble et du Docteur Raphaël Féraud d'Orange Labs à Lannion, et sous la direction du Docteur Patrick Maillé d'IMT Atlantique à Rennes.

## Contexte de la thèse

L'internet des objets (IoT) est un terme nouveau, mais en même temps ancien. L'expression "Internet des objets" a été inventée par le père de l'IoT, Kevin Ashton [1], lors d'une présentation qu'il a faite en 1999. Il l'a utilisé pour relier l'idée de l'identification par radiofréquence (RFID) au domaine alors nouveau de l'Internet [1]. Depuis lors, l'utilisation de ce terme s'est développée et des milliards d'équipements sont déjà déployés dans le monde entier. Les analystes de l'IoT prévoient que, d'ici 2025, il y aura probablement plus de 27 milliards de connexions IoT [2] permettant un large éventail d'applications différentes. Ce large déploiement d'équipements IoT et la variété des applications génèrent différents défis, principalement en termes de fiabilité et d'efficacité énergétique, ce qui pousse les chercheurs et les développeurs à concevoir différents schémas d'accès radio efficaces. De même, la croissance rapide du marché de l'apprentissage automatique et son applicabilité à un très large éventail d'applications ouvrent la voie à des équipements IoT intelligents.

En conséquence, dans cette thèse, nous visons à améliorer les performances des réseaux IoT, principalement en terme de *fiabilité*, tout en consommant le moins d'*énergie* possible et en améliorant donc la durée de vie moyenne des batteries. Nous proposons d'utiliser les techniques d'apprentissage par renforcement et, plus particulièrement, les bandits multi-bras multi-joueurs (MP-MABs) pour un mécanisme efficace d'allocation des ressources.

1. Kevin Ashton était cofondateur et directeur exécutif de l'Auto-ID Center.

*"L'internet des objets a le potentiel de changer le monde, tout comme l'internet l'a fait. Peut-être même plus."*

– Père de l'IoT, Kevin Ashton

**Internet des objets** Malgré la grande révolution de l'Internet des objets, il n'existe pas encore de définition universellement utilisée. Nous présentons ici la définition simple de l'IoT fournie par Wikipedia [3]: *"L'internet des objets décrit des objets physiques qui sont dotés de capteurs, de capacités de traitement, de logiciels et d'autres technologies qui se connectent et échangent des données avec d'autres équipements et systèmes via l'internet ou d'autres réseaux de communication"*. La pluralité des définitions de l'IoT provient des diverses applications et domaines de l'IoT. Tous les aspects de notre vie sont concernés. L'IoT peut en effet permettre de considérables améliorations dans les domaines de l'éducation, la santé ou la sécurité, ainsi que pour la prise de décision et la productivité des entreprises dans le commerce de détail, la fabrication et de nombreux autres secteurs. L'internet des objets et l'intelligence artificielle constituent une part importante de la quatrième révolution industrielle[4]. Désormais, toutes sortes d'objets du quotidien peuvent être connectés à l'internet, de sorte que l'IoT compte plus d'appareils et de capteurs intelligents que de personnes. Ces appareils et capteurs connectés collectent et partagent des données à des fins d'utilisation et d'évaluation par de nombreuses organisations, notamment des entreprises, des villes, des gouvernements, des hôpitaux et des particuliers, générant des quantités massives de données. Afin de fournir et de prendre en charge tous les types d'applications IoT, de nombreuses technologies aux caractéristiques différentes sont disponibles.

**Technologies sans fil pour l'IoT** Les équipements IoT utilisent la technologie qui correspond le mieux à leurs contraintes. L'IoT critique, qui couvre des applications telles que la sécurité routière et la chirurgie à distance dans le domaine de la santé, nécessite une fiabilité et une disponibilité élevées ainsi qu'une faible latence [5]. Ces applications sont mieux servies par les technologies à courte portée telles que Wi-Fi, Bluetooth et ZigBee qui offrent une couverture jusqu'à 1 km [6]. Les réseaux cellulaires tels que 2G/3G, 4G/LTE et 5G sont adaptés aux applications longue portée (jusqu'à 16 km) et haut débit [6]. Les réseaux grande distance et faible consommation d'énergie (LPWAN) conviennent quand à eux aux applications qui nécessitent une large zone de couverture et une longue durée de vie des batteries.

Une contrainte commune aux équipements IoT est la faible consommation d'énergie, comme dans le cas des compteurs intelligents et de la surveillance industrielle, car les équipements sont en général déployés sans accès direct à l'électricité et fonctionnent sur des batteries qui peuvent difficilement être changées. Les technologies LPWAN offrent, en général, une large zone de couverture à des objets connectés contraints en énergie, en débit, en complexité et donc en coût et la possibilité de déployer un grand nombre d'équipements.

**Le spectre des radiofréquences** En termes de spectre RF, nous distinguons deux types de technologie en fonction du spectre sur lequel elles opèrent : le spectre sous licence ou sans licence (bandes de fréquences exemptes de licence). Le spectre sous licence correspond à une partie de l'espace public des fréquences qui a été concédée par les autorités nationales ou régionales à une entreprise privée, généralement un opérateur de réseau mobile, à condition qu'elle fournisse un certain service au public, tel que la connectivité cellulaire [7]. L'Internet cellulaire des objets (CIoT) fournit des technologies utilisant des bandes sous licence pour des applications à longue portée, comme le standardise le projet de partenariat de troisième génération (3GPP) dans sa version 13. Ces technologies cellulaires comprennent le GSM à couverture étendue pour l'internet des objets (EC-GSM-IoT), l'évolution à long terme des communications de type machine de catégorie M1 (LTE MTC Cat M1 ou LTE-M) et l'IoT à bande étroite (NB-IoT). Nous nous référons au livre [7] pour plus de détails. Le spectre sous licence est toutefois généralement associé à des coûts élevés.

A l'inverse, le spectre sans licence correspond à des portions de l'espace public des fréquences dont on peut dire qu'elles restent publiques et donc exemptes de frais de licence [7]. Parmi les bandes sans licence les plus populaires figurent les bandes industrielles, scientifiques et médicales (ISM), centrées sur 2,4 GHz, 868/915 MHz, 433 MHz et 169 MHz, selon la région d'exploitation [5]. En n'obligeant pas les opérateurs à obtenir une licence coûteuse et une autorisation spéciale pour son utilisation, le spectre sans licence est une option peu coûteuse et accessible à tous pour répondre aux besoins de communication. Tout comme Wi-Fi, Bluetooth et ZigBee, les technologies LPWA fonctionnent généralement dans les bandes ISM. SIGFOX et LoRa sont les technologies LPWA les plus connues qui fonctionnent dans les bandes ISM sans licence. La technologie LoRa est présentée en détail au chapitre 5.

Les réseaux LPWA ont suscité un intérêt considérable de la part du marché et des

médias en raison de leur fonctionnement dans des bandes exemptes de licence et de leur capacité à assurer une longue durée de vie des batteries des équipements, une faible complexité de ces mêmes équipements et la connexion d'un très grand nombre d'objets. Cependant, avec la forte augmentation du déploiement de l'IoT dans le monde, un problème majeur de coexistence des systèmes se pose. À l'intérieur du spectre sans licence, les différents systèmes ne sont pas séparés dans le domaine des fréquences mais se chevauchent dans le sens où ils peuvent utiliser les mêmes ressources en fréquences à tout moment, ce qui provoque des interférences et donc des échecs de transmission.

Dans ce contexte, cette thèse vise à résoudre le problème suivant : *Comment les équipements IoT peuvent éviter les interférences entre eux et éviter les interférences avec les réseaux coexistants partageant le même spectre, afin de fournir une fiabilité élevée tout en respectant les contraintes IoT, et notamment en préservant une faible consommation d'énergie ?* Notre objectif est de rendre les equipements IoT intelligents en les programmant et en les configurant de manière à ce qu'ils puissent connaitre et choisir les meilleurs paramètres de fonctionnement (par exemple, la puissance d'émission, la fréquence radio) afin d'éviter les interférences et la congestion.

Nous proposons ici d'utiliser des algorithmes d'apprentissage par renforcement (RL) et plus particulièrement des algorithmes de bandit multi-bras multi-joueurs (MP-MAB).

## Courte introduction sur les Bandits Multi-Bras

Le terme MAB vient des machines à sous, connues sous le nom de bandits car elles prennent généralement votre argent. Il s'agit d'une classe de problèmes de RL qui se réfère à un jeu de décision en ligne où, dans la formulation classique du problème, telle que présentée dans [8, 9], un ensemble de plusieurs bras (actions) dans un certain environnement est associé à des séquences de récompenses (une séquence pour chaque bras) qui sont tirées aléatoirement et indépendamment selon une distribution fixe mais inconnue. Chaque séquence est disponible pour un agent (également appelé joueur) qui doit prendre une série d'actions tout en observant la séquence de récompenses correspondante. La figure 1 illustre ce cycle. L'objectif principal est de découvrir les meilleures actions, c'est-à-dire celles qui offrent les récompenses les plus élevées, et de les exploiter. En particulier, l'agent est confronté au dilemme "exploration-exploitation": il doit essayer toutes les actions pour savoir laquelle est la meilleure (exploration), mais il doit converger rapidement vers celle qu'il croit être la meilleure pour accumuler des récompenses (exploitation).

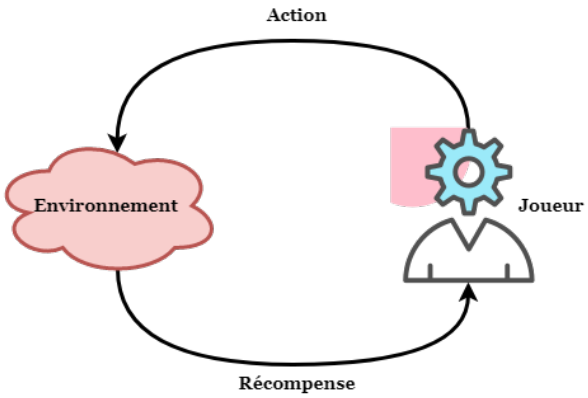



Figure 1 – Interaction entre agent et environnement dans un bandit multi-bras

Une *politique*, ou *stratégie d'allocation*, est un algorithme qui choisit la prochaine action à jouer en fonction de la séquence des jeux passés et des récompenses obtenues. Une politique optimale est la politique qui maximise la récompense cumulative (c'est-à-dire qui choisit le meilleur bras à chaque itération).

**Mesure de la performance** Si la politique optimale maximise la récompense cumulative, alors elle devrait toujours tirer l'un des bras optimaux (il peut ne pas être unique), mais cela n'est pas réaliste car le joueur ne connaît pas les véritables moyens, ni les bras optimaux. Une mesure populaire du succès d'une politique pour résoudre le dilemme exploration-exploitation est le *regret*, qui est la perte due au fait que la politique globalement optimale n'est pas toujours suivie. Cette métrique est détaillée dans la section 2.1.2.

**Applications** Les problèmes de bandits à bras multiples se posent dans de nombreux domaines d'application. L'étude [10] passe en revue les développements les plus récents dans de multiples applications réelles des MABs, telles que la santé, la finance et les télécommunications.

**Classification des MABs** En fonction du modèle de la fonction de récompense, il existe différentes variantes du modèle MAB, que nous distinguons ci-dessous :

- **Stochastique et Contradictoire**: dans les MABs *stochastiques* [11], les récompenses sont générées indépendamment à partir de distributions stochastiques inconnues associées à chaque bras, tandis que dans les MABs *contradictoires* [12], aucune hypothèse statistique n'est faite sur la nature du processus générant les récompenses

des bras, mais un adversaire, plutôt qu'un processus stochastique bien conduit, a un contrôle total sur les récompenses.

- **Stationnaire et Non-stationnaire**: les MABs stochastiques peuvent être classés en MABs *stationnaires* et *non-stationnaires*. Dans les MABs *stationnaires*, les récompenses sont générées à partir de distributions stochastiques qui ne changent pas dans le temps, tandis que dans les MABs *non-stationnaires*, la distribution stochastique d'au moins un bras peut changer à tout moment.

Identifier le modèle d'un MAB est utile pour formaliser un problème de bandit et le résoudre avec les bons outils. Dans cette thèse, nous nous intéressons principalement aux environnements *stochastiques-stationnaires*.

**Bandits multi-bras multi-joueurs** Le problème du bandit multi-bras multi-joueurs (MP-MAB) est une classe de problèmes MAB où, au lieu d'un agent unique, il existe un ensemble $[N]$ de $N$ joueurs, où tous les joueurs ont accès au même ensemble de bras $[K]$, et doivent prendre des décisions à certains instants pré-spécifiés et observer le résultat correspondant. Dans ce modèle, la notion de collision est introduite, c'est-à-dire que lorsque deux joueurs ou plus choisissent le même bras en même temps, ils subissent tous une collision. Différents modèles de collision ont été proposés, mais le plus simple consiste à donner une récompense de 0 à chacun des joueurs qui entrent en collision. Dans ce contexte, les joueurs doivent apprendre à accéder aux bras tout en maximisant leurs récompenses, ce qui nécessite d'éviter les collisions.

En plus de la minimisation du regret attendu cumulé, un autre objectif commun généralement étudié dans les MABs multi-joueurs est l'*équité*. Il a été récemment étudié dans [13] dans le sens où les algorithmes doivent garantir que pour tous les pas de temps $t$ (i.e. uniformément), chaque bras $k$ est tiré au moins $\lfloor r_k.t \rfloor$ fois en $t$ rounds.

Dans cette thèse, nous modélisons notre problème comme un MAB multi-joueurs, et plus spécifiquement un MP-MAB massif puisque le nombre de joueurs (c'est-à-dire les équipements dans un réseau IoT) est possiblement plus grand que le nombre de bras. Nous considérons l'*équité* dans le chapitre 4, mais dans un contexte différent, où elle fait référence à la récompense attendue cumulative de chaque joueur. La modélisation du problème est présentée en détail dans la section 2.2, et les algorithmes MP-MAB conçus pour les réseaux de communication sont ensuite présentés dans la section 2.3.

# Énoncé du problème

Comme indiqué précédemment, dans cette thèse, nous visons à optimiser les communications dans un réseau IoT. Les équipements finaux choisissent intelligemment leurs paramètres de communication à chaque transmission de manière à maximiser le taux de communication global du réseau. Pour cela, nous modélisons notre problème comme un MP-MAB où les équipements sont les joueurs et où toute ressource (ou combinaison de ressources [2]) qui caractérise les communications et satisfait nos conditions (présentées ci-dessous) peut être un bras (i.e. fréquence radio, puissance d'émission, facteur d'étalement pour LoRa, etc.). Dans la suite de ce document, nous désignerons les bras du réseau IoT par des "canaux" qui ne correspondent pas nécessairement à des canaux radio, et les termes "acteur", " nœud" et "équipement" sont équivalents.

Dans un réseau IoT, le nombre de nœuds est dans la plupart des cas supérieur au nombre de canaux. Il n'est donc pas réaliste de considérer un nombre de joueurs inférieur au nombre de bras, comme le supposent la plupart des travaux antérieurs. De plus, les nœuds ne transmettent pas à chaque créneau horaire, mais la fréquence d'envoi des paquets à la passerelle dépend de l'application (santé, sécurité, villes intelligentes, marketing, domotique...). Pour plusieurs applications temps réel, l'équipement doit envoyer un paquet lorsqu'un événement inconnu et non contrôlé se produit. Par exemple, l'équipement d'un utilisateur peut interagir avec son environnement en temps réel, pour obtenir un feu vert lorsque l'utilisateur se trouve face à un carrefour, une publicité lorsque l'utilisateur se trouve devant un magasin, un ticket lorsqu'il monte dans le bus... C'est pourquoi nous supposons dans la suite que chaque joueur a une probabilité d'envoyer un paquet à chaque pas de temps. Considérant que la probabilité d'envoyer un paquet dépend principalement du cas d'usage, nous supposons que chaque joueur connaît sa propre probabilité d'envoyer un paquet. Nous supposons que le nombre de joueurs est connu par la passerelle, ce qui est réaliste dans les protocoles IoT (la passerelle peut garder la trace de tous les appareils dont elle a reçu des paquets), et que la passerelle envoie cette information à chaque joueur au début du jeu.

Nous permettons aux appareils de partager des informations en envoyant des messages à d'autres appareils via la passerelle en utilisant le protocole IoT, où certains octets dans la charge utile de chaque paquet peuvent être dédiés au partage d'informations avec

2. un bras peut être une paire de ressources, par exemple (facteur d'étalement, puissance d'émission) pour LoRa

d'autres joueurs. Par exemple, dans les réseaux LoRa, la charge utile de chaque paquet peut contenir jusqu'à 255 octets [14, 15]. Nous supposons que, dans le même paquet, 8 octets de la charge utile peuvent être utilisés pour envoyer un message aux autres acteurs. Nous distinguons ici les deux termes : un *paquet* correspond aux transmissions régulières d'un équipement, et un *message* correspond à l'information partagée entre les joueurs. Lorsque l'équipement envoie un paquet, il est actif pendant de petites fenêtres de temps et peut recevoir des paquets. Il reçoit également des accusés de réception de la passerelle si sa transmission est réussie.

Nous considérons un réseau IoT avec un nombre fixe de nœuds finaux communiquant avec une seule passerelle suivant un protocole slotted-Aloha. Ce réseau intelligent sous contrôle coexiste géographiquement avec d'autres réseaux qui peuvent partager le même spectre et interférer avec lui. Les collisions sont prises en compte lorsque deux ou plusieurs nœuds envoient des données en même temps sur le même canal. Nous distinguons ici deux types de collisions :

- **Collision interne** : se produit entre les nœuds du réseau contrôlé.
- **Collision externe** : se produit entre un nœud du réseau contrôlé et d'autres nœuds de réseaux externes coexistants.

La figure 2 présente une illustration du problème de congestion, où les flèches de même couleur correspondent à des transmissions de paquets sur les mêmes canaux. Elle différencie les collisions internes et externes.

Par souci de généralité et de simplicité, nous considérons que lorsqu'une *collision interne* ou une *collision externe* se produit, tous les paquets des équipements en collision sont simplement perdus, tandis que l'issue d'une collision dépend du protocole IoT. Par exemple, dans le protocole LoRa [14, 15], le paquet ayant la plus grande puissance reçue peut être décodé tandis que les paquets ayant une puissance reçue inférieure sont perdus [16] (dans le chapitre 5, nous étudions une application sur les réseaux LoRa). Les *collisions externes* rendent les probabilités de transmission réussie (et donc les qualités des canaux) différentes pour chaque canal.

Bien que nous distinguions deux types de collisions, nous ne considérons pas que des collisions puissent se produire lorsque la passerelle envoie des accusés de réception. En effet, ces collisions en liaison descendante nécessitent qu'au moins deux accusés de réception soient envoyés par la passerelle au même moment à des équipements situés au même endroit, ce qui ne peut pas se produire avec une passerelle unique utilisant un protocole crénelé dans le temps, et qui serait improbable dans un réseau IoT réel, où

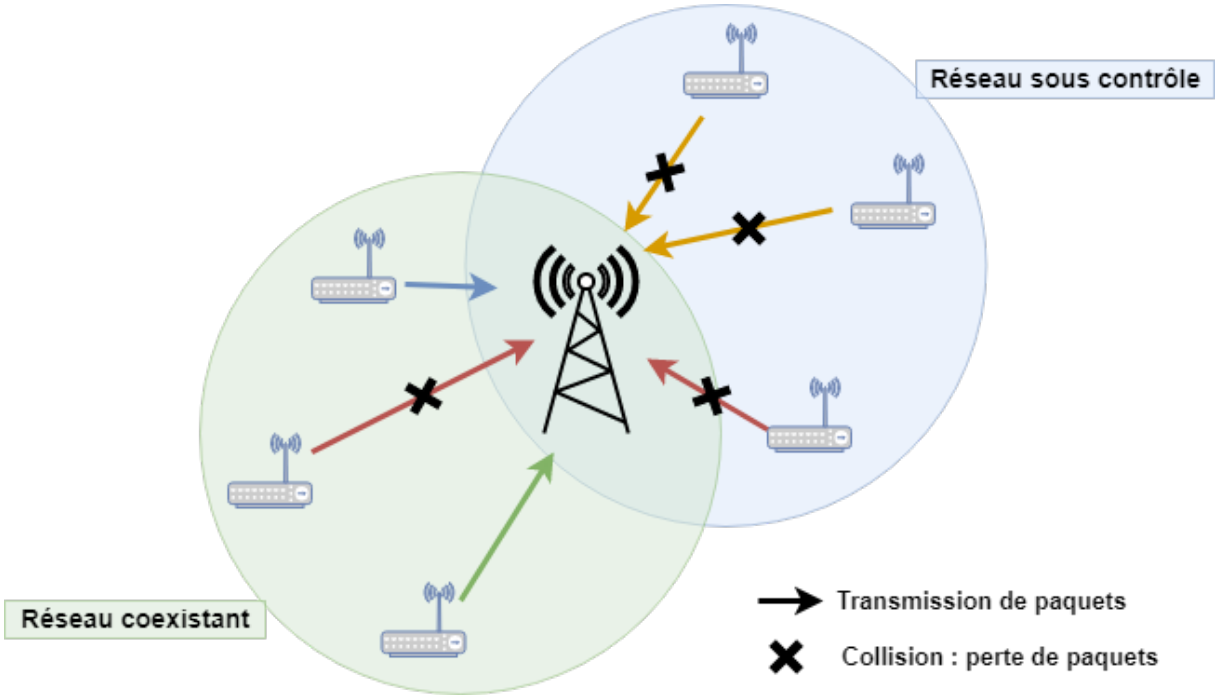



Figure 2 – Illustration des collisions internes et externes

un nombre fini de passerelles est déployé pour couvrir la zone maximale. De plus, les appareils ne peuvent pas détecter les canaux, c'est-à-dire qu'ils ne peuvent pas distinguer les collisions internes et externes, mais ils observent seulement le succès ou l'échec de leurs transmissions. Nous considérons que l'objectif est de maximiser le nombre attendu de transmissions réussies tout en consommant le moins d'énergie possible. En effet, comme les protocoles IoT utilisent les retransmissions pour assurer la *qualité de service* (QoS), augmenter les transmissions réussies conduit évidemment à diminuer le nombre de paquets envoyés et donc la consommation d'énergie. Notez cependant que, selon le problème considéré, chaque canal peut avoir une consommation d'énergie différente. Toujours dans un souci de généralité et de simplicité, nous considérons que les bras ont le même coût énergétique dans les chapitres 3 et 4. Pour situer ce scénario dans le cadre des bandits multi-joueurs à plusieurs bras, la récompense correspond à la réception ou non de l'accusé de réception de la passerelle.

Enfin, notez que seuls les équipements peuvent estimer la qualité des canaux, car la passerelle ne peut pas savoir que des paquets ont été envoyés par certains équipements si une collision se produit. Par conséquent, l'estimation de la qualité du canal doit être faite de manière *décentralisée*.

## Contributions de la thèse

Les contributions principales de cette thèse peuvent être résumées comme suit :

- Nous proposons d'abord de mettre sur liste noire les mauvais canaux dans les réseaux IoT afin d'optimiser le taux de communication réussie de tous les équipements. Pour cela, nous développons un algorithme d'exploration collaborative *CBAIMPB* grâce auquel les équipements coopèrent pour trouver, avec un certain niveau de confiance, un ensemble de canaux optimaux en utilisant des algorithmes d'identification des meilleurs bras. Nous étudions analytiquement et expérimentalement les performances de notre algorithme en termes de complexité d'échantillonnage et de coût de communication.
- La deuxième contribution représente le travail majeur de cette thèse. Afin de surmonter les faiblesses de la première approche proposée, nous proposons deux politiques qui assignent chaque joueur à un bras auquel il se tient pendant la phase d'exploitation : Decreasing-Order-Reward-Greedy (`DORG`) se concentre sur le nombre de communications réussies et nous montrons qu'il est optimal dans certains cas, tandis que Decreasing-Order-Fair-Greedy (`DOFG`) garantit en plus l'équité entre les joueurs jusqu'à un certain niveau. Afin d'implémenter une approximation des qualités du canal qui sont utilisées pour trouver les politiques, nous proposons un algorithme d'exploration décentralisé avec des échanges d'informations contrôlés entre les joueurs. Nous montrons que sa complexité d'échantillonnage est proche de l'optimum et que, lorsque `DORG` est optimal, son pseudo-regret, lorsqu'il utilise le modèle approximé, est optimal par rapport à l'horizon temporel $T$. Nous montrons également que `DOFG` est toujours équitable lorsqu'on utilise le modèle approximé. Enfin, nous fournissons des preuves expérimentales que les algorithmes proposés sont plus performants que l'état de l'art en termes d'équité et de succès des communications.
- Pour la contribution finale, nous redéveloppons un simulateur de réseau LoRa en langage C et testons les algorithmes développés. Nous étudions le taux de communication réussi et la consommation d'énergie d'un réseau LoRa, et prouvons que nos algorithmes sont plus performants que l'algorithme original Adaptive Data Rate (ADR) actuellement implémenté dans les réseaux LoRa.

Outre les contributions susmentionnées, nous fournissons également le cadre open-source de tous les algorithmes développés et les algorithmes MAB de pointe avec lesquels

nous avons comparé les nôtres, ainsi que le code du simulateur de réseau LoRa.

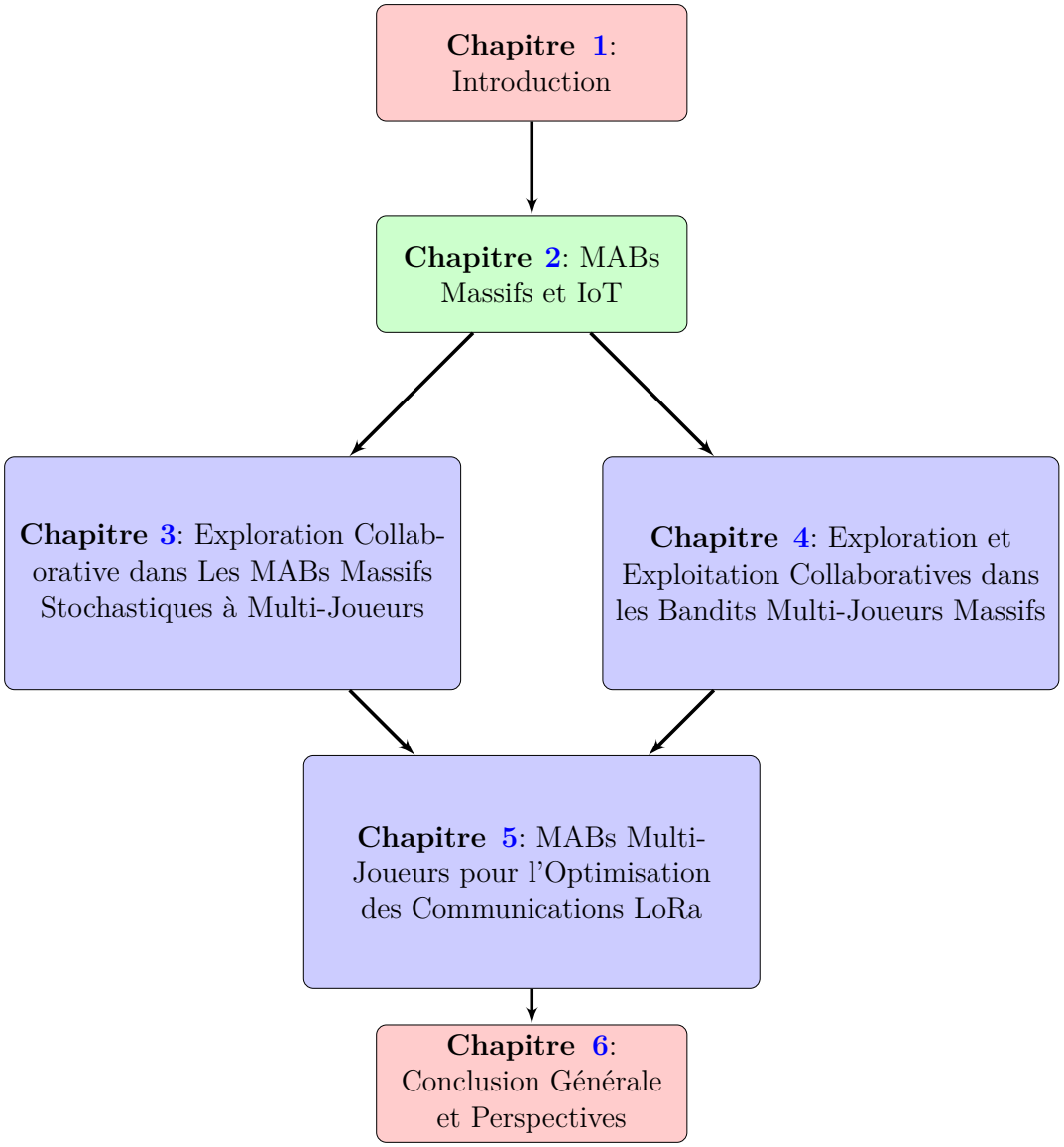



Figure 3 – La structure de la thèse

## Structure de la Thèse

La structure de cette thèse est présentée dans la figure 3 et peut être décrite comme suit: Après une chapitre introductif (chapitre 1), nous présentons au chapitre 2 les bandits multi-bras multi-joueurs et l'état de l'art de leurs applications pour les réseaux IoT. Le chapitre 3 présente ensuite notre première approche de la mise sur liste noire des canaux sous-optimaux, tandis que le chapitre 4 présente nos politiques `DORG` et `DOFG` et l'algorithme d'exploration collaborative. Dans le chapitre 5, nous présentons l'application des MABs aux réseaux LoRa en utilisant un simulateur de réseau LoRa. Enfin, nous concluons et ouvrons des perspectives au chapitre 6.

# Table of Contents

1

# INTRODUCTION

This manuscript concludes my PhD thesis, which started in January 2019 and finished by May 2022. My research was held in Orange Labs in Grenoble (France) in collaboration with IMT Atlantique in Rennes (France). This work was done under the supervision of Doctor Nadège Varsier from Orange Labs in Grenoble and Doctor Raphaël Féraud from Orange Labs in Lannion, and under the direction of Doctor Patrick Maillé from IMT Atlantique in Rennes.

In this introductory chapter, we first present the context of the thesis and the motivation behind it in section 1.1. We then introduce the main contributions of this work in section 1.2 and we terminate by presenting the structure of this manuscript in section 1.3.

## 1.1 Thesis Context

The Internet of Things (IoT) is a new, but at the same time an old term. The phrase "Internet of Things" came to life by the father of IoT Kevin Ashton [1] during a presentation he made in 1999. He used it to link the idea of radio frequency identification (RFID) to the then new topic Internet [1]. Since then, the use of this term has flourished and billions of devices are already deployed worldwide where IoT Analytics expect that by 2025, there will likely be more than 27 billion IoT connections [2] allowing a wide range of different applications. This wide and extensive expansion of IoT devices and the variety of applications generate different challenges mainly in terms of reliability and energy efficiency, which drive researchers and developers to design different efficient radio access schemes. Similarly, the rapid growth of the Machine Learning market and its applicability on a very wide range of applications paves the way to combine it with IoT, i.e. it is the time to make IoT devices intelligent.

For this sake, in this thesis we aim to improve the performance of IoT networks mainly in terms of *reliability* while consuming as little *energy* as possible and consequently

1. Kevin Ashton was cofounder and executive director of the Auto-ID Center.

improving the battery life. We propose to use Reinforcement Learning techniques and more specifically the *multi-player multi-armed bandits* (MP-MAB) for an efficient resource allocation mechanism.

> *"The Internet of Things has the potential to change the world, just as the Internet did. Maybe even more so."*
>
> – Father of IoT, Kevin Ashton

**Internet of Things** Despite the great revolution of Internet of Things, there is not yet a universally unified definition. We hereby present the simple definition of IoT provided by Wikipedia [3]: *"The Internet of things describes physical objects that are embedded with sensors, processing ability, software, and other technologies that connect and exchange data with other devices and systems over the Internet or other communications networks"*. The plurality of IoT definitions comes from the various applications and domains of IoT. It has intervened in all the aspects of our lives. It can greatly improve education, health, security, as well as decision-making and productivity of enterprises in retail, manufacturing and many other sectors. The Internet of Things along with artificial intelligence mark a significant part of the fourth industrial revolution [4]. Now, all kinds of everyday objects can be connected to the Internet, so that there are more smart devices and sensors on the IoT than there are people. These connected devices and sensors collect and share data for use and evaluation by many organizations including businesses, cities, governments, hospitals and individuals generating massive amounts of data. In order to provide and support all types of IoT applications, many different technologies with different characteristics are available.

**Wireless technologies for IoT** IoT devices use the technology that best fits their limitations. Critical IoT that covers applications including traffic safety and remote surgery in healthcare need high reliability, availability and low latency [5]. Such applications are best served by short-range technologies such as Wi-Fi, Bluetooth and ZigBee that provide coverage up to 1 km [6]. The cellular networks such as 2G/3G, 4G/LTE and 5G are suitable for long-range applications (up to 16 km) [6]. Long-range technologies such as Low-power Wide-Area Network (LPWAN) are suitable for applications that need wide coverage and long battery lifetime.

One common constraint in IoT devices is the low energy consumption such as in smart metering and industrial monitoring, as they will be deployed without direct power access and will be running on batteries that cannot be easily changed. LPWAN is a long-range technology that is characterized by its low energy consumption as the LPWAN transceivers can run on small, inexpensive batteries for up to 20 years. It provides wide coverage area especially in challenging indoor places such as in basements, and it is characterized by its low data rate, low cost and the ability to install massive number of devices.

**Radio frequency (RF) spectrum** In terms of RF spectrum, we distinguish between two types of technologies depending on the spectrum they operate in: licensed or unlicensed spectrum (license exempt frequency bands). Licensed spectrum corresponds to a part of the public frequency space that has been licensed by national or regional authorities to a private company, typically a mobile network operator, under the condition of providing a certain service to the public such as cellular connectivity [7]. The Cellular Internet of Things (CIoT) provides technologies using licensed bands for long-range applications as standardized by the 3rd Generation Partnership Project (3GPP) in its Release 13. It includes Extended Coverage GSM for Internet of Things (EC-GSM-IoT), Long Term Evolution Machine Type Communications Category M1 (LTE MTC Cat M1 or LTE-M) and Narrowband IoT (NB-IoT). The reader is referred to the book [7] for more details. Licensed spectrum is, however, commonly associated with high costs.

On the other hand, unlicensed spectrum corresponds to portions of the public frequency space that can be said to remain public and therefore free of licensing costs [7]. Among the most popular unlicensed bands are the Industrial, Scientific and Medical (ISM) bands that are centered at 2.4 GHz, 868/915 MHz, 433 MHz, and 169 MHz, depending on the region of operation [5]. By not requiring operators to obtain a costly license and special permission for its use, unlicensed spectrum is an inexpensive and barrier-free option for meeting communication requirements. Along with Wi-Fi, Bluetooth and ZigBee, LPWAN technologies commonly operate in the ISM bands. SIGFOX and LoRa are the most common known LPWAN technologies that operate in the unlicensed ISM bands. LoRa is presented in details in chapter 5.

LPWAN has attracted considerable market interest and media attention due to its operation in license-exempt bands and its support for long device battery life, low device complexity and high system capacity. However, with the great increase of IoT deployment the world is witnessing and the high demand on LPWAN, a major problem of systems'

coexistence arises. Inside the unlicensed band, the different systems are not separated in the frequency domain but are overlapping in the sense that they may use the same frequency resource at any time causing interference and hence transmission failures.

In this context, this thesis aims to solve the following problem: *How can IoT devices avoid interference amongst them and avoid interfering with coexisting networks sharing the same spectrum, in order to provide high reliability while respecting the IoT constraints, and in particular preserving low energy consumption?* Our goal is to make the IoT devices intelligent by programming and configuring them so that they are aware of the best operating parameters (e.g. transmitting power, radio frequency) such that they avoid interference and congestion.

