Sequential Learning in a strategical environment

Etienne Boursier

université
PARIS-SACLAY

Sequential Learning in a strategical environment

Thèse de doctorat de l'Université Paris-Saclay

Ecole Doctorale de Mathématique Hadamard (EDMH) n° 574
Spécialité de doctorat : Mathématiques appliquées
Unité de recherche : Centre Borelli (ENS Paris-Saclay), UMR 9010 CNRS
Référent : Ecole normale supérieure de Paris-Saclay

Thèse présentée et soutenue à Gif-sur-Yvette, le 30/09/2021, par

Etienne BOURSIER

Au vu des rapports de :

Alexandre Proutière Professeur, KTH	Rapporteur
Nicolas Vieille Professeur, HEC	Rapporteur

Composition du jury :

Nicolas Vieille Professeur, HEC	Président
Alexandre Proutière Professeur, KTH	Rapporteur
Sebastien Bubeck Directeur de recherche, Microsoft Research	Examinateur
Richard Combes Professeur Assistant, Centrale Supelec L2S	Examinateur
Shie Mannor Professeur, Technion	Examinateur
Lucie Ménager Professeure, Université Paris 2	Examinatrice
Vianney Perchet Professeur, CREST, ENSAE	Directeur
Marco Scarsini Professeur, LUISS	Invité

Thèse de doctorat

NNT: 2021UPASM034


école
normale
supérieure
paris–saclay


université
PARIS-SACLAY


CENTRE
BORELLI


Fondation mathématique
FMJH
Jacques Hadamard

Remerciements

Tout d'abord, je tiens à remercier mon directeur de thèse, Vianney Perchet, qui m'a encadré dès mon stage de fin d'études, et ce jusqu'à la fin de ma thèse, à mon plus grand plaisir. Tu as toujours su me rebooster lorsque cela était nécessaire et susciter mon intérêt sur de nouveaux problèmes, et ce dès notre première rencontre lorsque nous parlions déjà de multiplayer bandits et dilemme du prisonnier. Tu m'as également encouragé à participer à différents séminaires/conférences et discuter avec de nombreux chercheurs, ce qui m'a amené à faire de nombreuses rencontres positives. Ta disponibilité et ton attention m'ont permis d'entrer dans le monde de la recherche avec bienveillance et décontraction. Beaucoup de souvenirs resteront en mémoire dont de longues séances au tableau, mais aussi les petits plaisirs comme les Pisco sour *cathedral* à Lima ou lorsque nous nous plaignions mutuellement de certaines reviews.

Je remercie aussi Alexandre Proutière et Nicolas Vieille qui ont gentiment accepté d'être rapporteurs pour cette thèse. Je ne vous ai pas facilité la tâche avec plus de 250 pages de lecture pour l'été; vos commentaires éclairés ont clairement permis d'améliorer cette thèse. De plus, je tiens à remercier Sebastien Bubeck, Richard Combes, Shie Mannor et Lucie Ménager pour avoir accepté d'évaluer ma thèse. Vous compter au sein de mon jury de thèse est un honneur, tant vos divers travaux ont pu influencer ma thèse et m'influenceront dans le futur.

J'ai eu la chance de collaborer à plusieurs reprises avec Marco Scarsini. Travailler avec toi fut un réel plaisir, grâce notamment à ta bonne humeur et ton optimisme constants. Malgré plusieurs tentatives infructueuses, j'espère avoir la chance de te rendre un jour visite à Rome.

Lors de ma thèse, j'ai eu la chance de travailler avec de nombreux autres chercheurs. Merci à Emilie Kaufmann et Abbas Mehrabian qui ont accepté volontiers mon apport et mes suggestions sur un travail déjà bien abouti. Au delà des riches interactions que nous avons pu avoir au labo, dans les group meetings ou en dehors, je remercie aussi Pierre Perrault et Flore Sentenac pour ces longues séances de réflexion toujours aussi intéressantes. Merci à Michal Valko pour les divers échanges que nous avons pu avoir aux quatre coins du monde.

Malgré les restrictions sanitaires qui nous ont tenus à distance, j'ai rencontré de nombreuses personnes géniales à l'ENS Paris-Saclay (anciennement Cachan). Merci donc à Matthieu, Mathilde,

Firas, Alice, Antoine, Pierre P., Xavier, Tristan, Rémy, Pierre H., Batiste, Marie, Ludovic, Amir, Dimitri, Guillaume, Ioannis, Théo, Tina, Sylvain et ceux que j'oublie. Un merci particulier à Myrto, *co-bureau* de toujours.

J'ai aussi échangé avec de nombreuses personnes en conférence lorsque celles-ci étaient encore en présentiel. Je ne peux malheureusement pas toutes les citer, mais je salue en particulier Lilian, Mario, Joon, Thomas, Claire et Quentin. C'est aussi à NeurIPS que j'ai rencontré Nicolas Flammarion, avec qui je commence aujourd'hui mon post-doc. Merci de m'offrir cette chance.

Merci également à Alain Durmus, Alain Trouvé et Frédéric Pascal pour m'avoir permis d'enseigner dans leurs cours: ce fut très formateur (et parfois une bonne piqûre de rappel personnelle).

Je remercie aussi tout le personnel administratif et en particulier Virginie, Véronique et Alina qui m'ont, entre autres, permis d'assister aux différentes conférences et summer schools.

Une pensée particulière pour mes proches: mes parents et mes frères pour m'avoir supporté ces 26 années et accompagné dans cette aventure; ainsi que tous mes amis[1] pour les moments de pressions (au Gobelet) et de décompression. Être si bien entouré est une chance que je chéris.

Je m'excuse d'avance auprès de ceux que j'aurais pu oublier, ma mémoire est malheureusement faillible.

Pour finir, merci à toi Zineb pour tout ce que tu m'apportes depuis tant d'années. T'avoir au quotidien auprès de moi est un privilège.

[1] La liste est bien trop longue pour vous citer, mais vous vous reconnaîtrez.

Abstract

In sequential learning (or repeated games), data is acquired and treated on the fly and an algorithm (or strategy) learns to behave as well as if it got in hindsight the state of nature, e.g., distributions of rewards. In many real life scenarios, learning agents are not alone and interact, or interfere, with many others. As a consequence, their decisions have an impact on the others and, by extension, on the generating process of rewards. We study how sequential learning algorithms behave in strategic environments, when facing and interfering with each other. This thesis considers different problems, where interactions between learning agents arise and it proposes computationally efficient algorithms with good performance (small regret) guarantees for these problems.

When agents are cooperative, the difficulty of the problem comes from its decentralized aspect, as the different agents take decisions solely based on their observations. In this case, we propose algorithms that not only coordinate the agents to avoid negative interference with each other, but also leverage the interferences to transfer information between the agents, thus reaching performances similar to centralized algorithms. With competing agents, we propose algorithms with both satisfying performance and strategic (e.g., ε-Nash equilibria) guarantees.

This thesis mainly focuses on the problem of multiplayer bandits, which combines different connections between learning agents in a formalized online learning framework. Both for the cooperative and competing case, algorithms with performances comparable to the centralized case are proposed. Other sequential learning instances involving multiple agents are also considered in this thesis. We propose a strategy reaching centralized performances for decentralized queuing systems. In online auctions, we suggest to balance short and long term rewards with a utility/privacy trade-off. It is formalized as an optimization problem, that is equivalent to Sinkhorn divergence and benefits from the recent advances on Optimal Transport. We also study social learning with reviews, when the quality of the product varies over time.

Contents

Chapter 1

Introduction (version française)

1.1 Apprentissage en jeux répétés

Les jeux répétés formalisent les différentes interactions se produisant entre des joueurs (ou agents) participant à un jeu de manière répétée, à l'aide d'outils de théorie des jeux (Aumann et al., 1995; Fudenberg and Maskin, 2009). De nombreuses applications motivent ce type de problème, dont les enchères pour les publicités en ligne, l'optimisation du trafic dans des réseaux de transport, etc. Face à la recrudescence d'algorithmes d'apprentissage dans notre société, il est crucial de comprendre comment ceux-ci intéragissent. Alors que les paradigmes classiques d'apprentissage considèrent un seul agent dans un environnement fixe, cette hypothèse semble erronée dans de nombreuses applications modernes. Des agents *intelligents*, qui sont stratégiques et apprennent leur environnement, en effet intéragissent entre eux, influençant largement l'issue finale. Cette thèse explore différentes interactions possibles entre des agents intelligents dans un environnement stratégique et décrit les stratégies qui mènent typiquement à de bonnes performances dans ces configurations. Aussi, elle quantifie les différentes inefficiences en bien-être social qui résultent à la fois des considérations stratégiques, et d'apprentissage.

Les jeux répétés sont généralement formalisés comme suit. À chaque tour $t \in [T] \coloneqq \{1, \dots, T\}$, chaque joueur $m \in [M]$ choisit individuellement une stratégie (un montant d'enchère par exemple) $s^m \in \mathcal{S}^m$ où $\mathcal{S}^m$ est l'espace de stratégie. Le joueur m reçoit alors le gain (possiblement bruité) d'espérance $u_t^m(\boldsymbol{s})$, où u_t^m est sa fonction de gain associée à l'instant t et $\boldsymbol{s} \in \prod_{m=1}^M \mathcal{S}^m$ est le profil stratégique de l'ensemble des joueurs. Dans la suite de cette thèse, $\boldsymbol{s^{-m}}$ représente le vector $\boldsymbol{s}$ privé de sa m-ième composante.

Un joueur apprenant choisit à chaque nouveau tour sa stratégie, en fonction des ses précédentes observations. Celles-ci peuvent en effet permettre d'estimer l'environnement du jeu, c'est à dire les fonctions d'utilité u_t^m, ainsi que le profil de stratégie des autres joueurs $\boldsymbol{s^{-m}}$.

Maximiser son propre gain dans un environnement fixé à un joueur est au cœur des théories d'apprentissage et d'optimisation. Cela devient encore plus délicat lorsque plusieurs joueurs intéragissent entre eux dans des jeux répétés. Deux types d'interaction majeures entre ces joueurs sont possibles. Premièrement, le gain d'un joueur à chaque tour ne dépend pas seulement de sa propre action, mais également des actions des autres agents, et même potentiellement des issues des tours précédents. Dans ce cas, les joueurs peuvent soit rivaliser, ou bien coopérer, selon la nature du jeu. Deuxièmement, les joueurs peuvent aussi partager (dans une certaine mesure) leurs observations entre eux, influençant leur estimation de l'environnement du jeu. Cela peut soit accélérer l'apprentissage, ou biaiser l'estimation des différents paramètres.

Interaction dans les gains. Généralement, les fonctions de gain u_t dépendent du profil complet de stratégie des joueurs $\boldsymbol{s}$. Les objectifs des différents joueurs peuvent alors être antagonistes, puisqu'un profil donnant un gain conséquent à un certain joueur peut mener à des gains infimes pour un autre joueur. Le cas extrême correspond aux jeux à somme nulle pour deux joueurs, où les fonctions d'utilité vérifient $u^1 = -u^2$. Dans ce cas, les joueurs rivalisent entre eux et tentent de maximiser leurs gains individuels. Dans un jeu à un seul tour (non répété), les équilibres de Nash caractérisent des profils de stratégie intéressants pour des joueurs stratégiques. Un joueur déviant unilatéralement d'un équilibre de Nash subit, par définition, une diminution de gain.

Définition 1.1 (Equilibre de Nash)**.** *Un profil de stratégie $\boldsymbol{s}$ est un équilibre de Nash pour le jeu à un tour défini par les fonctions d'utilité $(u^m)_{m\in[M]}$ si*

$$\forall m \in [M], \forall s' \in \mathcal{S}^m, u^m(s', \boldsymbol{s^{-m}}) \leq u^m(s^m, \boldsymbol{s^{-m}}).$$

Dès lors ques les fonctions d'utilité u^m sont concaves et continues, l'existence d'un équilibre de Nash est garantie par le théorème de point fixe de Brouwer. C'est par exemple le cas si $\mathcal{S}^m$

est l'ensemble des distributions de probabilité sur un ensemble fini (qui est appelé **ensemble d'action** dans la suite).

Cette première considération stratégique mène à une première inefficience dans les décisions des joueurs, puisqu'ils maximisent leur gain individuel, au détriment du gain collectif. Le *prix de l'anarchie* (Koutsoupias and Papadimitriou, 1999) mesure cette inefficience comme le ratio de bien-être social entre la meilleure situation collective possible et le pire équilibre de Nash. Bien qu'atteindre la meilleure situation collective semble illusoire pour des agents égoïstes, considérer le pire équilibre de Nash peut être trop pessimiste. Le *prix de la stabilité* mesure plutôt cette inefficience comme le ratio de bien-être social entre la meilleure situation possible et le meilleur équilibre de Nash.