We hereby, propose the use of the multi-player multi-armed bandit (MP-MAB) which is a field of Reinforcement Learning (RL). MP-MAB learning is a sequential decision making process, where a set of players (learners), of which each interacts with the environment by recursively taking an action and then observing a reward which is a certain measure of success of this action, produced by the environment and affected by the selections of all players in the group. The goal of each player is to maximize its rewards, by trials and errors. MP-MAB is presented in details in Chapter 2.

## 1.2 Thesis Contributions

In this thesis, we model the IoT optimization problem as a massive MP-MAB, where the devices are the players and the communication parameters (such as radio channels) are the arms, with collisions possibly preventing packet reception. Unlike previous works in MP-MABs, a high number of players, possibly greater than the number of arms, aim to optimize their communications while not sensing any type of information except the success or failure of their transmissions. In this context, the main contributions of this thesis can be summarized as follows:

- We first propose to blacklist bad channels in IoT networks in order to optimize the successful transmission rate of all devices. For this sake we develop a collaborative exploration algorithm *CBAIMPB* by which the devices cooperate to find with a certain level of confidence a set of optimal channels by using best arms identification algorithms as subroutines. We study analytically and experimentally the performance of our algorithm in terms of sample complexity and communication cost.

- The second contribution represents the major work of this thesis. In order to overcome the weaknesses of the first proposed approach, we propose two greedy policies that assign each player to one arm that it sticks to during the exploitation phase: Decreasing-Order-Reward-Greedy (`DORG`) focuses on the number of successful transmission and we show that it is optimal in some cases, while Decreasing-Order-Fair-Greedy (`DOFG`) additionally guarantees fairness between players up to a certain level. In order to implement an approximation of the channel qualities that are used to find the greedy policies, we propose a decentralized exploration algorithm with controlled information exchanges between players. We show that its sample complexity is near optimal and when `DORG` is optimal, its pseudo-regret, when using the approximated model, is optimal with respect to the time horizon $T$. We also show that `DOFG` is still fair when using the approximated model. We finally provide experimental evidence that the proposed algorithms outperform the state-of-the-art in terms of fairness and successful transmission.
- At the end, we redevelop a LoRa network simulator in C language and test our developed algorithms. We study the reliability and the energy consumption of a LoRa network, and prove that our algorithms outperform the original Adaptive Data Rate (ADR) algorithm currently implemented in LoRa networks.

Along with the aforementioned contributions, we also provide the open-source framework of all the developed algorithms and the state-of-the-art MAB algorithms we compared ours with, as well as the LoRa network simulator code.

## 1.3 Thesis Structure

The structure of this thesis is presented in Figure 1.1 and can be described as follows: after this introductory chapter, we present in Chapter 2 the multi-player multi-armed bandits and the state of the art of their applications for IoT networks. Chapter 3 then presents our first approach of blacklisting sub-optimal channels while Chapter 4 introduces our greedy policies and the collaborative exploration algorithm. In Chapter 5, we present the application of MABs on LoRa networks using a LoRa network simulator. At the end, we conclude and open perspectives in Chapter 6.

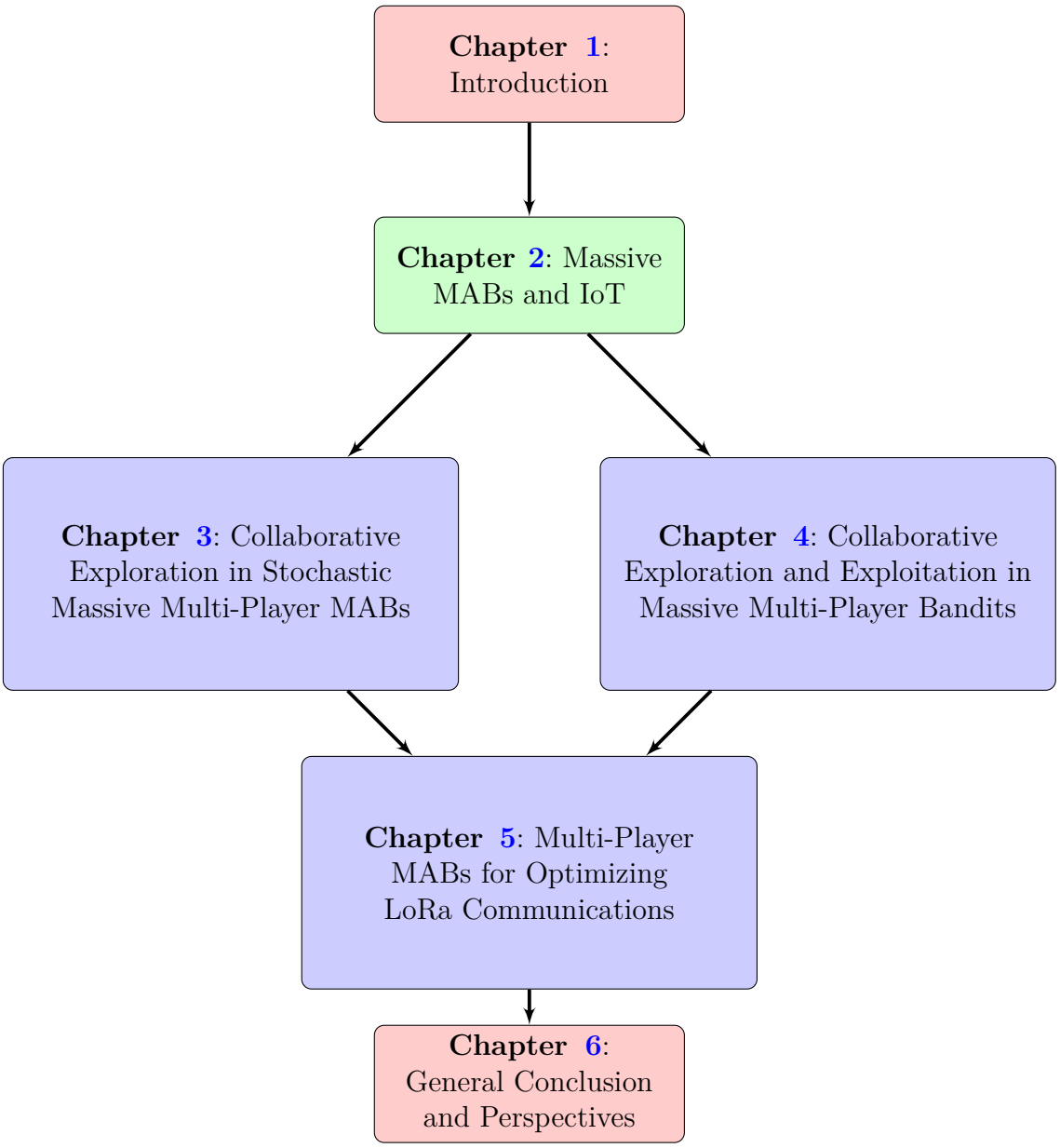



Figure 1.1 – The structure of the Thesis

2

# Massive MABs and IoT

**Key Takeaways:** In the previous chapter, we presented the high importance of IoT and its challenges. In this chapter, we present the multi-player multi-armed bandits which we aim to use to handle the optimization problem of the communications in IoT networks. IoT devices are considered as players and the communication parameters are the arms. The devices aim to optimize their communications by avoiding collisions with other devices inside and outside the network. Many previous works handled the problem of optimizing IoT communications using MABs, and are presented in this chapter.

The remaining of this chapter is organized as follows: section 2.1 presents an overview on multi-armed bandits including their classification, algorithms and applications. We then present the formulation of the problem we handle in section 2.2, and finally we present a review of the MAB applications on IoT networks in section 2.3.

## 2.1 Overview on Multi-Armed Bandits

As explained previously, in this thesis we make use of Reinforcement Learning techniques in order to optimize communications in IoT networks. More particularly, we focus on *multi-armed bandits* (MAB). MAB comes from slot machines, known as bandits as they typically take your money. It is a class of RL problems that refers to an online decision-making game where in the classical formulation of the problem, as presented in [8, 9], a set of several arms (actions) in a certain environment; each is associated to a sequence of rewards that are randomly and independently drawn according to a fixed but unknown distribution, is available to an agent (also called a player) that must take a sequence of actions while observing the corresponding sequence of rewards. Figure 2.1 illustrates this cycle. The main objective is to discover the best actions, that is, those offering the highest rewards, and to exploit them. In particular, the agent faces the so-called *exploration-*

*exploitation* dilemma: it must try all actions to learn what is the best (exploration) but needs to quickly converge to the (believed) best one to accumulate rewards (exploitation).

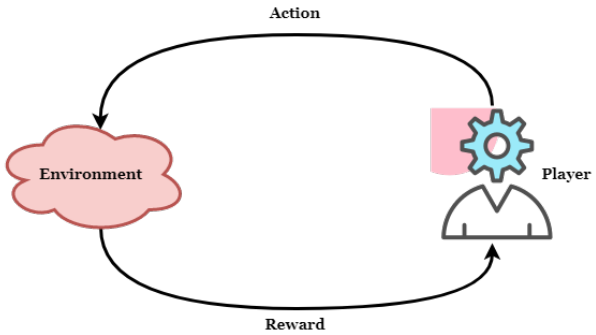



Figure 2.1 – Agent-environment interaction in a multi-armed bandit

A *policy*, or *allocation strategy* is an algorithm that chooses the next action to play based on the sequence of past plays and obtained rewards. An *optimal* policy is the policy that maximizes the cumulative reward (i.e. chooses the best arm at each iteration).

### 2.1.1 Classification of Multi-armed Bandits

Depending on the model of the reward function, there are different variants of the MAB model, of which we distinguish between:

- **Stochastic vs Adversarial**: in *stochastic* MABs [11], the rewards are generated independently from unknown stochastic distributions associated with each arm, whereas in *adversarial* MABs [12] no statistical assumptions are made whatsoever about the nature of the process generating the rewards of the arms, but an adversary, rather than a well-behaved stochastic process, has a complete control over the rewards.
- **Stationary vs Non-stationary**: stochastic MABs can be classified into *stationary* and *non-stationary* MABs. In *stationary* MABs, the rewards are generated from stochastic distributions that do not change in time, while in *non-stationary* the stochastic distribution of at least one arm may change at any time.

Identifying the model of a MAB is useful to formalize a bandit problem and solve it with the correct tools. In this thesis, we are mainly concerned with *stochastic-stationary* environments. In the following, we present some algorithms (i.e. policies) that are commonly used in solving MABs depending on their models.

### 2.1.2 Algorithms for Multi-Armed Bandits

Multi-armed bandits have been extensively studied in the literature that provided many algorithms to solve them. Before we present such algorithms, we first present how the performance of MAB algorithms (or policies) is studied.

#### Performance Metric

If the optimal policy maximizes the cumulative reward, then it should always pull one of the optimal arms (it can be not unique), but this is not realistic as the player does not know the true means, nor the optimal arms. A popular measure of a policy's success in addressing the *exploration-exploitation* dilemma is the *regret*, which is the loss due to the fact that the globally optimal policy is not followed all the times. Formally, let $[K]$ be the set of $K \geq 2$ arms, and $X_{i,t}$ be the unknown reward associated to arm $i$ at time $t$. Following policy $\pi$, at each time $t = 1, 2, ...$ the player selects an arm $k_t$ and receives the associated reward $X_{k_t,t}$. Let $t_k(t)$ denote the number of times the player selected arm $k$ up to time $t$. Then, the regret of policy $\pi$ after $T$ plays is defined by

$$R_\pi(T) = \max_{i\in[K]} \sum_{t=1}^{T} X_{i,t} - \sum_{t=1}^{T} X_{k_t,t} \tag{2.1}$$

and the expected regret is:

$$\mathbb{E}[R_\pi(T)] = \mathbb{E}\left[\max_{i\in[K]} \sum_{t=1}^{T} X_{i,t} - \sum_{t=1}^{T} X_{k_t,t}\right] \tag{2.2}$$

While the expected regret is the expectation of the regret with respect to the arm which is optimal on the sequence of the rewards, another important averaged value that compares to the optimal arm in expectation is the *pseudo-regret*, and it is defined as follows:

$$\overline{R}_\pi(T) = \max_{i\in[K]} \mathbb{E}\left[\sum_{t=1}^{T} X_{i,t} - \sum_{t=1}^{T} X_{k_t,t}\right] \tag{2.3}$$

In the stochastic setting, where the environment draws the rewards from probability distributions associated to the arms independently from the past and reveals them to the player, let $\theta_k$ denote the mean reward of arm $k$, $k^*$ denote any optimal arm, and $\theta^* := \theta_{k^*} := \max_{k\in[K]}(\theta_k)$ refer to the mean reward of an optimal arm. The remaining arms are called sub-optimal arms and let $\Delta_k = \theta^* - \theta_k$ denote the suboptimality parameter

of arm $k$. Then, the pseudo-regret of $\pi$ after $T$ plays is defined as follows.

**Definition 2.1: Pseudo-Regret**

For a policy $\pi$, a bandit problem of $K$ arms, where each arm $k$ has a mean $\theta_k$, the pseudo-regret at horizon $T$ is defined as:

$$\overline{R}_\pi(T) = \theta^*.T - \sum_{t=1}^{T} \mathbb{E}[\theta_{k_t}] \tag{2.4}$$

$$= \left(\sum_{k=1}^{K} \mathbb{E}[t_k(T)]\right) \theta^* - \mathbb{E}\sum_{k=1}^{K} t_k(T)\theta_k \tag{2.5}$$

$$= \sum_{k=1}^{K} \Delta_k \mathbb{E}[t_k(T)]. \tag{2.6}$$

In stochastic framework, the pseudo-regret presents the main quantity of interest. In the following, we present two famous pseudo-regret lower bounds that elucidate what are the best possible upper bounds that one can hope to achieve.

**Pseudo-Regret Lower Bounds** Lai and Robbins in [9], provided a *distribution-dependent* pseudo-regret lower-bound that states that player's pseudo-regret over $T$ plays can be as small as $\Omega(\log T)$. Their lower-bound applies to any one dimensional exponential distribution, but since in this work we are interested in Bernoulli distributions, we hereby present their lower bound restricted to Bernoulli families in Theorem 2.1, that is introduced and proved in [17].
First, we need to introduce the notion of Kullback-Leibler divergence for Bernoulli distributions in the following definition.

**Definition 2.2: Kullback-Leibler Divergence for Bernoulli distributions**

The Kullback-Leibler divergence between a Bernoulli of parameter $p \in [0, 1]$ and a Bernoulli of parameter $q \in [0, 1]$ is defined as

$$\mathrm{kl}(p, q) = p \log \frac{p}{q} + (1 - p) \log \frac{1 - p}{1 - q}. \tag{2.7}$$

Then the distribution-dependent pseudo-regret lower-bound is presented as follows.

**Theorem 2.1**

Consider a strategy that satisfies $\mathbb{E}[t_k(T)] = o(T^a)$ for any set of Bernoulli reward distributions, any arm $k$ with $\Delta_k > 0$, and any $a > 0$. Then, for any set of Bernoulli reward distributions the following holds:

$$\liminf_{T \to +\infty} \frac{\overline{R}_\pi(T)}{\log T} \geq \sum_{k:\Delta_k>0} \frac{\Delta_k}{\mathrm{kl}(\theta_k, \theta^*)} \tag{2.8}$$

They prove that this bound is optimal in the following sense: there does not exist a strategy for the player with a better asymptotic performance in any problem. This lower-bound is of high interest to design efficient algorithms, and a large number of research work on MAB algorithms has focused on finding algorithms whose regret upper bound matches this lower-bound asymptotically. When the regret upper-bound matches the lower-bound with the same constant as in the big-$\mathcal{O}$ notation, we say that the algorithm is *asymptotically optimal*, otherwise the algorithm is said to be *order-optimal* if it matches the lower-bound with a larger constant.

Another fundamental lower bound presented in theorem 2.2 is a *distribution-independent* lower bound that states that for certain problems there is no algorithm that performs better than $\Omega(\sqrt{KT})$ [18, 19].

**Theorem 2.2**

With a fixed time horizon $T$ and number of arms $K$, for any bandit algorithm $\pi$, there exists a bandit instance such that $\overline{R}_\pi(T) \geq \Omega(\sqrt{KT})$.

This lower bound is "*worst-case*", leaving open the possibility that certain bandit algorithms have low regret for many other problem instances.

Other than regret minimization, a commonly studied objective in MABs is *best-arms identification* which will be presented in chapter 3 .

### Algorithms for MABs

We hereby introduce the common strategies that deal with MABs.

**Stationary stochastic MABs.** The most popular bandit algorithm that deals with stationary stochastic MABs is the *Upper Confidence Bound (UCB)* algorithm [11]. It is based on the principle of optimism in face of uncertainty: when the expected reward of an arm is uncertain and the probability of it being the optimal action is high enough, the policy favours the selection of that arm. The more an arm is sampled, the more accurate is its estimate of rewards, which reduces the effect of optimism and eventually increases the selection of the action with highest mean reward. Based on this, it builds a measure of the uncertainty or variance in the estimate of each arm (i.e. UCB), then the arm with the greatest average reward plus the UCB is selected each time. UCB is *order-optimal* (i.e. its regret upper bound matches Lai and Robbins' lower bound [9]). *Upper Confidence Bound 1 Tuned (UCB1 Tuned)* [20] and *Kullback Leibler Upper Confidence Bound (KL-UCB)* [21] are two variants of UCB that are also used in stationary MABs and are respectively *order-optimal* and *asymptotically optimal*.

Another commonly used algorithm is *Thompson Sampling (TS)*. This is a Bayesian algorithm proposed by Thompson in 1933 in a medical context (online selection of the best treatment) [22]. It assumes a Bayesian prior distribution for each arm as a starting point. Then, at each round it samples from each arm distribution, chooses the action with highest sample, i.e. it chooses action with the probability that this action has the highest expected reward under the posterior distribution, and it updates its distribution using the Bayes's rule with the received reward. The updated distribution is called the posterior distribution. Authors in [23] provided finite analysis study of TS and proved it has *asymptotically-optimal* regret upper bound.

**Non-stationary stochastic MABs.** One of the popular used algorithms to deal with non-stationary stochastic MABs is *Discounted UCB*. It was first proposed by [24] and then it has been analyzed by [25]. It works by penalizing the past rewards by multiplying them with a discount factor in order to forget them and give more weight to new rewards. Another way to forget about old rewards is by using a sliding window of a fixed size $\tau$, so that only the last $\tau$ rewards are taken into account. This method is used in *sliding-window UCB* [26]. Many other algorithms that deal with non-stationarity are provided in the literature such as: *Switching Thompson Sampling (STS)* [27] , *Switching Thompson Sampling with Bayesian Aggregation (STSBA)* [28], *Discounted Thompson Sampling* [29], *Thompson Sampling with sliding window* [30] and *REXP3*[31]. Also the authors in [32]

propose several algorithms that handle non-stationarity with both *unique best arm* and *switching best arm* settings.

**Adversarial MABs** In this type of bandits, the rewards are generated by a process that cannot be considered stochastic. It can be seen as an adversary generating arbitrary rewards that make the player's policy achieve maximized regret. Hence, deterministic policies such as *UCB* cannot be applied since the adversary would easily know the selected arm and would assign it a low reward. So, we need more robust algorithms. The first and most well-known algorithm that deals with such bandits is *Exponential weights for Exploitation and Exploration (Exp3)* [33]. It selects an arm according to distributions assigned to each arm, where each is a mixture of the uniform distribution and a distribution which assigns to each action a probability mass exponential in the estimated cumulative reward for that action. It achieves an order-optimal regret upper-bound of the same order [33]. *Exp3.P* and *Exp3.S* are two variants of *Exp3* that are also used in adversarial environments [33].

The aforementioned algorithms were developed and analyzed for MAB problems with a single player. Another class of MABs is the multi-player MAB problem which is presented in the following section.

### 2.1.3 Multi-Player Multi-armed Bandits

The multi-player multi-armed bandit (MP-MAB) problem is a class of MAB problems where instead of a single agent, there exists a set $[N]$ of $N$ players, where all players have access to the same set of arms $[K]$, and have to make decisions at some pre-specified time instants and observe the corresponding outcome. In this model, the notion of collisions is introduced, i.e. whenever two or more players select the same arm at the same time, they all suffer from a collision. Different collision models have been proposed, but the simplest one consists in giving a 0 reward to each of the colliding players. In this context, the players must learn to access the arms while maximizing their rewards which necessitates avoiding collisions.

In addition to minimizing the cumulative expected regret, another common objective usually studied in multi-player MABs is *fairness*. It was recently studied in [13] in the sense that the algorithms must ensure that for all time steps $t$ (i.e. uniformly), each arm $k$ is pulled at least $\lfloor r_k.t \rfloor$ times in $t$ rounds.

In this thesis, we model our problem as a multi-player MAB, and more specifically a massive MP-MAB as the number of players (i.e. the devices in an IoT network) is possibly

greater than the number of arms. We consider *fairness* in Chapter 4, but in a different context, where it refers to the cumulative expected reward of each player. The modelling of the problem is presented in details in section 2.2, and MP-MAB algorithms designed for communication networks are then presented in section 2.3.

### 2.1.4 Applications of Multi-armed Bandits

Multi-armed bandit problems arise in a variety of application domains. The survey in [10] provides a review of top recent developments in multiple real-life applications of MABs such as in healthcare, finance, telecommunications and others. In this section we present some of such applications.

**Clinical Trials** The original application of MABs has been the design of "ethical" medical trials. An arm represents a treatment, and the reward follows a Bernoulli distribution: a 0 reward means the treatment did not heal the disease, and a 1 indicates a success. The mean of an arm here represents the mean success rate of a treatment. The doctor in a clinical trial aims to find the best treatment, which is the one with the highest mean in the shortest possible number of trials. For this sake, the doctor follows the "best arm identification" model (see Chapter 3 for details), while maximizing the rewards corresponds to maximizing the number of patients being successfully treated, i.e. attain useful scientific data while minimizing harm to the patients.

**Recommendation Services** In e-commerce and other digital domains, companies frequently want to offer personalised product recommendations to their users. They collect data on their customers' preferences and try to match up customers with the product that they are most likely to enjoy. For this sake, they would apply multi-armed bandits, where the arms correspond to items to recommend (e.g., ads, articles or movies) and the reward is the feedback from the customers. MABs then recommend products with the highest expected value of interest. In this context, authors in [34] study slowly-varying non-stationary models applied to recommender systems.

**Cognitive Radio (CR)** An application to cognitive radios has generated much interest and has been extensively studied [35, 36, 37, 38, 39, 40, 41]. As defined in Wikipedia Encyclopedia[1]: *a cognitive radio (CR) is a radio that can be programmed and configured*

1. https://en.wikipedia.org/wiki/Cognitive_radio

*dynamically to use the best wireless channels in its vicinity to avoid user interference and congestion. Such a radio automatically detects available channels in wireless spectrum, then accordingly changes its transmission or reception parameters to allow more concurrent wireless communications in a given spectrum band at one location. This process is a form of dynamic spectrum management.* This management can be handled with MABs that recommend the best transmission parameter for each communication based on last observations of previous communications, and more specifically with MP-MABs that moreover deal with multiple devices in a network.

Along with the aforementioned applications, a wide variety of successful applications of single and multi-player MABs can be found in the literature, such as A/B testing [42], Network routing [43], Dynamic pricing of items [44], tree search [45] , etc. As highlighted before, this thesis work focuses on IoT applications. In the following section, we formalize our bandit problem to best suit IoT networks.

## 2.2 Problem Statement

As previously stated, in this thesis we aim to optimize the communications in an IoT network. The end-devices will intelligently select their communication parameters at each transmission such that they maximize the overall successful transmission rate of the network. For this sake, we model our problem as a MP-MAB where the devices are the players and any resource (or combination of resources [2]) that characterizes the communications and satisfies our conditions (presented below) can be an arm (i.e. radio frequencies, transmitting power, spreading factor for LoRa, etc.). In the rest of this document, we will refer to the arms in IoT network by "channels" which does not necessarily correspond to radio channels, and the terms "player", "node" and "device" are equivalent.

In an IoT network, the number of nodes is in most cases greater than the number of channels, so considering a number of players less than the number of arms as assumed in most of previous works is unrealistic. Also, the nodes do not transmit at every time slot, but the frequency of sending packets to the gateway depends on the application (healthcare, security, smart cities, marketing, home automation...). Moreover, for several real-time applications, the device has to send a packet when an unknown and uncontrolled event occurs. For instance, a user's device can interact with its environment in real-time, to get a green light when the user faces a crossroad, an ad when the user is in front of

2. an arm can be a pair of resources, e.g. (spreading factor, transmitting power) for LoRa

a shop, a ticket when getting on the bus... That is why in the following we assume that each player has a probability of sending a packet at each time step. Considering that the probability of sending a packet depends mainly on the use case, we assume that each player knows its own probability of sending a packet. We assume that the number of players is known by the gateway, which is realistic in IoT protocols (the gateway can keep track of all the devices it has received packets from), and that the gateway sends this information to each player at the beginning of the game.

We allow the devices to share information by sending messages to other devices through the gateway using the IoT protocol, where some bytes in the payload of each packet can be dedicated to share information with other players. For example, in LoRa networks the payload of each packet can contain up to 255 bytes [14, 15], we assume that in the same packet 8 bytes of the payload can be used to send a message to other players. We hereby distinguish between the two terms: a *packet* that corresponds to the regular transmissions of a device, and a *message* that corresponds to the information shared between the players. When the device sends a packet, it is active during small time windows and may receive packets. It also receives acknowledgements from the gateway if its transmission is successful.

We consider an IoT network with a fixed number of end-nodes communicating with a single gateway following a slotted-Aloha protocol. This intelligent under-control network is geographically coexisting with other networks that may share the same spectrum and interfere with it. Collisions are taken into account when two or more nodes send data at the same time on the same channel. We hereby distinguish between two types of collisions:

- **Internal collision**: happens between the nodes in the controlled network
- **External collision**: happens between a node in the controlled network and other nodes in external coexisting networks

Figure 2.2 presents an illustration of the congestion problem, where the arrows of the same colors correspond to packet transmissions on the same channels. It differentiates between internal and external collisions.

For the sake of generality and simplicity, we consider that when an *internal collision* or an *external collision* occurs, all the packets of the colliding devices are simply lost, while the outcome of a collision depends on the IoT protocol. For instance in LoRa protocol [14, 15], the packet of the greatest received power might be decoded while the packets with lower received powers are lost [16] (in Chapter 5 we consider an application on LoRa networks). The *external collisions* make the probabilities of successful transmission (and

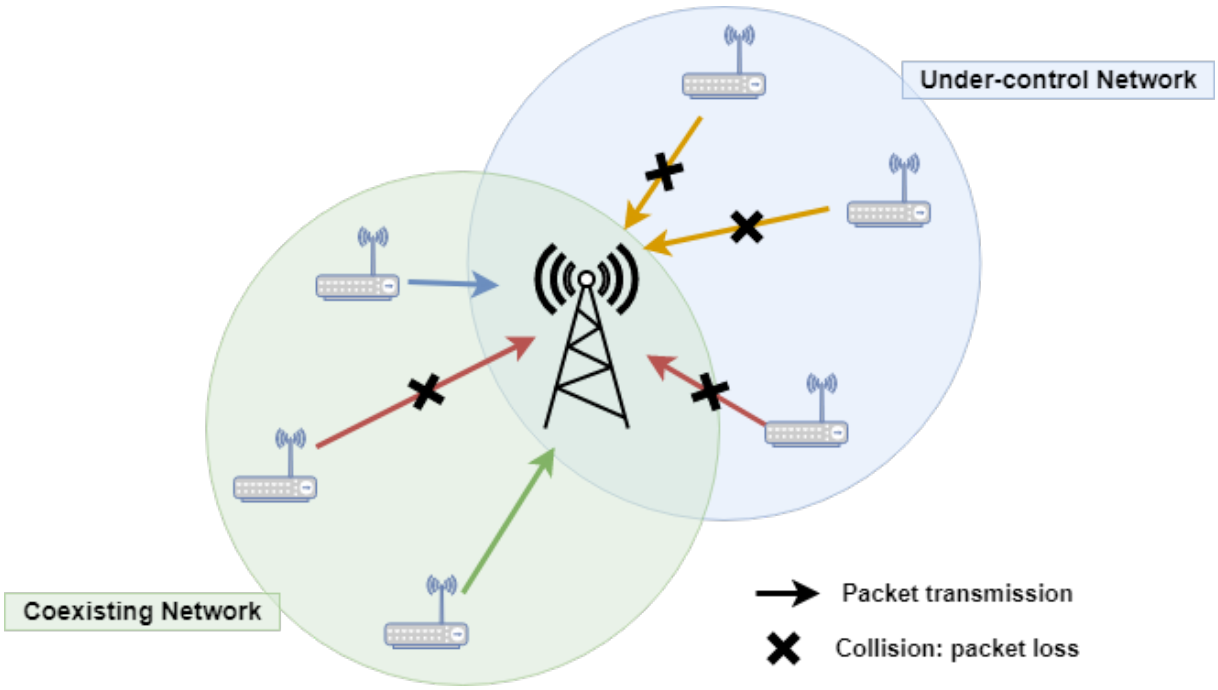



Figure 2.2 – Illustration of internal and external collisions

hence the channels' qualities) different for each channel.

Although we distinguish two types of collisions, we do not consider that collisions could occur when the gateway sends acknowledgements. Indeed, these downlink collisions require that at least two acknowledgements are sent from the gateway at the same time to devices located at the same place, which cannot happen with a unique gateway using a protocol slotted in time, and which would be unlikely in a real IoT network, where a finite number of gateways is positioned to cover the maximum area. Also, the devices cannot *sense* the channels; i.e. they cannot distinguish between internal and external collisions, but they only observe the success or failure of their transmissions.

We consider that the aim is to maximize the expected number of successful transmissions while consuming as little energy as possible. Indeed, as the IoT protocols use re-transmissions for ensuring the *quality of service* (QoS), increasing the successful transmissions obviously leads to decreasing the number of sent packets and hence the energy consumption. Notice however, that depending on the considered problem, each channel can have a different energy consumption. Again for the sake of generality and simplicity we consider that the arms have the same energy cost in Chapters 3 and 4. To set this scenario into the framework of multi-player multi-armed bandits, the reward corresponds to the reception or not of the acknowledgement from the gateway.

Finally, notice that, in the general case, the gateway cannot know that packets have been sent by some devices if a collision occurs. As a consequence, the estimation of the channel quality is done at the device side in a *decentralized* way.

**Notations** To ease the reading of the rest of this document, the reader is referred to the list of notations on page 4.

**Formulation** Formally, we consider a large set $[N]$ of $N$ devices (players) communicating with a unique gateway on a limited number $K$ of orthogonal channels ($N \geq K$), using an acknowledgement protocol slotted in time. Let $[K]$ denote the set of $K$ arms. At each time slot $t$ each player $n \in [N]$ has a constant probability $p_n$ to send a packet, such that $1 > p > p_n > 0$, where $p$ is the duty cycle that is imposed to the IoT network in order to share the free bandwidth with other users. Without loss of generality, in the following we assume that: the indices of players are sorted in decreasing order of their probability of sending a packet: $p_1 \geq ... \geq p_N$. At each time slot $t$, the set $\mathcal{N}_t$ of players sending packets is selected by $N$ independent Bernoulli samples: $\mathcal{N}_t := \{n \in [N] \text{ such that } a_n = 1, \text{ with } a_n \sim \mathcal{B}(p_n)\}$.

For a given time slot $t$, let $k_{t,n}$ (or $k_n$ when no confusion is possible) denote the arm played by player $n$. The transmission of a packet is successful if it does not collide with other packets. The random variable representing an external collision on arm $k$ is denoted by $E^k \sim \mathcal{B}(\theta^k)$ (equals 0 if collision, 1 otherwise). Similarly, internal collisions between the controlled players are represented by the random variables $(I^k)_{k\in[K]}$ (equals 0 if collision, 1 otherwise) and depend on the implemented policy. After playing arm $k$, player $n$ observes the binary outcome $Y_n^k = E^k I^k$, i.e., knows whether a collision occurred or not (through an acknowledgement) but cannot distinguish external and internal collisions.

Again, we will call a *policy* a (possibly randomized) way for players to select the channel to use for their next transmission. Formally, a policy $\pi$ will be a vector of probability distributions over the set of arms: $\pi = (\pi_1, ..., \pi_N)$, with $\pi_n = (\pi_n^1, ..., \pi_n^K)$, where $\pi_n^k \in [0, 1]$ denotes the probability that player $n$ chooses arm $k$ for sending a packet. We denote by $\mu_{n,\boldsymbol{\theta}}^k(\pi)$ the expected reward in model $\boldsymbol{\theta} = \{\theta^1, ..., \theta^K\}$ of playing arm $k$ by player $n$ while the other players follow policy $\pi$, where $\theta^k$ presents the mean reward of arm $k$. $\mu_{n,\boldsymbol{\theta}}^k(\pi)$ is the probability that no external collision occurs times the probability that no internal

collision occurs; and as the players are selecting the arms independently we then have,

$$\begin{aligned}\mu^k_{n,\boldsymbol{\theta}}(\pi) &= \mathbb{P}(\text{channel } k \text{ is not used externally}) \\ &\quad \times \mathbb{P}(\forall n' \neq n, n' \text{ is not using channel } k) \\ &= \theta^k \prod_{n'=1, n' \neq n}^{N} \mathbb{P}(n' \text{ is not using channel } k) \\ &= \theta^k \prod_{n'=1, n' \neq n}^{N} (1 - \mathbb{P}(n' \text{ is active}) \mathbb{P}(n' \text{ picks } k))\end{aligned}$$

Hence we get,

$$\mu^k_{n,\boldsymbol{\theta}}(\pi) = \theta^k \prod_{n'=1, n' \neq n}^{N} (1 - p_{n'} \pi^k_{n'}). \tag{2.9}$$

Equation (2.9) is basic for our work in the next chapters.

## 2.3 MP-MABs for Optimizing the Performance of Communication Networks: State of the Art

Multi-player multi-armed bandits have gained a great interest for optimizing the performance of communication networks in the last years, especially for dynamic spectrum access. In this section, we present recent works that handle the use of MP-MABs for optimizing the performance of communication networks.

**Opportunistic Spectrum Access (OSA)** OSA model is one of the most widely used models for dynamic spectrum access. Spectrum *sensing* is the enabling function for OSA. In OSA, as presented in [46], there are primary users PUs (licensed users) that have a strict priority over secondary users SUs (unlicensed users). In this model the SUs opportunistically access the spectrum bands of PUs which are temporally unused. Before transmission, the SUs in the OSA model need to know the busy/idle status of the spectrum bands which they are interested in. Thanks to their *spectrum sensing* capacity, they can analyze and access the unused spectrum bands of the PUs, i.e., the spectrum holes, so that the PUs' QoS will not be degraded.

*Decentralized multi-player multi-armed bandits* have been studied for OSA in [47, 48, 38, 49]. As the users can differentiate between internal and external collisions, the objective

of those works is to avoid collisions between concurrent SUs, that share the same channels, while choosing the best channels, i.e. with the highest probabilities to be free of PUs. This line of work makes the assumption that there are less players than channels ($N \leq K$), the collisions with other players are observed, and uses orthogonalization techniques to avoid collisions. In [40], the authors propose to use internal collisions to estimate in a first phase the number of players $N$ and the value of arms and then a Musical Chair approach (i.e. an approach where each player keeps hopping between the arms until it does not encounter internal collisions so it hangs on it) to allocate each player on a different $N$-best arm. In [50], the authors improve this approach by reducing the first phase to the estimation of the value of arms and then use a trekking approach to allocate each player on a different $N$-best arm without the knowledge of the number of players. In [39], the authors propose a communication protocol based on controlled collisions that achieves almost the same performance as a centralized algorithm. In [51], the authors improve this result by electing a leader that explores the arms and allocates other players on different estimated $N$-best arms. The leader communicates to other players the list of estimated $N$-best arms when it changes using the same communication protocol as in [39]. This algorithm is asymptotically optimal. An interesting extension of the problem setting was proposed in [52] for handling the case where the mean rewards of arms are not the same for each player.

This thread of research makes the assumption that *sensing* information is available, which is unrealistic for the low-cost and low-energy consumption IoT end-devices as sensing is known to be high-energy consuming as detailed in [53, 54]. Thus, the MAB model that uses *sensing* can no longer be applied. Also, it assumes that the number of players is small ($N \leq K$), which is also unrealistic for IoT networks where the number of devices is greater than the number of resources. We therefore, in this work, do not consider both assumptions, but rather consider that it is possible that $N \geq K$, and sensing information is not possible but the devices can only observe the success or failure of their transmissions.

**No sensing is possible** Other research works consider the case when *sensing* is not possible. In the same work in [39], the authors also propose an adaptation of their algorithm to the case where *sensing* is not allowed, that preserves the logarithmic behavior with respect to the time horizon. This approach has been improved in [55] thanks to the use of *Z-channel coding* for modelling no collision information, quantization of transmitted statistics and a tree structured communication, where a leader gathers the statistics and

then decides for all players the best set of arms. In [56], the authors define the multi-player stochastic multi-armed bandit as an anti-coordination game, where the goal is to quickly reach an approximate Nash equilibrium. Finally, in [41] the difficult case of non-stochastic multi-player multi-armed bandits is addressed.

Again, in all those works, the number of players is assumed to be below the number of channels, which is not realistic for IoT networks.

**Aloha-based Networks** The optimization problem we propose to solve is related to slotted Aloha protocol [57], where each player $n$ transmits a packet with a probability $p_n$ at the beginning of a slot. For instance in [58], the authors formulate the decentralized throughput maximization problem in an Aloha network with a single channel in a way that is close to our optimization problem. However that work considers a single channel, and the decision variable is the sending probability $p_n$ rather than the choice of the channel. If the probabilities of sending a packet are optimized, then the application constraints of IoT (frequency of sending packets or real-time packets) cannot be respected. In [59], the authors propose a best-response algorithm which solves the throughput maximization problem for multi-channels Aloha protocol. They notably show that the best-response algorithm converges to a Nash Equilibrium in a finite time. However they consider that the channel capacities and the strategies of other players are known, and that each player has the same probability to send a packet at each slot, which is unrealistic and restrictive for IoT networks.

**MP-MABs for IoT** Finally, motivated by IoT networks, in [60, 61] the authors propose a new problem setting where *sensing* is not allowed, the number of players is larger than the number of channels, and the players asynchronously play: each player has the same probability to send a packet at each time slot. The authors show experimentally that *selfish UCB*, which consists in each player independently playing *UCB* [11], works surprisingly well. This experimental result has been confirmed in the case of LoRa networks using stochastic and non-stochastic multi-armed bandits [62] or in the case of IEEE 802.15.4 time-slotted channel hopping protocol [63] but with single-agent MABs. Despite its good experimental performance, this algorithm has no theoretical guarantees, and it has been shown that *selfish UCB* can fail badly on some cases [61].

We hereby conclude that the literature lacks of new approaches that handle the communication optimization problem in IoT, while considering a realistic model of IoT networks.

3

# Collaborative Exploration in Stochastic Massive Multi-Player MABs

**Key Takeaways:** In this chapter, we introduce a general approach for the identification of poor-link quality channels. We develop and analyze a collaborative decentralized algorithm that aims to find a set of $m$ $(\epsilon, m)$-optimal arms using an *explore-m* algorithm (as introduced by [64]) as a subroutine, and hence blacklisting the suboptimal arms in order to improve the QoS of IoT networks while reducing their energy consumption. We prove analytically and experimentally that our algorithm outperforms selfish algorithms in terms of sample complexity with a low communication cost, and that although playing a smaller set of arms increases the collision rate, playing only the optimal arms improves the QoS of the network.

External interference may severely affect the radio channels. However, not all radio channels experience the same level of interference. Thus, as discussed in [65] in order to mitigate such inefficiency, the charged channels may be blacklisted. This concept allows IoT networks to operate only over high quality radio channels, blocking from use the heavily interfered channels. This technique has been used by a number of standardization bodies [66, 67]. In this chapter, we use the so-called *best-arms identification* algorithms, in order to identify and whitelist optimal channels (i.e. equivalently blacklisting sub-optimal channels). We develop and analyze a collaborative best-arms identification algorithm that mitigates the energy consumption. The main contributions of this chapter are:

- developing of a collaborative $(\epsilon, m)$-best arms identification algorithm
- providing a numerical and experimental analysis of the algorithm
- providing a C open-source framework of our developed algorithms

The remainder of this chapter is organized as follows: we first present the state of the art of the best-arms identification algorithms in section 3.1, then section 3.2 presents in details our collaborative algorithm that aims to find a set of m-optimal arms. In section 3.3 we provide a performance analysis of the developed algorithm. We complete and illustrate the analysis of our proposed algorithm in Section 3.4 with some experiments, and we conclude the chapter in Section 3.5.

## 3.1 Best-Arms Identification

Maximizing the aggregated expected reward of a player is one objective in MABs (i.e. minimizing the regret), another common objective is to identify the best arm(s) which correspond(s) to the arm(s) with the highest expected reward. A player learns the expected rewards of the arms during the exploration phase, and exploits the set of optimal arms afterwards. The problem of the best arms identification has been investigated thoroughly in the literature that studies several important questions arising in the probably approximately correct (PAC) framework [68]. The first question is when can an agent stop learning and start exploiting using the knowledge it obtained. The second question is which strategy leads to minimal learning time. In this context, the problem is studied in two distinct settings:

- *the fixed budget setting*: the duration of the exploration phase is fixed and is known by the forecaster, and the objective is to maximize the probability of returning the best arm, as in [69, 70, 71].
- *the fixed confidence setting*: the objective is to minimize the number of rounds needed to achieve a fixed confidence to return the best arm, as in [72, 73, 71, 74, 75].

In our work, we focus on the fixed confidence setting; we aim to find the set of optimal arms with a certain fixed *level of confidence* while minimizing the learning duration in order to save energy.

**Remark.** *An arm is said to be optimal with a confidence level* $1-\delta$ *(failure probability* $\delta$*) when this arm is optimal with probability at least* $1-\delta$*.*

The literature distinguishes between two problems: *explore-1* and *explore-m*, where the former looks for a single optimal arm and the latter looks for a set of $m$ arms. In the following we summarize the work done in both cases.

**Explore-1.** The work in [72, 70, 76, 32] studies the problem of identification of one optimal arm (i.e. *explore*-1) by a single agent. In [72], the authors propose at first the *Naive* algorithm where the player plays each arm a predefined number of times and chooses the one with the highest empirical average as the optimal arm with a certain confidence level $1-\delta$. Alternatively, the *Successive* and *Median Elimination* algorithms successively eliminate arms identified as suboptimal according to their empirical averages until only one is left, which is then labeled as the optimal one. The authors in [32] reformulate the MAB problem by generalizing it to the stationary stochastic, piecewise stationary and adversarial bandit problems in order to take into account the cases where the best arm changes over time. The work in [76] considers the problem of MP-MAB and presents the decentralized problem where a set of multiple players collaborate to find the optimal arm by asynchronously interacting with the same stochastic environment, while ensuring the privacy of players' shared information and controlling the communication cost. The authors' Decentralized Elimination algorithm uses any of the aforementioned or other *explore*-1 algorithms as a subroutine, and the players share their decisions in a decentralized manner to reach a global decision regarding the optimal arm. Our work in this thesis is built on this work, but instead we look for $m$ optimal arms rather than one.

**Explore-m.** The work in [64, 74, 73, 77] focuses on the *explore-m* problem. This work aims to find for a tolerance level $\epsilon$ and with a confidence level $1-\delta$, a set of $(\epsilon, m)-$optimal arms rather than a single arm, i.e. with a probability at least $1-\delta$, every arm in the set is $(\epsilon, m)-$optimal. An $(\epsilon, m)-$optimal arm is defined as follows:

**Definition 3.1: $(\epsilon, m)$-optimal arms**

Considering that the arms are indexed in the decreasing order of their average rewards: $\theta_1 \geq \theta_2 \geq .. \geq \theta_K$, for a given tolerance level $\epsilon \in (0, 1)$ an arm $k$ is said to be an $(\epsilon, m)$-optimal arm if:

$$\theta_k \geq \theta_m - \epsilon$$

We denote by $\mathcal{K}_{m,\epsilon}$ the set of $(\epsilon, m)$-best arms in $\mathcal{K}$.

The most common metric that measures the performance of any *explore-m* algorithm is the *sample complexity*, which refers to the number of samples needed to find the set of optimal arms. In this work, we define the sample complexity as presented below.

**Definition 3.2: Sample Complexity**

For a given $\delta \in (0,1)$, sample complexity is the total number of samples (or pulls) needed by all players to find a set of $m$ $(\epsilon, m)$-optimal arms with a confidence level $1-\delta$.

Developers aim to minimize the sample complexity of an algorithm in order to find the set of arms faster and minimize the energy consumption. For a given $\delta \in (0,1)$ and $\epsilon \in (0,1)$, the authors in [64] extend the *Naive* algorithm to find the $(\epsilon, m)$-best arms forming the *Direct* algorithm (Algorithm 1). It returns with a high probability $1-\delta$ a set of $m$ $(\epsilon, m)-$optimal arms with a sample complexity $O(\frac{K}{\epsilon^2}\log\frac{K}{\delta})$.

**Algorithm 1** Direct$(K, m, \delta, \epsilon)$ [64]

1: **for** all $k$ in $[K]$ **do**
2: Sample arm $k$ $\lceil \frac{2}{\epsilon^2}\log\frac{K}{\delta} \rceil$ times; let $\hat{\theta}_k$ be its average reward.
3: **end for**
4: Find $\mathcal{S} \subset [K]$ such that $|\mathcal{S}| = m$, and $\forall i \in \mathcal{S}, \forall j \in ([K]-\mathcal{S}) : (\hat{\theta}_i \geq \hat{\theta}_j)$
5: Return $\mathcal{S}$

They then present the *Incremental* algorithm that unlike *Direct*, proceeds through $m$ rounds. During each round, it selects an $(\epsilon, 1)$-optimal arm with a high probability by invoking the *median elimination* algorithm [72]. It ends up after $m$ rounds with a set of $m$ $(\epsilon, m)$-optimal arms. The sample complexity of both algorithms is improved with the third algorithm *Halving* which modifies the median elimination algorithm to identify $m$ arms instead of 1 and achieves a sample complexity of $O(\frac{K}{\epsilon^2}\log\frac{m}{\delta})$.

A more powerful algorithm with a lower sample complexity *LUCB* (Algorithm 2) is presented in [73]. It relies on the comparison of the lower and upper confidence bounds on the empirical averages of the arms. It starts by sampling each arm once to compute the first upper and lower confidence bounds (line 3), then at each round $t$ after sampling it computes the set $J(t)$ containing the $m$ arms of the highest empirical averages and considers two critical arms, $u_t$: the arm with the highest UCB not in $J(t)$ (i.e. among the $K-m$ arms with the lowest empirical averages), and $l_t$: the arm of the lowest LCB in $J(t)$ (i.e. among the $m$ arms with the highest empirical averages), (line 7). It stops exploring when the difference between the UCB and LCB of the critical arms is less than $\epsilon$: $U_{u_t}(t) - L_{l_t}(t) \leq \epsilon$ (line 5), and returns the $m$ arms of the highest empirical averages (line

10) . This algorithm works with any sampling strategy but the authors recommend the greedy sampling strategy with respect to the stopping rule that samples the two critical arms at each round as they are most likely to lead to a mistake.

**Algorithm 2** LUCB($K, m, \delta, \epsilon$) [73]

1: $t = 1$ (number of stage of the algorithm), $B(1) = \infty$ (stopping index)
2: **for** all $k$ in $[K]$ **do**
3: Sample arm $k$ and compute confidence bounds $U_k(1), L_k(1)$
4: **end for**
5: **while** $B(t) > \epsilon$ **do**
6: Draw an arm, t=t+1
7: Update confidence bounds, set $J(t)$ and arms $u_t, l_t$
8: $B(t) = U_{u_t}(t) - L_{l_t}(t)$
9: **end while**
10: Return $J(t)$

Another algorithm that is based on the upper and lower confidence bounds is called *Racing* and presented in [74]. *Racing* stated in Algorithm 3 was introduced first in the context of model selection for finding the (single) best model in [78]. In contrast to *LUCB*, it works by discarding and selecting arms recursively (discard the arms believed to be suboptimal and select the arms believed to be optimal) until $m$ arms are selected or $K - m$ arms are discarded. In Algorithm 3, $\mathcal{S}$ and $\mathcal{D}$ denote the sets of selected and discarded arms respectively. With *Racing* at each round the player draws all remaining arms (not selected nor discarded and contained in the set $\mathcal{R}$), and computes the set $J(t)$ that unlike *LUCB* contains the $m - |\mathcal{S}|$ empirical optimal arms, and $J(t)^c = \mathcal{R} - J(t)$ (line 4). It consequently finds the critical arms $u_t \in J(t)^c$ and $l_t \in J(t)$, then it selects the empirical best arm $k_b$ if its LCB is larger than the UCBs of all arms in $J(t)^c$, or to discard the empirical worst arm $k_w$ if its UCB is smaller than the LCBs of all arms in $J(t)$ (lines(7-11)).

The two algorithms *LUCB* and *Racing* use upper and lower confidence bounds on the mean of each arm based on Hoeffding's inequality. One has the intuition that the smaller these confidence regions are, the smaller the sample complexity of these algorithms will be. Consequently, in [74] the authors introduce the use of confidence regions based on Kullback-Leibler (KL) divergence [79] and define the *KL-Racing* and *KL-LUCB* algorithms that lead to an improved sample complexity.

As we mentioned before, we build our work in this chapter on the method presented in [76], where a collaborative, generic and decentralized algorithm is proposed to find

**Algorithm 3** Racing$(K, m, \delta, \epsilon)$ [74]

$\mathcal{R} = [K]$ set of remaining arms, $\mathcal{S} = \emptyset$ set of selected arms
$\mathcal{D} = \emptyset$ set of discarded arms. $t = 1$ (current round of the algorithm)
**while** $|\mathcal{S}| < m$ and $|\mathcal{D}| < K - m$ **do**
  Sample all the arms in $\mathcal{R}$, update confidence intervals, and compute $J(t)$ and $J(t)^c$
  Compute $u_t$ and $l_t$
  Compute $k_b$ and $k_w$ the best and worst empirical arms in $\mathcal{R}$ respectively
  **if** $(U_{u_t}(t) - L_{k_b}(t) < \epsilon)$ **or** $(U_{k_w}(t) - L_{l_t}(t) < \epsilon)$ **then**
    $k = \underset{k_b, k_w}{\operatorname{argmax}} \Big( (U_{u_t}(t) - L_{k_b}(t)) \mathbb{1}_{U_{u_t}(t) - L_{k_b}(t) < \epsilon}; (U_{k_w}(t) - L_{l_t}(t)) \mathbb{1}_{U_{k_w}(t) - L_{l_t}(t) < \epsilon} \Big)$
    if $k = k_b$ select $k$: $\mathcal{S} = \mathcal{S} \cup \{k\}$, else discard $k$:$\mathcal{D} = \mathcal{D} \cup \{k\}$
    Remove $k$: $\mathcal{R} = \mathcal{R} \setminus \{k\}$
  **end if**
  $t = t + 1$
**end while**
return $\mathcal{S}$ if $|\mathcal{S}| = m$, return $\mathcal{S} \cup \mathcal{R}$ otherwise

an $\epsilon$-approximation of the best arm using an explore-1 algorithm as a subroutine, while protecting the privacy of players' information contained in their shared messages against any adversary and controlling the communication cost. However, the problem that we address in this chapter is different, since we are looking for $m$ of the best arms (up to some $\epsilon > 0$) instead of one arm while using the aforementioned explore-m algorithms as a subroutine, while privacy is not a requirement. We use this last constraint relaxation to improve the performance in terms of sample complexity. Finally here, collisions occur, which was not considered by [76], and makes the problem harder since the true rewards of the arms cannot be observed in case of collisions. In the next section, we propose a collaborative algorithm in MP-MABs that aim to find a set of $(\epsilon, m)-$optimal arms.

## 3.2 Collaborative Exploration Algorithm

The goal of the collaborative exploration problem is to design an algorithm that minimizes the sample complexity to find a set of $m$ $(\epsilon, m)$-optimal arms, while controlling the number of exchanged messages between the players. The players are assumed to share some information through a single gateway (no direct node-to-node communication). By playing the arms and observing the corresponding rewards, the players decide what arms are optimal and eliminate the sub-optimal arms, then they share this information through the

gateway. Although sharing information would add more cost on the players, this should help decrease the sample complexity.

The basic idea behind our approach is that in order to get a set of optimal arms with a low failure probability $\delta$, each player finds a set of optimal arms but with a higher failure probability $\beta > \delta$ so the required number of samples by each player decreases. The players send to the gateway the set of arms they suggest to eliminate. However, the suboptimal arms are only really eliminated when at least a group of $\alpha$ players vote to eliminate them by sending "vote" messages. Formally, we have:

$$\mathcal{P}(\text{failure of one player}) < \beta, \tag{3.1}$$

the global decision is taken after the votes of at least $\alpha$ players, and as the players are voting independently, we have:

$$\mathcal{P}(\text{failure of } \alpha \text{ players}) < \beta^{\alpha} \tag{3.2}$$

$$\text{and we need, } \mathcal{P}(\text{failure of global decision}) < \delta \tag{3.3}$$

so we need,

$$\delta \leq \beta^{\alpha} \tag{3.4}$$

Consequently, the required number of players to eliminate an arm should be at least

$$\alpha \geq \left\lceil \frac{\log \delta}{\log \beta} \right\rceil .$$

### 3.2.1 Communication Protocol

The devices need to exchange some information in order to collaborate in our proposed approach. In order to share information, the players send messages directly to the gateway, and the latter will send usable information to all players.

In practice, a "vote" message can for example be of the form of a binary string $\lambda^n = (\lambda_1^n, ..., \lambda_K^n)$ of length $K$, sent by player $n$, indicating the indices of the arms player $n$ would like to eliminate: $\lambda_k^n = 1$ means player $n$ suggests to eliminate arm $k$. A "vote" message is sent to the gateway, and the latter waits until enough players vote to eliminate the same arms, then it sends the indices of the arms to be globally eliminated to all players.

The communication protocol that is used in the following is based on the same principle

as ALOHA, i.e. when a collision occurs the message is resent the next time the player is active.

### 3.2.2 ArmSelection Subroutine

We use in our proposed collaborative algorithm the explore-$m$ algorithms as subroutines. Those algorithms determine the players' sampling strategy of the arms, i.e. the exploration policy. Since the players cannot observe the real rewards of the played arms $\boldsymbol{\theta}$ (because of internal collisions), we introduce a new constraint on the used subroutines. Let $\rho_{n,k,\pi}$ be the probability that no collision happens on arm $k$ for player $n$ when all players follow policy $\pi$:

$$\rho_{n,k,\pi} = \prod_{n' \neq n} (1 - p_{n'}.\pi^k_{n'})$$

In order to get the same collision rate on all arms for all players, we start with a uniform exploration policy $\tilde{\pi}$, i.e., with $\forall n \in [N], \forall k \in [K], \tilde{\pi}^k_n = 1/K$, then for every player $n$ we have:

$$\rho_{n,k,\tilde{\pi}} = \rho_{n,\tilde{\pi}} := \prod_{n' \in \mathcal{N} \setminus \{n\}} \left(1 - \frac{p_{n'}}{K}\right)$$

We recall Equation (2.9) that represents the expected reward $\mu^k_{n,\boldsymbol{\theta}}(\pi)$ of the active player $n$ playing arm $k$, while the other players follow policy $\pi$:

$$\mu^k_{n,\boldsymbol{\theta}}(\pi) = \theta_k \prod_{n'=1, n' \neq n}^{N} (1 - p_{n'}\pi^k_{n'}). \tag{3.5}$$

which is the mean reward of arm $k$ multiplied with the probability that no other device plays the same arm $k$.

With that uniform exploration policy $\tilde{\pi}$, we have, for each player $n$, $\mu^k_{n,\boldsymbol{\theta}}(\tilde{\pi}) = \theta_k \rho_{n,\tilde{\pi}}$, so from (3.5)

$$\theta_m - \theta_k \leq \epsilon \Leftrightarrow \mu^m_{n,\boldsymbol{\theta}}(\tilde{\pi}) - \mu^n_{n,\boldsymbol{\theta}}(\tilde{\pi}) \leq \rho_{n,\tilde{\pi}}.\epsilon \tag{3.6}$$

As (3.6) illustrates, each player $n$ can use its observed values $Y^k_n$ to estimate $\mu^k_{n,\boldsymbol{\theta}}$, so as to find the set of $(\epsilon, m)$-best arms by looking for $(\epsilon.\rho_{n,\tilde{\pi}}, m)$-best arms. But this requires the knowledge of $\rho_{n,\tilde{\pi}}$ and hence the values of the players' active rates. Therefore, we will impose that during a first phase, the players exchange their active rates by sending them to the gateway, and the latter calculates and sends the value $\epsilon' = \rho_{\tilde{\pi}}.\epsilon := \prod_{n \in \mathcal{N}} \left(1 - \frac{p_n}{K}\right).\epsilon$

to all players. When player $n$ receives the value of $\epsilon'$, it calculates its value $\epsilon'_n = \rho_{n,\tilde{\pi}}.\epsilon := \prod_{n' \in \mathcal{N}/\{n\}} \left(1 - \frac{p_{n'}}{K}\right).\epsilon = \epsilon' / \left(1 - \frac{p_n}{K}\right)$.

Our algorithm works in epochs, we distinguish between two types of epochs:

• **Local elimination epoch** $l^n$ Using the ArmSelection subroutine every player $n$ finds a set of sub-optimal arms (once or iteratively), and locally eliminates them. Let $\overline{\mathcal{K}}^n(l^n)$and $\mathcal{K}^n(l^n)$ be the set of arms the player has locally eliminated and the set of remaining arms of player $n$ at epoch $l^n$ respectively. After each local elimination the epoch $l^n$ ends by the player's vote to eliminate this set of arms by sending messages.

• **Global elimination epoch** $l$ When enough players vote to eliminate the same arms, the arms are globally eliminated by all players at epoch $l$ and the set $\mathcal{K}(l)$ of arms remains.

The arm selection subroutine used in our algorithm is defined below:

**Definition 3.3: ArmSelection subroutine $\mathcal{A}$**

An ArmSelection subroutine $\mathcal{A}$ is an $(\epsilon, m)$-best arms identification algorithm that takes an approximation factor $\epsilon > 0$, a confidence level $1 - \beta < 1$ and a set of remaining arms $\mathcal{K}(l)$ as inputs. It is run by every player $n$: at every time slot it selects a remaining (not globally eliminated in $\mathcal{K}(l)$) arm to be played. Under specific conditions (depending on the subroutine used) it returns a set of suboptimal arms $\overline{\mathcal{K}}^n(l^n)$ locally eliminated by player $n$, so player $n$ votes to eliminate them and its epoch $l^n$ ends.

Using the ArmSelection subroutine, a device selects an arm and plays it by sending data to the gateway using the selected arm. Let $t^n$ be the total number of plays of player $n$. We denote by $\mathcal{H}_{t^n}$ the sequence of played arm indices and rewards for player $n$ up to play $t^n$, $\mathcal{H}_{t^n} = \{(k_1, y^n_{k_1}), (k_2, y^n_{k_2}), .., (k_{t^n}, y^n_{k_{t^n}})\}$. Let $f \in (0, 1]$, and $L$ be the total number of local eliminations of a single player, i.e. the value of $l^n$ when player $n$ finds a set of $m$ local optimal arms with failure probability $\beta$. We list below two properties that the ArmSelection subroutines should satisfy.

**Property 3.1: Remaining $(\epsilon, m)$-optimal arms**

For each player $n$, at each local elimination epoch $l^n$ the probability that there remain less than $m$ of the $(\epsilon, m)$-best arms (the arms in $\mathcal{K}_{m,\epsilon}$) in $\mathcal{K}^n(l^n)$ is small. More specifically,

$$\forall l^n \in \{1, .., L\}, \mathbb{P}(\{|\mathcal{K}^n(l^n) \cap \mathcal{K}_{m,\epsilon}| < m\}, \mathcal{K}^n(l^n - 1) \cap \mathcal{K}_{m,\epsilon} \geq m) \leq \beta f,$$

with $\beta$ the probability of failure of the used subroutine.

**Property 3.2: Finite Sample Complexity**

For any confidence level $1 - \beta < 1$ and approximation factor $\epsilon > 0$, the ArmSelection subroutine finds in a finite time a set of $(m, \epsilon)$-optimal arms. Formally,

$$\forall \beta \in (0, 1), \forall \epsilon > 0, \exists t^n \geq 1 \text{ s.t. } \mathbb{P}(\{\mathcal{K}^n(L) \subset \mathcal{K}_{m,\epsilon}\}|\mathcal{H}_{t^n}) \geq 1 - \beta$$

All the best-arms identification algorithms listed below satisfy the two properties. We consider three classes of $(\epsilon, m)$-optimal arms identification algorithms:

• **The fixed-design algorithms** use uniform sampling during a predetermined number of samples, such as *Direct* algorithm in [64] ($L = 1$ and $f = 1$) that eliminates the $k - m$ sub-optimal arms at the end of the sampling phase.

• **The successive elimination algorithms** are based on uniform sampling and arm eliminations. The arm, which cannot be an $(\epsilon, m)$-optimal arm with a high probability, is discarded from $\mathcal{K}^n(l^n)$. *Racing* in [74] and its variant *KL-Racing* are successive elimination algorithms ($L = K - m$ and $f = 1/(K - m)$).

• **The explore-then-commit algorithms** are based on adaptive sampling and a stopping rule. We focus on those of uniform sampling strategies. The stopping rule simply tests if the difference between the maximum of upper confidence bound of suboptimal arms and the lower confidence bound of the empirical best arm is higher than the approximation factor $\epsilon$. When the algorithm stops it eliminates the set of sub-optimal arms. $LUCB$ in [73] and its variant *KL-LUCB* in [74] are explore-then-commit algorithms ($L = 1$ and $f = 1$).

Again, due to collisions, in order to correctly estimate $\mu^k_{n,\boldsymbol{\theta}}$, the sampling strategy of any

used subroutine should be a uniform sampling strategy. In the following section we present our collaborative algorithm for best-arms identification.

### 3.2.3 Collaborative Best Arms Identification in Multi-Player Bandits

The Collaborative Best Arms Identification algorithm (see Algorithm 4) works as follows: it takes as inputs, the approximation factor $\epsilon$, the global failure probability $\delta$, the ArmSelection subroutine failure probability $\beta$, and the number of nearly-optimal arms to find $m$. Every player $n$ will run the ArmSelection subroutine $\mathcal{A}$ with an approximation factor $\epsilon'_{n,l} = \rho_{n,l}.\epsilon = \prod_{n' \in \mathcal{N}/\{n\}} \left(1 - \frac{p_{n'}}{|\mathcal{K}(l)|}\right).\epsilon$ at global elimination epoch $l$ in order to end up with a set of $(\epsilon, m)$-optimal arms. It outputs a common set of $m$ $(\epsilon, m)$-best arms for all players. The step $ack_n := \text{send}(s, k_n)$ used in our algorithm 4 means that the message $s$ is sent on channel $k_n$ to the gateway, and that a binary acknowledgement is waited for a given duration. It returns the value of the acknowledgement to player $n$ ($ack_n = 1$ if the message has been sent successfully and 0 otherwise).
The main steps of Algorithm 4 are the following:

- The players receive the gateway's messages even if they are not active and update their current sets of arms (line 2).
- The first time a player is active it sends its active rate to the gateway, and keeps sending it by selecting channels uniformly whenever it is active until it receives an acknowledgment (lines 10,12).
- Whenever a player $n$ receives the value of $\epsilon'$ from the gateway, it calculates its value of $\epsilon'_{n,l}$ (line 5).
- If an active player $n$ has sent its active rate successfully and has no new information to share with the gateway, it runs an ArmSelection subroutine with a failure probability $\beta$, and its approximation factor $\epsilon'_{n,l}$ when it is active. (line 19).
- If $\overline{\mathcal{K}}^n(l^n) \neq \emptyset$, player $n$ keeps trying to send the indexes of the arms in $\overline{\mathcal{K}}^n(l^n)$ to the gateway until it succeeds (lines 13-17).
- If enough players want to eliminate an arm, it is eliminated from the global set of arms $\mathcal{K}(l)$ with a low probability of failure $\delta$, and the gateway sends the updated set $\mathcal{K}(l)$ to all players (lines 24-28).
- When a player has found its set of $m$ optimal arms while the global set of optimal

**Algorithm 4** Collaborative Best Arms Identification in Multi-Player Bandits: CBAIMPB($\mathcal{K}$,$\mathcal{N}$,$\mathcal{A}$,$\epsilon$,$\delta$,$\beta$,$m$)

**Inputs:** $\mathcal{K}$, $\mathcal{N}$, $\epsilon \in (0,1]$, $\delta \in (0,1)$, $\beta \in (0,1)$, $m$, an ArmSelection subroutine $\mathcal{A}$
**Output:** a set of $m$ arms $\mathcal{K}(l)$
**Initialization:** $t := 1$, $l := 1$, $\mathcal{K}(l) := \mathcal{K}$, $\forall n \in \mathcal{N}$ $\epsilon'_n := 0$, $t^n := 1, l^n := 1, \mathcal{K}^n(l^n) := \mathcal{K}$,$ack1_n := 0, ack2_n^{l^n} := 0$, $\forall (n,k) \lambda_k^n := 0$

**repeat**
  every player $n \in \mathcal{N}$ gets the messages from the gateway if any and updates $\mathcal{K}^n(l^n)$
  **for** $n \in \mathcal{N}$ **do**
    **if** player $n$ receives $\epsilon'$ from the gateway **then**
      $\epsilon'_{n,l} := \dfrac{\epsilon'}{\left(1 - \frac{p_n}{|\mathcal{K}(l)|}\right)}$
    **end if**
  **end for**
  $\mathcal{N}_t$ is sampled from successive Bernoulli samples: $\mathcal{N}_t := \{n \in \mathcal{N} : a_n = 1 \text{ where } a_n \sim \mathcal{B}(p_n)\}$.
  **for** $n \in \mathcal{N}_t$ **do**
    **if** $ack1_n = 0$ **then**
      $k_n \sim \mathcal{U}(1, |\mathcal{K}(l)|)$
      $ack1_n = send(p_n, k_n)$ // $ack1_n$ indicates if $n$ has sent its active rate successfully
    **else if** $ack2_n^{l^n} = 0$ **and** $|\overline{\mathcal{K}}^n(l^n)| > 1$ **and** $|\mathcal{K}^n(l^n)| > m$ **then**
      $k_n \sim \mathcal{U}(1, |\mathcal{K}(l)|)$
      $\forall k \in \overline{\mathcal{K}}^n(l^n)$ $\lambda_k^n := 1$
      $ack2_n^{l^n} = send(\lambda^n, k_n)$ // $ack2_n$ indicates if $n$ has sent its last message $l^n$ successfully
      **if** $|\mathcal{K}^n(l^n)| > m$ **then** $l^n := l^n + 1$
    **else**
      $\overline{\mathcal{K}}^n(l^n) := \mathcal{A}(\epsilon'_{n,l}, \beta, \mathcal{K}(l))$ // if an active player has no information to send, it runs the ArmSelection subroutine and finds a set of non-optimal arms
      $\mathcal{K}^n(l^n) := \mathcal{K}^n(l^n) \setminus \overline{\mathcal{K}}^n(l^n)$
    **end if**
    **if** $|\mathcal{K}(l)| > m$ **then**
      **for** all $k \in \mathcal{K}(l)$ **do**
        **if** $\sum_{j=1}^N \lambda_k^j \geq \left\lceil \dfrac{\log \delta}{\log \beta} \right\rceil$ **then**
          $\mathcal{K}(l) := \mathcal{K}(l) \setminus \{k\}$, $l := l + 1$ // eliminate arm $k$ if enough players vote to eliminate it
        **end if**
      **end for**
      the gateway sends $\mathcal{K}(l)$ to all players
    **end if**
    **if** $|\mathcal{K}^n(l^n)| = m$ **and** $|\mathcal{K}(l)| > m$ **then**
      $t^n := 1, l^n := 1, \mathcal{K}^n(l^n) := \mathcal{K}$, $\overline{\mathcal{K}}^n(l^n) := \emptyset$ // resetting player $n$
    **end if**
  **end for**
  $t := t + 1$
**until** $\forall n \in \mathcal{N}$ $|\mathcal{K}^n(l^n)| = m$

has not been found yet, it is reset and restarts exploring the arms again so it can then vote as a new player (line 31).

- When there are only $m$ arms left in $\mathcal{K}(l)$, they are ($\epsilon$,m)-optimal arms with a high probability $1-\delta$, and the algorithm terminates (line 35).

## 3.3 Performance analysis

**Communication Cost** Theorem 3.1 states the upper bound of the communication cost (total number of sent messages for sharing information by all players) with a confidence level $1-\eta$ for obtaining a set of $(\epsilon, m)$-optimal arms with a high confidence level $1-\delta$. Due to collisions, the players need to send their messages several times until they succeed. In the ideal case when no collisions happen the players need to send at least $\gamma := \left\lceil \frac{\log \delta}{\log \beta} \right\rceil K - m + N$ messages. Theorem 3.1 takes into account the number of re-transmissions when collisions happen.

**Theorem 3.1**

Using an ArmSelection subroutine with a uniform sampling exploration strategy, the total number of sent messages by algorithm CBAIMPB to find a set of $(\epsilon, m)$-optimal arms is with a probability $1-\eta$ less than:

$$\gamma \left\lceil \frac{\log \eta/\gamma}{\log(1-\sum_{k=1}^{K} \frac{(1-p_1/K)^{N-1}}{K}\theta^k)} + 1 \right\rceil \text{ messages,} \tag{3.7}$$

with $\gamma = \left\lceil \frac{\log \delta}{\log \beta} \right\rceil K - m + N$.

*Proof.* An arm is eliminated when $\left\lceil \frac{\log \delta}{\log \beta} \right\rceil$ players vote to eliminate it. Hence, the number of sent messages to eliminate $K-m$ arms is at least $\left\lceil \frac{\log \delta}{\log \beta} \right\rceil (K-m)$. Considering the settings of no collisions at most $(\left\lceil \frac{\log \delta}{\log \beta} \right\rceil - 1).m$ messages are sent

to vote to eliminate the remaining $m$ arms (but they are not globally eliminated) and one extra message per player to share the active rates. Consequently at most $\left\lceil \frac{\log \delta}{\log \beta} \right\rceil .K - m + N$ messages are sent by all players using any ArmSelection subroutine if no collisions are taken into account.

On the other hand, considering the settings of collisions and re-transmissions, let $C(\gamma)$ be the random variable corresponding to the number of transmissions of player $n$ to send $\gamma$ messages. $C(1)$ follows a geometric distribution with a probability of success $p = \mu_n(\tilde{\pi}) = \sum_{k=1}^{K} \frac{\rho_n(\tilde{\pi})}{K} \theta_k$, and probability of failure $q = 1 - p$. Let $F$ be the number of failures before the success. We have:

$$\mathbb{P}(C(1) \leq F + 1) = 1 - q^F = 1 - \eta,$$
$$\implies F = \left\lceil \frac{\log \eta}{\log q} \right\rceil$$

Assuming that $p_1 \geq p_2, .., p_{N-1} \geq p_N$, we get $\rho_n(\tilde{\pi}) = \prod_{n' \neq n}(1 - p_{n'}/K) \leq (1 - p_1/K)^{N-1}$. Consequently, the total number of transmissions needed to successfully send $\gamma$ messages is with probability $1 - \eta$ :

$$C(\gamma) \leq \gamma \left\lceil \frac{\log \eta / \gamma}{\log(1 - \sum_{k=1}^{K} \frac{(1 - p_1/K)^{N-1}}{K} \theta^k)} + 1 \right\rceil \text{ transmissions.}$$

Substituting $\gamma$ by $\left\lceil \frac{\log \delta}{\log \beta} \right\rceil .K - m + N$, we get an upper bound on the total number of sent messages by the players using CBAIMPB algorithm and an ArmSelection subroutine of uniform sampling strategy.