Apprendre les équilibres de jeux répétés est donc crucial, puisqu'ils reflètent le comportement des agents connaissant parfaitement leur environnement. En particulier, c'est au cœur de nombreux problèmes en informatique et en économie (Fudenberg et al., 1998; Cesa-Bianchi and Lugosi, 2006). Une seconde inefficience vient de cette considération, puisque les joueurs doivent apprendre leur environnement et peuvent interférer l'un avec l'autre, ne convergeant potentiellement pas ou vers un mauvais équilibre. Les équilibres corrélés sont définis similairement aux équilibres de Nash, lorsque les stratégies $(s^m)_m$ sont des distributions de probabilité dont les réalisations jointes peuvent être corrélées. Il est connu que lorsque les fonctions d'utilité sont constantes dans le temps $u_t^m = u^m$, si tous les agents suivent des stratégies sans regret interne, leurs actions convergent en moyenne vers l'ensemble des équilibres corrélées (Hart and Mas-Colell, 2000; Blum and Monsour, 2007; Perchet, 2014). Cependant, on en sait beaucoup moins lorsque les fonctions d'utilité u_t^m dépendent aussi des issues des tours précédents, comme dans le cas des systèmes de queues décentralisés, étudié dans le Chapitre 7.

De plus, déterminer un équilibre de Nash peut-être trop coûteux en pratique (Daskalakis et al., 2009). C'est même le cas dans des jeux à somme nulle à deux joueurs, quand l'ensemble d'action est continu. Par exemple dans le cas d'enchères répétées, une action d'enchère est une fonction $\mathbb{R}_+ \to \mathbb{R}_+$ qui à chaque valeur d'objet associe un montant d'enchère. Apprendre les équilibres dans ce type de jeu semble alors déraisonnable et répondre de manière optimale à la stratégie de l'adversaire peut mener à une course à l'armement sans fin entre les joueurs. Nous proposons à la place au Chapitre 8 d'équilibrer entre le revenu à court terme obtenu en misant de manière avide, et le revenu à long terme en maintenant une certaine asymétrie d'informations entre les joueurs, qui est un aspect crucial des jeux répétés (Aumann et al., 1995).

Dans d'autres cas (par exemple l'allocation de ressources pour des réseaux radios ou informatiques), les joueurs ont intérêt à coopérer entre eux. C'est par exemple le cas si les joueurs répartissent équitablement le gain collectif entre eux, ou s'ils ont des intérêts communs en raison

des fonctions d'utilité (considèrez par exemple un jeu avec un prix d'anarchie égal à 1).

Dans les bandits à plusieurs joueurs, qui est l'axe de la Partie I, les joueurs choisissent un canal de transmission. Mais si certains joueurs utilisent le même canal à un certain instant, une *collision* se produit et aucune transmission n'est possible sur ce canal. Dans ce cas, les joueurs ont intérêt à se coordonner entre eux pour éviter les collisions et efficacement transmettre sur les différents canaux. En plus d'apprendre l'environnement du jeu, la difficulté vient aussi de la coordination entre les joueurs, tout en étant décentralisés et ayant une communication limitée, voire impossible. Lorsque les tours sont répétés, il devient cependant incertain si les joueurs ont réellement intérêt à coopérer aveuglément. En particulier, un joueur pourrait avoir intérêt à perturber le processus d'apprentissage des autres joueurs pour s'accorder le meilleur canal de transmission. Ce type de comportement peut malgré tout être prévenu, comme montré dans le Chapitre 6, en utilisant par exemple des stratégies punitives.

La coopération entre les joueurs semble encore plus encouragée dans les systèmes de queues décentralisés. Dans ce problème, les fonctions d'utilité dépendent aussi des issues des tours précédents. Leur conception assure que si un joueur a accumulé un plus petit gain que les autres joueurs jusqu'ici, il devient alors favorisé dans le futur et a la priorité sur les autres joueurs lorsqu'il accède à un serveur. Par conséquent, les joueurs ont aussi intérêt à partager les ressources entre eux, afin de ne pas dégrader leurs propres gains futurs.

Interaction dans les observations. Même lorsque les fonctions d'utilité ne dépendent pas des actions des autres joueurs, i.e. u_t^m ne dépend que de s^m, les joueurs peuvent intéragir en partageant des informations/observations entre eux. Dans ce cas, les joueurs n'ont pas intérêt à être compétitifs et ils partagent leurs informations uniquement pour que tous puissent apprendre plus vite l'environnement du jeu. Un tel phénomène apparaît par exemple dans le cas de bandits distribués, décrit en Section 3.6.1. Ce problème est similaire aux bandits à plusieurs joueurs, à l'exception de deux différences: il n'y a pas de collision ici, comme les fonctions d'utilité ne dépendent pas des actions des autre joueurs; et les joueurs sont assignés à un graphe et peuvent envoyer des messages à leurs voisins dans ce graphe. Ils peuvent donc envoyer leurs observations (ou une agrégation de ces observations) à leurs voisins, ce qui permet d'accélérer le processus d'apprentissage.

Même dans le cas général de jeux où les fonctions d'utilité dépendent du profil de stratégie complet $\boldsymbol{s}$, les joueurs coopératifs peuvent partager certaines informations afin d'accélérer l'apprentissage. C'est typiquement ce qui nous permet d'atteindre une performance quasi-centralisée dans le problème de bandits à plusieurs joueurs dans les Chapitres 4, 5 et 6.

Lorsque les joueurs coopèrent, le but est généralement de maximiser le revenu collectif. Comme expliqué ci-dessus, une inefficience d'apprentissage peut alors apparaître en raison des

différentes interactions entre les joueurs. Lorsqu'ils sont centralisés, c'est à dire qu'un agent central contrôle unilatéralement les décisions des autres joueurs, le problème est équivalent à un cas à un seul joueur et cette inefficience vient simplement de la difficulté d'apprentissage du problème. Mais lorsque les joueurs sont décentralisés, i.e. leurs décisions sont prises individuellement sans se concerter avec les autres, des difficultés supplémentaires apparaissent. Par exemple, les observations/décisions ne peuvent être mutualisées. Le but principal dans ces situations est alors de savoir si cette décentralisation apporte un coût supplémentaire, c'est à dire si le meilleur bien-être social possible dans le cas décentralisé est plus petit que dans le cas centralisé. C'est en particulier l'objectif des Chapitres 4 et 7, qui montrent que la décentralisation n'a globalement pas de coût, respectivement pour les problèmes de bandits à plusieurs joueurs homogènes et les systèmes séquentiels de queues. Le Chapitre 5 suggère également que ce coût est au maximum de l'ordre du nombre de joueurs pour le problème de bandits à plusieurs joueurs hétérogènes.

L'apprentissage social considère un problème différent de jeux répétés, où à chaque tour, un seul nouveau joueur ne joue que pour ce tour. Il choisit son action afin de maximiser son revenu espéré, en se basant sur les actions des précédents joueurs (et potentiellement un retour supplémentaire). Des comportements dits "de troupeau" peuvent alors se produire, où les agents n'apprennent jamais correctement leur environnement et finissent par prendre des décisions sous-optimales pour toujours. Ce type de problème illustre donc habilement comment des agents peuvent prendre des décisions optimales à court terme, menant à de très mauvaises situations collectives. Le Chapitre 9 montre à l'inverse que cette inefficience d'apprentissage est largement réduite lorsque les joueurs observent les revues des précédents consommateurs.

1.2 Bandits stochastiques à plusieurs bras

Les problèmes étudiés dans cette thèse sont complexes, puisqu'ils combinent des considérations d'apprentissage et de théorie des jeux. Le cadre d'apprentissage séquentiel et tout particulièrement de **Bandits à plusieurs bras** semble parfaitement adapté. Tout d'abord, il définit un problème formel et relativement simple d'apprentissage, pour lequel des résultats théoriques sont connus. De plus, son aspect séquentiel est similaire aux jeux répétés, et de nombreuses connexions existent entre les jeux répétés et les bandits (voir par exemple Cesa-Bianchi and Lugosi, 2006). Le problème de bandits est effectivement un cas particulier de jeux répétés, où un seul joueur joue contre la nature, qui génère les revenus de chaque bras.

Les bandits ont d'abord été introduits pour les essais cliniques (Thompson, 1933; Robbins, 1952) et ont été récemment popularisés pour ses applications aux systèmes de recommandation

en ligne. De nombreuses variations ont également été développées ces dernières années, incluant les bandits contextuels, combinatoriaux ou lipschitziens par exemple (Woodroofe, 1979; Cesa-Bianchi and Lugosi, 2012; Agrawal, 1995).

Cette section décrit rapidement le problème de bandits stochastiques, ainsi que les résultats et algorithmes principaux pour ce problème classique. Ceux-ci inspireront les algorithmes et résultats proposés tout au long de cette thèse. Nous renvoyons le lecteur à (Bubeck and Cesa-Bianchi, 2012; Lattimore and Szepesvári, 2018; Slivkins, 2019) pour des revues complètes des bandits.

1.2.1 Modèle et bornes inférieures

À chaque instant $t \in [T]$, l'agent tire un bras $\pi(t) \in [K]$ parmi un ensemble fini d'actions, où T est l'horizon du jeu. Lorsqu'il tire le bras k, il observe et reçoit le gain $X_k(t) \sim \nu_k$ de moyenne $\mu_k = \mathbb{E}[X_k(t)]$, où $\nu_k \in \mathcal{P}([0,1])$ est une distribution de probabilité sur $[0,1]$. Cette observation $X_k(t)$ est alors utilisée par l'agent pour choisir le bras à tirer aux prochains tours.

Les variables aléatoires $(X_k(t))_{t=1,\dots,T}$ sont indépendantes, identiquement distribuées et bornées dans $[0,1]$ dans la suite. Cependant, les résultats présentés dans cette section sont aussi valides dans le cas plus général de variables sous-gaussiennes.

Dans la suite, $x_{(k)}$ désigne la k-ième statistique ordonnée du vecteur $\boldsymbol{x} \in \mathbb{R}^n$, i.e., $x_{(1)} \geq x_{(2)} \geq \ldots \geq x_{(n)}$. Le but de l'agent est de maximiser son revenu cumulé. De manière équivalente, il minimise son regret, défini comme la différence entre le revenu maximal espéré obtenu par un agent connaissant *a priori* les distributions des bras et le revenu réellement accumulé par l'agent jusqu'à l'horizon T. Formellement, le regret est défini par

$$R(T) = T\mu_{(1)} - \mathbb{E}\left[\sum_{t=1}^{T} \mu_{\pi(t)}\right],$$

où l'espérance est sur les actions $\pi(t)$ de l'agent.

Le joueur n'observe que le gain $X_k(t)$ du bras tiré et pas ceux associés aux bras non-tirés. À cause de ce retour dit "bandit", le joueur doit équilibrer entre l'**exploration**, c'est à dire estimer les moyennes des bras en les tirant tous sufisamment, et l'**exploitation**, en tirant le bras qui apparaît comme optimal. Ce compromis est au cœur des problèmes de bandits et est aussi crucial dans les jeux répétés, comme il oppose élégamment revenus à court terme (exploitation) et long terme (exploration).

Une configuration de problème est fixée par les distributions $(\nu_k)_{k\in[K]}$.

Definition 1.1. *Un agent (ou algorithme) est asymptotiquement fiable si pour toute configuration de problème et $\alpha > 0$, $R(T) = o(T^\alpha)$.*

Le revenu cumulé est de l'ordre de $\mu_{(1)}T$ pour un algorithme asymptotiquement fiable. Le regret est alors un choix de mesure plus fin, puisqu'il capture le terme du deuxième ordre du revenu cumulé dans ce cas.

Déterminer le plus petit regret atteignable est une question fondamentale du problème de bandits. Tout d'abord, Théorème 1.1 borne inférieurement le regret atteignable dans le problème de bandits stochastiques classique.

Théorème 1.1 (Lai and Robbins 1985)**.** *Considérons une configuration de problème avec* $\nu_k = \mathrm{Bernoulli}(\mu_k)$*. Alors, tout algorithme asymptotiquement fiable a un regret asymptotique borné comme suit*

$$\liminf_{T\to\infty} \frac{R(T)}{\log(T)} \geq \sum_{k:\mu_k<\mu_{(1)}} \frac{\mu_{(1)}-\mu_k}{\mathrm{kl}\left(\mu_{(1)},\mu_k\right)},$$

où $\mathrm{kl}\,(p,q) = p\log\left(\frac{p}{q}\right) + (1-p)\log\left(\frac{1-p}{1-q}\right)$.