□

**Exploration Duration** For the analysis of the exploration duration of our algorithm, let $T_{\mathcal{A}}$ be the number of samples needed by the ArmSelection subroutine $\mathcal{A}$ to find a set of $(\epsilon', m)$-best arms with probability of failure $\beta$, and $T^*$ be the total number of time slots when the algorithm terminates. Let $\mathcal{N}_S$ be the set of the $S = \left\lceil \frac{\log \delta}{\log \beta} \right\rceil$ most likely players, and let $p^* = \min_{n \in \mathcal{N}_S} p_n$. Theorem 3.2 provides the exploration duration of the algorithm

CBAIMPB. This value depends on the ArmSelection subroutine used. Lemma 3.1 states the time duration of CBAIMPB when **Direct** [64] algorithm is used as an ArmSelection subroutine.

**Theorem 3.2**

Using an ArmSelection$(\beta, \delta, m, \epsilon)$ subroutine with a uniform sampling exploration strategy, with a probability at least $(1-\delta)(1-I_{1-p^*}(T^*-T_{\mathcal{A}}, 1+T_{\mathcal{A}}))^{\left\lceil \frac{\log \delta}{\log \beta} \right\rceil}$ CBAIMPB terminates after:

$$\mathcal{O}\left(\frac{1}{p^*}\left(T_{\mathcal{A}} + \sqrt{\frac{1}{2}\log\frac{S}{\delta}}\right)\right) \text{time slots} \tag{3.8}$$

where $I_a(b,c)$ denotes the incomplete beta function evaluated at $a$ with parameters b and c.

*Proof.* Let $T^*$ and $T_n$ respectively be the total number of time slots and the number of samples of player $n$ when the algorithm terminates. $T_n$ is a binomial random variable with parameters $p_n$ and $T^*$. Then we have:

$$\mathbb{E}[T_n] = p_n.T^* \tag{3.9}$$

Let $T_{\mathcal{A}}$ be the number of samples needed by the ArmSelection subroutine to find a set of $(\epsilon', m)$-best arms, and let $\mathcal{B}_{\delta,\beta}$ be the set of the $S = \left\lceil \frac{\log \delta}{\log \beta} \right\rceil$ players that have the highest $T_n$. The algorithm does not stop if the following event occurs: $E_1 = \{\exists n \in \mathcal{B}_{\delta,\beta}, T_n < T_{\mathcal{A}}\}$.

Applying Hoeffding's inequality, we get:

$$\mathcal{P}(T_n - p_n.T^* \leq -\epsilon) \leq \exp^{-2\epsilon^2} = \frac{\delta}{S} \tag{3.10}$$

Then, when $E_1$ does not occur, $\forall n \in \mathcal{B}_{\delta,\beta}, T_n \geq T_{\mathcal{A}}$, so we get that with a probability at most $\delta$ every player $n \in \mathcal{B}_{\delta,\beta}$ has:

$$T_{\mathcal{A}} - p_n.T^* \leq -\sqrt{\frac{1}{2}\log\frac{S}{\delta}} \tag{3.11}$$

Then, when $E_1$ does not occur we have with a probability at most $\delta$:

$$T^* \geq \frac{1}{p_{\delta,\beta}}.(T_{\mathcal{A}} + \sqrt{\frac{1}{2}\log\frac{S}{\delta}}) \tag{3.12}$$

where $p_{\delta,\beta} = \min_{n\in\mathcal{B}_{\delta,\beta}} p_n$

Equivalently, if $E_1$ does not occur we have with a probability at least $1-\delta$:

$$T^* \leq \frac{1}{p_{\delta,\beta}}.(T_{\mathcal{A}} + \sqrt{\frac{1}{2}\log\frac{S}{\delta}}) \tag{3.13}$$

Let $\mathcal{N}_S$ be the set of the $S$ most likely players. Let $n^* = \text{argmin}_{n\in\mathcal{N}_S} p_n$, and $p^* = \min_{n\in\mathcal{N}_S} p_n$. We consider the following event: $E_2 = \{n^* \notin \mathcal{B}_{\delta,\beta}\}$. $E_2$ is equivalent to the event $\{T_{n^*} < T_{\mathcal{A}}\}$. Then we have:

$$\mathbb{P}(T_{n^*} < T_{\mathcal{A}}) = I_{1-p*}(T^* - T_{\mathcal{A}}, 1 + T_{\mathcal{A}}), \tag{3.14}$$

where $I_a(b,c)$ denotes the incomplete beta function evaluated at $a$ with parameters $b$ and $c$. Equation (3.14) comes from the relation between the incomplete beta function and the cumulative binomial distribution.

We have, $\mathbb{P}(p_{\delta,\beta} = p^*) = \mathbb{P}(\forall n \in \mathcal{N}_S, \mathbb{P}(T_n \geq T_{\mathcal{A}}))$.

Finally, knowing $|\mathcal{N}_S| = S = \left\lceil \frac{\log\delta}{\log\beta} \right\rceil$, with a probability at least $(1 - I_{1-p*}(T^* - T_{\mathcal{A}}, 1 + T_{\mathcal{A}}))^{\left\lceil \frac{\log\delta}{\log\beta} \right\rceil}$, we have $p_{\delta,\beta} = p^*$.

□

**Lemma 3.1**

With a probability at least $(1-\delta)(1-I_{1-p*}(T^*-T_{\mathcal{A}}, 1+T_{\mathcal{A}}))^{\left\lceil \frac{\log \delta}{\log \beta} \right\rceil}$, the collaborative direct algorithm stops after:

$$\mathcal{O}\left(\frac{1}{p^*}\left(\frac{K}{{\epsilon'_{n^\dagger}}^2}\log(\frac{K}{\beta}) + \sqrt{\frac{1}{2}\log\frac{S}{\delta}}\right)\right)\text{time slots} \tag{3.15}$$

where $n^\dagger = \text{argmin}_{n\in\mathcal{N}}\, p_n$.

*Proof.* The Direct algorithm in [64] finds with a probability at least $1-\beta$ a set of $m$ $(\epsilon', m)$-optimal arms with:

$$\mathcal{O}\left(\frac{K}{\epsilon'^2}\log(\frac{K}{\beta})\right) samples$$

Let $n^\dagger = \text{argmin}_{n\in\mathcal{N}}\, p_n$, so for every player $n \in \mathcal{N}$, we have:

$$\rho_{n^\dagger,\tilde{\pi}} = \prod_{n'\in\mathcal{N}/\{n^\dagger\}}\left(1-\frac{p_{n'}}{|\mathcal{K}|}\right) \leq \rho_{n,\tilde{\pi}} = \prod_{n'\in\mathcal{N}/\{n\}}\left(1-\frac{p_{n'}}{|\mathcal{K}|}\right)$$
$$\implies \epsilon'_{n^\dagger} \leq \epsilon'_n$$

Hence we get that every player $n$ finds with a probability at least $1-\beta$ a set of $m$ $(\epsilon', m)$-optimal arms with:

$$\mathcal{O}\left(\frac{K}{{\epsilon'_n}^2}\log(\frac{K}{\beta})\right) \leq \mathcal{O}\left(\frac{K}{{\epsilon'_{n^\dagger}}^2}\log(\frac{K}{\beta})\right) samples$$

Then, by substitution Theorem 2 completes the proof.

□

## 3.4 Experimental analysis

In order to illustrate and complete the analysis of our algorithm *CBAIMPB*, we implemented the aforementioned algorithms in C and developed an open-source framework that is available here [1]. We conducted several experiments that are presented below.

**Experiment 1: Cooperation vs Selfishness.** We first compare *CBAIMPB* performance using the Explore-$m$ algorithms *Direct*, *LUCB* and *Racing* as subroutines with their selfish versions. We run the algorithms with different values of $N$ and $K = 10$, such that $\forall k, \theta^k \sim \mathcal{U}(0, 1)$. Each player $n$ has a probability to be active $p_n = 1/N$. We consider $\delta = 0.1$, $\beta = 0.9$, $\epsilon = 0.2$ and $m = 4$. We study the sample complexity as well as the communication cost of our algorithm with different ArmSelection subroutines. The results are averaged over 40 experiments and the figures show 95% confidence intervals.

Figure 3.1 (a) clearly shows that our cooperative algorithm with any ArmSelection subroutine outperforms the selfish versions of them in terms of sample complexity. Regarding the subroutines, Racing outperforms LUCB and the latter has a lower sample complexity than Direct algorithm in either the cooperative or the selfish versions. On the other hand, Racing has the highest communication cost among the three algorithms as shown in Figure 3.1 (b). This is because it is a successive elimination algorithm where the players eliminate one arm successively and they send one message after each elimination, while LUCB and Direct algorithms are of the explore-then-commit and fixed-design algorithms respectively and they eliminate all the suboptimal arms when the stopping condition is provided so one message is then sent.

**Experiment 2: successful Transmission Rate.** After the players find the set of optimal arms, they need to exploit this set so that they increase their successful transmission rate, i.e. the fraction of the successfully sent packets with respect to the total number of packets. In order to study the advantage of playing a set of optimal arms instead of playing all the arms (that would increase the collision rate), we compare the successful transmission rate and the collision rate of all the players achieved by the two scenarios. For simplicity, the exploitation policy we use is the uniform policy. We run the exploitation phase with various values of $N$, such that the distribution of players is uniform and the upper bound of the distribution is chosen such that the internal collision rate does not exceed 0.2 when the number of players reaches 1300 and play the arms uniformly, so $\forall n, p_n \sim \mathcal{U}(5.4.10^{-4}, 3.8.10^{-3})$. In **scenario 1**, the players share a set of $K = 10$ arms, such

1. https://github.com/IoT-MABs

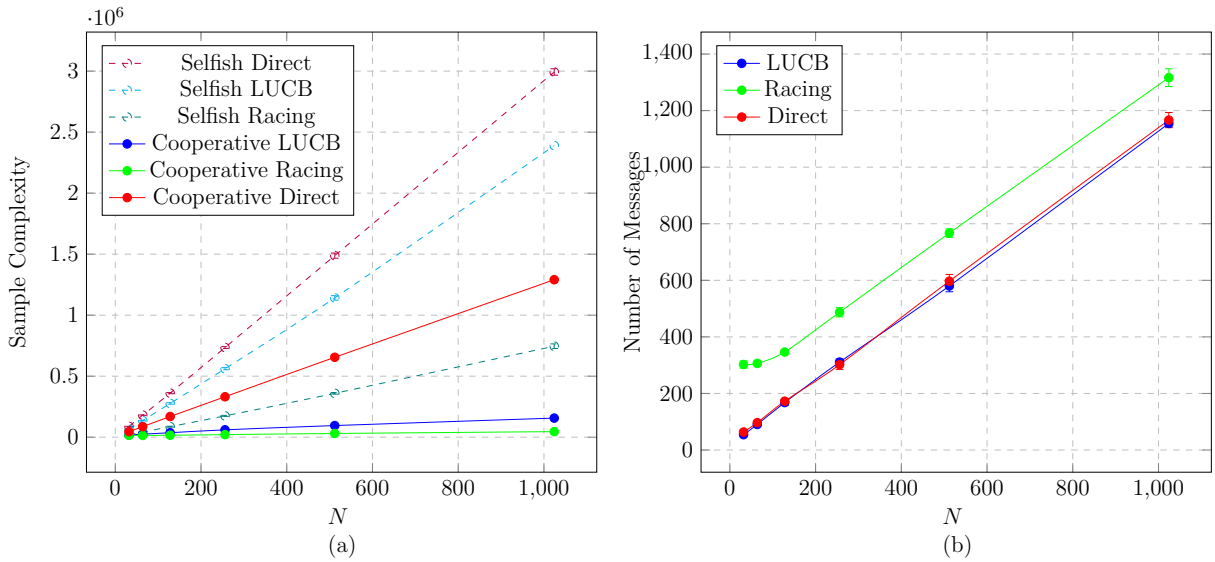



Figure 3.1 – (a) Sample complexity (cooperation vs selfishness), (b) Communication cost as a function of the number of players $N$

that $\forall k, \theta^k \sim \mathcal{U}(0,1)$. In **scenario 2**, the players find and play a set of $(\epsilon = 0.1, m = 4)$-optimal arms of the 10 previously played arms. The exploitation phase lasts for a time horizon $T = 10^6$ time slots. The results are averaged over 40 trials and the figures show 95% confidence intervals.

Figure 3.2 (a) clearly shows the advantage of playing a set of optimal arms instead of playing all available arms. Although with a smaller set of arms the internal collision rate increases as shown in Figure 3.2 (b), playing less arms of the highest qualities significantly increases the successful transmission rate.

**Experiment 3: Effect of the size of the optimal arms set.** The change in the successful transmission rate depends on the value of $m$ that should be carefully tuned and of course on the exploitation policy the players follow. In order to study the effect of the size of the set of optimal arms, we fix the value of $N = 1000$ and modify the value of $m < K$. The settings of this experiment are the same as in the previous one. Figure 3.3 shows the evolution of the successful transmission rate as well as the internal and external collision rates as a function of the size of the set of optimal arms $m$. We can notice that the successful transmission rate increases with the increase of $m$ until a certain value then it starts decreasing. This result is compatible with the variation of the collision rate. With low values of $m$, the internal collision rate is high due to the competition between the players on a small set of arms unlike the external collision rate which is low due to playing

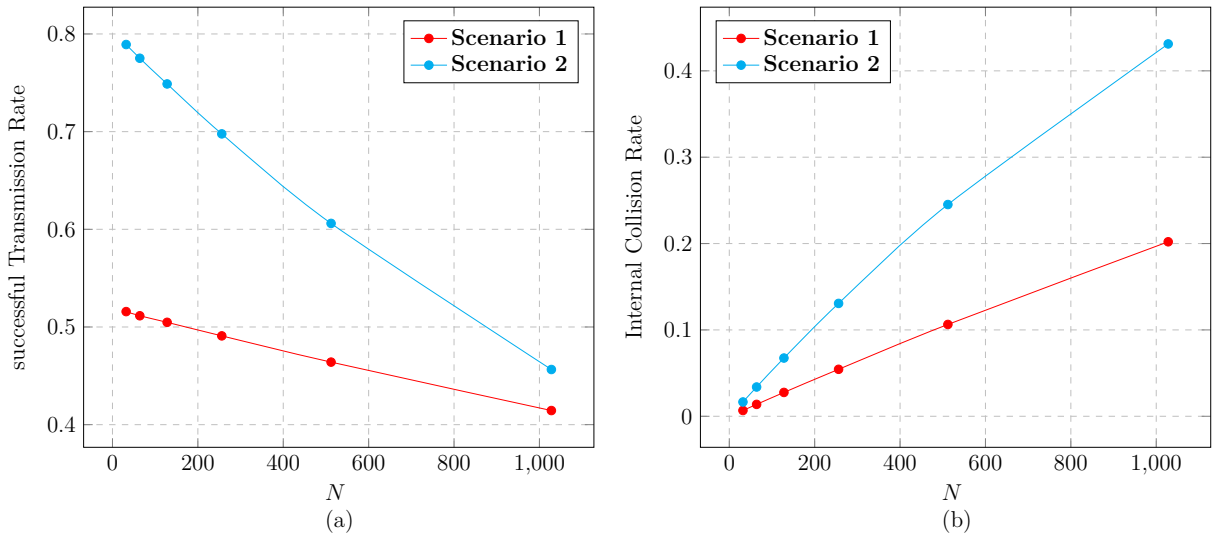



Figure 3.2 – (a) successful transmission rate, (b) Internal collision rate as a function of $N$

the optimal arms. As $m$ increases, normally the internal collision rate decreases while the external collision rate increases. Consequently, we conclude that the value of $m$ should be tuned such that we compromise between the internal and external collision rates that depend on the players' number and distribution, the arms' number and distribution and on the exploitation strategy followed by the players.

## 3.5 Conclusion

For the sake of identifying the optimal channels in an IoT network, we formulate our problem as a MP-MAB problem, and we design and analyze a new approach that aims to find a set of $m$ optimal arms by running Explore-$m$ algorithms that sample the arms uniformly as subroutines. Our approach takes into account collisions between players and does not assume any type of sensing or constraints on the number of players. The players collaborate by sharing some information, and we show that the communication cost is relatively low. We also prove experimentally that our algorithm outperforms the selfish versions in terms of sample complexity, and that although playing a smaller set of arms increases the collision rate, playing only the optimal arms increases the successful transmission rate. However, we also show that this result is highly dependent on the size of the optimal arms set $m$. It is also dependent on the players' number and distribution, the arms' number and distribution, as well as the exploitation strategy followed by the players. As a result, depending on the problem settings, we need to optimize the value of

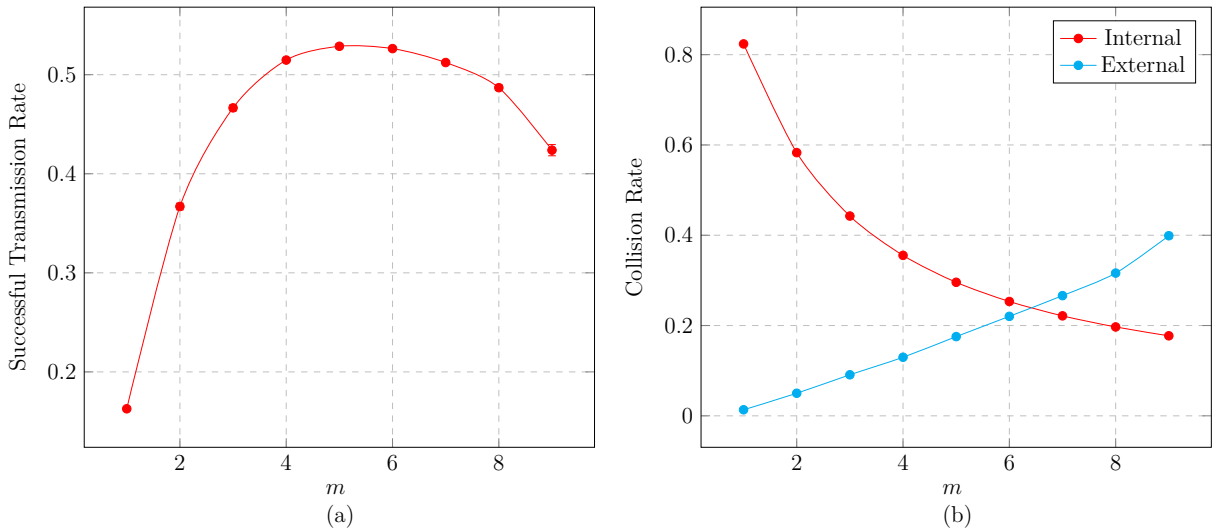



Figure 3.3 – (a) successful transmission rate, (b) Internal and external collision rate as a function of $m$

$m$ and the exploitation strategy which is still an open work for the future. Instead, in the next chapter, we propose an exploration algorithm followed by exploitation policies that improve the successful transmission rate.

The work accomplished in this chapter was published in the following article: *"Collaborative exploration in stochastic multi-player bandits."* Hiba Dakdouk, Raphaël Féraud, Nadège Varsier, and Patrick Maillé. In Asian Conference on Machine Learning, pp. 193-208. PMLR, 2020.

4

# Collaborative Exploration and Exploitation in Massive Multi-Player Bandits

**Key Takeaways:** In this chapter, we propose a new approach to optimize the performance of Internet of Things (IoT) networks. As the optimization problem is intractable, we propose two greedy policies: the first one focuses only on the number of successful transmissions, while the second one also takes into account fairness between players. In order to implement an approximation of the targeted policies, we propose an *explore-then-exploit* approach; where for estimating the mean reward of the arms, we propose a decentralized exploration algorithm with controlled information exchanges between players. Then we state that the regret of the estimated target policy is optimal with respect to the time horizon $T$. Finally, we provide some experimental evidence that the proposed algorithms outperform several baselines.

In the previous chapter, we presented a collaborative algorithm that aims to find a set of $m$-optimal arms to be exploited afterwards in order to optimize the successful transmission rate in an IoT network. However, the previous study lacks the optimization of the value of $m$, as well as the exploitation strategy of the $m$-optimal arms. In this chapter, we consider the same realistic settings of an IoT network as in the previous chapter, and we propose new policies to optimize the successful transmission rate of the network. The main contributions of this chapter are:

- We propose two deterministic target greedy policies: `DORG` (decreasing-order-reward-greedy) that aims to optimize the number of successful transmissions, while `DOFG` (decreasing-order-fair-greedy) guarantees in addition some fairness between players.
- We study the optimality of `DORG`, and the fairness of `DOFG`.

- We propose a decentralized collaborative exploration algorithm that outputs with a high probability an approximation of the mean rewards of the arms.
- We provide a deep performance analysis of the proposed algorithms.
- We provide a regret lower bound of any *explore-then-exploit* algorithm.
- We provide a numerical analysis comparing our approach with several state-of-the-art methods.
- We provide an open-source framework of our algorithms.

The remainder of this chapter is organized as follows. In section 4.1, we present the problem and we provide the two greedy policies in sections 4.2 and 4.3. Section 4.4 presents preliminary experiments that study the performance of the two greedy policies. In section 4.5, we study the *explore-then-exploit* algorithms, and we present the collaborative exploration algorithm in section 4.6 along with its analysis. Section 4.7 contains the established experiments and their results. Finally, we conclude the chapter in section 4.8.

To ease the reading of this chapter we recommend to refer to the list of notations on page 4.

## 4.1 Problem Formulation

Considering the same settings mentioned before, we first formulate the objective function to optimize. We recall Equation (2.9) that represents the expected reward in model $\boldsymbol{\theta}$ of playing arm $k$ by player $n$ while the other players follow policy $\pi$:

$$\mu^k_{n,\boldsymbol{\theta}}(\pi) = \theta^k \prod_{n'=1, n' \neq n}^{N} (1 - p_{n'} \pi^k_{n'}). \tag{2.9}$$

Equation (2.9) shows the difficulty of the studied problem: the mean reward of an arm for a given player depends on the probabilities of the other players to send a packet and on the policies they follow. The aggregated average reward in model $\boldsymbol{\theta} = \{\theta^1, ..., \theta^K\}$ per time slot over all players $\mu_{\boldsymbol{\theta}}(\pi)$ is:

$$\mu_{\boldsymbol{\theta}}(\pi) = \sum_{k=1}^{K} \theta^k \sum_{n=1}^{N} p_n . \pi^k_n \prod_{n' \in [N] \setminus \{n\}} (1 - p_{n'} \pi^k_{n'}). \tag{4.1}$$

Equation (4.1) represents our objective function to be optimized. We aim to find a policy $\pi$ followed by the players such that equation (4.1) is maximized. This performance metric

corresponds to the expected number of successful uplink transmissions per time slot in an IoT network. We can notice that this optimization problem with respect to $\pi$ has a solution, since the objective function is continuous and the set of decision variables is compact. However, the problem itself is not convex, and hence classical convex optimization methods cannot be applied. While suspected, proving the NP-hardness of the problem remains an open question.

Above all, we show in Theorem 4.1 that at least one solution is a *deterministic* policy, where each player sticks to a single arm that it will always play for sending packets.

**Definition 4.1: Deterministic policy**

A policy $\pi$ is said to be deterministic, when $\forall n \in [N], \exists\ k_n$, such that $\pi_n^{k_n} = 1$ and $\forall k' \neq k_n,\ \pi_n^{k'} = 0$.

**Theorem 4.1**

There exists a policy maximizing the overall network utility (equation (4.1)) that is deterministic.

*Proof.* We may write the global objective as:

$$\mu(\pi) = \tag{4.2}$$

$$\sum_{k=1}^{K} \underbrace{\theta^k}_{\text{mean reward of arm k}} \sum_{n=1}^{N} \underbrace{p_n.\pi_n^k}_{\text{probability that player n chooses arm k}} \underbrace{\prod_{n'=1, n' \neq n}^{N} (1 - p_{n'}.\pi_{n'}^k)}_{\text{probability that no collision occurs}} \tag{4.3}$$

Let us assume that $\pi^* = \{\pi_n\}_{n \in [N]}$ is optimal. Let us fix all player policies but player $n$'s. Then, we notice that $\mu(\pi)$ is linear (see (4.2)) in each $\pi_n^k, k = 1, ..., K$, meaning that the maximum is achieved for any $k_n^* \in \text{argmax}_{k \in [K]} \frac{\partial \mu(\pi)}{\partial \pi_n^k}$, and therefore the optimal policy may have been chosen so that $\pi_n$ is deterministic: $\pi_n^{k^*} = 1$ and $\forall k \neq k^*, \pi_n^k = 0$. The same reasoning can be repeated for the other players, so that there exists an optimal policy that is deterministic.

□

Based on Theorem 4.1, from now on, we will only consider deterministic policies, and we refer to the arm assigned to player $n$ by $k_n$. The expected reward per time slot in model $\boldsymbol{\theta} = \{\theta^1, ..., \theta^K\}$ of any deterministic policy $\pi$ can then be written as:

$$\begin{aligned}\mu(\pi) &= \sum_{n=1}^{N} p_n \theta^{k_n} \prod_{n' \neq n, \text{ s.t. } k_{n'}=k_n} (1-p_{n'}) \\ &= \sum_{k=1}^{K} \theta^k \underbrace{\prod_{n\in[N], \text{ s.t. } k_n=k} (1-p_n)}_{z^k} \underbrace{\sum_{n\in[N], \text{ s.t. } k_n=k} \frac{p_n}{1-p_n}}_{\ell^k}\end{aligned} \tag{4.4}$$

where $\begin{cases} z^k & \text{is the probability that all players assigned to arm } k \text{ do not send packets} \\ \ell^k & \text{is the sum of the activation odds for all players assigned to arm } k \end{cases}$

In the following, we will propose two deterministic policies that aim to maximize the network utility (equation (4.1)).

## 4.2 Reward greedy algorithm

We first present the *Reward Greedy* algorithm which is motivated by Lemma 4.1 stated below.

**Lemma 4.1**

For a deterministic policy $\pi$, let $\mu(\pi[n])$ denote the expected reward when only players $1, ..., n$ are playing (all players $n' > n$ are deactivated). Then we have the recursive expression:

$$\mu_{\boldsymbol{\theta}}(\pi[n]) = \mu_{\boldsymbol{\theta}}(\pi[n-1]) + p_n \theta^{k_n} \left(1 - \ell^{k_n}_{[n-1]}\right) z^{k_n}_{[n-1]},$$

where $z^k_{[n]}$ is the probability that arm $k$ is not used by any of the first $n$ players, and $\ell^k_{[n]}$ is the sum of activation odds of the $n$ first players for arm $k$.

*Proof.* We have:

$$\begin{aligned}
\mu(\pi[n]) &= \mu(\pi[n-1]) + \mu(\pi[n]) - \mu(\pi[n-1]) \\
&= \mu(\pi[n-1]) + \sum_{k\in[K]} \theta^k z^k_{[n]} \ell^k_{[n]} - \sum_{k\in[K]} \theta^k z^k_{[n-1]} \ell^k_{[n-1]} \\
&= \mu(\pi[n-1]) + \theta^{k_n} z^{k_n}_{[n]} \ell^{k_n}_{[n]} - \theta^{k_n} z^{k_n}_{[n-1]} \ell^{k_n}_{[n-1]} \qquad (4.5) \\
&= \mu(\pi[n-1]) + \theta^{k_n} \left( z^{k_n}_{[n]} \ell^{k_n}_{[n]} - z^{k_n}_{[n-1]} \ell^{k_n}_{[n-1]} \right) \\
&= \mu(\pi[n-1]) + \theta^{k_n} \left( (1-p_n)\, z^{k_n}_{[n-1]} \left( \ell^{k_n}_{[n-1]} + \frac{p_n}{1-p_n} \right) - z^{k_n}_{[n-1]} \ell^{k_n}_{[n-1]} \right) \\
&= \mu(\pi[n-1]) + \theta^{k_n} \left( -p_n z^{k_n}_{[n-1]} \ell^{k_n}_{[n-1]} + p_n z^{k_n}_{[n-1]} \right) \\
&= \mu(\pi[n-1]) + p_n \theta^{k_n} z^{k_n}_{[n-1]} \left( 1 - \ell^{k_n}_{[n-1]} \right),
\end{aligned}$$

where the line (4.5) comes from the fact that $z^k_{[n]} = z^k_{[n-1]}$ and $\ell^k_{[n]} = \ell^k_{[n-1]}$ for all $k \neq k_n$.

□

Lemma 4.1 reveals a recursion relation over $n$ of the expected total reward. Under the assumption that the problem parameters are known, Lemma 4.1 paves the way to the definition of Algorithm 5, which is a recursive algorithm that assigns player $n$ to arm $k_n$ (Line 2) such that the right-hand term of the recursive equation in Lemma 4.1 is maximized.

**Algorithm 5** Reward Greedy
(`DORG` if players are sorted in $p_n$ decreasing order)

**Inputs:** $[K]$, $[N]$, $\{\theta^k\}_{k\in[K]}$, $\{p_n\}_{n\in[N]}$
**Output:** $\pi$
**Init:** per-arm inactivity probabilities: $z^k = 1$.
**Init:** per-arm activation odds sums: $\ell^k = 0$.

1: **for** $n = 1$ to $N$ **do**
2: Set $k_n \in \operatorname{argmax}_{k\in[K]} \theta^k z^k (1-\ell^k)$.
3: Update $z^{k_n} \leftarrow z^{k_n}(1-p_n)$.
4: Update $\ell^{k_n} \leftarrow \ell^{k_n} + \frac{p_n}{1-p_n}$.
5: Set $\pi^{k_n}_n = 1$, and $\forall k \neq k_n, \pi^k_n = 0$.
6: **end for**

The result is highly dependent on the order in which the players are added to the pool

(an experimental evidence is presented in section 4.4), and we refer to the algorithm by `DORG` which stands to decreasing-order-reward-greedy when the players are sorted in the decreasing order of $p_n$. Besides, Theorem 4.2 suggests the algorithm can lead to an actual optimum.

We first present Lemma 4.2 that is used by Theorem 4.2.

**Lemma 4.2**

As long as $\ell^k_{n-1} \leq 2$, the reward-greedy criterion for Algorithm 5 decreases as we add a new player $n$:

$$z^k_{[n]}\left(1-\ell^k_{[n]}\right) \leq z^k_{[n-1]}\left(1-\ell^k_{[n-1]}\right). \tag{4.6}$$

*Proof.* We look at the difference:

$$\begin{aligned}
\forall k \neq k_n, \quad z^k_{[n]}\left(1-\ell^k_{[n]}\right) - z^k_{[n-1]}\left(1-\ell^k_{[n-1]}\right) &= 0 && (4.7)\\
z^{k_n}_{[n]}\left(1-\ell^{k_n}_{[n]}\right) - z^{k_n}_{[n-1]}\left(1-\ell^{k_n}_{[n-1]}\right) &= (1-p_n) z^{k_n}_{[n-1]}\left(1-\ell^{k_n}_{[n-1]} - \frac{p_n}{1-p_n}\right) \\
&\quad - z^{k_n}_{[n-1]}\left(1-\ell^{k_n}_{[n-1]}\right) && (4.8)\\
&= (1-p_n) z^{k_n}_{[n-1]}\left(1-\ell^{k_n}_{[n-1]}\right) - p_n z^{k_n}_{[n-1]} \\
&\quad - z^{k_n}_{[n-1]}\left(1-\ell^{k_n}_{[n-1]}\right) && (4.9)\\
&= -p_n z^{k_n}_{[n-1]}\left(1-\ell^{k_n}_{[n-1]}\right) - p_n z^{k_n}_{[n-1]} && (4.10)\\
&= -p_n z^{k_n}_{[n-1]}\left(2-\ell^{k_n}_{[n-1]}\right) && (4.11)
\end{aligned}$$

Since $p_n$ and $z^{k_n}_{[n-1]}$ are always positive, we may conclude.

□

Then, Theorem 4.2 comes to show that there exists an ordering over the players so that Algorithm 5 returns an optimal policy.

**Theorem 4.2**

If $\sum_{n\in[N]} \frac{p_n}{1-p_n} \leq K+1$, then there exists an ordering over players $\sigma^* : [N] \to [N]$ such that Algorithm 5 returns an optimal policy.

*Proof.* Lemma 4.2 states that as long as $\ell^k_{n-1} \leq 2$, the reward-greedy criterion for Algorithm 5 decreases as we add a new player $n$.

We prove below that this Lemma applies for all picked arms if $\sum_{n\in[N]} \frac{p_n}{1-p_n} \leq K+1$. By *reductio ad absurdum*, we assume that $\sum_{n\in[N]} \frac{p_n}{1-p_n} \leq K+1$ and that there exists some arm $k$ and some player ordering $\sigma$ (not necessarily $\sigma^*$) such that $\pi^*(\sigma(N)) = k$ and $\ell^k_{\sigma([N-1])} > 2$, where $\pi^*$ is an optimal policy and $\sigma([N-1])$ denotes the $N-1$ first indexes in the $\sigma$ reordering. Then, there must exist an arm $k'$ for which $\ell^{k'}_{\sigma([N-1])} < 1$, otherwise we would have $\sum_{n\in[N]} \frac{p_n}{1-p_n} > \sum_{n\in[N-1]} \frac{p_{\sigma(n)}}{1-p_{\sigma(n)}} > K+1$. It means that, for $k'$, the reward-greedy criterion $z^{k'}_{\sigma([N-1])} \left(1 - \ell^{k'}_{\sigma([N-1])}\right)$ is positive, and therefore larger than that of $k$: $z^{k}_{\sigma([N-1])} \left(1 - \ell^{k}_{\sigma([N-1])}\right)$, which is negative. As Lemma 4.1 states that the reward-greedy criterion is incrementally optimal, it means that $k'$ would have been a strictly better arm for player $\sigma(N)$, which contradicts the assumption that $\pi^*$ is optimal.

Let an optimal policy $\pi^*$ be given, and let us construct the player ordering $\sigma^*$ such that Algorithm 5 applied on the $\sigma^*$ ordering returns $\pi^*$. It is direct to understand that Algorithm 5 applied on a $\sigma^*$ player ordering would retrieve $\pi^*$. Indeed, Algorithm 6 makes it so the players are ordered to be incrementally optimal. The last piece of the proof is to check the existence of a player $\sigma^*(n)$ assigned to a reward-greedy arm on line 2.

**Algorithm 6** Reconstruction of a player ordering that allows Algorithm 5 to return $\pi^*$

**Inputs:** $[K]$, $[N]$, $\{\theta^k\}_{k\in[K]}$, $\{p_n\}_{n\in[N]}$, $\pi^*$

**Output:** $\sigma^*$ such that Algorithm 5 returns $\pi^*$

**Init:** per-arm inactivity probabilities: $z^k = 1$.

**Init:** per-arm activation odds sums: $\ell^k = 0$.

**Init:** Set of players remaining to be assigned: $\mathcal{N} = [N]$.

1: **for** $n = 1$ to $N$ **do**
2: Let $\sigma^*(n)$ be an element of $\mathcal{N}$ such that $\pi^*(\sigma^*(n)) \in \operatorname{argmax}_{k\in[K]} \theta^k z^k(1-\ell^k)$.
3: Update $\mathcal{N} \leftarrow \mathcal{N} - \{\sigma^*(n)\}$.
4: Update $z^{k_n} \leftarrow z^{k_n}\left(1 - p_{\sigma^*(n)}\right)$.
5: Update $\ell^{k_n} \leftarrow \ell^{k_n} + \frac{p_{\sigma^*(n)}}{1-p_{\sigma^*(n)}}$.
6: **end for**

Again by *reductio ad absurdum*, we assume that there is no remaining player that $\pi^*$ assigned to a reward-greedy arm $k^*$. Then, it means that until the last selection, this arm will not be picked and another arm $k$ will be picked instead. We showed at the beginning of the proof that the reward-greedy criterion is only decreasing as the arms are being selected, and that the reward-greedy criterion of an arm not being selected, such as $k^*$, is constant. So it means that $\pi^*(\sigma^*(N))$ should be $k^*$, hence, the contradiction.

We may therefore conclude the proof by stating that Algorithm 6 will never fail to construct $\sigma^*$ and that Algorithm 5 applied to the $\sigma^*$ player ordering will return $\pi^*$.

□

Note that when $\forall n, p_n = p$ (the settings studied in [60]), Theorem 4.2 states that `DORG` returns an optimal policy. The precondition of Theorem 4.2 clearly holds in IoT networks, where the duty cycle $p$ is commonly set to less than 0.01.

Although, `DORG` works on optimizing the network utility, it does not guarantee any fairness between the players. In the following section, we present a new policy that takes into account the fairness between the players along with optimizing the network utility.

## 4.3 Fairness greedy algorithm

A consequence of Theorem 4.1 is that the resource assignment of an optimal deterministic policy is a Pareto optimum: as the network utility is maximum, if a user increases its own utility (equation (2.9)) another user has necessarily to decrease its utility (due to equation (4.1)). Notice that a Pareto optimum does not provide any guarantee about the *fairness* of the resource allocation among players. In this section, we design a policy to ensure *fairness* among players which is defined as follows:

**Definition 4.2: $\alpha$-fairness**

A policy $\pi$ is said to be $\alpha$-fair if $\frac{\min_{n\in[N]} \mu_{n,\boldsymbol{\theta}}(\pi)}{\max_{n\in[N]} \mu_{n,\boldsymbol{\theta}}(\pi)} \geq \alpha$, where $\mu_{n,\boldsymbol{\theta}}(\pi) = \sum_{k=1}^{K} \pi_n^k . \mu_{n,\boldsymbol{\theta}}^k(\pi)$

**Algorithm 7** Fairness Greedy
(`DOFG` if players are sorted in $p_n$ decreasing order)

**Inputs:** $[K]$, $[N]$, $\{\theta^k\}_{k\in[K]}$, $\{p_n\}_{n\in[N]}$
**Output:** $\pi$
**Init:** per-arm inactivity probabilities: $z^k = 1$.

1: **for** $n = 1$ to $N$ **do**
2: Let $k_n \in \operatorname{argmax}_{k\in[K]} \theta^k z^k$
3: Update $z^{k_n} \leftarrow z^{k_n}(1 - p_n)$
4: Set $\pi_n^{k_n} = 1$, and $\forall k \neq k_n, \pi_n^k = 0$.
5: **end for**

Building a fair policy can be done by balancing the load with respect to the mean rewards of the arms. The fairness greedy algorithm (see Algorithm 7) assigns sequentially each player to the arm that maximizes the reward of the arm times the probability of no internal collision. The player scheduling also plays an important role and we prove a lower bound on the fairness of Algorithm 7, when players are sorted in decreasing order of $p_n$. In that case we coin this algorithm `DOFG`, which stands for decreasing-order-fair-greedy.

In Theorem 4.3 we study the fairness of `DOFG`.

**Theorem 4.3**

`DOFG` generates $\alpha$-fair policies, with $\alpha \geq 1 - \max_{n\in[N]} p_n$.

*Proof.* For every arm, we have the following equality:

$$\mu_n(\pi^\dagger) = \theta^{k_n} \prod_{n' \neq n, \text{ s.t. } k_{n'}=k_n} (1 - p_{n'}) = \frac{\theta^{k_n} z^{k_n}}{1 - p_n}. \tag{4.12}$$

We prove now that $\min_{n\in[N]} \mu_n(\pi^\dagger) = \mu_N(\pi^\dagger)$. We proceed by induction. The base case is direct for $N = 1$. Now, we prove the induction step by assuming that it is true for $N$ and prove it for $N + 1$. We have to distinguish two cases whether $k_N$ equals $k_{N+1}$ or not.

Case $k_N = k_{N+1}$, then from Equation 4.12, we have $\mu_{N+1}(\pi^\dagger) = \frac{1-p_N}{1-p_{N+1}}\mu_N(\pi^\dagger)$. Since we know by construction that $p_{N+1} \leq p_N$, we may conclude that $\mu_{N+1}(\pi^\dagger) \leq \mu_N(\pi^\dagger)$.

Case $k_N \leq k_{N+1}$, then stating that $\mu_{N+1}(\pi^\dagger) > \mu_N(\pi^\dagger)$ would imply that $k_N$ was not optimally selecting the arm at the previous step, which brings a contradiction.

Let us assume without loss of generality that player $N$ has been assigned to arm $K$. Since $\pi^\dagger_N$ has been chosen so that to maximize $\theta^k z^k$ at iteration $N$, it means that:

$$\min_{n\in[N]} \mu_n(\pi^\dagger) = \mu_N(\pi^\dagger) \geq \max_{k\in[K]} \theta^k z^k. \tag{4.13}$$

We also know that:

$$\begin{aligned}
\max_{n\in[N]} \mu_n(\pi^\dagger) &= \max_{n\in[N]} \frac{\theta^{k_n} z^{k_n}}{1 - p_n} && (4.14)\\
&\leq \frac{\max_{k\in[K]} \theta^k z^k}{1 - \max_{n\in[N]} p_n} && (4.15)\\
&\leq \frac{1}{1 - p_1} \min_{n\in[N]} \mu_n(\pi^\dagger), && (4.16)
\end{aligned}$$

which concludes the demonstration.

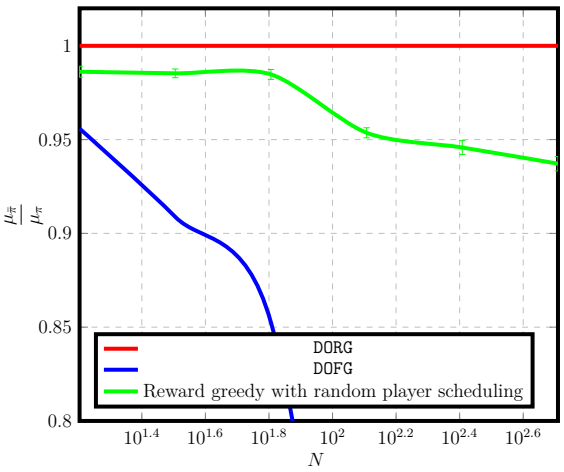
□

Theorem 4.3 implies that when the probability of sending packets of the most frequent player is not high, which is the case in IoT networks, `DOFG` is a fair policy.

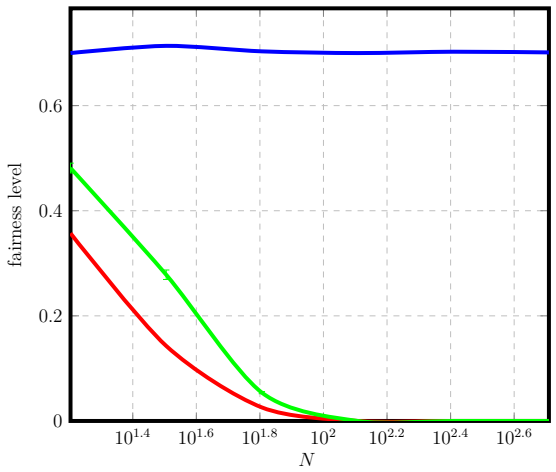
In the following section, we provide an experimental evidence on the performance and fairness of `DORG` and `DOFG`.

## 4.4 Preliminary Experiments

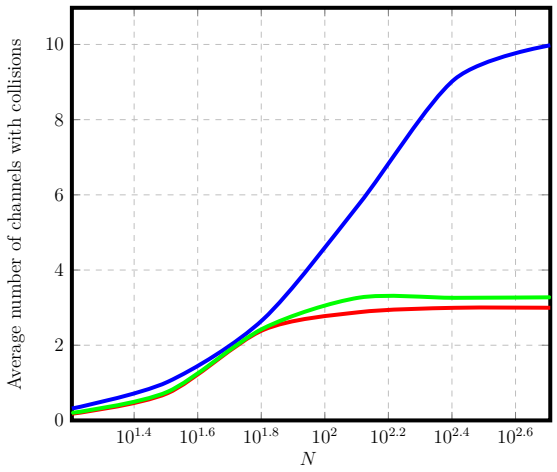
We perform several experiments to study and compare the performance of DORG and DOFG as explained below.




(a) Expected reward ratio w.r.t DORG




(b) Level of fairness between players




(c) Average number of channels with internal collisions

Figure 4.1 – Experiment 1: with a fixed number of arms $K = 10$, and for different values of $N$ (ranging from 16 to 512 on a log scale), the performance of DORG, DOFG, and Reward Greedy (Algorithm 5) with random ordering is compared.

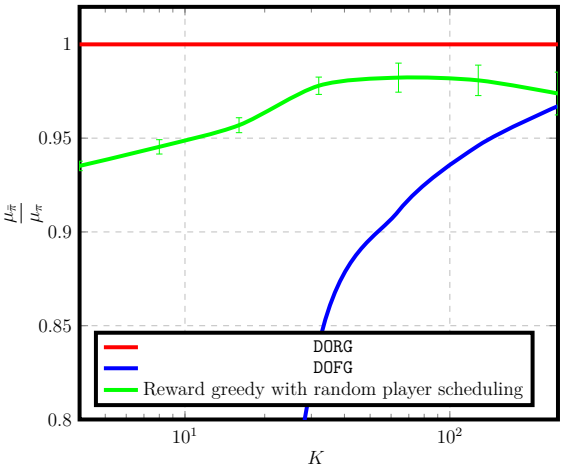
**Experiment 1.** The problem parameters are sampled as follows: $\forall n \in [N], p_n \sim \mathcal{U}(0, 0.3)$ [1] and $\forall k \in [K], \theta^k \sim \mathcal{U}(0, 1)$. Figure 4.1 compares the performance of DORG, DOFG, and Reward Greedy (Algorithm 5) with random ordering, where each point is

1. Such high values for $p_n$ are used to graphically observe the expected properties.

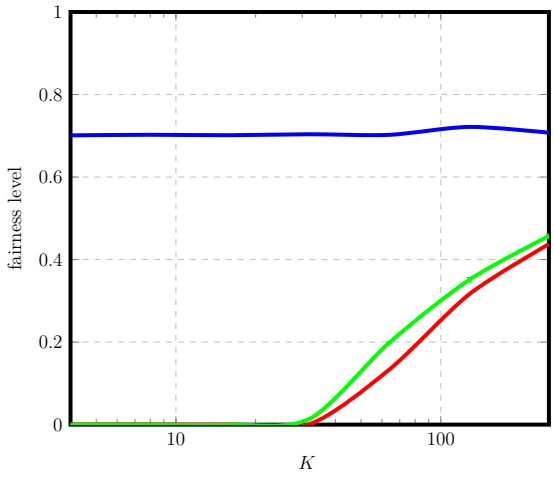
the average of 10,000 runs. Figure 4.1a that compares the expected reward ratio of the algorithms with respect to `DORG`, where $\bar{\pi}$ denotes the policy to be compared with `DORG`, reveals that sorting the players in decreasing order is a good policy. However, it has to be noted that the difference between `DORG` and a random ordering is much thinner when $p_n$ are smaller, as expected in a real setting. We also notice that `DOFG` expected reward loss, as compared to `DORG`, is below 20% until $N \approx 75$. Figure 4.1b illustrates the result of Theorem 4.3, and indicates that the fairness lower bound is tight. It also shows that, while `DOFG` only loses 20% rewards when $N \approx 75$ as compared to `DORG`, its fairness is approximately 30 times larger.

Further, on figure 4.1c, we notice that the expected number of channels experiencing internal collisions per time slot stops increasing as $N$ grows around $N = 100$. It is the moment when the channels get completely saturated. $N = 100$ coincides with the point where the fairness gets to 0 on figure 4.1b. We explain this phenomenon as follows: each channel $k$ fills up, up to the point when $\ell^k > 1$. When all the channels reach this point, adding new players to the network actually decreases the expected reward, and `DORG`'s strategy condemns the arms with the lowest $\theta^k$ and use them as a garbage bin for new players. These channels get so crowded that there is a collision on it with a very high probability, in order to keep the other channels functionally unspoiled. In comparison, to guarantee fairness `DOFG` does not throw away players on a bin channel.

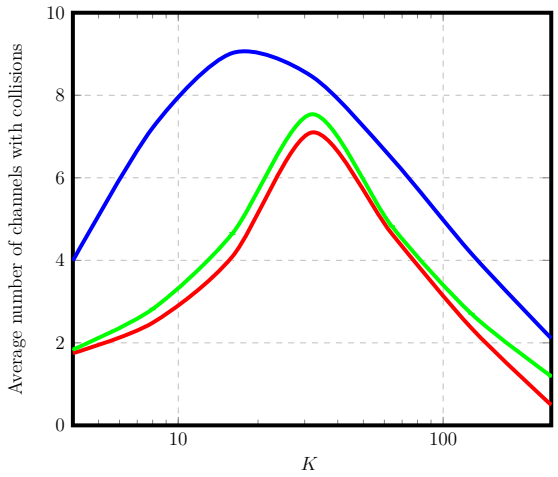
**Experiment 2.** The same experiment is carried out but with fixed $N = 200$ and different values of $K$ ranging from 4 to 256 are conducted, and the results are presented in Figure 4.2. As obvious, the performance of any algorithm improves as the number of arms increases since the number of internal collisions decreases. The fairness level increases with DORG and the reward greedy with random ordering as $K$ increases since the players have greater opportunities to be located on arms with high mean rewards and hence they experience close expected rewards. As we notice in Figure 4.2c, the number of channels with internal collisions starts decreasing at a certain point. This is because the players are more and more assigned to different arms.




(a) Expected reward ratio w.r.t DORG



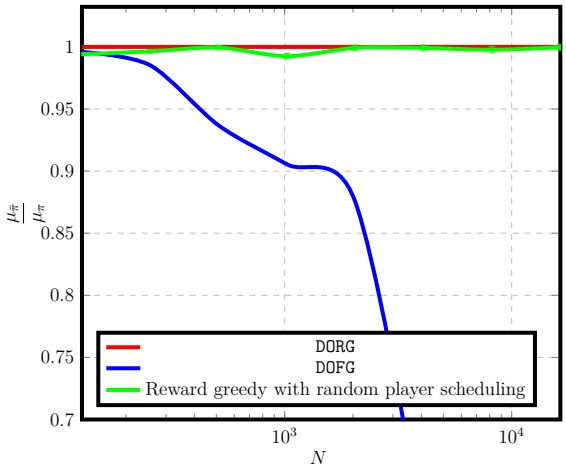

(b) Level of fairness between players



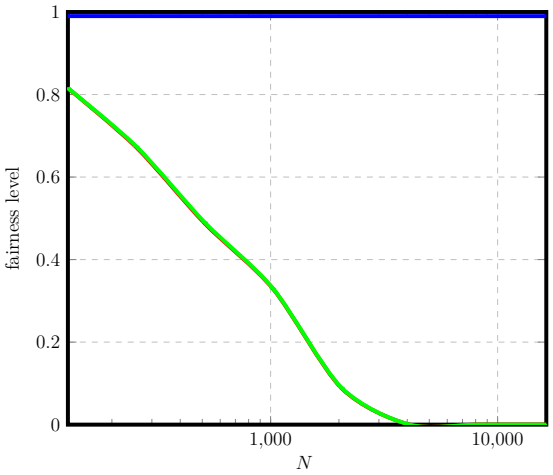

(c) Average number of channels with internal collisions

Figure 4.2 – Experiment 2: with a fixed number of players $N = 200$, and for different values of $K$ (ranging from 4 to 256 on a log scale), the performance of DORG, DOFG, and Reward Greedy (Algorithm 5) with random ordering is compared.

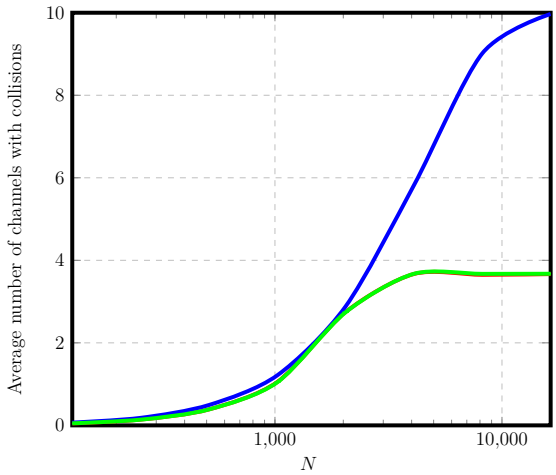
**Experiment 3.** We studied the case of lower values of $p_n$. The same experiments were carried out with a fixed number of arms $K = 10$ and different values of $N$ ranging between 128 and 16384, but with $\forall n \in [N], p_n \sim \mathcal{U}(0, 0.01)$. The results are presented in Figure 4.3. Compared to Figure 4.2, we can notice that the results of the Reward greedy with random player scheduling tends to those of DORG when the values of $p_n$ are smaller which is the case in IoT networks.




(a) Expected reward ratio w.r.t `DORG`




(b) Level of fairness between players




(c) Average number of channels with internal collisions

Figure 4.3 – Experiment 3: with a fixed number of arms $K = 10$, and for different values of $N$ (ranging from 128 to 16384 on a log scale), the performance of `DORG`, `DOFG`, and Reward Greedy (Algorithm 5) with random ordering is compared.

In the previous experiments, we assumed that the model $\boldsymbol{\theta}$ and the probabilities to send packets of the players $p_n$ are known. However, this is not true in IoT networks. Consequently, this information will be learnt during a preceding exploration phase presented and studied in the following.

## 4.5 Explore-then-Exploit Approach

The choice of the policy depends on the metric to be maximized: for maximizing network utility, `DORG` policy (Algorithm 5) should be used, while to guarantee some fairness among players, `DOFG` policy (Algorithm 7) is to be used. However both policies necessitate the model $\boldsymbol{\theta}$, which is unknown: hence exploration is necessary. To maximize the objective metric, we propose an *explore-then-exploit* approach: an exploration algorithm shares the probabilities of sending packets of players and outputs an $\epsilon$-approximation of the model $\boldsymbol{\theta}$ (i.e. estimations of the arms' mean rewards) with high probability for a sufficiently small $\epsilon$, and then a target policy is used during the exploitation phase (`DORG` or `DOFG`).

**Definition 4.3: $\epsilon$-approximation**

$\hat{\theta}^k$ is said to be an $\epsilon$-approximation of arm $k$, if the difference between it and $\theta^k$ is less than $\epsilon : |\theta^k - \hat{\theta}^k| \leq \epsilon$.

Several approaches are available for exploration. A first approach could be a *selfish exploration*, where each player explores the value of arms with the packets it has to send, and then computes the target policy. The drawback of *selfish exploration* is the exploration time, which is driven by the least frequent player. As the probability of sending packets of players is needed to compute the target policies in the exploitation phase, the communication cost of *selfish exploration* is the number of transmissions needed to share the probabilities of sending packets of players. Another approach could be to perform a *follow-the-leader exploration* for estimating the mean rewards of arms: the gateway assigns one single player to be in charge of exploration while the others do not send packets during exploration phase in order to avoid internal collisions. As in IoT networks, the devices have to respect the duty cycle, which can be of the same order as the probabilities of sending packets of devices, it is advantageous to choose the most frequent player as the leader. Follow-the-leader exploration explores faster than *selfish exploration* with the packets the leader has to send, and the communication cost is only increased by $K$ to send its approximation of the mean reward of each arm. The main drawback of *follow-the-leader exploration* is that during the exploration phase the packets of other players are lost.

We rather propose a *decentralized collaborative exploration algorithm* (Algorithm 8) of lower sample complexity, exploration duration and low communication cost. Since the gateway cannot observe the collisions (packet losses), the learning (exploration) should be

done at the device (player) side, so our algorithm for exploring the mean rewards of arms is *decentralized* and performed with the packets that the devices have to send. For computing the exploration policy on each player, the probabilities of sending a packet have to be shared at the beginning of the exploration phase. In order to reduce the exploration time needed to find an $\epsilon$-approximation of each arm, each player is responsible of a predefined number of samples $t_n^*$ for each arm according to its probability of sending a packet, so that all players would finish their estimations almost at the same time. At the end of the exploration phase, each player sends its $\epsilon$-approximation of each arm to other players through the gateway. Then, the target policy can be computed in a centralized way (by the gateway) or separately within each player. Our proposed algorithm is presented in details in the following.

## 4.6 Collaborative Exploration in Multi-Player Bandits

The basic idea of our proposed algorithm is *collaboration*: the players (nodes in a network) divide the exploration mission between them, then they share the results at the end by sending messages. As stated in section 2.2, a message can be sent with a packet at the same time slot. It contains information to be sent to other players through the gateway. We recall that in order to estimate the target policy, the players need to know two things: the probabilities of sending packets of all players and the estimations of the mean rewards of the arms. In the sake of simplifying notations, let's assume that $p_1 > p_2, ..., p_{N-1} > p_N$ in the following.

### 4.6.1 Description of the algorithm

Algorithm 8 presents our *collaborative exploration* algorithm. The sampling strategy used is the *Uniform Policy* (line 4) $\tilde{\pi}$: $\forall n, \forall k, \pi_n^k = \frac{1}{K}$. Then, player $n$ can estimate the mean rewards of the arms using:

$$\hat{\theta}_n^k = \frac{\hat{\mu}_n^k(\tilde{\pi})}{\rho_n^k(\tilde{\pi})}, \text{ where} \tag{4.17}$$

$$\rho_n^k(\tilde{\pi}) = \prod_{n'=1, n' \neq n}^{N} (1 - p_{n'} \pi_{n'}^k) = \prod_{n'=1, n' \neq n}^{N} (1 - p_{n'}/K) \tag{4.18}$$

**Algorithm 8** Collaborative Exploration in Multi-Player Multi-Armed Bandits

**Inputs:** $[K]$, $[N]$, $\epsilon \in [0,1]$, $\delta \in (0,1)$

**Output:** $\hat{\theta} = \{\hat{\theta}^k, \forall k \in [K]\}$

**Init:** $t := 0$; $\forall n \in [N] : t_n^* := \infty$, $ack1_n := 0$; $\forall (n,k) \in [N] \times [K]$: $ack2_n^k := 0, ack3_n^k := 0$

**repeat**
  $\mathcal{N}_t := \{n \in [N], a_n \sim \mathcal{B}(p_n), a_n = 1\}$
  **for** $n \in \mathcal{N}_t$ **do**
    $k_n \sim \mathcal{U}(1, K)$
    $Y_n^{k_n}(t_n^{k_n}) := I_n^{k_n} E^{k_n}$
    $\hat{\mu}_n^{k_n}(\tilde{\pi}) := \sum_{t=1}^{t=t_n^{k_n}} Y_n^{k_n}(t)/t_n^{k_n}$
    $t_n^{k_n} := t_n^{k_n} + 1$
    **if** $ack1_n = 0$ **then**
      $ack1_n := send(p_n)$
    **else**
      **if** $\forall i \in [N], ack1_i = 1$ **then**
        $\forall i, t_i^* := \left\lceil \dfrac{p_n \log(2K/\delta)}{2\epsilon^2 (\prod_{n' \neq 1}(1 - p_{n'}/K))^2 \sum_{i=1}^N p_i} \right\rceil$
      **end if**
      **if** $\exists k, t_n^k \geq t_n^*$ **then**
        **if** $ack2_n^k = 0$ **then**
          $ack2_n^k := \text{send}(\hat{\theta}_n^k)$
        **else if** $ack3_n^k = 0$ **then**
          $ack3_n^k := \text{send}(t_n^k)$
        **end if**
      **end if**
    **end if**
  **end for**
  $t = t + 1$
**until** $\exists \mathcal{N}' \subset \mathcal{N}, \begin{cases} \forall k \sum\limits_{n \in \mathcal{N}'} t_n^k \geq \sum\limits_{n \in \mathcal{N}} t_n^* \\ \forall k \sum\limits_{n \in \mathcal{N}'} ack2_n^k = |\mathcal{N}'| \end{cases}$
all players calculate $\hat{\theta}^k := \dfrac{\sum_{n \in \mathcal{N}'} \hat{\theta}_n^k t_n^k}{\sum_{n \in \mathcal{N}'} t_n^k}$

We can notice that in order to correctly estimate the means of the arms, each player needs to compute the value of $\rho_n(\tilde{\pi})$ which depends on the probabilities of sending packets of all players in the network (Equation (4.18)), so the exploration phase starts by sharing the probabilities of sending packets between the players.

**Algorithm 8.** Every player $n$ sends its probability to the gateway that forwards it to all other players (line 9). The function send($s$) means that message $s$ is broadcast to other players through the gateway on a channel chosen uniformly over $K$. The function send($s$) returns 1 if an acknowledgement is received from the gateway and 0 otherwise. When player $n$ receives the probabilities of all other players (i.e. all players successfully send their probabilities), it computes the required number of samples of each arm $t_n^*$ (lines 11,12) according to Lemma 4.3. When player $n$ samples an arm $k$ at least $t_n^*$ times, it sends its estimation $\hat{\theta}_n^k$ and $t_n^k$ to other players (lines 14-18) each in a distinct message (distinct time slots). $\hat{\theta}_n^k$ is computed according to equation (4.17). The exploration phase ends when the arms have been sampled enough by a subset of players (line 24) and the estimations of this subset have been successfully sent. Finally, the players compute the global estimations of the arms by combining the received local ones (line 25).

**Lemma 4.3**

With Algorithm 8, to obtain with a probability $1-\delta$ an $\epsilon$-approximation of the mean rewards of arms, every player $n$ needs to sample each arm at least

$$t_n^* = \left\lceil \frac{p_n \log(2K/\delta)}{2\epsilon^2 (\prod_{n' \neq 1}(1 - p_{n'}/K))^2 \sum_{i=1}^{N} p_i} \right\rceil \text{ times.}$$

where the player of the greatest probability to send packets is indexed by 1.

*Proof.* Due to equations 2.9 and 4.17, for a given probability of failure $\delta \in [0,1]$, and a given approximation factor $\epsilon$, $\forall n \in [N]$, $\forall k \in [K]$ we have:

$$P(|\mu^k - \hat{\mu}_n^k| \geq \epsilon) \leq \frac{\delta}{K} \iff P(|\theta^k - \hat{\theta}_n^k| \geq \epsilon'_n) \leq \frac{\delta}{K}, \tag{4.19}$$

where $\epsilon'_n = \epsilon . \prod_{n' \neq n}(1 - p_{n'}/K)$.
Applying Hoeffding's inequality:

$$P(|\theta_n^k - \hat{\theta}_n^k| \geq \epsilon'_n) \leq 2e^{-2t_n^k {\epsilon'_n}^2}. \tag{4.20}$$

Therefore for obtaining an $\epsilon$-approximation of arm $k$ on player $n$ with a probability

$1-\frac{\delta}{K}$:

$$t_n^k \geq \frac{\log(2K/\delta)}{2\epsilon_n'^2} \iff t_n^k \geq \frac{\log(2K/\delta)}{2\epsilon^2(\prod_{n'\neq n}(1-p_{n'}/K))^2}$$

For the sake of simplifying notations, we assume that $p_1 > p_2, ..., p_{N-1} > p_N$, then We have:

$$\frac{\log(2K/\delta)}{2\epsilon^2(\prod_{n'\neq n}(1-p_{n'}/K))^2} \leq \frac{\log(2K/\delta)}{2\epsilon^2(\prod_{n'\neq 1}(1-p_{n'}/K))^2} = t^\dagger$$

Now, as Algorithm 8 shares the estimations of the $N$ players for finding $\epsilon$-approximation of arm $k$ with high probability, we need $\sum_{n=1}^N t_n^* = t^\dagger$ samples. Hence, if each player samples arm $k$ at least $t_n^* = \left\lceil \frac{p_n \log(2K/\delta)}{2\epsilon^2(\prod_{n'\neq 1}(1-p_{n'}/K))^2 \sum_{i=1}^N p_i} \right\rceil$ times, an $\epsilon$-approximation of arm $\theta^k$ is obtained with a probability $1-\frac{\delta}{K}$.

□

We analyze the performance of our exploration algorithm in the following.

### 4.6.2 Analysis of the algorithm

Due to internal and external collisions, messages sent by the players might be lost. In this case, the corresponding player does not receive an acknowledgement and hence it keeps sending its message until it observes a success. In Lemma 4.4, we provide an upper bound on the number of transmissions needed to successfully send $m$ messages.

**Lemma 4.4**

In Algorithm 8, so that player $n$ sends successfully $m$ messages, with a probability $1-\delta$ player $n$ needs to issue a number of transmissions $C(m)$, which is at most:

$$m\left\lceil \frac{\log(m/\delta)}{\log\left(1-\sum_{k=1}^K \frac{(1-p_1/K)^{N-1}}{K}\theta^k\right)^{-1}} + 1 \right\rceil \text{ transmissions.}$$

*Proof.* Let $C(1)$ be the random variable corresponding to the number of transmissions of player $n$ to successfully send one message. $C(1)$ follows a geometric distribution with a probability of success $p = \mu_n(\tilde{\pi}) = \sum_{k=1}^{K} \frac{\rho_n(\tilde{\pi})}{K}\theta^k$, and probability of failure $q = 1 - p$. Let $F$ be the number of failures before the success. We have:

$$\mathbb{P}(C(1) \leq F + 1) = 1 - q^F = 1 - \delta,$$
$$\implies F = \left\lceil \frac{\log \delta}{\log q} \right\rceil$$

Assuming that $p_1 \geq p_2, .., p_{N-1} \geq p_N$, we get $\rho_n(\tilde{\pi}) = \prod_{n' \neq n}(1 - p_{n'}/K) \leq (1 - p_1/K)^{N-1}$. Consequently, for sending $m$ messages, with a probability $1 - \delta$ any player needs at most :

$$C(m) \leq m \left\lceil \frac{\log \delta/m}{\log(1 - \sum_{k=1}^{K} \frac{(1-p_1/K)^{N-1}}{K}\theta^k)} + 1 \right\rceil \text{ transmissions.}$$

□

**Communication Cost.** The communication cost presents the number of transmissions needed to successfully send the messages of Algorithm 8. Theorem 4.4 states an upper bound on the total number of transmissions issued by the $N$ players for sharing the probabilities of sending packets, and for sharing their estimations that is in the order of $O\left(NK \log \frac{NK+N}{\delta}\right)$.

**Theorem 4.4**

When Algorithm 8 stops, the total number of transmissions issued by the players is, with probability $1 - \delta$, less than $C(N(1 + 2K))$, where

$$C(m) = m \left\lceil \frac{\log m/\delta}{\log\left(1 - \sum_{k=1}^{K} \frac{(1-p_1/K)^{N-1}}{K}\theta^k\right)^{-1}} + 1 \right\rceil.$$

*Proof.* The required number of messages to send during Algorithm 8 is at most $N(1+2K)$. Using Lemma 4.4, the total number of transmissions done by all players to send successfully their messages is with probability $1-\delta$:

$$C(N(1+2K)) \leq N(1+2K)\left\lceil \frac{\log \delta/(N(1+2K))}{\log(1-\sum_{k=1}^{K}\frac{(1-p_1/K)^{N-1}}{K}\theta^k)} + 1 \right\rceil \tag{4.21}$$

□

**Exploration Duration.** Theorem 4.5 states an upper bound on the number of time slots needed by all players to finish their estimations of the mean rewards of the arms and to share them. The left term in $O\left(K/\epsilon^2 \log K/\delta\right)$ is the dominating term of the upper bound of the sample complexity. It is near optimal in comparison to the lower bound of $K$ biased coin estimations in $\Omega\left(K/\epsilon^2 \log 1/\delta\right)$ [80]. The right term of the upper bound in $O\left(K/p_N\sqrt{\log NK/\delta}\right)$ mainly depends on the least frequent player. This is due to the fact that, in the worst case, before stopping Algorithm 8 has to wait until the least frequent player has sent its estimations of the arms.

**Theorem 4.5**

With a probability at least $1-\delta$, Algorithm 8 stops while finding the $\epsilon$-approximations of model $\boldsymbol{\theta} = \{\theta^1, ..., \theta^K\}$ at:

$$t^* \leq \frac{K \log 2K/\delta}{2\epsilon^2(1-p_1/K)^{2N-2}\sum_{i=1}^{N} p_i} + \frac{K}{p_N}\left(\sqrt{\frac{1}{2}\log\frac{NK}{\delta}} + C(3)\right),$$

where $p_N = \min_{n\in[N]} p_n$, $p_1 = \max_{n\in[N]} p_n$, and $C(3)$ is the needed number of sent messages to successfully send 3 messages.

*Proof.* A player $n$ stops while finding its estimations when it plays each arm $k$ at least $t_n^*$ times (Lemma 4.3). Let $t_n^k$ be the number of plays of arm $k$ by player $n$ before the algorithm stops at time $t^*$. $t_n^k$ is a binomial random variable with parameters $t^*$ and $p_n/K$. Then we have:

$$\mathbb{E}[t_n^k] = \frac{p_n}{K}.t^* \tag{4.22}$$

The estimation does not terminate if this event occurs: $E = \{\exists n \in [N], \exists k \in [K], t_n^k < t_n^* + C(3)\}$.

Applying Hoeffding's inequality we get:

$$\mathcal{P}(t_n^k - \frac{p_n}{K}.t^* \leq -\epsilon) \leq \exp^{-2\epsilon^2} = \frac{\delta}{NK} \tag{4.23}$$

Hence, when $E$ does not occur $\implies$ we have with probability at most $\delta$:

$$\begin{aligned}
&\forall n \quad t_n^* + C(3) - \frac{p_n}{K}.t^* \leq -\sqrt{\frac{1}{2}\log\frac{NK}{\delta}} \\
\implies &\forall n \quad t^* \geq \left(\sqrt{\frac{1}{2}\log\frac{NK}{\delta}} + C(3) + p_n \frac{\log 2K/\delta}{2\epsilon^2(\prod_{n'\neq 1}(1-p_{n'}/K))^2 \sum_{i=1}^N p_i}\right)\frac{K}{p_n} \\
\implies &\forall n \quad t^* \geq \frac{K}{p_n}\left(\sqrt{\frac{1}{2}\log\frac{NK}{\delta}} + C(3)\right) + K\frac{\log 2K/\delta}{2\epsilon^2(\prod_{n'\neq 1}(1-p_{n'}/K))^2 \sum_{i=1}^N p_i} \\
\implies &t^* \geq \frac{K}{p_N}\left(\sqrt{\frac{1}{2}\log\frac{NK}{\delta}} + C(3)\right) + K\frac{\log 2K/\delta}{2\epsilon^2(1-p_1/K)^{2N-2} \sum_{i=1}^N p_i},
\end{aligned}$$

Then, when $E$ does not occur and hence the estimation terminates, we have with probability at least $1-\delta$:

$$t^* < \frac{K}{p_N}\left(\sqrt{\frac{1}{2}\log\frac{NK}{\delta}} + C(3)\right) + K\frac{\log 2K/\delta}{2\epsilon^2(1-p_1/K)^{2N-2} \sum_{i=1}^N p_i},$$

where $p_N$ and $p_1$ are respectively the lowest and the greatest probability of sending a packet among the players.

□

**Regret Analysis.** In this section, we provide upper and lower bounds on the pseudo-regret. The pseudo-regret is defined as follows:

**Definition 4.4: Pseudo-regret**

Let $\pi_t$ be a policy generated at time $t$ by an algorithm, and $\mu_{\boldsymbol{\theta}}(\pi_t)$ be its value in model $\boldsymbol{\theta} = \{\theta^1, ..., \theta^K\}$, we define the pseudo-regret with respect to an optimal policy $\pi^*_{\boldsymbol{\theta}}$ as $R(T) = \sum_{t=1}^{T}(\mu_{\boldsymbol{\theta}}(\pi^*_{\boldsymbol{\theta}}) - \mu_{\boldsymbol{\theta}}(\pi_t))$.

For the regret study of our exploration algorithm we make use of Lemma 4.5.

**Lemma 4.5**

The expected instantaneous regret in the model $\boldsymbol{\theta}$ of the target policy $\pi^*_{\hat{\boldsymbol{\theta}}}$ using the estimated model $\hat{\boldsymbol{\theta}}$ with respect to the optimal policy $\pi^*_{\boldsymbol{\theta}}$ using the true model $\boldsymbol{\theta}$ is upper bounded by:

$$\mu_{\boldsymbol{\theta}}(\pi^*_{\boldsymbol{\theta}}) - \mu_{\boldsymbol{\theta}}(\pi^*_{\hat{\boldsymbol{\theta}}}) \leq 2K\epsilon, \tag{4.24}$$

where $\mu_{\boldsymbol{\theta}}(\pi)$ denotes the mean reward of the policy $\pi$ in the model $\boldsymbol{\theta}$.