Une borne inférieure similaire existe pour des distributions générales ν_k, mais cette version plus simple suffit à notre propos. La borne inférieure ci-dessus est asymptotique pour une configuration fixée et est dite configuration-dépendante. Cependant, le regret maximal à l'instant T sur toutes les configurations possibles peut toujours être linéaire en T. Cela correspond au pire cas, où la configuration considérée est la pire, pour l'horizon fini fixé égal à T. Lorsque l'on fait référence à cette quantité, on parle alors de regret minimax, qui est borné inférieurement comme suit.

Théorème 1.2 (Auer et al. 1995)**.** *Pour tous les algorithmes et horizons* $T \in \mathbb{N}$*, il existe toujours une configuration telle que*

$$R(T) \geq \frac{\sqrt{KT}}{20}.$$

1.2.2 Algorithmes de bandits classiques

Cette section décrit les algorithmes de bandits classiques suivants: ε-greedy, Upper Confidence Bound (UCB), Thompson Sampling et Explore-then-commit (ETC). La plupart des algorithmes dans le reste de la thèse sont inspirés de ceux-ci, comme ils sont relativement simples et offrent de bonnes performances. Des bornes supérieures de leur regret sont données sans preuve; elles s'appuient principalement sur l'inégalité de concentration suivante, qui permet de borner l'erreur d'estimation de la moyenne empirique d'un bras.

Lemme 1.1 (Hoeffding 1963)**.** *Pour des variables aléatoires indépendantes* $(X_s)_{s\in\mathbb{N}}$ *dans* $[0,1]$*:*

$$\mathbb{P}\left(\frac{1}{n}\sum_{s=1}^{n} X_s - \mathbb{E}[X_s] \geq \varepsilon\right) \leq e^{-2n\varepsilon^2}.$$

Les notations suivantes sont utilisées dans le reste de la section:

- $N_k(t) = \sum_{s=1}^{t-1} \mathbb{1}(\pi(s) = k)$ est le nombre de tirages du bras k jusqu'à l'instant t;
- $\hat{\mu}_k(t) = \frac{\sum_{s=1}^{t-1} \mathbb{1}(\pi(s)=k) X_k(t)}{N_k(t)}$ est la moyenne empirique du bras k avant l'instant t;
- $\Delta = \min\{\mu_{(1)} - \mu_k > 0 \mid k \in [K]\}$ est l'écart de sous-optimalité et représente la difficulté du problème.

Algorithme ε-greedy

L'algorithme ε-greedy décrit par Algorithme 1.1 est définie par une suite $(\varepsilon_t)_t \in [0,1]^{\mathbb{N}}$. Chaque bras est d'abord tiré une fois. Ensuite à chaque tour t, l'algorithme explore avec probabilité ε_t, auquel cas un bras est aléatoirement de manière uniforme. Sinon, l'algorithme exploite, i.e., le bras avec la plus grande moyenne empirique est tiré.

Algorithme 1.1: ε-greedy

Entrées: $(\varepsilon_t)_t \in [0,1]^{\mathbb{N}}$

pour $t = 1, \dots, K$ **faire** tirer le bras t

pour $t = K+1, \dots, T$ **faire** $\begin{cases} \text{tirer } k \sim \mathcal{U}([K]) \text{ avec probabilité } \varepsilon_t; \\ \text{tirer } k \in \arg\max_{i \in [K]} \hat{\mu}_i(t) \text{ sinon.} \end{cases}$

Quand $\varepsilon_t = 0$ pour tout t, l'algorithme est appelé greedy (ou glouton), puisqu'il tire toujours de manière "gloutonne" le meilleur bras empirique. L'algorithme greedy entraîne généralement un regret de l'ordre de T, comme le meilleur bras peut-être sous-estimé dès son premier tirage et n'est alors plus tiré.

En choisissant une suite (ε_t) appropriée, on obtient alors un regret sous-linéaire, comme donné par le Théorème 1.3.

Théorème 1.3 (Slivkins 2019, Théorème 1.4)**.** *Pour une certaine constante universelle positive c_0, l'algorithme ε-greedy avec probabilités d'exploration $\varepsilon_t = \left(\frac{K \log(t)}{t}\right)^{1/3}$ a un regret borné par*

$$R(T) \leq c_0 K \log(T)^{1/3} T^{2/3}.$$

Si l'écart de sous-optimalité $\Delta = \min\{\mu_{(1)} - \mu_k > 0 \mid k \in [K]\}$ est connu, la suite $\varepsilon_t = \min(1, \frac{CK}{\Delta^2 t})$ pour une constante suffisamment large C donne un regret configuration-dépendant logarithmique en T.

Algorithme UCB

Comme expliqué ci-dessus, choisir naïvement le meilleur bras empirique entraîne un regret considérable. Contrairement à greedy, l'algorithme UCB choisit le bras k maximisant $\hat{\mu}_k(t)+B_k(t)$ à chaque instant, où le terme $B_k(t)$ est une certaine borne de confiance. UCB, donné par l'Algorithme 1.2 ci-dessous, biaise donc positivement les estimées des moyennes des bras. Grâce à cela, le meilleur bras ne peut être sous-estimée (avec grande probabilité), évitant donc les situations d'échec de l'algorithme greedy décrites ci-dessus.

Algorithme 1.2: UCB

pour $t = 1, \dots, K$ **faire** tirer le bras t
pour $t = K+1, \dots, T$ **faire** tirer $k \in \arg\max_{i\in[K]} \hat{\mu}_i(t) + B_i(t)$

Théorème 1.4 borne le regret de l'algorithme UCB avec son choix de borne de confiance le plus commun.

Théorème 1.4 (Auer et al. 2002a)**.** *L'algorithme UCB avec* $B_i(t) = \sqrt{\frac{2\log(t)}{N_i(t)}}$ *verifie les bornes de regret configuration-dépendante et minimax suivantes, pour certaines constantes universelles positives* c_1, c_2

$$R(T) \leq \sum_{k:\mu_k<\mu_{(1)}} \frac{8\log(T)}{\mu_{(1)} - \mu_k} + c_1, \tag{1.1}$$

$$R(T) \leq c_2\sqrt{KT\log(T)}.$$

L'algorithme UCB a donc un regret configuration-dépendant optimal, à une constante multiplicative près, et lorsque les moyennes des bras ne sont pas arbitrairement proches de 0 ou 1. En utilisant des bornes de confiance plus fines, un regret configuration-dépendant optimal est en fait possible pour UCB (Garivier and Cappé, 2011). Dans la suite de cette thèse, une borne similaire à l'Équation (1.1) est dite optimale à un facteur constant près par abus de notation.

Algorithme Thompson sampling

L'algorithme Thompson sampling décrit par Algorithme 1.3 adopte un point de vue Bayésien. Pour une distribution *a posteriori* $\boldsymbol{p}$ des moyennes des bras $\boldsymbol{\mu}$, il échantillonne aléatoirement un vecteur $\boldsymbol{\theta} \sim \boldsymbol{p}$ et choisit un bras dans $\arg\max_{k\in[K]} \theta_k$. La distribution *a posteriori* est alors mise à jour en utilisant le gain observé, selon la règle de Bayes.

Théorème 1.5 (Kaufmann et al. 2012)**.** *Il existe une fonction* f*, dépendant uniquement du vecteur des moyennes* $\boldsymbol{\mu}$ *telle que pour toute configuration et* $\varepsilon > 0$*, le regret de l'algorithme*

Algorithme 1.3: Thompson sampling

$\boldsymbol{p} = \otimes_{k=1}^{K} \mathcal{U}([0,1])$ // Uniforme *a priori*
pour $t = 1, \dots, T$ **faire**
  Échantillonner $\boldsymbol{\theta} \sim \boldsymbol{p}$
  Tirer $k \in \arg\max_{k \in [K]} \theta_k$
  Mettre à jour p_k comme la distribution *a posteriori* de μ_k
fin

Thompson sampling est borné comme suit

$$R(T) \leq (1+\varepsilon) \sum_{k:\mu_k < \mu_{(1)}} \frac{\mu_{(1)} - \mu_k}{\mathrm{kl}\left(\mu_k, \mu_{(1)}\right)} \log(T) + \frac{f(\boldsymbol{\mu})}{\varepsilon^2}.$$

Bien qu'il vienne d'un point de vue Bayésien, Thompson sampling atteint des performances fréquentistes optimales, lorsqu'il est initialisé avec une distribution uniforme *a priori*. La preuve de cette borne supérieure est délicate. Échantillonner selon la distribution *a posteriori* $\boldsymbol{p}$ peut être coûteux en terme de calcul à chaque tour. Cependant, dans certains cas comme des gains binaires ou gaussiens, la mise à jour et l'échantillonnage de la distribution *a posteriori* est très simple. Dans le cas général, une substitution de la distribution *a posteriori* peut être utilisée, à partir des cas binaires et gaussiens. L'intérêt de ce type d'algorithmes pour les bandits combinatoriaux est illustré par Perrault et al. (2020), bien que ce travail n'est pas discuté dans cette thèse.

Algorithme Explore-then-commit

Alors que les algorithmes ci-dessus combinent exploration et exploitation à chaque instant, l'algorithme ETC sépare clairement les deux en phases distinctes. D'abord, tous les bras sont explorés. Seulement une fois que le meilleur bras est détecté (avec grande probabilité), l'algorithme commence sa phase d'exploitation et tire ce bras jusqu'à l'horizon final T.

Séparer de manière distincte exploration et exploitation entraîne un plus grand regret. En particulier, si tous les bras sont explorés le même nombre de fois (exploration uniforme), la borne configuration-dépendante croît en $\frac{1}{\Delta^2}$.

Pour remédier à cela, l'exploration est adaptée à chaque bras comme décrit dans Algorithme 2.4. Cette version plus fine de l'algorithme ETC est appelée *éliminations successives* (Perchet and Rigollet, 2013). Un bras k est *éliminé* lorsqu'il est détecté comme sous-optimal, c'est à dire quand il existe un bras i tel que $\hat{\mu}_k + B_k(T) \leq \hat{\mu}_i - B_i(T)$, pour des bornes de confiances $B_i(t)$. Quand cette condition est vérifiée, le bras k est moins bon que le bras i avec grande probabilité; il n'est alors plus joué. Avec cette exploration adaptative, le regret devient optimal à un facteur près comme donné par Théorème 1.6.

Algorithme 1.4: Éliminations successives

$\mathcal{A} \leftarrow [K]$ `// bras actifs`
tant que $\#\mathcal{A} > 1$ **faire**
  tirer tous les bras dans $\mathcal{A}$ une fois
  pour *tout* $k \in \mathcal{A}$ *tel que* $\hat{\mu}_k + B_k(T) \leq \max_{i \in \mathcal{A}} \hat{\mu}_i - B_i(T)$ **faire** $\mathcal{A} \leftarrow \mathcal{A} \setminus \{k\}$
fin
répéter tirer le seul bras dans $\mathcal{A}$ **jusqu'à** $t = T$

Théorème 1.6 (Perchet and Rigollet 2013)**.** *Algorithme 1.4 avec* $B_i(t) = \sqrt{\frac{2\log(T)}{N_i(t)}}$ *a un regret borné comme suit*

$$R(T) \leq 324 \sum_{k:\mu_k<\mu_{(1)}} \frac{\log(T)}{\mu_{(1)} - \mu_k},$$
$$R(T) \leq 18\sqrt{KT\log(T)}.$$

En plus d'avoir une regret plus large qu'UCB et Thompson sampling (d'un facteur constant), l'algorithme éliminations successives nécessite la connaissance *a priori* de l'horizon T. Connaître l'horizon T n'est pas trop restrictif dans les problèmes de bandits (Degenne and Perchet, 2016a) et cette connaissance est donc supposée dans le reste de cette thèse. D'un autre côté, cet algorithme a l'avantage d'être simple car les phases d'exploration et d'exploitation sont clairement séparées, ce qui sera utile pour le problème de bandits à plusieurs joueurs en Partie I.

1.3 Aperçu et Contributions

Le but de cette thèse est d'étudier les jeux répétés avec des agents apprenant et décentralisés. Pour la majorité des problèmes considérés, le but est de fournir de bonnes stratégies d'apprentissage séquentiel, par exemple des algorithmes avec un faible regret. Pour des raisons pratiques, les calculs faits par ces algorithmes doivent être efficaces, ce qui est assuré et illustré par des expériences numériques dans la plupart des cas.

La formalisation des bandits pour étudier les relations entre plusieurs agents apprenant amène au problème de bandits à plusieurs joueurs, qui est le principal problème de cette thèse et en particulier de la Partie I. La Partie II quant à elle considère différents problèmes indépendants, afin d'explorer les différents types d'interactions qui peuvent intervenir entre des agents apprenant. Le contenu de chaque chapitre est décrit ci-dessous.