*Proof.*

$$\mu_{\boldsymbol{\theta}}(\pi^*_{\boldsymbol{\theta}}) - \mu_{\boldsymbol{\theta}}(\pi^*_{\hat{\boldsymbol{\theta}}}) = \mu_{\boldsymbol{\theta}}(\pi^*_{\theta}) - \mu_{\hat{\boldsymbol{\theta}}}(\pi^*_{\theta}) + \mu_{\hat{\boldsymbol{\theta}}}(\pi^*_{\theta}) - \mu_{\hat{\boldsymbol{\theta}}}(\pi^*_{\hat{\boldsymbol{\theta}}}) + \mu_{\hat{\boldsymbol{\theta}}}(\pi^*_{\hat{\boldsymbol{\theta}}}) - \mu_{\boldsymbol{\theta}}(\pi^*_{\hat{\boldsymbol{\theta}}}) \tag{4.25}$$

Then, we have:

- $\mu_{\boldsymbol{\theta}}(\pi^*_{\theta}) - \mu_{\hat{\boldsymbol{\theta}}}(\pi^*_{\theta}) = \sum_{k=1}^{K} z^k l^k \theta^k - \sum_{k=1}^{K} z^k l^k \hat{\theta}^k \leq K\epsilon$,
- $\mu_{\hat{\boldsymbol{\theta}}}(\pi^*_{\boldsymbol{\theta}}) - \mu_{\hat{\boldsymbol{\theta}}}(\pi^*_{\hat{\boldsymbol{\theta}}}) \leq 0$, since $\pi^*_{\hat{\boldsymbol{\theta}}}$ is the best policy in the model $\hat{\boldsymbol{\theta}}$.
- $\mu_{\hat{\boldsymbol{\theta}}}(\pi^*_{\hat{\boldsymbol{\theta}}}) - \mu_{\boldsymbol{\theta}}(\pi^*_{\hat{\boldsymbol{\theta}}}) = \sum_{k=1}^{K} \hat{z}^k \hat{l}^k \hat{\theta}^k - \sum_{k=1}^{K} \hat{z}^k \hat{l}^k \theta^k \leq K\epsilon$.

□

Theorem 4.6 states that in the setting proposed by [60], the regret of Algorithm 8 followed by `DORG` is in $O\left(T^{2/3}\left((\log KT)/(1-p/K)^N N\right)\right)$.

**Theorem 4.6**

When $\delta = 1/T, \epsilon = K/\sqrt[3]{T}$, $\forall n \in [N], p_n = p$, the pseudo-regret with respect to the optimal policy $\pi^*_{\boldsymbol{\theta}}$ of Algorithm 8 followed by the policy $\pi^*_{\hat{\boldsymbol{\theta}}}$ is upper bounded by:

$$R(T) \leq T^{2/3}\left(2K^2 + \frac{\log 2KT}{2(1-p/K)^{2N-2}Np}\right) + \frac{K^2}{p}\left(\sqrt{\frac{1}{2}\log NKT} + C(3)\right) + K$$

*Proof.* Let $T$ be the time horizon, $\tilde{\pi}$ be the uniform policy used in Algorithm 8, which outputs an $\epsilon$-approximation with high probability of $\boldsymbol{\theta}$, and $\pi^*_{\boldsymbol{\theta}}$ be the optimal policy. Let $t^*$ be the stopping time of the exploration phase. Then, the pseudo-regret with respect to a target policy $\pi^*_{\boldsymbol{\theta}}$ of Algorithm 8 is expressed as:

$$E[R(T)] = t^*(\mu_{\boldsymbol{\theta}}(\pi^*_{\boldsymbol{\theta}}) - \mu_{\boldsymbol{\theta}}(\pi_u)) + (T - t^*)(\mu_{\boldsymbol{\theta}}(\pi^*_{\boldsymbol{\theta}}) - \mu_{\boldsymbol{\theta}}(\pi^*_{\hat{\boldsymbol{\theta}}})), \tag{4.26}$$

where $\mu_{\boldsymbol{\theta}}(\pi^*_{\hat{\boldsymbol{\theta}}})$ denotes the mean reward in the model $\boldsymbol{\theta}$ of the optimal policy using the estimated model $\hat{\boldsymbol{\theta}}$. The left term of equation 4.26 is the instantaneous pseudo-regret of the exploration policy $\pi_{\mathcal{U}}$, and the right term is the instantaneous pseudo-regret of the estimated optimal policy $\pi^*_{\hat{\boldsymbol{\theta}}}$.

Theorem 4.5 allows us to upper-bound the stopping time of Algorithm 8 with $t^*$ on an event of high probability $1 - \delta$:

$$t^* \leq \frac{K}{p_N}\left(\sqrt{\frac{1}{2}\log\frac{NK}{\delta}} + C(3)\right) + K\frac{\log 2K/\delta}{2\epsilon^2(1-p_1/K)^{2N-2}\sum_{i=1}^N p_i} \tag{4.27}$$

When $\forall n \in [N], p_n = p$, we have:

$$t^* \leq \frac{K}{p}\left(\sqrt{\frac{1}{2}\log\frac{NK}{\delta}} + C(3)\right) + K\frac{\log 2K/\delta}{2\epsilon^2(1-p/K)^{2N-2}Np} \tag{4.28}$$

The instantaneous pseudo-regret of uniform policy with respect to the optimal

policy $\pi^*_{\boldsymbol{\theta}}$ is upper bounded by:

$$\mu_\theta(\pi^*_{\boldsymbol{\theta}}) - \mu_{\boldsymbol{\theta}}(\tilde{\pi}) \leq K$$

and on the other hand we know by Lemma 4.5 that:

$$\mu_{\boldsymbol{\theta}}(\pi^*_{\boldsymbol{\theta}}) - \mu_{\boldsymbol{\theta}}(\pi^*_{\hat{\boldsymbol{\theta}}}) \leq 2K\epsilon \tag{4.29}$$

Then the pseudo-regret is controlled by the trivial upper bound $KT$ on the complementary event of probability less than $\delta$:

$$E[R(T)] \leq t^*(\mu_\theta(\pi^*_{\boldsymbol{\theta}}) - \mu_{\boldsymbol{\theta}}(\tilde{\pi})) + (T - t^*)(\mu_{\boldsymbol{\theta}}(\pi^*_{\boldsymbol{\theta}}) - \mu_{\boldsymbol{\theta}}(\pi^*_{\hat{\boldsymbol{\theta}}})) + \delta KT \tag{4.30}$$

Then, by setting $\delta = 1/T$, the pseudo-regret of Algorithm 8 followed by a policy $\pi^*_{\hat{\boldsymbol{\theta}}}$ is:

$$\begin{aligned}
E[R(T)] &\leq Kt^* + (T - t^*) \times 2K\epsilon + K && (4.31)\\
&\leq Kt^* + 2K\epsilon T + K && (4.32)\\
&\leq \frac{K^2}{p}\left(\sqrt{\frac{1}{2}\log NKT} + C(3)\right) + \frac{K^2 \log 2KT}{2\epsilon^2(1 - p/K)^{2N-2}Np} + 2K\epsilon T + K && (4.33)\\
& && (4.34)
\end{aligned}$$

Finally, by setting $\epsilon = K/\sqrt[3]{T}$, we conclude the proof:

$$E[R(T)] \leq T^{2/3}\left(2K^2 + \frac{\log 2KT}{2(1 - p/K)^{2N-2}Np}\right) + \frac{K^2}{p}\left(\sqrt{\frac{1}{2}\log NKT} + C(3)\right) + K \tag{4.35}$$

□

To show how tight this upper bound is we provide in Theorem 4.7 a lower bound on

the pseudo-regret of any exploration algorithm that outputs an $\epsilon-$approximation of each arm $\theta^k$ and is followed by an optimal policy that uses the estimated model.

**Theorem 4.7**

When $\epsilon = K/\sqrt[3]{T}$ , there exists a model $\boldsymbol{\theta} = \{\theta^1, ..., \theta^k\}$ and a distribution of players $p_1, ..., p_N$ such that the pseudo-regret with respect to the deterministic optimal policy $\pi^*_{\boldsymbol{\theta}}$ of any exploration algorithm that, with probability at least $1 - 1/T$, outputs an $\epsilon$-approximation of every arm $\theta^k$ and which is followed by the optimal policy using the estimated model is at least:

$$R(T) \geq \Omega\left(T^{2/3}\frac{\log T}{N}\right).$$

*Proof.* In the following we show that a lower bound holds for a class of models $\boldsymbol{\theta}$ and distribution of players $p_1, ..., p_N$. Without loss of generality, we assume in the following that:

- $\theta^1 \geq \theta^2, ..., \theta^{K-1} \geq \theta^K$,
- $p_1 \geq p_2, ..., p_{N-1} \geq p_N$.

**Choice of a class of problems.** The most difficult point for evaluating a regret lower bound is that in the general case, the optimal policy, which maximizes the mean reward (see equation (4.4)), is unknown. For handling this point we choose a particular class of problems, where $N = K+1$. Then, we assume that the distribution of players and the mean rewards of arms are such that:

$$\begin{cases} \forall k \in [K-1] \quad \theta^k = \theta^{k+1} + \epsilon, \\ p_1 > p_2 = ... = p_K > p_{K+1}, \\ p_1(1-p_{K+1}) + p_{K+1}(1-p_1) = p_2, \\ p_2(1-p_{K+1}) + p_{K+1}(1-p_2) > p_2, \\ \forall k \in [K] \quad \frac{\epsilon}{2p_k} < \theta^k. \end{cases} \tag{4.36}$$

**The optimal policy.** When $\frac{\epsilon}{2p_k} < \theta^k$ (equation (4.36)), superposing players on any arm provides less reward than spreading players on the arms. Indeed, let $\Delta_s$ be the gap between the mean reward of two players $k_1, k_2, k_1 < k_2 \leq K$ assigned on different arms, and the mean reward of two players assigned on the same arm:

$$\begin{aligned}
\Delta_s &= p_{k_1}\theta^{k_1} + p_{k_2}\theta^{k_2} - p_{k_1}\theta^{k_1}(1-p_{k_2}) - p_{k_2}\theta^{k_1}(1-p_{k_1}), && (4.37)\\
&= p_{k_2}(\theta^{k_2} - \theta^{k_1}) + 2p_{k_1}p_{k_2}\theta^{k_1}, && (4.38)\\
&= -p_{k_2}\epsilon + 2p_{k_1}p_{k_2}\theta^{k_1} > 0. && (4.39)
\end{aligned}$$

Let $\Delta_{1,2}$ be the difference between the mean reward of policy that assigns player $K+1$ on arm 1 and the one that assigns it on arm 2.

$$\begin{aligned}
\Delta_{1,2} &= (p_1(1-p_{K+1}) + p_{K+1}(1-p_1))\theta^1 + p_2\theta^2 && (4.40)\\
&\quad - p_1\theta^1 - (p_2(1-p_{K+1}) + p_{K+1}(1-p_2))\theta^2 && (4.41)\\
&= p_2\theta^1 - p_1\theta^1 + p_2\theta^2 - (p_2(1-p_{K+1}) + p_{K+1}(1-p_2))\theta^2 < 0 && (4.42)
\end{aligned}$$

Now let $\Delta_{2,k}$ be the difference between the mean reward of policy that assigns player $K+1$ on arm 2 and the one that assigns it on arm $k > 2$.

$$\begin{aligned}
\Delta_{2,k} &= (p_2(1-p_{K+1}) + p_{K+1}(1-p_2))\theta^2 + p_2\theta^k && (4.43)\\
&\quad - p_2\theta^2 - (p_2(1-p_{K+1}) + p_{K+1}(1-p_2))\theta^k && (4.44)\\
&= (p_2(1-p_{K+1}) + p_{K+1}(1-p_2))(\theta^2 - \theta^k) - p_2(\theta^2 - \theta^k) > 0 && (4.45)
\end{aligned}$$

Hence, when equation (4.36) holds, the optimal assignment of players over arms is:

$$\pi^*_{\boldsymbol{\theta}} = (p_1, \theta^1), (p_2, p_{K+1}, \theta^2), ..., (p_{K-1}, \theta^{K-1}), (p_K, \theta^K). \quad (4.46)$$

**The optimal exploration policy.** As an $\epsilon$-approximation of each arm is needed to compute the optimal policy. The optimal exploration policy plays each arm the same expected (with respect to the distribution of players $\boldsymbol{p}$) number of times. When

equation (4.36) holds, any optimal exploration policy belongs to the following set:

$$\pi_E^* \in \{m \in [K], \forall n \in [K] \setminus \{1\}, k \in [K] \setminus \{m\} : (p_n, \theta^k), (p_1, p_{K+1}, \theta^m)\}. \tag{4.47}$$

Hence any other assignment of players over arms generates more collisions.

**Pseudo-regret decomposition.** Let $T$ be the time horizon. Let $\pi_E^*$ be the optimal (in term of sample complexity) exploration policy that outputs an $\epsilon$-approximation with high probability of $\boldsymbol{\theta}$, i.e. each arm $\theta^k$, and $\pi_{\boldsymbol{\theta}}^*$ be the optimal policy. We consider the time $t^*$, where the optimal exploration algorithm $\pi_E^*$ outputs exactly an $\epsilon$-approximation of model $\boldsymbol{\theta}$. Then, the pseudo-regret with respect to the deterministic policy $\pi_{\boldsymbol{\theta}}^*$ is expressed as:

$$R(T) = t^*(\mu_{\boldsymbol{\theta}}(\pi_{\boldsymbol{\theta}}^*) - \mu_{\boldsymbol{\theta}}(\pi_E^*)) + (T - t^*)(\mu_{\boldsymbol{\theta}}(\pi_{\boldsymbol{\theta}}^*) - \mu_{\boldsymbol{\theta}}(\pi_{\hat{\boldsymbol{\theta}}}^*)), \tag{4.48}$$

where $\mu_{\boldsymbol{\theta}}(\pi_{\hat{\boldsymbol{\theta}}}^*)$ denotes the mean reward in the model $\boldsymbol{\theta}$ of the optimal policy using the estimated model $\hat{\boldsymbol{\theta}}$.

**Lower bound of the right term.** The right term equation (4.48) is the instantaneous regret of the estimated optimal policy $\pi_{\hat{\boldsymbol{\theta}}}^*$. For stating a lower bound on this term, we lower bound it by the minimal gap between the optimal policy and the estimated optimal policy when a mistake in the ranking of two arms is done:

$$\mu_{\boldsymbol{\theta}}(\pi_{\boldsymbol{\theta}}^*) - \mu_{\boldsymbol{\theta}}(\pi_{\hat{\boldsymbol{\theta}}}^*) \geq \min_{k \in [K], \hat{\theta}^{k+1} > \hat{\theta}^k} \left(\mu_{\boldsymbol{\theta}}(\pi_{\boldsymbol{\theta}}^*) - \mu_{\boldsymbol{\theta}}(\pi_{\hat{\boldsymbol{\theta}}}^*)\right), \tag{4.49}$$

The minimal gap, between the mean reward of the optimal policy (see equation (4.46)) and a policy where an arm is not well ranked, is obtained when the ranks of arms 2 and 3 are inverted.

$$\begin{aligned}\mu_{\boldsymbol{\theta}}(\pi^*_{\boldsymbol{\theta}}) - \mu_{\boldsymbol{\theta}}(\pi^*_{\hat{\boldsymbol{\theta}}}) &\geq (p_2(1-p_{K+1}) + p_{K+1}(1-p2))\theta^2 + p_2\theta^3 \\ &\quad - p_2\theta^2 - (p_2(1-p_{K+1} + p_{K+1}(1-p2))\theta^3 \\ &\geq c_{\boldsymbol{p}}\epsilon, \text{ where } c_{\boldsymbol{p}} > 0. \end{aligned} \tag{4.50}$$

**Lower bound of the left term.** The left term of equation (4.48) is the instantaneous regret of the optimal exploration policy $\pi^*_E$. The optimal exploration policy cannot be the optimal policy since estimating $\epsilon$-approximations of arms necessitates to play the same expected number of times the arms, and hence assigning $p_1$ and $p_{K+1}$ on the same arm, which is not optimal. There are three possibilities:

- $p_1$ and $p_{K+1}$ are on arm 1:

$$\begin{aligned}\mu_{\boldsymbol{\theta}}(\pi^*_{\boldsymbol{\theta}}) - \mu_{\boldsymbol{\theta}}(\pi^*_E) \geq \quad & p_1\theta^1 + (p_2(1-p_{K+1}) + p_{K+1}(1-p_2))\theta^2 \\ & - (p_1(1-p_{K+1}) + p_{K+1}(1-p_1))\theta^1 - p_2\theta^2,\end{aligned}$$

- $p_1$ and $p_{K+1}$ are on arm $m \in [K] \setminus \{1,2\}$:

$$\begin{aligned}\mu_{\boldsymbol{\theta}}(\pi^*_{\boldsymbol{\theta}}) - \mu_{\boldsymbol{\theta}}(\pi^*_E) \geq \quad & p_1\theta^1 + p_m\theta^m + (p_2(1-p_{K+1}) + p_{K+1}(1-p_2))\theta^2 \\ & - p_2\theta^1 - (p_1(1-p_{K+1}) + p_{K+1}(1-p_1))\theta^m - p_2\theta^2,\end{aligned}$$

- $p_1$ and $p_{K+1}$ are on arm 2:

$$\begin{aligned}\mu_{\boldsymbol{\theta}}(\pi^*_{\boldsymbol{\theta}}) - \mu_{\boldsymbol{\theta}}(\pi^*_E) \geq \quad & p_1\theta^1 + (p_2(1-p_{K+1}) + p_{K+1}(1-p_2))\theta^2 \\ & - p_2\theta^1 - (p_1(1-p_{K+1}) + p_{K+1}(1-p_1))\theta^2.\end{aligned}$$

Hence we have:

$$\mu_{\boldsymbol{\theta}}(\pi^*_{\boldsymbol{\theta}}) - \mu_{\boldsymbol{\theta}}(\pi^*_E) \geq c_{\boldsymbol{\theta},\boldsymbol{p}}, \tag{4.51}$$

where $c_{\boldsymbol{\theta},\boldsymbol{p}} > 0$ is a constant depending on the problem parameters $\boldsymbol{\theta}$ and $p_1, ..., p_N$.

**Lower bound of the regret.** Now, injecting the lower bound of $\mu_{\boldsymbol{\theta}}(\pi^*_{\boldsymbol{\theta}}) - \mu_{\boldsymbol{\theta}}(\pi^*_E)$ (equation (4.51)) and the lower bound of $\mu_{\boldsymbol{\theta}}(\pi^*_{\boldsymbol{\theta}}) - \mu_{\boldsymbol{\theta}}(\pi^*_{\hat{\boldsymbol{\theta}}})$ (equation (4.50)) in the pseudo-regret decomposition (equation (4.48)), we obtain:

$$R(T) \geq t^* c_{\boldsymbol{\theta},\boldsymbol{p}} + (T - t^*) c_{\boldsymbol{p}} \epsilon, \tag{4.52}$$

$$\geq t^* c_{\boldsymbol{\theta},\boldsymbol{p}} + T\epsilon\Delta_{\boldsymbol{p}} - t^* c_{\boldsymbol{p}} \epsilon. \tag{4.53}$$

The lower bound of number of samples for finding a bias $\epsilon$ of a coin is $\Omega\left(1/\epsilon^2 \log 1/\delta\right)$ [80]. At each time step, a maximum of $N$ players are sampled. Hence, the time $t^*$ where $\pi^*_E$ finds exactly an $\epsilon$-approximation of each arm $\theta^k$ is at least:

$$\Omega\left(\frac{K}{N\epsilon^2} \log \frac{1}{\delta}\right) \Leftrightarrow \exists c_1 > 0, t^* = c_1 \frac{K}{N\epsilon^2} \log \frac{1}{\delta}. \tag{4.54}$$

We have:

$$R(T) \geq c_1 c_{\boldsymbol{\theta},\boldsymbol{p}} \frac{K}{N\epsilon^2} \log \frac{1}{\delta} + T c_{\boldsymbol{p}} \epsilon - c_1 c_{\boldsymbol{p}} \epsilon \frac{K}{N\epsilon} \log \frac{1}{\delta}. \tag{4.55}$$

Finally setting $\delta = 1/T$ and $\epsilon = \sqrt{K}/\sqrt[3]{T}$, obtain:

$$E[R(T)] \geq \Omega\left(T^{2/3} \frac{\log T}{N} + T^{2/3} - \frac{K^{1/2}}{N} T^{1/3} \log T\right). \tag{4.56}$$

Hence, we have:

$$E[R(T)] \geq \Omega\left(T^{2/3} \frac{\log T}{N}\right). \tag{4.57}$$

□

Theorem 4.7 reveals the difficulty of the studied problem in comparison to multi-armed bandits. Indeed, in the case of bandit the pseudo-regret lower bound of *explore-then-exploit* algorithms is in $\Omega(\sqrt{KT \log T})$ [81], and in the case of multi-player bandit there

exists an *explore-then-exploit* algorithm with a regret upper bound in $O(K\sqrt{T\log T})$ [39]. The difference in power of $T$ of the pseudo-regret lower bounds of bandits and massive multi-player bandits is due to the fact that in the problem studied, the whole model $\boldsymbol{\theta}$ is needed to compute the optimal policy, and not only the best arm: when the exploration stops, there is no guarantee that the arms are sufficiently sampled to compute the optimal policy without mistakes of assignment of players over arms. The independence of $K$ of the pseudo-regret lower bound of massive multi-player bandits is due to the fact that at each time step $K$ players can sample the arms.

**Fairness Analysis.** As `DOFG` will be preceded by an exploration phase and will be based on the estimations of the arms rather than the real values of the arms' reward means, its fairness level would be affected. We hereby present the new level of fairness achieved by `DOFG`$(\hat{\theta})$ preceded by Algorithm 8.

**Theorem 4.8**

Applying Algorithm 8 followed by `DOFG` ( Algorithm 7) on $\hat{\boldsymbol{\theta}}$ returns an $\alpha$-fair policy in the true model $\boldsymbol{\theta}$, with

$$\alpha \geq 1 - p_1 - \frac{2K\epsilon}{\max_{n\in[N]} \frac{\hat{\theta}^{k_n} z^{k_n}}{1-p_n}}.$$

*Proof.* Theorem 4.3 states that the policy returned by Algorithm 7, denoted as $\pi^\dagger$ has the following fairness guarantees:

$$\hat{\alpha} = \frac{\min_{n\in[N]} \mu_{n,\hat{\boldsymbol{\theta}}}(\pi^\dagger)}{\max_{n\in[N]} \mu_{n,\hat{\boldsymbol{\theta}}}(\pi^\dagger)} \geq 1 - \max_{n\in[N]} p_n, \tag{4.58}$$

with $\mu_{n,\hat{\boldsymbol{\theta}}}(\pi^\dagger)$ denoting the expectation of rewards received by player $n$ in estimated model $\hat{\boldsymbol{\theta}}$ when following policy $\pi^\dagger$. We may write it as follows:

$$\mu_{n,\hat{\boldsymbol{\theta}}}(\pi^\dagger) = \hat{\theta}^{k_n} \prod_{n',\text{ s.t. } k_{n'}=k_n} (1-p_{n'}) = \frac{\hat{\theta}^{k_n} z^{k_n}}{1-p_n}. \tag{4.59}$$

We therefore get:

$$\alpha = \frac{\min_{n\in[N]} \mu_{n,\boldsymbol{\theta}}(\pi^\dagger)}{\max_{n\in[N]} \mu_{n,\boldsymbol{\theta}}(\pi^\dagger)} \tag{4.60}$$

$$= \frac{\min_{n\in[N]} \frac{\theta^{k_n} z^{k_n}}{1-p_n}}{\max_{n\in[N]} \frac{\theta^{k_n} z^{k_n}}{1-p_n}} \tag{4.61}$$

$$\geq \frac{\min_{n\in[N]} \frac{\hat{\theta}^{k_n} z^{k_n}}{1-p_n} - \|\boldsymbol{\theta}-\hat{\boldsymbol{\theta}}\|_\infty}{\max_{n\in[N]} \frac{\hat{\theta}^{k_n} z^{k_n}}{1-p_n} + \|\boldsymbol{\theta}-\hat{\boldsymbol{\theta}}\|_\infty} \qquad \text{since } \frac{z^{k_n}}{1-p_n} \leq 1, \forall n \tag{4.62}$$

$$= \hat{\alpha} - \frac{2\|\boldsymbol{\theta}-\hat{\boldsymbol{\theta}}\|_\infty}{\max_{n\in[N]} \frac{\hat{\theta}^{k_n} z^{k_n}}{1-p_n} + \|\boldsymbol{\theta}-\hat{\boldsymbol{\theta}}\|_\infty} \tag{4.63}$$

$$\geq 1 - \max_{n\in[N]} p_n - \frac{2\|\boldsymbol{\theta}-\hat{\boldsymbol{\theta}}\|_\infty}{\max_{n\in[N]} \frac{\hat{\theta}^{k_n} z^{k_n}}{1-p_n}} \tag{4.64}$$

□

Theorem 4.8 implies that, using $\epsilon$-approximations of arms, with high probability `DOFG` still has the same fairness guarantee minus a term that decreases with $\epsilon$.

## 4.7 Simulation and Results

In order to illustrate and complete the analysis of the aforementioned algorithms, we first compare the performance of *collaborative exploration* (Algorithm 8) with *selfish exploration*, where each player explores selfishly, and with *follow-the-leader exploration (FtL)*, where only the most frequent player explores. Then we compare *collaborative exploration* followed by `DORG`$(\hat{\boldsymbol{\theta}})$ and `DOFG`$(\hat{\boldsymbol{\theta}})$, with *selfish UCB* [60] and *selfish Exp3* [19], which respectively consist in independently playing *UCB* and *Exp3* on each player, and with CBAIMPB, where the players find $(\epsilon', m)-$ optimal arms and exploit them uniformly with $m = 5, \epsilon' = 0.2$. We run simulations with various values of $N$, and $K = 10$, such that $\forall k, \theta^k \sim \mathcal{U}(0,1)$. The distribution of players is uniform and the upper bound of the distribution is chosen such that the internal collision rate does not exceed 0.15 when the number of players reaches 1300 while playing the arms uniformly, so $\forall n, p_n \sim \mathcal{U}(3.10^{-4}, 2.2.10^{-3})$. $\delta = 0.05$, $\epsilon = 0.1$. The curves are averaged over 40 trials and run on $10^6$ time steps.

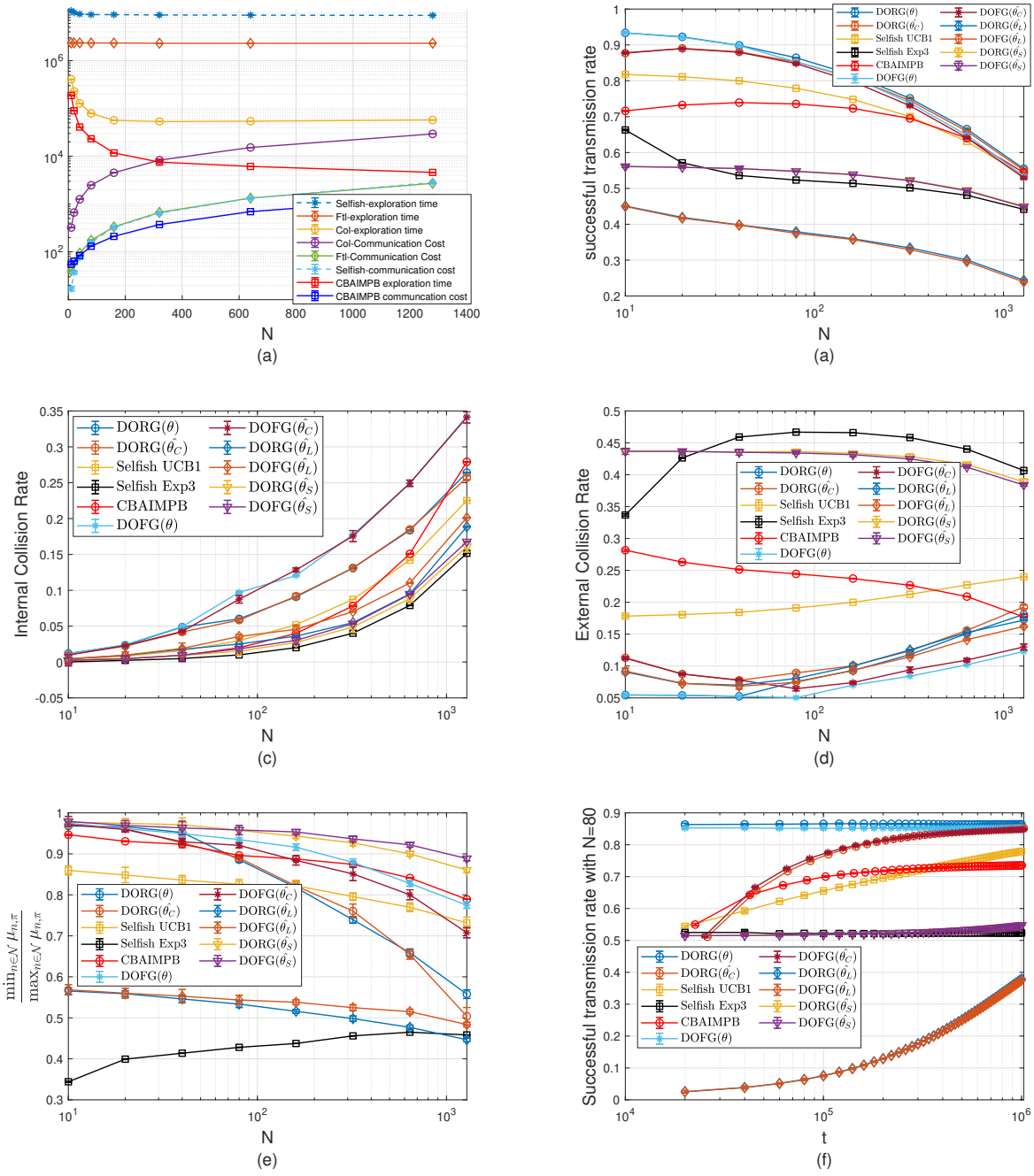



Figure 4.4 – (a) exploration phase, (b) successful transmission rate, (c) internal collision rate, (d) external collision rate, (e) fairness, (f) successful transmission rate versus time. The successful transmission and collision rates are cumulative over time. $\hat{\theta}_{\mathbf{C}}$ when *collaborative exploration* is used, $\hat{\theta}_{\mathbf{S}}$ when *selfish exploration* is used, and $\hat{\theta}_{\mathbf{L}}$ when *follow-the-leader exploration* is used. $\theta$ is the ground truth.

In figure 4.4a, we observe that the exploration time of *collaborative exploration* is two orders of magnitude less than *follow-the-leader exploration* and three orders of magnitude less than *selfish exploration* but one order of magnitude more than *CBAIMPB*, which stops exploration when it finds the best arms. Concerning the communication cost, we observe that the communication cost of the *collaborative exploration* is only one order of magnitude greater than other exploration algorithms, however it is more than two times less than the upper bound stated in Theorem 4.4, which is in the order of $O\left(NK \log \frac{NK+N}{\delta}\right)$. This is due to the fact that the stopping condition of Algorithm 8 does not imply that all players have been sampled enough, but that the arms have been sampled enough. As a consequence, all the estimations of all players do not need to be shared, but only those of players that have finished their estimations.

The performance differences of the exploration policies affect the whole performance of $\texttt{DORG}(\hat{\boldsymbol{\theta}})$ and $\texttt{DOFG}(\hat{\boldsymbol{\theta}})$, which consist of the exploration algorithm followed by the corresponding exploitation phase. That is why in figures 4.4b and 4.4f, the successful transmission rate when using *selfish exploration* and *follow-the-leader exploration* are dramatically less than the one of *collaborative exploration*. In figures 4.4b and 4.4f, $\texttt{DOFG}(\boldsymbol{\theta})$ is slightly outperformed in terms of successful transmission rate by $\texttt{DORG}(\boldsymbol{\theta})$. $\texttt{DORG}(\hat{\boldsymbol{\theta}})$ and $\texttt{DOFG}(\hat{\boldsymbol{\theta}})$ exhibit the same behavior, and we can notice that $\texttt{DORG}(\hat{\boldsymbol{\theta}})$ and $\texttt{DOFG}(\hat{\boldsymbol{\theta}})$ clearly outperform *selfish UCB1*, *selfish Exp3* and *CBAIMPB*, and tend to perform as well as $\texttt{DORG}(\boldsymbol{\theta})$ and $\texttt{DOFG}(\boldsymbol{\theta})$ as $N$ increases (figure 2b). This improvement is due to their low external collision rate (figure 2d) thanks to playing more the best arms, while because of playing more the best arms their internal collision rate is higher (figure 2c). Finally, while *Selfish Exp3* is theoretically better suited for our problem setting, it is clearly outperformed by *Selfish UCB*.

Concerning fairness, $\texttt{DOFG}(\hat{\boldsymbol{\theta}})$ clearly outperforms *selfish UCB1*, *selfish Exp3* and $\texttt{DORG}(\hat{\boldsymbol{\theta}})$, while $\texttt{DORG}(\hat{\boldsymbol{\theta}})$ is outperformed by them when $N$ is high (Figure 4.4e). *CBAIMPB* offers a high fairness between players due to the uniform selection of the arms by all players during both exploration and exploitation phases. The use of *selfish exploration* leads to high fairness level due to its very long uniform exploration phase, in contrast to *follow-the-leader exploration* that suffers of very low fairness level due to the fact that during the exploration time, only the leader can send packets. The observed fairness of $\texttt{DOFG}(\boldsymbol{\theta})$ in figure 4.4e differs from the theoretical one (Theorem 4.3). This is due to the fact that the mean rewards of players are observed on a finite number of time slots ($10^6$). As time passes the observed fairness tends to the theoretical fairness as shown below.

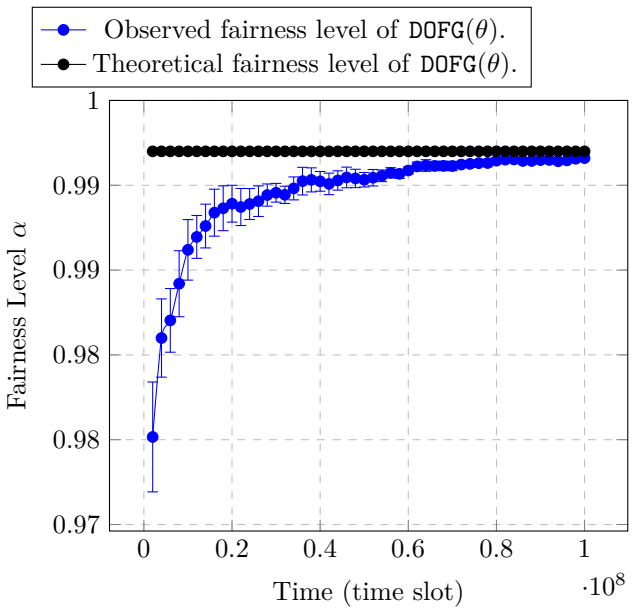



Figure 4.5 – Fairness level achieved by DOFG($\theta$) as a function of time with 10 players.

**Fairness Convergence.** Figure 4.5 shows the progress of the fairness level achieved by DOFG($\theta$) policy as time passes. The experimental settings are the same as those in section 4.7. The black plot corresponds to the theoretical fairness level proved in Theorem 4.3. In order to reach the theoretical fairness level, the observed mean rewards of all players have to reach their expected values. Due to the low probabilities of sending packets of the players, this would take a long time. As shown by figure 4.5, the observed fairness tends to the theoretical fairness in $10^8$ time steps for 10 players.

The source code of all experiments accomplished in this chapter are available in an open-source framework [2] published under the GNU GPL v2.0 license.

## 4.8 Conclusion

With the aim of optimizing transmission in IoT networks we have proposed an *explore-then-exploit* approach. We have proposed two target policies DORG and DOFG that are efficient with any number of players, and can handle internal and external collisions without *sensing*. We have shown that DORG in the setting proposed in [60] (when $\forall n, p_n = p$) is optimal, and that DOFG is fair. Then, we showed that using an $\epsilon$-approximation of the

2. https://github.com/Orange-OpenSource/MAB_IoT

model $\boldsymbol{\theta}$, the pseudo-regret lower bound of $\texttt{DORG}(\hat{\boldsymbol{\theta}})$ is optimal with respect to $T$ and that $\texttt{DOFG}(\hat{\boldsymbol{\theta}})$ is fair up to an additive term that decreases with $\epsilon$. Our experiments confirm the good behavior of *selfish UCB* and *CBAIMPB*, but show that both are outperformed in terms of network successful transmission rate by $\texttt{DORG}(\hat{\boldsymbol{\theta}})$ and $\texttt{DOFG}(\hat{\boldsymbol{\theta}})$, and in terms of fairness by $\texttt{DOFG}(\hat{\boldsymbol{\theta}})$. This work can be extended in many directions: studying *explore-and-exploit* approach for the proposed problem, handling an evolving number of active players, handling more general non-stationary environments, using an efficient change point detection [82], or by adapting *Exp3-Coop* [83] to competitive access to arms, handling players with different mean rewards of arms, handling the energy cost of each arm using the approaches developed in [52, 84], or using contextual bandits such as [85, 86]... Finally, showing the NP-Hardness of the optimization problem stated in equation (4.1) is an open problem.

The work accomplished in this chapter is submitted to the *38th Conference on Uncertainty in Artificial Intelligence (UAI) 2022*.

5

# Multi-Player MABs for Optimizing LoRa Communications

**Key Takeaways:** *Long-Range Wide Area Network* (LoRaWAN), a key technology for the IoT, is a fast-growing communication system due to its advantages in optimizing battery lifetime, capacity, range and cost. However, it faces many constraints including energy consumption and quality of service. In this chapter, we present an efficient way to manage the trade-off between energy consumption and packet losses of LoRa nodes using MP-MAB algorithms for nodes to adjust their emission parameters. We implement our reinforcement learning methods on a LoRa network simulator, and show that such learning techniques largely outperform the *Adaptive Data Rate* (ADR) algorithm, currently implemented in LoRa devices, in terms of energy consumption and packet losses.

In the previous chapters, we presented MP-MAB algorithms that aim to optimize the communications in IoT networks. In this chapter, we study the efficiency of such algorithms on one of the most deployed IoT technologies i.e. LoRaWAN, using a LoRa network simulator. The contribution of this chapter is two-fold:

- We redevelop and extend a LoRa network simulator to simulate large intelligent networks
- We model the LoRa communications optimization problem as a massive MP-MAB
- We study the performance of MP-MAB algorithms compared to ADR algorithm in terms of both energy consumption and packet loss

The remainder of this chapter is organized as follows. We first provide an overview of the LoRaWAN technology and the ADR algorithm in section 5.1. In section 5.2 we describe the LoRa network simulator used for experimentation. In Section 5.3 we demonstrate how we can apply massive multi-player multi-armed bandits to optimize the trade-off

between energy consumption and packet loss in a LoRa network. Section 5.4 includes the experiments and the numerical analysis, and we finally conclude in Section 5.5.

## 5.1 Overview on LoRaWAN Technology

LoRaWAN is a LPWAN protocol designed to wirelessly connect battery operated 'things' to the Internet, and targets key IoT requirements such as bi-directional communications, end-to-end security, mobility and localization services. LoRaWAN is designed to optimize LPWANs for battery lifetime, capacity, range, and cost. LoRa is the physical layer or the wireless modulation used to provide the long range communication link. While many wireless systems use the *Frequency Shift Keying* (FSK) modulation [87] for its high efficiency in achieving low power, LoRa instead uses the *chirp spread spectrum* (CSS) modulation [88] which maintains the same low power characteristics as FSK modulation but also significantly increases the communication range; a LoRa gateway provides wide coverage reaching 2-5km, 15km and 45km in urban, suburban and rural areas respectively[89]. It was developed by **Cycleo** [1] and acquired by **Semtech** [2], the founding member of the **LoRa Alliance**.

### 5.1.1 Network Architecture

Unlike many existing deployed networks that utilize a mesh network architecture, LoRaWAN nodes utilize the star architecture as shown in Figure 5.1. In a mesh network, the end-nodes receive and forward the information of other nodes in order to increase the communication range, however this adds complexity and reduces the battery lifetime. Unlikely, LoRa star topology preserves battery lifetime while long-range connectivity can be achieved. The LoRaWAN network end-nodes are not associated with a specific gateway, but alternatively the data transmitted by a node can be received by multiple gateways that forward this data to the network server. The latter will filter redundant received packets and schedule acknowledgements through the optimal gateway.

1. a French startup created in 2009 and based in Grenoble
2. a California-based semiconductor company, founded in 1960

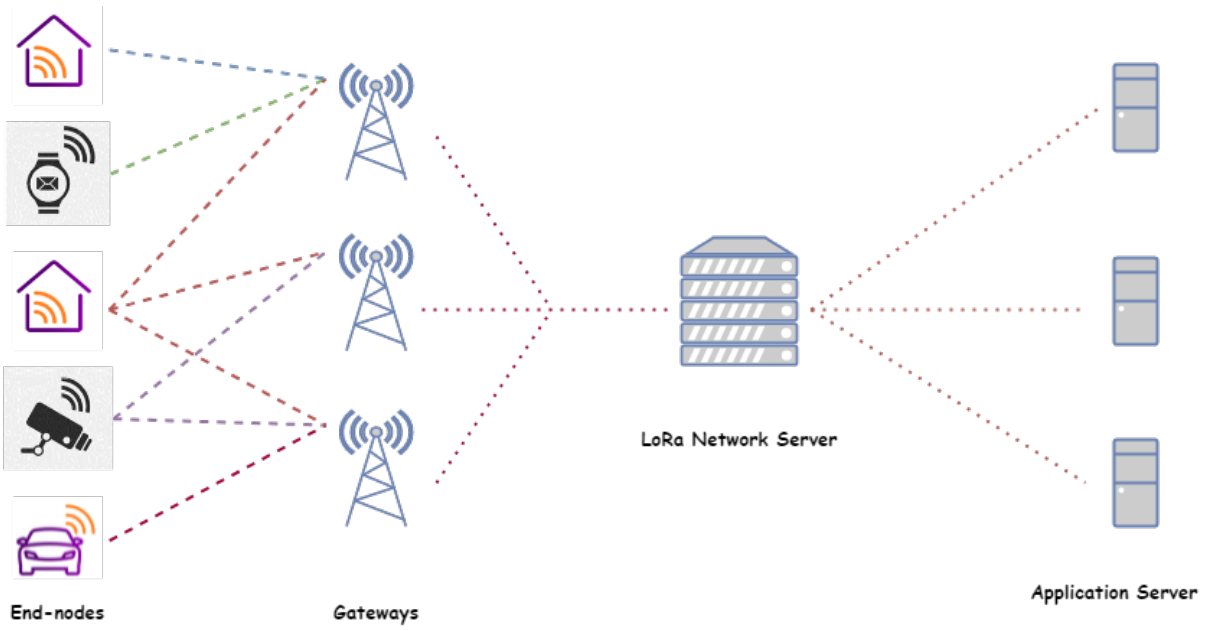



Figure 5.1 – LoRa Network Architecture

### 5.1.2 LoRaWAN Regional Parameters

LoRaWAN as any LPWAN technology operates in the unlicensed *Industrial Scientific and Medical* (ISM) frequency band. Two types of channel plans are considered:

- Dynamic channel plan: in which the majority of channels are defined after the join process
- Fixed channel plan: where the majority of channels (or all channels) are defined statically and known prior to the join process.

The LoRaWAN specification varies slightly from region to region based on the different regional spectrum allocations and regulatory requirements. In **Europe**, LoRaWAN operates in the $863 - 870$ MHz frequency band referred to by **EU868**. It defines 16 channels with 125/250 kHz bandwidth in a dynamic channel plan. The maximum output power allowed by the *European Telecommunications Standards Institute* (ETSI) in Europe is +14 dBm. There are duty cycle [3] restrictions under ETSI ($< 1\%$) but no maximum transmission or channel dwell time limitations [4][90].

3. It is a maximum percentage of time during which an end-device can occupy a channel
4. It is the amount of time needed to transmit on a frequency

### 5.1.3 LoRaWAN Classes

A LoRaWAN network distinguishes between a basic LoRaWAN (called Class A) and optional features (Class B, Class C ...). All LoRaWAN end-devices should implement at least Class A functionality, and they may implement Class B and/or Class C functionalities [91].

#### Bi-directional end-devices (Class A)

Class A end-devices allow bi-directional communications whereby each uplink (UL) transmission is followed by one or two short downlink (DL) receive windows (RX1 and RX2). If no packet is received in RX1, the end-device shall open RX2. They start a communication whenever they have data to send (ALOHA-type of protocol). This is the lowest power end-device system for applications that only require DL shortly after the end-device has sent an UL. A DL from the server at any other time will have to wait until the next scheduled UL.

#### Bi-directional end-devices with scheduled receive slots (Class B)

Class B devices open extra receive windows at scheduled times in addition to the two random receive windows of Class A.

#### Bi-directional end-devices with maximal receive slots (Class C)

Class C devices have almost continuously open receive windows, only closed when transmitting. They need more power to operate than Class A or Class B, but they feature the lowest latency for communications between servers and end-devices.

### 5.1.4 Network Capacity

The ability to manage the communication parameters of the end-devices in a LoRaWAN network allows to increase the capacity of the network. LoRaWAN nodes follow a pseudo-random channel hopping at each transmission so that simultaneous messages on multiple channels can be received. The resulting frequency diversity makes the system more robust to interference. The LoRa CSS modulation technique supports 6 orthogonal spreading factors (SF) corresponding to 6 different data rates: SF7 (50 kbps) to SF12 (300 bps). The signals are practically orthogonal to each other when different spreading factors are

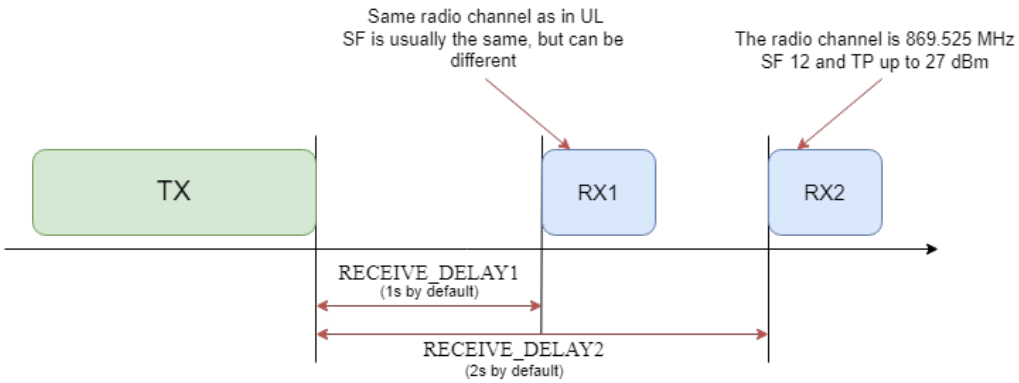



Figure 5.2 – Illustration of receive windows for DL emissions

utilized, so the gateway is able to receive multiple different data rates on the same channel at the same time.

### 5.1.5 Channel Mask

The network server (NS) can limit an end-device to a given set of uplink frequency channels amongst all the usable channels. It might use this strategy to reserve some channels for certain classes of applications. However, it is recommended not to constrain an end-device with a single channel, but keeping the possibility to hop across 2 channels at least.

### 5.1.6 Acknowledgement and Retransmission Procedures

LoRaWAN distinguishes between 2 types of frames: **confirmed** frames that shall be acknowledged by the receiver and **unconfirmed** frames that shall not be acknowledged. When the network receives a confirmed UL, it should send an acknowledgement using one of Class A receive windows (RX1 and RX2) opened by the end-device after the send operation (see Figure 5.2). UL confirmed and unconfirmed frames are transmitted `nbTrans` times unless a valid Class A DL is received. The default value of `nbTrans` is 1, which corresponds to a single transmission of each frame, and the valid range is $[1, 15]$. The end-device should perform channel hopping between retransmissions and should wait after each repetition until the receive windows have expired. The delay between retransmissions may be different for each end-device. For confirmed frames, an end-device waits `RETRANSMIT_TIMEOUT` seconds after `RECEIVE_DELAY2` seconds have elapsed after the end of the previous UL to receive an acknowledgement before it sends a new UL (retransmission or a new frame)[91].

### 5.1.7 Adaptive Data Rate

Rate adaptation is an essential feature in LoRaWAN. The data rate used by an end-point is dictated by the central NS based on the analysis of history of the received signal quality of the end-node. *Adaptive Data Rate* (ADR) is only suitable for static devices. It should not be applied on mobile devices since the radio channel changes dramatically with every frame. In this part, we present a simple baseline way to implement this decision mechanism recommended by **Semtech** [92] and to the best of our knowledge most of NSs operate based on it. Its performance has been evaluated in [93]. This algorithm in its present form is limited to EU868 ISM band, and to 6 data rates (SF12/125kHz to SF7/125kHz).

Each UL transmission might be received by several gateways that forward the frame to the NS. For rate adaptation, the NS considers the set of the Signal to Noise Ratio (SNR) of the last 20 received transmissions of each end-node denoted by $SNR_{20}$. For each transmission, it considers the SNR value that corresponds to the maximum value of the various SNRs reported by the different gateways who received this given frame.

When the end-device is in stable conditions, it sets the ADR bit in the frame header to 1 informing the NS that it is ready to receive ADR commands. Then the NS starts collecting the information about the UL transmissions of the device, and after 20 transmissions the NS runs the ADR algorithm presented in Algorithm 9, where:

- **margin_db** is the installation margin of the network which is a device specific static parameter. It is typically 10 dB in most networks [92].
- **SNR(SF)** is the required SNR to successfully demodulate a frame. It is a function of the SF of the end-device's last received frame and presented in Table 5.1

In Algorithm 9: the NS computes the number of steps $N_{step}$ to perform (line 2). If $N_{step}$ is positive, it means that the SNR values are high, so the NS decreases the end-node's SF in order to decrease the time on air and save energy. When the lowest SF is reached and there are still steps to perform, the NS decreases the transmitting power (TP) (lines 3-11). If $N_{step}$ is negative, and so the SNR values are low, the NS increases the TP of the end-node (lines 12-16), but it does not try to increase the SF since the end-device implements automatic data rate decay. At the end, the NS provides the SF and TP the end-device should use starting from the next transmission.

On the other hand, the end-device periodically needs to validate that the network still receives the uplink frames. Therefore, each time the UL frame counter is incremented, the end-device shall increment an `ADRACKCnt` counter. After `ADR_ACK_LIMIT`

**Algorithm 9** NS ADR algorithm for a given end-device

**Inputs:** SF, TP, margin_db, $\text{SNR}_{20}$, $\text{SF}_{\text{min}} = 7$, $\text{TP}_{\text{min}} = 2$ dBm
**Output:** SF, TP
**Init:** $\text{SNR}_{\text{max}} = \max(\text{SNR}_{20})$

$\text{SNR}_{\text{margin}} = \text{SNR}_{\text{max}} - \text{SNR}(\text{SF}) - \text{margin\_db}$
$\text{N}_{\text{step}} := \left\lfloor \frac{\text{SNR}_{\text{margin}}}{3} \right\rfloor$
**if** $\text{N}_{\text{step}} > 0$ **then**
  **while** $\text{N}_{\text{step}} > 0$ **and** $\text{SF} > \text{SF}_{min}$ **do**
    $\text{SF} := \text{SF} - 1$
    $\text{N}_{\text{step}} := \text{N}_{\text{step}} - 1$
  **end while**
  **while** $\text{N}_{\text{step}} > 0$ **and** $\text{TP} > \text{TP}_{min}$ **do**
    $\text{TP} := \text{TP} - 3$ dB
    $\text{N}_{\text{step}} := \text{N}_{\text{step}} - 1$
  **end while**
**else**
  **while** $\text{N}_{\text{step}} < 0$ **and** $\text{TP} < \text{TP}_{max}$ **do**
    $\text{TP} := \text{TP} + 3$ dB
    $\text{N}_{\text{step}} := \text{N}_{\text{step}} + 1$
  **end while**
**end if**

Table 5.1 – Spreading factors and their corresponding required SNR values to successfully demodulate

| SF | Required SNR (in dB) |
|---|---|
| SF7 | -7.5 |
| SF8 | -10 |
| SF9 | -12.5 |
| SF10 | -15 |
| SF11 | -17.5 |
| SF12 | -20 |

uplinks (`ADRACKCnt` $\geq$ `ADR_ACK_LIMIT`) without receiving a downlink response from the network, the end-device shall set the ADR acknowledgment request bit `ADRACKReq` on uplink transmissions. The Network is then required to respond with a DL frame within the next `ADR_ACK_DELAY` frames. Upon receipt of any DL, the end-device shall clear the `ADRACKReq` bit and reset the `ADRACKCnt` counter. If no DL is received within the next

`ADR_ACK_DELAY` uplinks (i.e., after a total of `ADR_ACK_LIMIT` + `ADR_ACK_DELAY` transmitted frames), the end-device shall try to regain connectivity by first setting TP to the default power (the maximum TP), then switching to the next lower data rate that provides a longer radio range. The end-device shall further lower its data rate (increase the SF) step by step every time `ADR_ACK_DELAY` uplink frames are transmitted. Once the end-device has reached the default data rate (the lowest data rate), and transmitted for `ADR_ACK_DELAY` uplinks with `ADRACKReq` =1 without receiving a DL, it shall re-enable all default uplink frequency channels.

Notice that the ADR scheme is a heuristic and is not based on any optimization objective, it modifies SF and TP depending on the SNR values. It also treats each device individually regardless of other devices in the network. In this work, we contrarily aim to optimize the global network capacity using massive MP-MAB using a LoRa network simulator described in the next section.

## 5.2 LoRa Network Simulator

The authors in [16] have developed a LoRa network simulator in Matlab. We extended this simulator to cover massive MP-MABs and run simulations for any time horizon T. The new version is written in C and available at [94]. It is described below.

### 5.2.1 Network Structure

A hexagonal distribution of the LoRa gateways with omnidirectional layout is modelled as shown in Figure 5.3. The gateways are located at the centers of the hexagonal cells and the nodes are uniformly distributed in the cell coverage. The number of gateways and nodes per cell are configurable as well as their heights. The inter-sight distance $d$ is also configurable depending on the simulated environment.

### 5.2.2 Network Operation

Although LoRaWAN uses pure ALOHA rather than slotted ALOHA (SA), authors in [95] show that SA outperforms pure ALOHA in terms of packet error rate (PER), throughput and collisions. Therefore, we choose SA rather than pure ALOHA to examine RL techniques because they both need time synchronization.

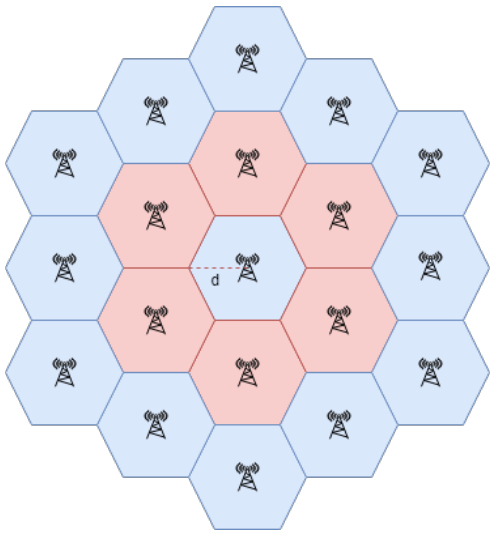



Figure 5.3 – Hexagonal distribution of gateway with inter-site distance $d$

Each node $n$ transmits at the beginning of a time slot with a fixed probability $p_n$. The time slot is of a configurable duration that respects a duty cycle of 1%.

We consider devices of class A. The devices always receive an acknowledgement if their UL is successful. In case of a packet loss, an end-device $n$ retransmits its packet in the next time slots with a probability $p_n^\dagger > p_n$ whose value depends on the application. The maximum possible number of retransmissions `nbTrans` is configurable and depends on the device.

We consider the ADR algorithm presented previously in section 5.1.7 with `ADR_ACK_DELAY` $= 32$ and `ADR_ACK_LIMIT` $= 64$ as recommended by LoRa Alliance [90] for all regions.

### 5.2.3 Transmission Success and Collision Rules

The success of a transmission mainly depends on two important metrics: the **Received Signal Strength Indicator** (RSSI) which characterizes the power level of a received radio signal, and the **Signal to Interference and Noise Ratio** (SINR). A packet is successfully received by a gateway if it does not collide with any other packets, and if its RSSI is strictly greater than the antenna sensitivity. The antenna sensitivity depends on the SF of the sent transmission as reported in Table 5.2.

On the other hand, a collision may occur when two or more frames sent on the same radio channel are received simultaneously. Considering two colliding frames: frame $a$ and frame $b$, two types of collisions are modelled:

Table 5.2 – Spreading factors and corresponding gateway antenna sensitivities [96]

| SF | LoRa gateway antenna sensitivity (dBm) |
|---|---|
| SF7 | -123 |
| SF8 | -126 |
| SF9 | -129 |
| SF10 | -132 |
| SF11 | -134.5 |
| SF12 | -137 |

Table 5.3 – Spreading factors and corresponding inter-SF collision threshold [97]

| SF | Inter-SF collision threshold (dB) |
|---|---|
| SF7 | -7.5 |
| SF8 | -9 |
| SF9 | -13.5 |
| SF10 | -15 |
| SF11 | -18 |
| SF12 | -22.5 |

- **Intra-SF collisions**: occurs when the colliding frames are of the same SF. The frame with the highest power will be decoded if it is at least 6 dB higher than the other LoRa frame: $\mathbf{RSSI}_a - \mathbf{RSSI}_b \geq 6$ **dB**
- **Inter-SF collisions**: occurs when the colliding frames are of different SFs ($SF_a \neq SF_b$). The frame is demodulated if the power difference is strictly greater than the inter-SF collision threshold which depends on the SF of the corresponding frame (see Table 5.3): frame "a" is demodulated if: $\mathbf{RSSI}_a - \mathbf{RSSI}_b > \mathbf{Thr}(\mathbf{SF}_a)$

### 5.2.4 Propagation Model

Propagation is modeled by the universal Okumura-Hata model, which is an accurate and widely used propagation model for predicting path loss in urban areas. Adaptations to rural and suburban areas are also added as recommended by ETSI for GSM 900 MHz [98]. This model takes into account the effects of diffraction, reflection and scattering caused

by city structures. It is generally used for frequency ranges of 150 MHz to 1500 MHz, for a link distance varying from 1 km to 20 km and for antenna heights varying from 30 m to 200 m and from 1 m to 10 m for the transmitter and the base station antenna respectively [99]. Typical indoor penetration losses are considered (18 dB, 15 dB, 12 dB and 10 dB for dense urban, urban, suburban and rural environments respectively) along with additional 6 dB loss for deep indoor environments [100, 101].

### 5.2.5 Environment Modeling

Two main environmental aspects are modeled: shadowing and fast fading. Shadowing is the effect causing the received signal power to fluctuate due to objects obstructing the propagation path between the transmitter and the receiver. The resulting loss is modeled as a random variable following a log-normal distribution with a standard deviation of 12 dB (resp., 6 dB) for outdoor (resp., indoor) settings. Fast fading or Rayleigh fading is the variation of the signal power due to multipath propagation, and its resulting loss is modeled using a Rayleigh distribution.

## 5.3 Modelling LoRa Communications as a Massive Multi-Player Multi-Armed Bandit

We model the LoRa communications as a massive MP-MAB. At each transmission, a node selects the corresponding SF and TP , and then observes a reward. We have a set of 30 arms of pairs of (SF, TP) corresponding to the 6 possible spreading factors (SF7, SF8, SF9, SF10, SF11 and SF12) and 5 transmitting power (2 dBm, 5 dBm, 8 dBm, 11 dBm and 14 dBm). Minimizing the energy consumption while maintaining a high packet delivery ratio (PDR) are two incompatible objectives: as the SF and TP increase the PDR increases and energy consumption increases. That is why our approach for handling energy consumption is to introduce a parametric function used to penalize high-energy consuming arms. We first normalize the values of the energy consumption of each arm with respect to the largest possible consumed energy (the arm with the highest power and greatest SF (SF12, 14 dBm)). Let $e^k \in (0, 1]$ be the value of the normalized energy consumed on arm $k$. The values of $e^k$ are presented in Table 5.4. We can notice that the energy consumption increases as TP and SF increase.

We consider the following function of arm $k$:

$$\xi_{\alpha,q}(e^k) = (1 - \alpha e^k)^q \tag{5.1}$$

$\xi_{\alpha,q}(e^k)$ is a decreasing function of the energy consumption $e^k$. The parameters $\alpha \in [0, 1)$ and $q \geq 1$ allow to shape it. We will consider this function in the reward function in order to penalize high-energy consuming arms with low rewards.

Table 5.4 – The normalized energy consumption per arm $e^k$, where the colors from blue to red correspond to the values from low to high

| | | SF7 | SF8 | SF9 | SF10 | SF11 | SF12 |
|---|---|---|---|---|---|---|---|
| TP (in dBm) | 2 | 0.0026 | 0.0046 | 0.0092 | 0.016 | 0.032 | 0.0631 |
| | 5 | 0.0052 | 0.0092 | 0.0183 | 0.031 | 0.063 | 0.1259 |
| | 8 | 0.0104 | 0.0183 | 0.0366 | 0.063 | 0.126 | 0.2512 |
| | 11 | 0.0208 | 0.0365 | 0.073 | 0.125 | 0.251 | 0.5012 |
| | 14 | 0.0416 | 0.0728 | 0.1457 | 0.25 | 0.5 | 1 |

To handle packet delivery, the used propagation model takes into account all conditions impacting it. As mentioned previously, a packet is successfully received if it does not collide with any internal or external transmissions, and the RSSI is strictly greater than the antenna sensitivity. To model packet delivery, we consider three random variables for every arm $k$:

- $E^k \in \{0, 1\}$ denotes the event *'no external collision occurs'*
- $I_n^k \in \{0, 1\}$ denotes the event *'no internal collision occurs for node $n$'*
- $D_n^k \in \{0, 1\}$ denotes the event *'no decoding error occurs'*

Consequently, the event *'transmission is successful'* for node $n$ is denoted $T_n^k \in \{0, 1\}$, such that:

$$T_n^k = E^k I_n^k D_n^k. \tag{5.2}$$

To handle both energy consumption and packet delivery we combine Equations (5.1) and (5.2) in the reward function of node $n$ playing arm $k$ below:

$$R_n^k(\alpha, q) = (1 - \alpha e^k)^q T_n^k \tag{5.3}$$

Table 5.5 – The network configuration and input parameters

| | |
|---|---|
| Channel Frequency | 868 MHz |
| Bandwidth | 125 kHz |
| Number of Gateways | 1 |
| Gateway noise figure | 3 dB |
| Gateway antenna gain | 5 dBi |
| Indoor penetration loss | 15 dB |
| Additional deep indoor loss | 6 dB |
| Gateway antenna height | 30 m |
| End-device height | 1.5 m |
| End-device antenna gain | 0 dBi |
| Targeted C/N after despreading | 6 dB |

Note that, other forms of the reward functions can be also considered and further investigated in the future. After each transmission, a node $n$ observes the success or failure of its transmission and computes the corresponding reward. It selects the SF and TP of its next transmission depending on the followed strategy. In the next section, with extensive experiments using the LoRa network simulator, we compare the performance of several MAB algorithms with the ADR algorithm in terms of both energy consumption and packet delivery.

## 5.4 Numerical Analysis

For our simulations, we consider a network operating in the LoRa European band 863−870 MHz. For simplicity we consider only one gateway and assume all transmissions are done on one frequency channel (868 MHz). The network configuration and input parameters are summarized in Table 5.5. We consider the worst case of a deep indoor LoRa network in an urban city. The frame size is 11 bytes (4 bytes of payload for the consumption index and 7 bytes Zigbee Cluster Library application protocol overhead) [16] corresponding to a smart metering application.

We consider a set of $N = 400$ end nodes where each node $n$ has a fixed probability $p_n$ to send a packet at the beginning of a time slot. The distribution of the nodes is uniformly chosen such that $\forall n, p_n \sim \mathcal{U}(7.10^{-4}, 5.10^{-3})$. We consider the maximum number of transmissions `nbTrans` =8. In case of a packet loss of any node $n$, it will increase its probability to send packets to $p_n^{\dagger} = p_n \times 8$ in order to be able to retransmit it before a

new packet is needed to be sent. The communication parameters of the retransmissions are chosen according to the policy the nodes follow.

In such settings, we compare the performance of ADR algorithm with *selfish* UCB [60], *collaborative exploration* followed by DORG or DOFG, CBAIMPB, and Exp3 [19], i.e. a commonly-used algorithm in non-stochastic environments. We consider the reward function of (5.3) for the MAB algorithms with penalty factor $\alpha = 0.5$ and $q = 4$ due to the very slow increase of energy values near 0 and very fast increase near 1 as shown in Table 5.4.

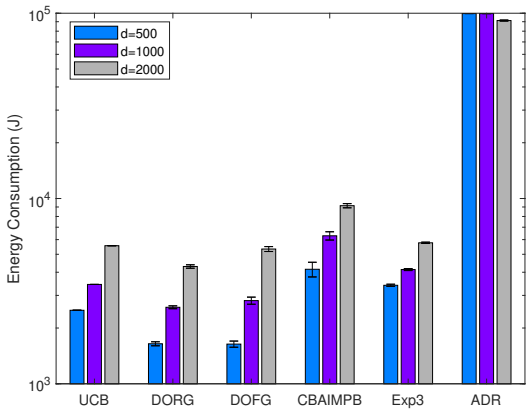
Although DORG and DOFG assume that the mean rewards of the arms are the same for all the nodes which necessitates that all nodes be located at the same distance from the gateway, we consider here that the nodes are uniformly distributed in the hexagonal cell region centered by the gateway. We consider 3 different inter-sight distances $d = \{500, 1000, 2000\}$. To simulate external traffic, we consider $S = 200$ static devices located in the same area, each sends packets with a fixed probability $p = 0.01$. Our experiments are designed so that $5.10^5$ packets are sent by the network in total during one trial, and the figures present the averaged values over 40 trials with 95% confidence intervals.

We perform two different experiments, each considering different external traffic.

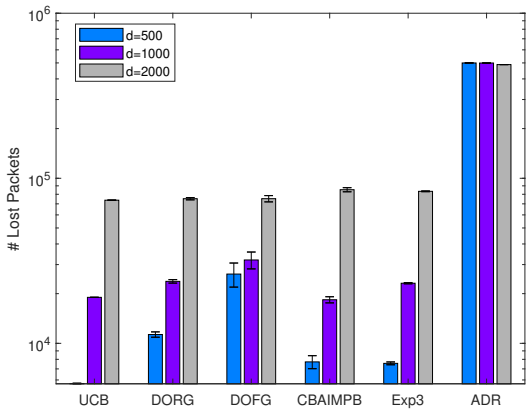
**Experiment 1** In the first experiment, we consider that the external nodes select an arm $k$ for each transmission with a probability $l^k \sim \mathcal{U}(0, 1)$, such that $\sum_{k=1}^{K} l^k = 1$, which makes the environment stationary.

In Fig. 5.4 we present the average values of the total energy consumed by the end nodes, the total number of lost packets and the total sum of rewards gained by the end-devices. It clearly shows that the nodes when implementing the ADR algorithm suffer of very high energy consumption and packet loss compared to the learning methods with any inter-site distance. This directly leads to greater sum of rewards for all the learning methods, and implies that MAB algorithms guarantee better management of the trade-off between energy consumption and packet loss, and provides a better QoS.

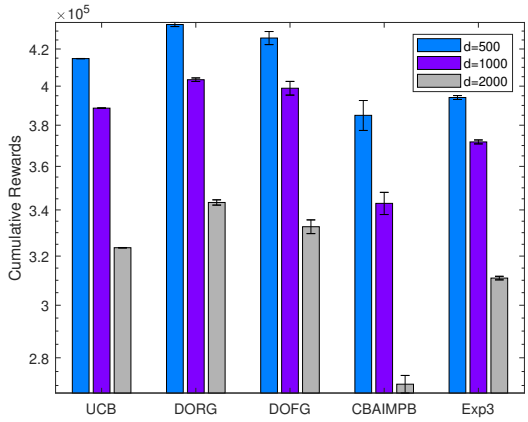
As described in section 5.2.3, inter-SF and intra-SF collisions may occur and lead to packet losses. Also, the propagation model introduces a decoding error, which depends on the topography, the position of the node, and the position of the gateway. Notice that this realistic propagation model violates two assumptions made by the theoretical model described in previous chapters: the channels are orthogonal, and the arms' qualities are the same for all nodes. Retransmissions are also not taken into account in the utility




(a) Energy consumption




(b) Total number of lost packets



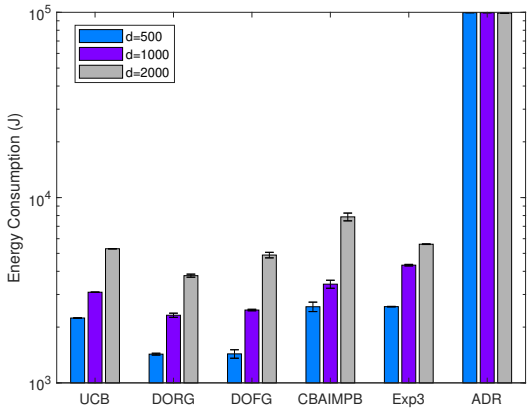

(c) The total rewards of the network

Figure 5.4 – **Experiment 1**: Performance of the LoRa network with end nodes distributed in hexagonal areas centered by the gateway with three different radii and external nodes following a fixed policy

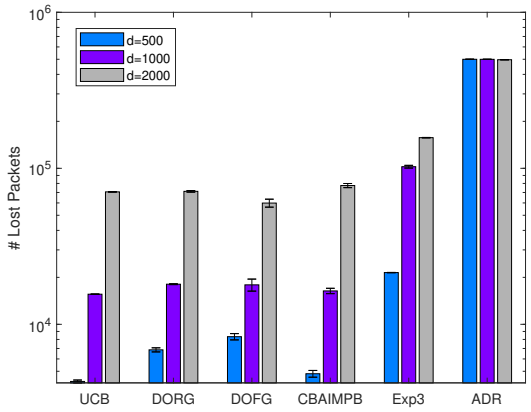
function (equation 4.1), and hence in the target policies DORG and DOFG.

Above all, despite there still being a gap between the theoretical model and the true model, DORG and DOFG highly outperform ADR in terms of energy consumption and packet loss and slightly outperform UCB by compromising energy consumption and packet loss (see Fig. 5.4c), while the latter which is a selfish algorithm developed for single-agent MABs shows to be highly robust against collisions and can compete with multi-player MABs. We also notice that stochastic algorithms outperform Exp3 even though its underlying assumptions are not violated.

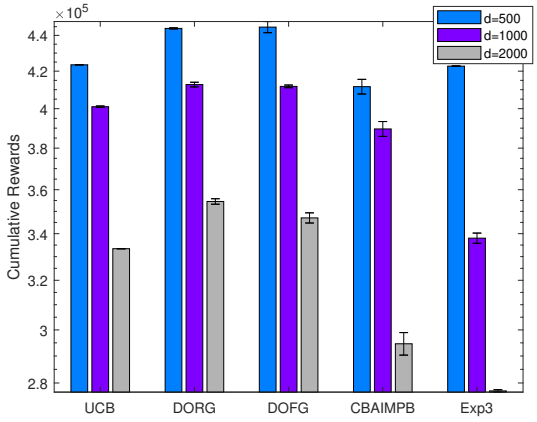
**Experiment 2** In this experiment, we consider that the external nodes are LoRa devices that follow the ADR algorithm. The results are presented in Figure 5.5.