Partie I, Multiplayer Bandits

Cette partie s'intéresse au problème de bandits à plusieurs joueurs.

Chapitre 3, Multiplayer bandits: a survey. Ce chapitre présente le problème de bandits à plusieurs joueurs et étudie de manière exhaustive l'état de l'art en bandits à plusieurs joueurs, incluant les Chapitres 4, 5 et 6, ainsi que des travaux ultérieurs par différents auteurs.

Chapitre 4, SIC-MMAB: Synchronisation Involves Communication in Multiplayer Multi-Armed Bandits. Bien que les joueurs soient décentralisés, ils peuvent toujours communiquer implicitement entre eux en utilisant les informations de collision comme des bits. Cette observation est ici exploitée pour proposer un algorithme décentralisé qui renforce les collisions entre les joueurs pour établir une communication entre eux. Un regret similaire aux algorithmes centralisés optimaux est alors atteint. Bien que quasi-optimal en théorie, cet algorithme n'est pas satisfaisant, puisqu'un tel niveau de communication est très coûteux en pratique. Nous suggérons que la formulation usuelle des bandits à plusieurs joueurs mène vers ce type d'algorithme et en particulier l'hypothèse statique, selon laquelle les joueurs commencent et terminent tous le jeu au même moment. Nous étudions ensuite un nouveau problème dynamique et proposons un algorithme avec un regret logarithmique dans ce cas, sans utiliser de communication directe entre les joueurs.

Chapitre 5, A Practical Algorithm for Multiplayer Bandits when Arm Means Vary Among Players. Ce chapitre considère le cas hétérogène, où les moyennes de chaque bras varient selon le joueur. Pour atteindre l'appariement optimal entre joueurs et bras, un niveau minimum de communication est nécessaire entre les joueurs. Ce chapitre propose donc un algorithme efficace pour le cas hétérogène à la fois en terme de regret et de calcul. Cela est réalisé en renforçant les collisions parmi les joueurs et en améliorant le protocole de communication initialement proposé dans le Chapitre 4.

Chapitre 6, Selfish Robustness and Equilibria in Multi-Player Bandits. Alors que la majorité des travaux sur le problème de bandits à plusieurs joueurs supposent des joueurs coopératifs, ce chapitre considère le cas de joueurs stratégiques, maximisant leur revenu individuel cumulé de manière égoïste. Les algorithmes existants ne sont pas adaptés à ce contexte, comme un joueur malveillant peut facilement interférer avec l'exploration des autres joueurs afin de largement augmenter son propre revenu.

Nous proposons donc un premier algorithme, ignorant les collisions après l'initialisation, qui est à la fois un $\mathcal{O}(\log(T))$-équilibre de Nash (robuste aux joueurs égoïstes) et a un regret collectif comparable aux algorithmes non-stratégiques. Lorsque les collisions sont observées, les algorithmes existants peuvent en fait être adaptées en stratégies *Grim-Trigger*, qui sont aussi des $\mathcal{O}(\log(T))$-équilibres de Nash, tout en maintenant des garanties de regret similaires aux

algorithmes coopératifs originaux. Avec des joueurs hétérogènes, l'appariement optimal ne peut plus être atteint et nous minimisons alors une notion adaptée et pertinente de regret.

Partie II, Other learning instances

Cette partie étudie des problèmes indépendants qui illustrent les différents types d'interaction entre des agents apprenant décrits en Section 1.1.

Chapitre 7, Decentralized Learning in Online Queuing Systems. Ce chapitre étudie le problème séquentiel de systèmes de queues, initalement motivé par le routage de paquets dans les réseaux informatiques. Dans ce problème, les queues reçoivent des paquets selon différents taux et envoient répététivement leurs paquets aux serveurs, chacun d'entre eux ne pouvant traiter au plus qu'un seul paquet à la fois. La stabilité du système (i.e., si le nombre de paquets restants est borné) est d'un intérêt vital et est possible dans le cas centralisé dès lors que le ratio entre taux de service et taux d'arrivée est strictement plus grand que 1. Avec des joueurs égoïstes, Gaitonde and Tardos (2020a) ont montré que les queues minimisant leur regret sont stables lorsque ce ratio est plus grand que 2. La minimisation du regret cependant mène à des comportements à court terme et ignore les effets long terme dûs à la propriété de report propre à cet exemple de jeu répété. En revanche, lorsque les joueurs minimisent des coûts à long terme, Gaitonde and Tardos (2020b) ont montré que tous les équilibres de Nash sont stables tant que le ratio des taux est plus grand que $\frac{e}{e-1}$, qui peut alors être vu comme le prix de l'anarchie pour ce jeu. Cependant, le coût d'apprentissage reste inconnu et nous soutenons dans ce chapitre qu'un certain niveau de coopération est nécessaire entre les queues pour garantir la stabilité avec un ratio plus petit que 2 lorsqu'elles apprennent. Par conséquent, nous proposons un algorithme d'apprentissage décentralisé, stable pour tout ratio plus grand que 1, ce qui implique que la décentralisation n'entraîne pas de coût supplémentaire ici.

Chapitre 8, Utility/Privacy Trade-off as Regularized Optimal Transport. Dans les enchères pour la publicité en ligne, le comissaire-priseur et les enchérisseurs sont répététivement en concurrence. Déterminer les équilibres de Nash est ici trop coûteux en terme de calcul, comme les espaces d'action sont continus. S'adapter aux nouvelles stratégies des autres joueurs mène à une course à l'armement entre le comissaire-priseur et les enchérisseurs. À la place, ce chapitre propose d'équilibrer naturellement le revenu à court terme, en maximisant sa propre utilité de manière avide, et le revenu à long terme en cachant certaines informations privées dont la divulgation pourrait être exploitée par les autres joueurs. Ce problème est formalisé par un cadre Bayésien de compromis entre utilité et confidentialité, dont on montre qu'il est équivalent à un

problème de minimisation de divergence de Sinkhorn. Cette équivalence permet de calculer ce minimum efficacement, en utilisant les différents outils développés par les théories de transport optimal et d'optimisation.

Chapitre 9, Social Learning in Non-Stationary Environments. Ce chapitre considère l'apprentissage social avec revues, où des consommateurs hétérogènes et Bayésiens décident l'un après l'autre d'acheter un objet de qualité inconnue, en se basant sur les revues de précédents acheteurs. Les précédents travaux supposent que la qualité de l'objet est constante dans le temps et montrent que son estimée converge vers sa vraie valeur sous de faibles hypothèses. Ici, nous considérons un modèle dynamique où la qualité peut changer par moments. Le coût supplémentaire dû à la structure dynamique se révèle être logarithmique en le taux de changement de la qualité, dans le cas de caractéristiques binaires. Cependant, l'écart entre les modèles statique et dynamique lorsque les caractéristiques ne sont plus binaires demeure inconnu.

Chapter 2

Introduction

2.1 Learning in repeated games

Repeated games formalize the different interactions occurring between players (or agents) repeatedly taking part in a game instance, using game theoretical tools (Aumann et al., 1995; Fudenberg and Maskin, 2009). Many applications derive from this kind of problem, including bidding for online advertisement auctions, resource allocation in radio or computer networks, minimizing travelling time in transportation networks, etc. Facing the surge of learning algorithms in our society, it is of crucial interest to understand how these algorithms interact. While the classical learning paradigms consider a single agent in a fixed environment, this assumption seems inaccurate in many modern applications. *Smart* agents, which are strategic and learn their environment, indeed interact between each other, highly influencing the final outcome. This thesis aims at exploring these different possible interplays between learning agents in a strategic environment and at describing the typical strategies that yield good performances in these settings. It also measures the different inefficiencies in social welfare stemming from both strategic

and learning considerations.

Repeated games are generally formalized as follows. At each round $t \in [T] \coloneqq \{1, \dots, T\}$, each player $m \in [M]$ individually chooses a strategy (a bidding amount for example) $s^m \in \mathcal{S}^m$ where $\mathcal{S}^m$ is the strategy space. She then receives a possibly noisy reward of expectation $u_t^m(\boldsymbol{s})$ where u_t^m is her associated reward function at time t and $\boldsymbol{s} \in \prod_{m=1}^M \mathcal{S}^m$ is the strategy profile of all players. In the following, $\boldsymbol{s^{-m}}$ represents the vector $\boldsymbol{s}$, except for its m-th component.

A learning player chooses at each new round her strategy based on her past observations. These observations can indeed help in estimating both the game environment, i.e., the utility functions u_t^m, and the other players strategy profile $\boldsymbol{s^{-m}}$.

Maximizing one's sole reward in a single player, fixed environment is at the core of optimization and learning theories and becomes even more intricate when several players are interacting with each other in repeated games. Two major types of interaction between these players can happen. First, the reward of a player at each round does not solely depend on her action, but also on other agents' actions and even potentially on past outcomes. In this case, players can either compete or cooperate, depending on the game's nature. Secondly, players can also share (to some extent) their observations with each other, influencing their estimation of the game environment. This can either lead to a faster global learning, or bias the parameters estimations.

Interaction in outcomes. Generally, the reward functions u_t depend on the complete strategy profile of the players $\boldsymbol{s}$. The different players objectives might then be antagonistic, as any strategy profile yielding a large reward for some player can yield a low reward for another player. The extreme case corresponds to zero-sum games for two players, where the utility functions verify $u^1 = -u^2$. In this case, players compete with each other and aim at maximizing their individual reward. In a single round game, Nash equilibria characterize interesting strategy profiles for strategic players. A player unilaterally deviating from a Nash equilibrium indeed suffers a decrease in her reward.

Definition 2.1 (Nash equilibrium)**.** *A strategy profile $\boldsymbol{s}$ is a Nash equilibrium for the single round game defined by the utility functions $(u^m)_{m\in[M]}$ if*

$$\forall m \in [M], \forall s' \in \mathcal{S}^m, u^m(s', \boldsymbol{s^{-m}}) \leq u^m(s^m, \boldsymbol{s^{-m}}).$$

As soon as the utility functions u^m are concave and continuous, the existence of a Nash equilibrium is guaranteed by Brouwer fixed point theorem. It is for instance the case if $\mathcal{S}^m$ is the set of probability distributions over some finite set (which is called the **action space** in the following).

This strategic consideration thus leads to a first inefficiency in the players' decisions, as they maximize their individual reward, at the expense of the collective reward. The *price of anarchy* (Koutsoupias and Papadimitriou, 1999) measures this inefficiency as the social welfare ratio between the best possible collective situation and the worst Nash equilibrium. Although reaching the best collective outcome might be illusory for selfish agents, considering the worst Nash equilibrium might be too pessimistic. Instead, the *price of stability* (Schulz and Moses, 2003) measures the inefficiency by the social welfare ratio between the best possible situation and the best Nash equilibrium.

Learning equilibria in repeated games is thus of crucial interest, as they nicely reflect the behavior of agents perfectly knowing their environment. It is in particular at the core of many problems in computer science and economics (Fudenberg et al., 1998; Cesa-Bianchi and Lugosi, 2006). A second inefficiency stems from this consideration, as players need to learn their environment and might interfere with each other, potentially converging to no or bad equilibria. A correlated equilibrium is defined similarly to a Nash equilibrium, when the strategies $(s^m)_m$ are probability distributions whose joint realizations can be correlated. It is known that when the utility functions are constant in time $u_t^m = u^m$, if all agents follow no internal regret strategies, their actions converge in average to the set of correlated equilibria (Hart and Mas-Colell, 2000; Blum and Monsour, 2007; Perchet, 2014). Yet little is known when the utility functions u_t^m also depend on the outcomes of previous rounds as in decentralized queuing systems, which are studied in Chapter 7.

Moreover, computing a Nash equilibrium might be too expensive in practice (Daskalakis et al., 2009). It is even the case in two players zero-sum games when the action space is continuous. For example in repeated auctions, a bidding action is a function $\mathbb{R}_+ \to \mathbb{R}_+$ which for every item value, returns some bidding amount. Learning equilibria in this kind of game thus seems unreasonable and optimally responding to the adversary's strategy leads to an endless arm race between the players. We instead propose in Chapter 8 to balance between the short term revenue earned by greedily bidding, and the long term revenue by maintaining some level of information asymmetry between the players, which is a crucial aspect of repeated games (Aumann et al., 1995).

In other cases (e.g., resource allocation in radio or computer networks), the players have an interest in cooperating with each other. This for example happens if players equally split their collective reward, or if they have common interests by design of the utility functions (assume for example a game with a price of anarchy equal to 1).

In multiplayer bandits, which is the focus of Part I, the players choose a channel for transmission. But if several players query the same server at some time step, a *collision* occurs and no

transmission happens on this channel. In this case, the players have interest in coordinating with each other to avoid collisions and efficiently transmit on the different channels. Besides learning the game environment, the difficulty here comes from coordinating the players with each other, while being decentralized and limited in communication. When repeating the rounds, it however becomes unclear whether players have an interest in blindly cooperating. Especially, a player could have an interest in disturbing the learning process of other players in order to grant oneself the best transmitting channel. This kind of behavior can however be prevented here as shown in Chapter 6 using, for example, Grim-Trigger strategies.

Cooperation between the players seems even more strongly enforced in decentralized queuing systems. In this problem, the utility functions also depend on the outcomes of previous rounds. Their design actually ensures that if some player cumulated a smaller reward than the other players, she gets favored in the future and is prioritized over the other players when querying some server. Consequently, players also have interest in sharing the resources with each other, to not degrade their future own rewards.

Interaction in observations. Even when the reward functions are independent of the other players' actions, i.e., u_t^m only depends on s^m, players can interact by sharing some information/observations with each other. In that case, players have no interest in competing and they only share their information to improve each other's estimation of the game environment. Such a phenomenon for example happens in distributed bandits, described in Section 3.6.1. This problem is similar to the multiplayer bandits except for two features: there are no collisions here, as the utility functions do not depend on each other's action, and players are assigned to a graph and can send messages to their neighbours. They can thus send their observations (or an aggregated function of these observations) to their neighbours, which allows to speed up the learning process.

Even in general games where the utility functions depend on the whole strategy profile $\boldsymbol{s}$, cooperative players can share some level of information in order to improve the learning rate. This is typically what allows to reach near centralized performances in the multiplayer bandits problem in Chapters 4 to 6.

When players are cooperating, the goal is generally to maximize the collective reward. As explained above, some learning inefficiency might emerge because of the different interactions between the players. When they are centralized, i.e., a central agent unilaterally controls the decisions of all the players, this is equivalent to the single player instance and this inefficiency solely comes from the learning difficulty of the problem. But when the players are decentralized, that is their decisions are individually taken without consulting with each other, additional difficulties arise, e.g., the observations/decisions cannot be mutualized. The main question in these

settings is thus generally whether decentralization yields some additional cost, i.e., whether the maximal attainable social welfare in the decentralized setting is smaller than in the centralized setting. This is especially the focus of Chapters 4 and 7, which show that decentralization has roughly no cost in homogeneous multiplayer bandits and online queuing systems, respectively. Chapter 5 also suggests that this cost scales at most with the number of players in heterogeneous multiplayer bandits.

Social learning considers a different instance of repeated games, where at each round, a new single agent plays for this sole round. A player chooses her action to maximize her expected reward, based on the former players' actions (and potentially an additional feedback). Situations of herding can then happen, where the agents never learn correctly their environment and end up taking suboptimal decisions for ever. This problem instance thus nicely illustrates how myopic agents can take decisions leading to bad collective situations. Chapter 9 on the other hand shows that this learning inefficiency is largely mitigated under mild assumptions when players observe the reviews of the previous consumers.

2.2 Stochastic Multi-Armed Bandits

The problems studied in this thesis are intricate as they combine both game theoretical and learning considerations. The framework of sequential (or online) learning and especially **Multi-Armed Bandits** (MAB) seems well adapted. On the first hand, it defines a formal and rather simple instance of learning, for which theoretical results are known. On the other hand, its sequential aspect is similar to repeated games and many connections exist between repeated games and MAB (see e.g., Cesa-Bianchi and Lugosi, 2006). MAB is indeed a particular instance of repeated games, where a single agent plays against the nature, which generates the rewards of each arm.

MAB was first introduced for clinical trials (Thompson, 1933; Robbins, 1952) and has been recently popularised thanks to its applications to online recommendation systems. Many extensions have also been developed in the past years, such as contextual, combinatorial or lipschitz bandits for example (Woodroofe, 1979; Cesa-Bianchi and Lugosi, 2012; Agrawal, 1995).

This section shortly describes the stochastic MAB problem, as well as the main results and algorithms for this classical instance, which will give insights for the proposed algorithms and results all along this thesis. We refer the reader to (Bubeck and Cesa-Bianchi, 2012; Lattimore and Szepesvári, 2018; Slivkins, 2019) for extensive surveys on MAB.

2.2.1 Model and lower bounds

At each time step $t \in [T]$, the agent pulls an arm $\pi(t) \in [K]$ among a finite set of actions, where T is the game horizon. When pulling the arm k, she observes and receives the reward $X_k(t) \sim \nu_k$ of mean $\mu_k = \mathbb{E}[X_k(t)]$, where $\nu_k \in \mathcal{P}([0,1])$ is a probability distribution on $[0,1]$. This observation $X_k(t)$ is then used by the agent to choose the arm to pull in the next rounds.

The random variables $(X_k(t))_{t=1,\dots,T}$ are independent, identically distributed and bounded in $[0,1]$ in the following. Yet, the results presented in this section also hold for the more general class of sub-gaussian variables.

In the following, $x_{(k)}$ denotes the k-th order statistics of the vector $\boldsymbol{x} \in \mathbb{R}^n$, i.e., $x_{(1)} \geq x_{(2)} \geq \ldots \geq x_{(n)}$. The goal of the agent is to maximize her cumulated reward. Equivalently, she aims at minimizing her regret, which is the difference between the maximal expected reward of an agent knowing beforehand the arms' distributions and the actual earned reward until the game horizon T. It is formally defined as

$$R(T) = T\mu_{(1)} - \mathbb{E}\left[\sum_{t=1}^{T} \mu_{\pi(t)}\right],$$

where the expectation holds over the actions $\pi(t)$ of the agent.

The player only observes the reward $X_k(t)$ of the pulled arm and not those associated to the non-pulled arms. Because of this bandit feedback, the player must balance between **exploration**, i.e., estimating the arm means by pulling all arms sufficiently, and **exploitation**, by pulling the seemingly optimal arm. This trade-off is at the core of MAB and is also crucial in repeated games, as it nicely opposes short term (exploitation) with long term (exploration) rewards.

A problem instance is fixed by the distributions $(\nu_k)_{k\in[K]}$.

Definition 2.2. *An agent (or algorithm) is asymptotically consistent if for every problem instance and $\alpha > 0$, $R(T) = o\left(T^{\alpha}\right)$.*

The cumulated reward is of order $\mu_{(1)}T$ for an asymptotically consistent algorithm. The regret is instead a more refined choice of measure, since it captures the second order term of the cumulated reward in this case.

Determining the smallest achievable regret is a fundamental question for bandits problem. First, Theorem 2.1 lower bounds the achievable regret in the classical stochastic MAB.

Theorem 2.1 (Lai and Robbins 1985)**.** *Consider a problem instance with Bernoulli distributions $\nu_k = \mathrm{Bernoulli}(\mu_k)$, then any asymptotically consistent algorithm has an asymptotic regret*

bounded as follows

$$\liminf_{T\to\infty} \frac{R(T)}{\log(T)} \geq \sum_{k:\mu_k<\mu_{(1)}} \frac{\mu_{(1)}-\mu_k}{\mathrm{kl}\left(\mu_{(1)},\mu_k\right)},$$

where $\mathrm{kl}\left(p,q\right) = p\log\left(\frac{p}{q}\right) + (1-p)\log\left(\frac{1-p}{1-q}\right)$.

A similar lower bound holds for general distributions ν_k, but this simpler version is sufficient for our purpose. The above lower bound holds asymptotically for a fixed instance and is referred to as an instance dependent bound. However, the maximal regret incurred at time T over all the possible instances might still be linear in T. This corresponds to the worst case, where the considered instance is the worst for the fixed, finite horizon T. When specifying this quantity, we instead refer to the minimax regret, which is lower bounded as follows.

Theorem 2.2 (Auer et al. 1995)**.** *For all algorithms and horizon $T \in \mathbb{N}$, there exists a problem instance such that*

$$R(T) \geq \frac{\sqrt{KT}}{20}.$$

2.2.2 Classical bandit algorithms

This section describes the following classical bandit algorithms: ε-greedy, Upper Confidence Bound (UCB), Thompson Sampling and Explore-then-commit (ETC). Most algorithms in the following chapters will be inspired from them, as they are rather simple and yield good performances. Upper bounds of their regret are provided without proofs; they mostly rely on the following concentration inequality, which allows to bound the estimation error of the empirical mean of an arm.

Lemma 2.1 (Hoeffding 1963)**.** *For independent random variables $(X_s)_{s\in\mathbb{N}}$ in $[0,1]$:*

$$\mathbb{P}\left(\frac{1}{n}\sum_{s=1}^{n} X_s - \mathbb{E}[X_s] \geq \varepsilon\right) \leq e^{-2n\varepsilon^2}.$$

The following notations are used in the remaining of this section

- $N_k(t) = \sum_{s=1}^{t-1} \mathbb{1}\left(\pi(s)=k\right)$ is the number of pulls on arm k until time t;
- $\hat{\mu}_k(t) = \frac{\sum_{s=1}^{t-1} \mathbb{1}(\pi(s)=k)X_k(t)}{N_k(t)}$ is the empirical mean of arm k before time t;
- $\Delta = \min\{\mu_{(1)} - \mu_k > 0 \mid k \in [K]\}$ is the suboptimality gap and represents the hardness of the problem.

ε-greedy algorithm

The ε-greedy algorithm described in Algorithm 2.1 is defined by a sequence $(\varepsilon_t)_t \in [0,1]^{\mathbb{N}}$. Each arm is first pulled once. Then at each round t, the algorithm explores with probability ε_t, meaning it pulls an arm chosen uniformly at random. Otherwise, it exploits, i.e., it pulls the best empirical arm.

Algorithm 2.1: ε-greedy algorithm

input: $(\varepsilon_t)_t \in [0,1]^{\mathbb{N}}$

for $t = 1, \dots, K$ **do** pull arm t

for $t = K+1, \dots, T$ **do** $\begin{cases} \text{pull } k \sim \mathcal{U}([K]) \text{ with probability } \varepsilon_t; \\ \text{pull } k \in \arg\max_{i \in [K]} \hat{\mu}_i(t) \text{ otherwise.} \end{cases}$

When $\varepsilon_t = 0$ for all t, it is called the greedy algorithm, as it always greedily pulls the best empirical arm. The greedy algorithm generally incurs a regret of order T, as the best arm can be underestimated after its first pull and never be pulled again.

Appropriately choosing the sequence (ε_t) instead leads to a sublinear regret, as given by Theorem 2.3.

Theorem 2.3 (Slivkins 2019, Theorem 1.4)**.** *For some positive universal constant c_0, ε-greedy algorithm with exploration probabilities $\varepsilon_t = \left(\frac{K \log(t)}{t}\right)^{1/3}$ has a regret bounded as*

$$R(T) \leq c_0 K \log(T)^{1/3} T^{2/3}.$$

If the suboptimality gap $\Delta = \min\{\mu_{(1)} - \mu_k > 0 \mid k \in [K]\}$ is known, choosing the sequence $\varepsilon_t = \min(1, \frac{CK}{\Delta^2 t})$ for a sufficiently large constant C leads to a logarithmic in T instance dependent regret.

Upper confidence bound algorithm

As explained above, greedily choosing the best empirical arm leads to a considerable regret. The UCB algorithm instead chooses the arm k maximizing $\hat{\mu}_k(t) + B_k(t)$ at each time step, where the term $B_k(t)$ is some confidence bound. UCB, given by Algorithm 2.2 below, thus positively bias the empirical means. Thanks to this, the best arm cannot be underestimated with high probability, thus avoiding the failing situations of the greedy algorithm described above.

Algorithm 2.2: UCB algorithm

for $t = 1, \dots, K$ **do** pull arm t

for $t = K+1, \dots, T$ **do** pull $k \in \arg\max_{i \in [K]} \hat{\mu}_i(t) + B_i(t)$

Theorem 2.4 bounds the regret of the UCB algorithm with its most common choice of confidence bound.

Theorem 2.4 (Auer et al. 2002a)**.** *The UCB algorithm with* $B_i(t) = \sqrt{\frac{2\log(t)}{N_i(t)}}$ *verifies the following instance dependent and minimax bounds, for some positive universal constants* c_1, c_2

$$R(T) \leq \sum_{k:\mu_k<\mu_{(1)}} \frac{8\log(T)}{\mu_{(1)} - \mu_k} + c_1, \tag{2.1}$$
$$R(T) \leq c_2\sqrt{KT\log(T)}.$$

The UCB algorithm thus has an optimal instance dependent regret, up to some constant factor, when the arm means are bounded away from 0 and 1. Using finer confidence bounds, an optimal instance dependent regret is actually reachable for the UCB algorithm (Garivier and Cappé, 2011). In the following of this thesis, regret bounds similar to Equation (2.1) are said optimal up to constant factors by abuse of notation.