(a) Energy consumption




(b) Total number of lost packets




(c) The total rewards of the network

Figure 5.5 – **Experiment 2**: Performance of the LoRa network with end nodes distributed in hexagonal areas centered by the gateway with three different radii and external nodes following ADR algorithm

We notice that the results are very similar to those in the previous experiment: all MAB algorithms outperform ADR, and our developed algorithms outperform other state-of-the-art MAB algorithms. This is due to the fact that the nodes following ADR algorithm do not change their selected arms frequently, but as shown in figure 5.6 the ratio of the number of changes of the selected arms by all the external nodes with respect to the number of plays at any distance is less than 0.15. This reveals that if there exist some nodes not following our collaborative algorithms but the ADR, they will lose.

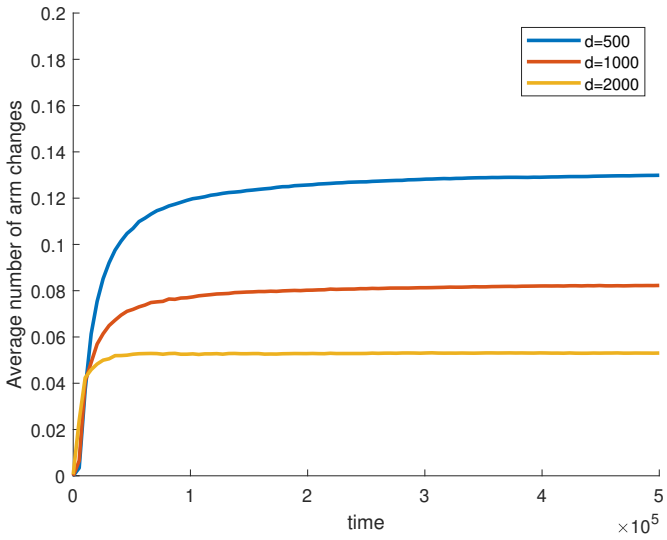



Figure 5.6 – Average number of arm changes with respect to the number of plays

On the other hand, notice that the *explore-then-exploit* algorithms are more appropriate for low-complexity devices (used in IoT networks) than UCB, Exp3 and other classic MAB algorithms, since after the exploration phase ends no computation takes place at the device side, unlike with UCB where the devices keep computing confidence bounds to find the next arm to select.

The source code of all the experiments accomplished in this chapter are available in an open-source framework [5] published under the GNU GPL v2.0 license .

## 5.5 Conclusion

In this chapter, we presented an application of MPMAB algorithms on LoRa networks in the sake of optimizing their performance. For this purpose, we propose to replace the ADR algorithm with massive MP-MAB algorithms to manage the trade off between the energy consumption and the packet loss by selecting the spreading factor and the transmitting power of the transmissions. Using a LoRa simulator that meets the LoRaWAN standards, we experimentally show that the MPMABs outperform the standard ADR algorithm by managing the trade off between the energy consumption and packet loss and achieving

5. https://github.com/Orange-OpenSource/MAB_IoT

high reduction of both metrics at different distances from the gateway.

The work accomplished in this chapter is to be submitted to the *IEEE Internet of Things Journal*.

6

# General Conclusion and Perspectives

This final chapter concludes the manuscript by summarizing the accomplished work and our contributions in section 6.1 and opening new perspectives in section 6.2.

## 6.1 Conclusion

This thesis contributes to the research field of optimizing IoT networks performance using machine learning. We organize this manuscript in six core chapters, including three with technical contributions. We started first by introducing the motivations behind our work. We provided a theoretical background on IoT, its importance, applications, constraints, and limitations. We aimed in this work to make IoT devices intelligent by embedding light decision-making algorithms, so that they can learn the surrounding environment and select efficiently the communication parameters such that they avoid interference with other devices to ensure reliability while consuming as little energy as possible. For this sake, we used Reinforcement Learning techniques, and in particular multi-player multi-armed bandit algorithms.

After the introduction, we provided a theoretical background on MABs before presenting the modeling of the IoT optimization problem as a MP-MAB problem. In our model, the devices in an IoT network are the players, the arms are the channels (i.e. the communication parameters) and the reward is a Bernoulli random variable presenting the success or failure of a transmission. The devices aim to maximize the number of successful transmissions so they minimize the number of retransmissions and hence save energy. We model congestion in the sense that if two players play the same arm at the same time, a collision happens and all colliding packets are lost. Both internal and external interference are modeled where the external interference affects the arms qualities (reward means) and are different from one arm to another. Unlike most of the work in the literature handling

dynamic spectrum access, we assumed that the devices cannot sense information i.e. they only observe the success or failure of their transmissions without distinguishing internal from external collisions, and the number of players is possibly greater than the number of arms. We also consider that the players can share information together by sending messages.

In such settings, we developed algorithms that aim to maximize the successful transmission rate of the IoT network while consuming as little energy as possible, and we provided numerical and experimental analysis of their performance.

Our first contribution is presented with a channel blacklisting approach provided in Chapter 3. We developed a collaborative algorithm that aims to find a set of optimal arms (best channels) with a certain confidence level. The basic idea of this approach is that the players run best arms identification algorithms, that are already presented and analyzed in the literature, as subroutines. The players share the output of the subroutines together until they obtain a set of a predefined number of optimal arms with a high level of confidence. The devices exploit the obtained set of optimal arms afterwards. We provided a numerical and experimental analysis of the algorithm in terms of both sample complexity and communication cost. We showed that playing the optimal arms uniformly achieves a greater successful transmission rate than playing the whole set of arms although the internal collision rate is greater. However, we showed that this result is highly dependent on the size of the optimal arms set. Also it depends on the problem settings and the exploitation strategy to be followed. Therefore, the number of optimal arms to find should be optimized as well as the exploitation strategy to be followed.

To overcome the limitations of the first approach, we provided a new approach that is presented in Chapter 4. We presented first the objective function we need to optimize, which corresponds to the expected number of successful transmissions per time slot in an IoT network, and we aim to find the policy followed by the players such that the objective function is maximized. However, the optimization problem is intractable, but we showed that at least one solution is a *deterministic* policy. Also, the fact that the channels are shared with a number of players that is greater than the number of channels raises the problem of *fairness* between players. We therefore proposed two greedy policies: Decreasing-Order-Reward-Greedy (`DORG`) that focuses on the number of successful transmissions, and Decreasing-Order-Fair-Greedy (`DOFG`) that also guarantees fairness between players. We proved that `DORG` is optimal in some cases, and `DOFG` is fair up to a certain level that depends on the players' distribution. We then provided some preliminary

experiments that study the performance of both policies.

Since the policies require the values of the channel qualities, we propose a decentralized exploration algorithm with controlled information exchanges between players that estimates these values. Its basic idea is to distribute the exploration task on the players according to their probabilities of sending packets, so that they finish the estimation almost at the same time. We showed that its sample complexity is near optimal and when `DORG` is optimal, its pseudo-regret, when using the approximated model, is optimal with respect to the time horizon $T$. We also showed that `DOFG` is still fair when using the approximated model. We finally provided experimental evidence that the proposed algorithms outperform the state-of-the-art in terms of fairness and successful transmissions.

Finally, in order to test our algorithms on IoT networks, we provided an application on LoRa networks in chapter 5. For this sake, we redeveloped a LoRa network simulator to be adapted to our settings and tested our algorithms on it. After an overview on LoRaWAN including the adaptive data rate (ADR) algorithm that is already implemented in LoRa devices, we presented the LoRa network simulator including the environment and propagation model, congestion model and others. We then presented our model of LoRa communications as a massive MP-MAB, where the players are the end-devices and the arms are couples of the spreading factors and transmitting power forming 30 different arms. As the energy consumption is different from one arm to another, we formulated a reward function that takes into account both transmission's success or failure and the energy consumed on the arm. In the experiments, we considered that our network is coexisting with other devices sharing the same spectrum and sending to the same gateway. We compared our algorithms with the ADR algorithm in terms of both successful transmission rate and energy consumption, where we showed the high out-performance of our algorithms in both terms.

## 6.2 Perspectives

This work opens several avenues for future work on massive MP-MABs for IoT networks. The general model of the problem as well as the contributions in this work can be further extended in different directions.

**Problem model** In this work, we assumed a fixed number of players and we considered a stationary stochastic environment. One direction to improve this work can be by

handling more general non-stationary environments and handling an evolving number of active players. For this, we might use efficient change point detection such as Bayesian Online Change Point Detector proposed in [102] and improved in [82], or by adapting the algorithm *Exp3-Coop* proposed in [83] to competitive access to arms. We also assumed that the arms qualities are the same for all players in the network and they have the same energy cost, which is not totally realistic in IoT. Indeed, the quality of the arms may change from one player to another according to their geographical positions, also as we have seen in Chapter 5 in LoRa networks, the arms do not always have the same energy cost. To deal with this, one direction can be to use the approaches developed in [52, 84] or use the contextual bandits such as in [85, 86] which is an extension of the MAB model by making the decision conditional on the state of the environment, so the decision is not only optimized based on previous observations, but is also personalized for every situation.
Besides, in this work we considered a slotted-ALOHA transmission protocol where nodes send at the beginning of the fixed-duration time slots. But, since the time-on-air of packets varies (depending on the selected spreading factor), considering slotted-ALOHA necessitates long-duration time slots which decreases the performance by creating collisions, i.e. the long-duration of the time slot can be enough for multiple consecutive transmissions on the same low SF (such as SF7) instead of one transmission at the beginning of the time slot. Future works could overcome this by considering sub-slotting: one time slot can be divided into several sub-slots of durations that depend on the time-on-air of the transmission ( 1 sub-slot for SF12, 2 sub-slots for SF11, 4 for SF10,..etc.).

**Best arms identification for sub-optimal channels blacklisting** As previously noted, our first approach that aims to find a set of optimal channels is lacking the optimization of the size of the optimal set and the exploitation strategy. The network performance afterwards is highly dependent on the size of the optimal set $m$, and the exploitation strategy to be followed. Depending on the number of players and their distribution, and with a uniform exploitation strategy, one simple idea can be to upper bound the internal collision rate and hence compute the value of $m$. On the other hand, knowing the value of $m$, the players can be distributed evenly over the optimal arms based on their probabilities to send packets, so that the internal collision rate on each arm is almost the same.

**Experimentation** On the experimental side, we tested MP-MABs on a LoRa network simulator considering that only one gateway is available. Future work could consider

multiple gateways receiving packets and forwarding them to the network server as in reality. Also, as stated before, different reward functions can be considered.

Moreover, a great contribution could be to establish real experiments testing different MAB and MP-MABs as the literature is poor in such experiments. One way to achieve this is by using the so-called *pycom LoPy*[4] which is a Micropython-programmable board that works with LoRa, Sigfox, WiFi and Bluetooth. Using such boards, one can establish a network and test the algorithms.

---

Thanks for reading this document.

# List of Figures

# List of Tables

# List of Algorithms

# BIBLIOGRAPHY


[1] ASHTON Kevin. *That 'Internet of Things' Thing*. Online at https://www.rfidjournal.com/that-internet-of-things-thing. Accessed: 2022-02-15.

[2] IoT Analytics Research. *State of IoT 2021: Number of connected IoT devices growing 9% to 12.3 billion globally, cellular IoT now surpassing 2 billion*. Online at https://iot-analytics.com/wp/wp-content/uploads/2021/09/Global-IoT-market-forecast-in-billion-connected-iot-devices-min.png. Accessed: 2022-02-15.

[3] Wikipedia. *Internet of things*. Online at https://en.wikipedia.org/wiki/Internet_of_things. Accessed: 2022-02-15.

[4] Arpan Pal, Arijit Mukherjee, and Swarnava Dey. « Future of healthcare—sensor data-driven prognosis ». In: *Wireless World in 2050 and Beyond: A Window into the Future!* Springer, 2016, pp. 93–109.

[5] Noushin Poursafar, Md Eshrat E Alahi, and Subhas Mukhopadhyay. « Long-range wireless technologies for IoT applications: A review ». In: *2017 Eleventh International Conference on Sensing Technology (ICST)*. IEEE. 2017, pp. 1–6.

[6] Katrina M Miranda, Michael G Espey, and David A Wink. « A rapid, simple spectrophotometric method for simultaneous detection of nitrate and nitrite ». In: *Nitric oxide* 5.1 (2001), pp. 62–71.

[7] Olof Liberg, Marten Sundberg, Eric Wang, Johan Bergman, and Joachim Sachs. *Cellular Internet of things: technologies, standards, and performance*. Academic Press, 2017.

[8] Herbert Robbins. « Some aspects of the sequential design of experiments ». In: *Bulletin of the American Mathematical Society* 58.5 (1952), pp. 527–535.

[9] Tze Leung Lai and Herbert Robbins. « Asymptotically efficient adaptive allocation rules ». In: *Advances in applied mathematics* 6.1 (1985), pp. 4–22.

[10] Djallel Bouneffouf and Irina Rish. « A survey on practical applications of multi-armed and contextual bandits ». In: *arXiv preprint arXiv:1904.10040* (2019).

[11] Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. « Finite-time analysis of the multiarmed bandit problem ». In: *Machine learning* 47.2-3 (2002), pp. 235–256.

[12] Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire. « Gambling in a rigged casino: The adversarial multi-armed bandit problem ». In: *Proceedings of IEEE 36th Annual Foundations of Computer Science*. IEEE. 1995, pp. 322–331.

[13] Vishakha Patil, Ganesh Ghalme, Vineet Nair, and Yadati Narahari. « Achieving Fairness in the Stochastic Multi-Armed Bandit Problem. » In: *AAAI*. 2020, pp. 5379–5386.

[14] Aloÿs Augustin, Jiazi Yi, Thomas Clausen, and William Mark Townsley. « A study of LoRa: Long range & low power networks for the internet of things ». In: *Sensors* 16.9 (2016), p. 1466.

[15] N. Sornin and A. Yegin. *LoRaWAN$^{TM}$ Specification, V1.0.3*. July 2018.

[16] Nadège Varsier and Jean Schwoerer. « Capacity limits of LoRaWAN technology for smart metering applications ». In: *2017 IEEE international conference on communications (ICC)*. IEEE. 2017, pp. 1–6.

[17] Sébastien Bubeck and Nicolo Cesa-Bianchi. « Regret analysis of stochastic and nonstochastic multi-armed bandit problems ». In: *arXiv preprint arXiv:1204.5721* (2012).

[18] Aleksandrs Slivkins. « Introduction to multi-armed bandits ». In: *arXiv preprint arXiv:1904.07272* (2019).

[19] Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire. « The nonstochastic multiarmed bandit problem ». In: *SIAM journal on computing* 32.1 (2002), pp. 48–77.

[20] Jean-Yves Audibert, Rémi Munos, and Csaba Szepesvári. « Variance estimates and exploration function in multi-armed bandit ». In: *CERTIS Research Report 07–31*. 2007.

[21] Olivier Cappé, Aurélien Garivier, Odalric-Ambrym Maillard, Rémi Munos, and Gilles Stoltz. « Kullback-Leibler upper confidence bounds for optimal sequential allocation ». In: *The Annals of Statistics* (2013), pp. 1516–1541.

[22] William R Thompson. « On the likelihood that one unknown probability exceeds another in view of the evidence of two samples ». In: *Biometrika* 25.3/4 (1933), pp. 285–294.

[23] Emilie Kaufmann, Nathaniel Korda, and Rémi Munos. « Thompson sampling: An asymptotically optimal finite-time analysis ». In: *International conference on algorithmic learning theory*. Springer. 2012, pp. 199–213.

[24] Levente Kocsis and Csaba Szepesvári. « Discounted ucb ». In: *2nd PASCAL Challenges Workshop*. Vol. 2. 2006.

[25] Aurélien Garivier and Eric Moulines. « On upper-confidence bound policies for switching bandit problems ». In: *International Conference on Algorithmic Learning Theory*. Springer. 2011, pp. 174–188.

[26] Aurélien Garivier and Eric Moulines. « On upper-confidence bound policies for non-stationary bandit problems ». In: *arXiv preprint arXiv:0805.3415* (2008).

[27] Joseph Mellor and Jonathan Shapiro. « Thompson sampling in switching environments with Bayesian online change detection ». In: *Artificial Intelligence and Statistics*. PMLR. 2013, pp. 442–450.

[28] Réda Alami, Odalric Maillard, and Raphael Féraud. « Memory bandits: a bayesian approach for the switching bandit problem ». In: *NIPS 2017-31st Conference on Neural Information Processing Systems*. 2017.

[29] Vishnu Raj and Sheetal Kalyani. « Taming non-stationary bandits: A Bayesian approach ». In: *arXiv preprint arXiv:1707.09727* (2017).

[30] Francesco Trovo, Stefano Paladino, Marcello Restelli, and Nicola Gatti. « Sliding-window thompson sampling for non-stationary settings ». In: *Journal of Artificial Intelligence Research* 68 (2020), pp. 311–364.

[31] Omar Besbes, Yonatan Gur, and Assaf Zeevi. « Stochastic multi-armed-bandit problem with non-stationary rewards ». In: *Advances in neural information processing systems* 27 (2014), pp. 199–207.

[32] Robin Allesiardo, Raphaël Féraud, and Odalric-Ambrym Maillard. « The non-stationary stochastic multi-armed bandit problem ». In: *International Journal of Data Science and Analytics* 3.4 (2017), pp. 267–283.

[33] Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire. « The nonstochastic multiarmed bandit problem ». In: *SIAM journal on computing* 32.1 (2002), pp. 48–77.

[34] Jonathan Louëdec, Laurent Rossi, Max Chevalier, Aurélien Garivier, and Josiane Mothe. « Algorithme de bandit et obsolescence: un modèle pour la recommandation ». In: (2016).

[35] Lifeng Lai, Hai Jiang, and H Vincent Poor. « Medium access in cognitive radio networks: A competitive multi-armed bandit framework ». In: *2008 42nd Asilomar Conference on Signals, Systems and Computers*. IEEE. 2008, pp. 98–102.

[36] Keqin Liu and Qing Zhao. « Distributed learning in multi-armed bandit with multiple players ». In: *IEEE Transactions on Signal Processing* 58.11 (2010), pp. 5667–5681.

[37] Animashree Anandkumar, Nithin Michael, Ao Kevin Tang, and Ananthram Swami. « Distributed algorithms for learning and cognitive medium access with logarithmic regret ». In: *IEEE Journal on Selected Areas in Communications* 29.4 (2011), pp. 731–745.

[38] Orly Avner and Shie Mannor. « Concurrent Bandits and Cognitive Radio Networks ». In: *ECML PKDD*. Berlin, Heidelberg: Springer-Verlag, 2014.

[39] Etienne Boursier and Vianney Perchet. « SIC-MMAB: Synchronisation Involves Communication in Multiplayer Multi-Armed Bandits ». In: *Advances in Neural Information Processing Systems 32*. 2019, pp. 12048–12057.

[40] Jonathan Rosenski, Ohad Shamir, and Liran Szlak. « Multi-Player Bandits – a Musical Chairs Approach ». In: *ICML*. 2016.

[41] Sébastien Bubeck, Yuanzhi Li, Yuval Peres, and Mark Sellke. « Non-Stochastic Multi-Player Multi-Armed Bandits: Optimal Rate With Collision Information, Sublinear Without ». In: (2019). URL: https://arxiv.org/abs/1904.12233.

[42] Shafi Kamalbasha and Manuel JA Eugster. « Bayesian A/B testing for business decisions ». In: *Data science–analytics and applications*. Springer, 2021, pp. 50–57.

[43] Yi Gai, Bhaskar Krishnamachari, and Rahul Jain. « Learning multiuser channel allocations in cognitive radio networks: A combinatorial multi-armed bandit formulation ». In: *2010 IEEE Symposium on New Frontiers in Dynamic Spectrum (DySPAN)*. IEEE. 2010, pp. 1–9.

[44] Ravi Ganti, Matyas Sustik, Quoc Tran, and Brian Seaman. « Thompson sampling for dynamic pricing ». In: *arXiv preprint arXiv:1802.03050* (2018).

[45] Levente Kocsis and Csaba Szepesvári. « Bandit based monte-carlo planning ». In: *European conference on machine learning*. Springer. 2006, pp. 282–293.

[46] Ying-Chang Liang. « Opportunistic Spectrum Access ». In: *Dynamic Spectrum Management: From Cognitive Radio to Blockchain and Artificial Intelligence*. Singapore: Springer Singapore, 2020, pp. 19–40. ISBN: 978-981-15-0776-2. DOI: 10.1007/978-981-15-0776-2_2. URL: https://doi.org/10.1007/978-981-15-0776-2_2.

[47] Keqin Liu and Qing Zhao. « Distributed Learning in Multi-Armed Bandit With Multiple Players ». In: *IEEE Transactions on Signal Processing* 58.11 (2010), pp. 5667–5681.

[48] Animashree Anandkumar, Nithin Michael, and Ao Tang. « Opportunistic spectrum access with multiple users: Learning under competition ». In: *2010 Proceedings IEEE INFOCOM*. IEEE. 2010, pp. 1–9.

[49] Naumaan Nayyar, Dileep Kalathil, and Rahul Jain. « On regret-optimal learning in decentralized multiplayer multiarmed bandits ». In: *IEEE Transactions on Control of Network Systems* 5.1 (2016), pp. 597–606.

[50] Manjesh Kumar Hanawal and Sumit Darak. « Multi-player bandits: A trekking approach ». In: *IEEE Transactions on Automatic Control* (2021).

[51] Po-An Wang, Alexandre Proutiere, Kaito Ariu, Yassir Jedra, and Alessio Russo. « Optimal algorithms for multiplayer multi-armed bandits ». In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2020, pp. 4120–4129.

[52] Etienne Boursier, Vianney Perchet, Emilie Kaufmann, and Abbas Mehrabian. « A Practical Algorithm for Multiplayer Bandits when Arm Means Vary Among Players ». In: *AISTATS*. 2020.

[53] Tevfik Yucek and Huseyin Arslan. « A survey of spectrum sensing algorithms for cognitive radio applications ». In: *IEEE Communications Surveys Tutorials* 11.1 (2009), pp. 116–130. DOI: 10.1109/SURV.2009.090109.

[54] Mansi Subhedar and Gajanan Birajdar. « Spectrum sensing techniques in cognitive radio networks: A survey ». In: *International Journal of Next-Generation Networks* 3.2 (2011), pp. 37–51.

[55] Chengshuai Shi, Wei Xiong, Cong Shen, and Jing Yang. « Decentralized multi-player multi-armed bandits with no collision information ». In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2020, pp. 1519–1528.

[56] Gábor Lugosi and Abbas Mehrabian. « Multiplayer bandits without observing collision information ». In: *arXiv preprint arXiv:1808.08416* (2018).

[57] Dimitri P Bertsekas, Robert G Gallager, and Pierre Humblet. *Data networks*. Vol. 2. Prentice-Hall International New Jersey, 1992.

[58] Xin Wang and Koushik Kar. « Distributed algorithms for max-min fair rate allocation in Aloha networks ». In: *Proceedings of the 42nd Annual Allerton Conference*. Citeseer. 2004.

[59] Kobi Cohen, Amir Leshem, and Ephraim Zehavi. « Game theoretic aspects of the multi-channel ALOHA protocol in cognitive radio networks ». In: *IEEE Journal on Selected Areas in Communications* 31.11 (2013), pp. 2276–2288.

[60] Rémi Bonnefoi, Lilian Besson, Christophe Moy, Emilie Kaufmann, and Jacques Palicot. « Multi-Armed Bandit Learning in IoT Networks: Learning Helps Even in Non-stationary Settings ». In: *Cognitive Radio Oriented Wireless Networks*. Ed. by Paulo Marques, Ayman Radwan, Shahid Mumtaz, Dominique Noguet, Jonathan Rodriguez, and Michael Gundlach. Springer International Publishing, 2018, pp. 173–185.

[61] Lilian Besson and Emilie Kaufmann. « Multi-Player Bandits Revisited ». In: *Proceedings of Algorithmic Learning Theory*. Vol. 83. 2018, pp. 56–92.

[62] Raouf Kerkouche, Réda Alami, Raphaël Féraud, Nadège Varsier, and Patrick Maillé. « Node-based optimization of LoRa transmissions with Multi-Armed Bandit algorithms ». In: *2018 25th International Conference on Telecommunications (ICT)*. IEEE. 2018, pp. 521–526.

[63] Hiba Dakdouk, Erika Tarazona, Reda Alami, Raphaël Féraud, Georgios Z Papadopoulos, and Patrick Maillé. « Reinforcement Learning Techniques for Optimized Channel Hopping in IEEE 802.15. 4-TSCH Networks ». In: *Proceedings of the 21st ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*. 2018, pp. 99–107.

[64] Shivaram Kalyanakrishnan and Peter Stone. « Efficient Selection of Multiple Bandit Arms: Theory and Practice. » In: *ICML*. Vol. 10. 2010, pp. 511–518.

[65] Thomas Watteyne, Ankur Mehta, and Kris Pister. « Reliability through frequency diversity: why channel hopping makes sense ». In: *Proceedings of the 6th ACM symposium on Performance evaluation of wireless ad hoc, sensor, and ubiquitous networks*. 2009, pp. 116–123.

[66] « IEEE Standard for Low-Rate Wireless Networks ». In: *IEEE Std 802.15.4-2015 (Revision of IEEE Std 802.15.4-2011)* (2016), pp. 1–709. DOI: 10.1109/IEEESTD.2016.7460875.

[67] Jianping Song, Song Han, Al Mok, Deji Chen, Mike Lucas, Mark Nixon, and Wally Pratt. « WirelessHART: Applying wireless technology in real-time industrial process control ». In: *2008 IEEE Real-Time and Embedded Technology and Applications Symposium*. IEEE. 2008, pp. 377–386.

[68] Leslie G Valiant. « A theory of the learnable ». In: *Communications of the ACM* 27.11 (1984), pp. 1134–1142.

[69] Sébastien Bubeck, Rémi Munos, and Gilles Stoltz. « Pure exploration in multi-armed bandits problems ». In: *International conference on Algorithmic learning theory*. Springer. 2009, pp. 23–37.

[70] Jean-Yves Audibert, Sébastien Bubeck, and Remi Munos. « Best Arm Identification in Multi-Armed Bandits ». In: *COLT - The 23rd Conference on Learning Theory*. Nov. 2010, pp. 41–53.

[71] Victor Gabillon, Mohammad Ghavamzadeh, and Alessandro Lazaric. « Best arm identification: A unified approach to fixed budget and fixed confidence ». In: *NIPS-Twenty-Sixth Annual Conference on Neural Information Processing Systems*. 2012.

[72] Eyal Even-Dar, Shie Mannor, and Yishay Mansour. « Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems ». In: *Journal of machine learning research* 7.Jun (2006), pp. 1079–1105.

[73] Shivaram Kalyanakrishnan, Ambuj Tewari, Peter Auer, and Peter Stone. « PAC Subset Selection in Stochastic Multi-armed Bandits. » In: *ICML*. Vol. 12. 2012, pp. 655–662.

[74] Emilie Kaufmann and Shivaram Kalyanakrishnan. « Information complexity in bandit subset selection ». In: *Conference on Learning Theory*. 2013, pp. 228–251.

[75] Aurélien Garivier and Emilie Kaufmann. « Optimal best arm identification with fixed confidence ». In: *Conference on Learning Theory*. PMLR. 2016, pp. 998–1027.

[76] Raphaël Féraud, Réda Alami, and Romain Laroche. « Decentralized Exploration in Multi-Armed Bandits ». In: *ICML*. 2019.

[77] Kwang-Sung Jun and Robert D Nowak. « Anytime Exploration for Multi-armed Bandits using Confidence Information. » In: *ICML*. 2016, pp. 974–982.

[78] Oded Maron and Andrew W Moore. « The racing algorithm: Model selection for lazy learners ». In: *Lazy learning*. Springer, 1997, pp. 193–225.

[79] Solomon Kullback and Richard A Leibler. « On information and sufficiency ». In: *The annals of mathematical statistics* 22.1 (1951), pp. 79–86.

[80] Martin Anthony and Peter L. Bartlett. *Neural Network Learning: Theoretical Foundations*. 1st. USA: Cambridge University Press, 1999. ISBN: 052111862X.

[81] Aurélien Garivier, Tor Lattimore, and Emilie Kaufmann. « On explore-then-commit strategies ». In: *Advances in Neural Information Processing Systems* 29 (2016), pp. 784–792.

[82] Réda Alami, Odalric Maillard, and Raphael Féraud. « Restarted Bayesian Online Change-point Detector achieves Optimal Detection Delay ». In: *International Conference on Machine Learning*. PMLR. 2020, pp. 211–221.

[83] Nicolo Cesa-Bianchi, Claudio Gentile, Yishay Mansour, and Alberto Minora. « Delay and cooperation in nonstochastic bandits ». In: *Conference on Learning Theory*. PMLR. 2016, pp. 605–622.

[84] Lydia T Liu, Horia Mania, and Michael Jordan. « Competing bandits in matching markets ». In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2020, pp. 1618–1628.

[85] Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. « Improved algorithms for linear stochastic bandits ». In: *Advances in neural information processing systems* 24 (2011), pp. 2312–2320.

[86] Raphaël Féraud, Robin Allesiardo, Tanguy Urvoy, and Fabrice Clérot. « Random forest for the contextual bandit problem ». In: *Artificial intelligence and statistics*. PMLR. 2016, pp. 93–101.

[87] Wikipedia contributors. *Frequency-shift keying*. [Online; accessed 22-November-2021]. 2021. URL: https://en.wikipedia.org/wiki/Frequency-shift_keying.

[88] Wikipedia contributors. *Chirp spread spectrum*. [Online; accessed 22-November-2021]. 2021. URL: https://en.wikipedia.org/wiki/Chirp_spread_spectrum.

[89] Jonathan de Carvalho Silva, Joel JPC Rodrigues, Antonio M Alberti, Petar Solic, and Andre LL Aquino. « LoRaWAN—A low power WAN protocol for Internet of Things: A review and opportunities ». In: *2017 2nd International Multidisciplinary Conference on Computer and Energy Science (SpliTech)*. IEEE. 2017, pp. 1–6.

[90] LoRa Alliance. « RP002-1.0.3 LoRaWAN Regional Parameters ». In: *white paper, May* 20 (2021).

[91] LoRa Alliance. « LoRaWAN L2 1.0.4 Specification (TS001-1.0.4) ». In: *white paper, October* (2020).

[92] Semtech Corporation. *LoRaWAN – simple rate adaptation recommended algorithm.* [Online; accessed 25-November-2021]. URL: https://www.thethingsnetwork.org/forum/uploads/default/original/2X/7/7480e044aa93a54a910dab8ef0adfb5f515d14a1.pdf.

[93] Riccardo Marini, Walter Cerroni, and Chiara Buratti. « A Novel Collision-Aware Adaptive Data Rate Algorithm for LoRaWAN Networks ». In: *IEEE Internet of Things Journal* 8.4 (2021), pp. 2670–2680.

[94] Dakdouk Hiba. *LoRaSim_MABs: A LoRa network simulator with applied MAB algorithms.* Code at `https://github.com/IoT-MABs/LoRaSim_MABs.git`. 2019–2021.

[95] Zulfiqar Ali, Shagufta Henna, Adnan Akhunzada, Mohsin Raza, and Sung Won Kim. « Performance Evaluation of LoRaWAN for Green Internet of Things ». In: *IEEE Access* 7 (2019), pp. 164102–164112. DOI: 10.1109/ACCESS.2019.2943720.

[96] Semtech. *Understanding the LoRa Adaptive Data Rate.* Available on: semtech.com/LoRa, December 2019.

[97] Antoine Waret, Megumi Kaneko, Alexandre Guitton, and Nancy El Rachkidy. « LoRa throughput analysis with imperfect spreading factor orthogonality ». In: *IEEE Wireless Communications Letters* 8.2 (2018), pp. 408–411.

[98] TR ETSI. « Digital cellular telecommunications system (Phase 2+); Radio Network Planning Aspects (3GPP TR 03.30 version 8.4.0 Release 1999) ». In: *ETSI TR 101 362 V8.4.0* (June 2005).

[99] Abraham Deme, Danjuma Dajab, MMA Buba Bajoga, and D Choji. « Hata-Okumura Model Computer Analysis for Path Loss Determination at 900MHz for Maiduguri, Nigeria ». In: *Mathematical Theory and Modeling* 3.3 (2013), pp. 1–9.

[100] Lúcio Ferreira, Martijn Kuipers, Carlos Rodrigues, and Luis M Correia. « Characterisation of Signal Penetration into Buildings for GSM and UMTS ». In: *2006 3rd International Symposium on Wireless Communication Systems*. IEEE. 2006, pp. 63–67.

[101] Ignacio Rodriguez, Huan C Nguyen, Niels TK Jørgensen, Troels B Sørensen, Jan Elling, Morten B Gentsch, and Preben Mogensen. « Path loss validation for urban micro cell scenarios at 3.5 GHz compared to 1.9 GHz ». In: *2013 IEEE global communications conference (GLOBECOM)*. IEEE. 2013, pp. 3942–3947.

[102] Paul Fearnhead and Zhen Liu. « On-line inference for multiple changepoint problems ». In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 69.4 (2007), pp. 589–605.

DOCTORAT BRETAGNE LOIRE / MATHSTIC

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

**Titre : Bandits Massifs Multi-Bras Multi-Joueurs pour les Réseaux de l'Internet des Objets**

**Mot clés :** Internet des Objects, Apprentissage par renforcement, Bandit multi-joueurs multi-bras

**Résumé :** Cette thèse de doctorat étudie le problème d'optimisation de la performance des réseaux de l'Internet des objets (IoT). L'objectif est de maximiser le succès des transmissions dans les réseaux de l'IoT, en proposant des algorithmes de prise de décision dynamiques efficaces pouvant être intégrés dans les futurs équipements IoT, tout en respectant leurs contraintes de faible complexité et de faible consommation d'énergie. Pour cela, l'apprentissage par renforcement (RL) est utilisé et le problème d'optimisation est modélisé comme un problème de bandit multi-joueurs multi-bras (MP-MAB), adapté aux réseaux IoT et permettant de surmonter de nombreuses hypothèses irréalistes dans le cadre des réseaux IoT précédemment effectuées dans la littérature. Dans cette thèse, deux approches différentes sont proposées pour traiter le problème d'optimisation. La première approche permet de blacklister les mauvais canaux de propagation d'un réseau en utilisant un algorithme collaboratif d'identification des meilleurs bras. La seconde approche consiste en deux politiques différentes qui attribuent de manière récursive chaque équipement IoT à un canal ; la première politique se concentre sur le nombre de transmissions réussies tandis que l'autre garantit un niveau d'équité entre les équipements. Dans un premier temps, nous avons effectué l'étude numérique et expérimentale des différents algorithmes développés pendant cette thèse afin de montrer qu'ils étaient capables de surclasser les autres algorithmes de la littérature. Dans un second temps, une partie importante du travail a consisté en l'application des algorithmes développés au problème concret de choix de la puissance d'émission et du facteur d'étalement dans un réseau LoRa, en analysant les performances en termes de qualité de service et de consommation d'énergie à l'aide d'un simulateur de réseau LoRa réaliste entièrement redéveloppé en C durant la thèse.

**Title: Multi-Player Multi-Armed Bandits for Internet of Things Networks**

**Keywords:** Internet of Things, Reinforcement Leaning, Multi-player multi-armed bandits

**Abstract:** This PhD thesis studies the optimization problem of Internet of Things (IoT) networks performance. We aim to maximize the successful transmissions in IoT networks, by proposing efficient dynamic decision-making algorithms that can be embedded in future IoT devices, while respecting the low complexity and low energy consumption constraints in IoT devices. For this sake, we use Reinforcement Learning (RL), and we model the optimization problem as a massive multi-player multi-armed bandit (MP-MAB) problem to best suit IoT networks, while overcoming many unrealistic assumptions previously made in the literature. In this manuscript, we propose two different approaches to handle the optimization problem. The first blacklists bad channels after a collaborative best-arms identification algorithm. The second consists of two different policies that recursively assign each device to one channel; where one policy focuses on the number of successful transmissions while the other guarantees a level of fairness between the devices. We provide both numerical and experimental studies of our developed algorithms, and show their out-performance over other algorithms proposed in the literature. Furthermore, we test our algorithms using a realistic LoRa network simulator entirely redeveloped in C during the thesis, and show the gain they achieve in terms of both successful transmissions and energy consumption compared to other already implemented algorithms.