Thompson sampling algorithm

The Thompson sampling algorithm described in Algorithm 2.3 originally adopts a Bayesian point of view. From some posterior distribution $\boldsymbol{p}$ on the arm means $\boldsymbol{\mu}$, it samples the vector $\boldsymbol{\theta} \sim \boldsymbol{p}$ and pulls an arm in $\arg\max_{k\in[K]} \theta_k$. It then updates its posterior distribution using the observed reward, according to the Bayes rule.

Algorithm 2.3: Thompson sampling algorithm

$\boldsymbol{p} = \otimes_{k=1}^{K} \mathcal{U}([0,1])$ `// Uniform prior`
for $t = 1, \dots, T$ **do**
  Sample $\boldsymbol{\theta} \sim \boldsymbol{p}$
  Pull $k \in \arg\max_{k\in[K]} \theta_k$
  Update p_k as the posterior distribution of μ_k
end

Theorem 2.5 (Kaufmann et al. 2012)**.** *There exists a function* f *depending only on the means vector* $\boldsymbol{\mu}$ *such that for every problem instance and* $\varepsilon > 0$*, the regret of Thompson sampling algorithm is bounded as*

$$R(T) \leq (1+\varepsilon) \sum_{k:\mu_k<\mu_{(1)}} \frac{\mu_{(1)} - \mu_k}{\mathrm{kl}\left(\mu_k, \mu_{(1)}\right)} \log(T) + \frac{f(\boldsymbol{\mu})}{\varepsilon^2}.$$

Despite coming from a Bayesian point of view, it thus reaches optimal frequentist performances, when initialized with a uniform prior. Proving this upper bound is rather intricate.

Sampling from the posterior distribution $\boldsymbol{p}$ might be computationally expensive at each time step. Yet in special cases, e.g., binary or gaussian rewards, the posterior update is very simple. In the general case, a *proxy* of the exact posterior can be used, by deriving results from the binary or gaussian case. The interest of Thompson sampling for combinatorial bandits is well illustrated in (Perrault et al., 2020), although this work is not discussed in this thesis.

Explore-then-commit algorithm

While the above algorithms combine exploration and exploitation at each round, the ETC algorithm instead clearly separates both in two distinct phases. It first explores all the arms. Only once the best arm is detected (with high probability), it enters the exploitation phase and pulls this arm until the final horizon T.

Distinctly separating the exploration and the exploitation phase leads to a larger regret bound. Especially, if all the arms are explored the same amount of time (uniform exploration), the instance dependent bound scales with $\frac{1}{\Delta^2}$.

Instead, the exploration is adapted to each arm as described in Algorithm 2.4. This finer version of ETC is referred to as Successive Eliminations (Perchet and Rigollet, 2013). An arm k is *eliminated* when it is detected as suboptimal, i.e., when there is some arm i such that $\hat{\mu}_k + B_k(T) \leq \hat{\mu}_i - B_i(T)$, for confidence bounds $B_i(T)$. When this condition holds, the arm k is worse than the arm i with high probability; it is thus not pulled anymore. With this adaptive exploration, the regret bound is optimal up to some constant factor as given by Theorem 2.6.

Algorithm 2.4: Successive Eliminations algorithm

$\mathcal{A} \leftarrow [K]$ `// active arms`
while $\#\mathcal{A} > 1$ **do**
  pull all arms in $\mathcal{A}$ once
  for *all* $k \in \mathcal{A}$ *such that* $\hat{\mu}_k + B_k(T) \leq \max_{i \in \mathcal{A}} \hat{\mu}_i - B_i(T)$ **do** $\mathcal{A} \leftarrow \mathcal{A} \setminus \{k\}$
end
repeat pull only arm in $\mathcal{A}$ **until** $t = T$

Theorem 2.6 (Perchet and Rigollet 2013)**.** *Algorithm 2.4 with* $B_i(t) = \sqrt{\frac{2\log(T)}{N_i(t)}}$ *has a regret bounded as*

$$R(T) \leq 324 \sum_{k:\mu_k<\mu_{(1)}} \frac{\log(T)}{\mu_{(1)} - \mu_k},$$

$$R(T) \leq 18\sqrt{KT\log(T)}.$$

Besides yielding a larger regret than UCB and Thompson sampling (of constant order), Successive Eliminations requires a prior knowledge of the horizon T. Knowing the horizon T is

not too restrictive in bandits problem (Degenne and Perchet, 2016a) and is thus assumed in the remaining of this thesis. On the other hand, Successivation Eliminations has the advantage of being simple since it clearly separates both exploration and exploitation, which will be useful for multiplayer bandits in Part I.

2.3 Outline and Contributions

The goal of this thesis is to study repeated games with decentralized learning agents. For most of the considered problems, it aims at providing good sequential learning strategies, e.g., small regret algorithms. For practical reasons, these strategies have to be computationally efficient, which is ensured and illustrated by numerical experiments in most of the cases.

Using the MAB formalization to study relations between multiple learning agents leads to the multiplayer bandits problem, which is the main focus of this thesis and particularly of Part I. On the other hand, Part II considers different and independent problems, exploring the different types of interactions that can happen between learning agents. The content of each chapter is described below.

Part I, Multiplayer Bandits

This part focuses on the problem of multiplayer bandits.

Chapter 3, Multiplayer bandits: a survey. This chapter introduces the problem of multiplayer bandits and extensively reviews the multiplayer bandits literature, including Chapters 4 to 6 and subsequent works by different authors.

Chapter 4, SIC-MMAB: Synchronisation Involves Communication in Multiplayer Multi-Armed Bandits. Although players are decentralized, they can still implicitly communicate with each other using collision information as bits. This observation is here leveraged to propose a decentralized algorithm that enforces collisions between players to allow communication between them. It then achieves a regret bound similar to the smallest achievable regret in the centralized case. Although theoretically efficient, this algorithm is not satisfying, as such a level of communication is very costly in practice. We suggest that the usual formulation of the multiplayer bandits leads to this kind of algorithm and in particular the static assumption, which assumes that all players start and end the game at the same time. We then study a new dynamic setting and propose a logarithmic regret algorithm for this setting, using no direct communication between the players.

Chapter 5, A Practical Algorithm for Multiplayer Bandits when Arm Means Vary Among Players. This chapter considers the heterogeneous case, where the arm means vary among the players. Reaching the optimal matching between the players here requires some minimal level of communication among them. This chapter thus proposes an efficient algorithm for the heterogeneous case, both in terms of regret and computation, by enforcing collisions between the players and improving the communication protocol proposed in Chapter 4.

Chapter 6, Selfish Robustness and Equilibria in Multi-Player Bandits. While the multi-player bandits literature mostly focuses on cooperative players, this chapter considers the case of strategic players, selfishly maximizing their individual cumulated reward. Existing algorithms are not adapted to this setting, as a malicious player can easily interfere with the exploration of the other players in order to significantly increase her own reward.

We thus propose a first algorithm, ignoring the collision information after some initialization, which is both a $\mathcal{O}(\log(T))$-Nash equilibrium (robust to selfish players) and has a collective regret comparable to non strategic algorithms. When collisions are observed, existing algorithms can actually be adapted to *Grim Trigger* strategies, which are also $\mathcal{O}(\log(T))$-Nash equilibria, while maintaining the regret bounds of the original cooperative algorithms. With heterogeneous players, reaching the optimal matching becomes hopeless and we instead minimize an adapted and relevant notion of regret.

Part II, Other learning instances

This part studies independent problems illustrating the different types of interaction between learning agents described in Section 2.1.

Chapter 7, Decentralized Learning in Online Queuing Systems. This chapter studies the problem of online queuing systems, originally motivated by packet routing in computer networks. In this problem, queues receive packets at different rates and repeatedly send packets to servers, each of them treating at most one packet at a time. The stability of the system (i.e., whether the number of remaining packets is bounded) is of crucial interest and is possible in the centralized case as long as the ratio between service rates and arrival rates is larger than 1. With selfish players, Gaitonde and Tardos (2020a) showed that queues minimizing their regret are stable when this ratio is above 2. Regret minimization however leads to myopic behaviors, ignoring the long term effects due to the carryover feature proper to this repeated game instance. By contrast, when minimizing long term costs, Gaitonde and Tardos (2020b) showed that all Nash equilibria are stable as long as the ratio of rates is larger than $\frac{e}{e-1}$, which can then be seen

as the price of anarchy of the considered game. Yet the cost of learning remains unknown and we argue in this chapter that some level of cooperation is required between the queues to ensure stability with a ratio below 2 when learning. As a consequence, we propose a decentralized learning strategy, that is stable for any ratio of rates larger than 1, implying that decentralization yields no additional cost here.

Chapter 8, Utility/Privacy Trade-off as Regularized Optimal Transport. In online advertisement auctions, the auctioneer and the bidders are repeatedly competing. Determining the Nash equilibria is here too costly in terms of computation, as the action spaces are continuous. Adapting to the new strategies of the other players leads to an arm race between the auctioneer and the bidders. This chapter instead proposes to naturally balance between short term reward, earned by greedily maximizing one's utility, and long term reward by hiding some private information whose disclosure could be leveraged by the other players. This problem is generally formalized as a Bayesian framework of utility/privacy trade-off, which is shown to be equivalent to Sinkhorn divergence minimization. This equivalence leads to efficient computations of this minimum, using the different tools developed in Optimal Transport and optimization theories.

Chapter 9, Social Learning in Non-Stationary Environments. This chapter considers social learning with reviews, where heterogeneous Bayesian consumers decide one after the other whether to buy an item of unknown quality, based on the previous buyers' reviews. Previous works assume the item quality to be constant in time and show that its estimate converges to its true value under mild assumptions. We here consider a dynamical model where the quality might change at some point. The additional cost due to the dynamical structure is shown to be logarithmic in the changing rate of the quality, in the case of binary features. Yet, the gap between static and dynamical models when the features belong to more complex sets remains unknown.

2.4 List of Publications

With the exception of Chapter 3, the chapters of this thesis are based either on publications in proceedings of maching learning conferences or works currently submitted, as listed below.

Advances in Neural Information Processing Systems (NeurIPS)

- Chapter 4: *"SIC-MMAB: Synchronisation Involves Communication in Multiplayer Multi-Armed Bandits"*, Etienne Boursier and Vianney Perchet (2019).

- Chapter 7: *"Decentralized Learning in Online Queuing Systems"*, Flore Sentenac*, Etienne Boursier* and Vianney Perchet (2021).

Conference on Learning Theory (COLT)

- Chapter 6: *"Selfish robustness and equilibria in multi-player bandits"*, Etienne Boursier and Vianney Perchet (2020).

International Conference on Artificial Intelligence and Statistics (AISTATS)

- Chapter 5: *"A Practical Algorithm for Multiplayer Bandits when Arm Means Vary Among Players"*, Etienne Boursier, Emilie Kaufmann, Abbas Mehrabian and Vianney Perchet (2020).

- Chapter 8: *"Utility/Privacy Trade-off through the lens of Optimal Transport"*, Etienne Boursier and Vianney Perchet (2020).

Working papers

- Chapter 9: *"Social Learning from Reviews in Non-Stationary Environments"*, Etienne Boursier, Vianney Perchet and Marco Scarsini (2020).

The author also participated in the following published works, that are not discussed in this thesis.

- *"Statistical efficiency of thompson sampling for combinatorial semi-bandits"*, Pierre Perrault, Etienne Boursier, Michal Valko and Vianney Perchet (NeurIPS 2020).

- *"Making the most of your day: online learning for optimal allocation of time"*, Etienne Boursier, Tristan Garrec, Vianney Perchet and Marco Scarsini (NeurIPS 2021).

*Equal contributions

Part I

Multiplayer Bandits

Chapter 3

Multiplayer bandits: a survey

3.1 Introduction

The problem of multiplayer bandits has known a recent interest. Motivated by cognitive radio networks, it considers multiple decentralized players on a single Multi-Armed Bandits instance. When several of them pull the same arm at some round, a *collision* occurs and causes a decrease in the received reward, which makes the problem much more intricate.

Many works on multiplayer bandits thus emerged, considering different models, objectives or algorithmic techniques. Because of the recency of the problem, the large diversity of the literature and the different communities involved (learning theory and communication networks), gathering and structuring altogether the existing works remains missing.

The goal of this survey is thus multiple. It first aims at placing the current state of the art in multiplayer bandits. It also aims at contextualizing altogether the existing works, according to their studied models, their objectives and used techniques. Finally, this survey also provides comprehensive explanations of the main existing algorithms and results.

For the sake of conciseness, this survey does not provide detailed proofs, but simple insights, of the different presented results[1]. Similarly, it does not extensively describe the mentioned algorithms but only describes them as simple and clear as possible.

Section 3.2 first presents the motivations leading to the design of the multiplayer bandits model. The most classical version of multiplayer bandits is then described in Section 3.3, along with a first base study including the centralized case and a lower bound of the incurred regret. Section 3.4 then presents the different results known for this model. In particular, collision information can be abusively used to reach regrets similar to the centralized case. Section 3.5 then presents the several practical considerations that can be added to the model, in the hope of leading to more natural algorithms. Finally, Section 3.6 mentions the Multi-agent bandits, Competing bandits and Queuing Systems problems, which all bear similarities with Multiplayer bandits, either in the model or in the used algorithms. Tables 3.3 and 3.4 in Section 3.7 summarize the main results presented in this survey.

3.2 Motivation for cognitive radio networks

The concept of cognitive radio has been first developed by Mitola and Maguire (1999) and can be defined as a radio capable of learning its environment and choosing dynamically the best wireless channels for transmission. Especially, cognitive radio should lead to a more efficient bandwidth usage rate. The concept of cognitive radio thus covers many different applications.

[1]The detailed proofs can be found in the corresponding cited papers.

Two major cognitive radio models appear to be closely related to multiplayer bandits and each of them still represent several different applications. We refer the reader to (Zhao and Sadler, 2007) for a survey on different cognitive radio models.

A first common approach to cognitive radio is **Opportunistic Spectrum Access** (OSA), which considers licensed bands, where Primary Users (PU) have preferential access to designated channels (e.g., frequency bands). In practice, many of these bands remain largely unused and Secondary Users (SU) then have the possibility to access these channels when let free by the PUs. Assuming the SUs are equipped with a *spectrum sensing* capacity, they can first sense the presence of a PU on a channel to give priority to PUs. If no PU is using the channel, SUs can then decide to transmit on this channel. Such devices yet have limited capabilities; in particular, they proceed in a decentralized network and cannot sense different channels simultaneously. This last restriction justifies the bandit feedback assumed in the considered models.

The second model related to multiplayer bandits is for **Internet of Things** (IoT) networks, where the devices have even lower power capabilities and thus cannot sense the presence of another user before transmitting. Moreover, there is no more licensed bands and all devices are then SUs (no PU). Still, these devices can perform some form of learning as they determine afterwards whether their transmission was successful. As a consequence, models for OSA and IoT still share strong similarities, as shown in Section 3.3.1.

Using a Multi-Armed bandits model for cognitive radios was first suggested by Jouini et al. (2009), Jouini et al. (2010), and Liu and Zhao (2008). In these first attempts in formalizing the problem, a single SU (player) repeatedly chooses among a choice of K channels (arms) for transmission. The success of transmission is then given by a random variable $X_k(t) \in \{0, 1\}$, where the sequence $(X_k(t))_t$ can be i.i.d. (stochastic model) or a Markov chain for instance. A successful transmission corresponds to $X_k = 1$, against $X_k = 0$ if transmission failed, e.g., the channel was occupied by a PU. The goal of the SU is then to maximize its number of transmitted bits, or in bandits lingo, to minimize its regret.

Shortly after, Liu and Zhao (2010) extended this model to multiple SUs, taking into account the interaction between SUs in cognitive radio networks. The problem becomes more intricate as SUs interfere when transmitting on the same channel. The event of multiple SUs simultaneously using the same channel is called a *collision*.

Different proof-of-concepts later justified the use of Reinforcement Learning, and especially Multi-Armed bandits model, for both OSA (Robert et al., 2014; Kumar et al., 2018b) and IoT networks (Bonnefoi et al., 2017). We refer to (Marinho and Monteiro, 2012; Garhwal and Bhattacharya, 2011) for surveys on the different research directions for cognitive radios and to (Jouini, 2012; Besson, 2019) for more details on the link between OSA and Multi-Armed

bandits.

3.3 Baseline problem and first results

This section describes the classical model of multiplayer bandits and gives first results, which are inferred from the centralized case.

3.3.1 Model

This section describes the general multiplayer bandits problem, with several variations of observation and arm means setting, as well as notations used all along this survey. Harder, more realistic variations are discussed in Section 3.5. The model and notations described here will be used in the remaining of Part I.

We consider a bandit problem with M players and K arms, where $M \leq K$. To each arm-player pair is associated an i.i.d. sequence of rewards $(X_k^m(t))_{t\in[T]}$, where X_k^m follows a distribution in $[0,1]$ of mean μ_k^m. At each round $t \in [T] \coloneqq \{1,\dots,T\}$, all players pull simultaneously an arm. We denote by $\pi^m(t)$ the arm pulled by player m at time t, who receives the individual reward

$$r^m(t) \coloneqq X^m_{\pi^m(t)}(t) \cdot (1 - \eta_{\pi^m(t)}(t)),$$

$$\text{where } \eta_k(t) = \mathbb{1}\left(\#\left\{m \in [M] \mid \pi^m(t) = k\right\} > 1\right) \text{ is the collision indicator.}$$

The players are assumed to know the horizon T and use a common numbering of the arms.

A *matching* $\pi \in \mathcal{M}$ is an assignment of players to arms, i.e., mathematically, it is a one to one function $\pi : [M] \to [K]$. The (expected) utility of a matching is then defined as

$$U(\pi) \coloneqq \sum_{m=1}^{M} \mu^m_{\pi(m)}.$$

The performance of an algorithm is measured in terms of collective regret, which is the difference between the maximal expected reward and the algorithm cumulative reward:

$$R(T) \coloneqq TU^* - \sum_{t=1}^{T}\sum_{m=1}^{M} \mathbb{E}[\mu^m_{\pi^m(t)} \cdot (1 - \eta_{\pi^m(t)}(t))],$$

where $U^* = \max_{\pi\in\mathcal{M}} U(\pi)$ is the maximal utility. In the following, the problem difficulty is related to the suboptimality gap Δ where

$$\Delta(\pi) \coloneqq U^* - U(\pi)$$

$$\text{and } \Delta \coloneqq \min \{\Delta(\pi) \mid \Delta(\pi) > 0\}.$$

In contrast to the classical bandits problem where only the received reward at each time step can be observed, algorithms might differ in the information observed at each time step, which leads to four different settings[2], described in Table 3.1 below.

Setting	Full sensing	Statistic sensing	Collision sensing	No sensing
Feedback	$\eta_{\pi^m(t)}(t)$ and $X^m_{\pi^m(t)}(t)$	$X^m_{\pi^m(t)}(t)$ and $r^m(t)$	$\eta_{\pi^m(t)}(t)$ and $r^m(t)$	$r^m(t)$

Table 3.1: Different observation settings considered. *Feedback* represents the observation of player m for round t

The different settings can be motivated by different applications, or purely for theoretical purposes. For example, statistic sensing models the OSA problem, where SUs first sense the presence of a PU before transmitting on the channel; while no sensing models IoT networks, where devices have more limited capacities as explained in Section 3.2.

The no sensing setting is obviously the hardest one, since a 0 reward can either corresponds to a low channel quality or a collision with another player.

This description corresponds to the *heterogeneous* setting, where the arm means vary among the players. In practice, it can be due to several factors such as the presence of devices of heterogeneous nature (especially in modern IoT networks) or the spatial aspect that may affect signals quality.

In the following, the easier *homogeneous* setting is also considered, in which the arm means are common to all players, i.e., $\mu_k^m = \mu_k$ for all $m, k \in [M] \times [K]$. In this case, the maximal expected reward is given by

$$\max_{\pi \in \mathcal{M}} U(\pi) = \sum_{k=1}^{M} \mu_{(k)},$$

which largely facilitates the learning problem.

The statistics $(X_k^m(t))$ can be either common or different between homogeneous players depending on the literature. In the following, we consider by default common statistics between players (i.e., $X_k^m(t) = X_k(t)$) and precise when otherwise. Note that this has no influence in both collision and no sensing settings.

[2]Bubeck and Budzinski (2020) also consider a fifth setting where only $X^m_{\pi^m(t)}(t)$ is observed in order to completely ignore collision information.

3.3.2 Centralized case

To set baseline results, first consider in this section the easier centralized model, where all players in the game described in Section 3.3.1 are controlled by a common central agent. It becomes trivial for this central agent to avoid collisions between players as she unilaterally decides the arms they pull. The difficulty is thus only to learn which is the optimal matching π in this simplified setting.

Bandits with multiple plays. In the homogeneous setting where the arm means do not vary across players, the centralized case reduces to bandits with multiple plays, where a single player has to pull M arms among a set of K arms at each round. Anantharam et al. (1987a) introduced this problem long before multiplayer bandits and provided an asymptotic lower bound for this problem, given by Theorem 3.1 below.

Komiyama et al. (2015) later showed that a Thompson Sampling (TS) based algorithm reaches this exact regret bound in the specific setting of multiple plays bandits.

Combinatorial bandits. More generally, multiple plays bandits as well as the heterogeneous centralized setting are particular instances of combinatorial bandits (Gai et al., 2012), where the central agent plays an action (representing several arms) $a \in \mathcal{A}$ and receives $r(\boldsymbol{\mu}, a)$ for reward. We here consider the simple case of linear reward $r(\boldsymbol{\mu}, a) = \sum_{k \in a} \mu_k$.

In the homogeneous case, $\mathcal{A}$ was all the subsets of $[K]$ of size M. In the heterogeneous case however, MK arms are considered instead of K (one arm per pair (m, k)) and $\mathcal{A}$ represents the set of matchings between players and arms.

Chen et al. (2013) proposed the CUCB algorithm, which yields a $\mathcal{O}\left(\frac{M^2 K}{\Delta} \log(T)\right)$ regret in the heterogeneous setting (Kveton et al., 2015). While CUCB performs well for any correlation between the arms, Combes et al. (2015) leverages the independence of arms with ESCB to reach a $\mathcal{O}\left(\frac{\log^2(M) MK}{\Delta} \log(T)\right)$ regret in this specific setting. ESCB however suffers from computational inefficiencies in general, as it requires to compute upper confidence bounds for every action. Thompson Sampling strategies remedy this problem, while still having $\mathcal{O}\left(\frac{\log^2(M) MK}{\Delta} \log(T)\right)$ regret for independent arms (Wang and Chen, 2018). Degenne and Perchet (2016b) and Perrault et al. (2020) respectively extended ESCB and combinatorial TS for the intermediate case of neither independent nor fully correlated arms.

3.3.3 Lower bound

This section describes the different lower bounds known in multiplayer bandits, which are derived from the centralized case.

As mentioned in Section 3.3.2, Anantharam et al. (1987a) provided a lower bound for the centralized homogeneous setting. This setting is obviously easier than the decentralized homogeneous multiplayer problem, so that this bound also holds for the latter.

Definition 3.1. *An algorithm is asymptotically consistent if for every instance (given by $\boldsymbol{\mu}, K, M$) and for every $\alpha > 0$, $R(T) = o\left(T^{\alpha}\right)$.*

Theorem 3.1 (Anantharam et al. 1987a)**.** *For any asymptotically consistent algorithm and any instance of homogeneous multiplayer bandits where arms follow Bernoulli distributions such that $\mu_{(M)} > \mu_{(M+1)}$,*

$$\liminf_{T\to\infty} \frac{R(T)}{\log(T)} \geq \sum_{k>M} \frac{\mu_{(k)} - \mu_{(M)}}{\mathrm{kl}\left(\mu_{(M)}, \mu_{(k)}\right)}.$$

Combes et al. (2015) proved a lower bound for general combinatorial bandits, depending on a problem constant $c(\boldsymbol{\mu}, \mathcal{A})$, determined as the solution of an optimization problem. Luckily for the specific case of matchings, its value is simplified. Especially, for some heterogeneous problem instances, any asymptotically consistent algorithm regret is $\Omega\left(\frac{KM}{\Delta}\log(T)\right)$.

Note that the lower bound is tight in the homogeneous case, i.e., an algorithm matches this regret bound, while there remains a $\log^2(M)$ gap between the known lower and upper bounds in the heterogeneous setting. In the centralized case, studying the heterogeneous setting is already more intricate than the homogeneous one. This difference seems even larger when considering decentralized algorithms as shown in the following sections.

It was first thought that the decentralized problem was harder than the centralized one, and especially in the homogeneous setting that an additional M factor, the number of players, would appear for all decentralized algorithms (Liu and Zhao, 2010; Besson and Kaufmann, 2018a). This actually only holds if the players do not use any information from the collisions with other players (Besson and Kaufmann, 2019), but as soon as the players use this information, only the centralized bound holds.

3.4 Reaching centralized optimal regret

This section shows how this collision information has been used in the literature, from a coordination tool to a communication tool between players, until reaching a near centralized performance in theory. In the following, all algorithms are written from the point of view of a single player to highlight their decentralized aspects.

3.4.1 Coordination routines

The main challenge of multiplayer bandits comes from additional loss due to collisions between players. The players cannot try solely to minimize their individual regret without considering the multiplayer environment, as they would encounter a large amount of collisions. In this direction, Besson and Kaufmann (2018a) studied the behavior of the SELFISH algorithm, where players individually follow a UCB algorithm. Although it yields good empirical results on average, players appear to incur a linear regret in some runs. Section 4.C proves the inefficiency of SELFISH for machines with infinite precision. It yet remains to be proved for machines with finite precision.

The first attempts at proposing algorithms for multiplayer bandits considered the homogeneous setting, as well as the existence of a pre-agreement between players (Anandkumar et al., 2010). If players are assumed to have distinct ranks $j \in [M]$ beforehand, the player j then just focuses on pulling the j-th best arm. Anandkumar et al. (2010) proposed a first algorithm using an ε-greedy strategy. Instead of targeting the j-th best arm, players can instead rotate in a delayed fashion on the M-best arms. For example, when player 1 targets the k-th best arm, player j targets the k_j-th best arm where $k_j = k + j - 1 \pmod M$. Liu and Zhao (2010) used a UCB-strategy with rotation among players.

This kind of pre-agreement among players is however undesirable, and many works instead suggested that the players use collision information for coordination. Especially, a significant goal of multiplayer bandits is to *orthogonalise* players, i.e., reach a state where all players pull different arms and no collision happens.

A first routine for orthogonalisation, called RAND ORTHOGONALISATION is given by Algorithm 3.1 below. Each player pulls an arm uniformly at random among some set (the M-best arms or all arms for instance). If she encounters no collision, she continues pulling this arm until receiving a collision. As soon as she encounters a collision, she then restarts sampling uniformly at random. After some time, all players end pulling different arms with high probability. Anandkumar et al. (2011) and Liu and Zhao (2010) used this routine when selecting an arm among the set of the M largest UCB indexes to limit the number of collisions between players.

Avner and Mannor (2014) used a related procedure with an ε-greedy algorithm, but instead of systematically resampling after a collision, players resample only with a small probability p. When a player gives up an arm by resampling after colliding on it, she marks it as occupied and stops trying to pull it for a long time.

Rosenski et al. (2016) later introduced a faster routine for orthogonalisation, MUSICAL CHAIRS described by Algorithm 3.2. Players sample at random as RAND ORTHOGONALISATION, but as soon as a player encounters no collision, she remains idle on this arm until the

end of the procedure, even if she encounters new collisions afterwards. This routine is faster since players do not restart each time they encounter a new collision.

Rosenski et al. (2016) used this routine with a simple Explore-then-Commit (ETC) algorithm. Players first pull all arms $\log(T)/\Delta^2$ times so that they know the M best arms afterwards, while sampling uniformly at random. Players then play musical chairs on the set of M best arms and remain idle on their attributed arm until the end. Joshi et al. (2018) proposed a similar strategy, but used MUSICAL CHAIRS directly at the beginning of the algorithm so that players rotate over the arms even during the exploration, avoiding additional collisions.

Algorithm 3.1: RAND ORTHOGONALISATION

input: time T_0, set $\mathcal{S}$
$\eta_k(0) \leftarrow 1$
for $t \in [T_0]$ **do**
  if $\eta_k(t-1) = 1$ **then**
    Sample $k \sim \mathcal{U}(\mathcal{S})$
  end
  Pull arm k
end

Algorithm 3.2: MUSICAL CHAIRS

input: time T_0, set $\mathcal{S}$
stay $\leftarrow$ False
for $t \in [T_0]$ **do**
  if not(stay) **then**
    Sample $k \sim \mathcal{U}(\mathcal{S})$
  end
  Pull arm k
  if $\eta_k(t) = 0$ **then**
    stay $\leftarrow$ True
end

Besson and Kaufmann (2018a) adapted both routines with a UCB strategy. They show that even in the statistic sensing setting where collisions are not directly observed, these routines can be used for orthogonalisation. Lugosi and Mehrabian (2018) even used MUSICAL CHAIRS with no sensing, but require the knowledge of a lower bound of $\mu_{(M)}$. Indeed, for arbitrarily small means, observing only zeros on an arm might not be due to collisions. While the ETC algorithm proposed by Rosenski et al. (2016) assumes the knowledge of Δ, Lugosi and Mehrabian (2018) removed this assumption by instead using a Successive Accept and Reject (SAR) algorithm (Bubeck et al., 2013)[3] with epochs of increasing sizes. At the end of each epoch, players eliminate the arms that appear suboptimal and accept arms that appear optimal. The remaining arms still have to be explored in the next phases. To avoid collisions on accepted arms, players proceed to MUSICAL CHAIRS at the beginning of each new epoch.

Kumar et al. (2018a) proposed an ETC strategy based on MUSICAL CHAIRS. However, they do not require the knowledge of M when assigning the M best arms to players, but instead use a scheme where players improve their current arm when possible.

[3]It is a direct extension of the Successive Eliminations algorithm, that eliminates suboptimal arms similarly and accept optimal arms as soon as they appear among the top-M arms (with high probability).

With a few exceptions (Avner and Mannor, 2014; Kumar et al., 2018a), the presented algorithms require the knowledge of the number of players M at some point, as the players must exactly target the M best arms. While some of them assume M to be a priori known, others estimate it. Especially, uniform sampling rules are useful here, since the number of players can be deduced from the collision probability (Anandkumar et al., 2011; Rosenski et al., 2016; Lugosi and Mehrabian, 2018). Indeed, assume all players are sampling uniformly at random among all arms. The probability to collide for a player at each round is exactly $1 - (1 - 1/K)^{M-1}$. If this probability is estimated tightly enough, the number of players is then exactly estimated.

Joshi et al. (2018) proposed another routine to estimate M. If all players except one are orthogonalized and rotate over the K arms while the remaining one stays idle on a single arm, the number of collisions observed by this player during a window of K rounds is then $M - 1$. Joshi et al. (2018) also proposed this routine with no sensing, in which case some lower bound on μ has to be known similarly to (Lugosi and Mehrabian, 2018).

Heterogeneous setting. All the previous algorithms reach a sublinear regret in the homogeneous setting. Reaching the optimal matching in the heterogeneous setting is yet much harder with decentralized algorithms and the first works on this topic thus only proposed solutions reaching Pareto optimal matchings. A matching is Pareto optimal if no player can change her assigned arm to increase her expected reward, without decreasing the expected reward of any other player.

Avner and Mannor (2019) and Darak and Hanawal (2019) both proposed algorithms with similar ideas to reach a Pareto optimal matching. First, the players are orthogonalized. The time is then divided in several windows. In each window, with small probability p, a player becomes a leader. The leader then suggests to switch with the player pulling her currently preferred arm (in UCB index). If this player refuses, the leader then tries to switch for her second preferred arm, and so on. This algorithm thus finally reaches a Pareto optimal matching when all arms are well estimated.

3.4.2 Enhancing communication

The works of Section 3.4.1 used collision information as tool for coordination, i.e., to avoid collisions between players. Yet, a richer level of information seems required to reach the optimal allocation in the heterogeneous case. Indeed, the sole knowledge of other players preferences order is not sufficient to compute the best matching between players and arms. Instead, players need to be able to exchange detailed information on their arm means.

For this purpose, Kalathil et al. (2014) assumed that players were able to send real numbers

to each others at some rounds. The players can then proceed to a Bertsekas Auction algorithm (Bertsekas, 1992) by bidding on arms to end up with the optimal matching. Especially, the algorithm works in epochs of doubling size. Each epoch starts by a decision phase, where players bid according to UCB indexes of their arms. After this phase, players are attributed an ε-optimal matching for these indexes and pull this matching for the remaining of the epoch. This algorithm was later improved and adapted to ETC and Thompson sampling strategies (Nayyar et al., 2016).

Although these works provide first algorithms with a sublinear regret in the heterogeneous setting, they assume undesirable communication possibilities between players. Actually, this kind of communication is possible through collision observations. In the following of this section, we consider the collision sensing setting if not specified, so that a collision is systematically detected.

Communication via Markov chains.

Bistritz and Leshem (2020) adapted a Markov chain dynamic (Marden et al., 2014) for multiplayer bandits to attribute the best matching to players. Here as well the algorithm works with epochs of increasing sizes. Each epoch is divided in an exploration phase where players estimate the arm means; a Game of Thrones (GoT) phase, in which players follow a Markov chain dynamic to determine the best estimated matching; and an exploitation phase where players pull the matching attributed by the GoT phase. This algorithm reaches a $\log^{1+\delta}(T)$ regret for any choice of parameter $\delta > 0$ and even with several optimal matchings.

The main interest of the algorithm comes from the GoT phase, described in Algorithm 3.3, which allows the players to determine the best matching using only collision information. In this phase, players follow a decentralized game, where they tend to explore more when discontent (state D) and still explore with a small probability when content (state C). When the routine parameters ε and c are well chosen, players tend to visit more often the best matching according to the estimated means $\hat{\mu}_k^j$ so far. In particular, each player, while content, pulls her assigned arm in the optimal matching most often. This phase thus allows to estimate the optimal matching between arms and players as proved by Bistritz and Leshem (2020).

Youssef et al. (2020) extended this algorithm to the multiple plays setting, where each player can pull several arms at each round.

This algorithm is a very elegant way to assign the optimal matching to decentralized players. However, it suffers from a large dependency in other problem parameters than T, as the GoT phase requires the Markov chain to reach its stationary distribution. Moreover, the algorithm requires a good tuning of the GoT parameters ε and c, which depends on the suboptimality gap

Algorithm 3.3: Game of Thrones subroutine

input: time T_0, starting arm $\overline{a}_t$, player j, parameters ε and c

$S_t \leftarrow C$; $u_{\max} \leftarrow \max_{k\in[K]} \hat{\mu}_k^j$

for $t = 1, \dots, T_0$ **do**

  if $S_t = C$ **then** pull k with probability $\begin{cases} 1-\varepsilon^c \text{ if } k = \overline{a}_t \\ \varepsilon^c/(K-1) \text{ otherwise} \end{cases}$

  else pull $k \sim \mathcal{U}([K])$

  if $k \neq \overline{a}_t$ *or* $\eta_k(t) = 1$ *or* $S_t = D$ **then**

    $\overline{a}_t, S_t \leftarrow \begin{cases} k, C \text{ with probability } \frac{\hat{\mu}_k^j \eta_k(t)}{u_{\max}} \varepsilon^{u_{\max} - \hat{\mu}_k^j \eta_k(t)} \\ k, D \text{ otherwise} \end{cases}$

end

return *arm the most played, that resulted in being content*

Δ.

Collision Information as bits.

In Chapter 4, we suggest with SIC-MMAB algorithm that the collision information $\eta_k(t)$ can be interpreted as a bit sent from a player i to a player j, if they previously agreed that at this time, player i was sending a message to player j. For example, a collision represents a 1 bit, while no collision a 0 bit.

Such an agreement is possible if the algorithm is well designed and different ranks in $[M]$ are assigned to the players. These ranks are here assigned using an initialization procedure that first orthogonalises the players with Musical chairs. The number of players M and different ranks are then estimated in a time $\mathcal{O}\left(K^2\right)$, using a procedure close the one of Joshi et al. (2018) described in Section 3.4.1.

Homogeneous setting. After this initialization, the SAR based algorithm runs in epochs of doubling size. Each epoch is divided in an exploration phase, where players pull accepted arms and arms to explore. In the communication phase, players then send to each other their empirical means (truncated up to a small error) in binary, using collision information as bits. From then, players have shared all their statistics, and can accept/eliminate in common the optimal/suboptimal arms. These epochs go on, until M arms have been accepted. The players then pull these arms until T, with no collision.

Note that the communication regret of SIC-MMAB can directly be improved by using a leader gathering all the information and giving the arms to pull to other players, as done in Chapter 5.

As the players share their statistics altogether, we show that the centralized lower bound was achievable with decentralization, contradicting first intuitions. The algorithm however presents an additional $MK\log(T)$ regret due to the initialization. Wang et al. (2020) later improved this initialization, so that its regret is only of order K^2M^2. Their algorithm thus matches the theoretical lower bound for the homogeneous setting.

Theorem 3.2 (Wang et al. 2020)**.** DPE1 *algorithm, in the homogeneous with collision sensing setting such that* $\mu_{(M)} > \mu_{(M+1)}$*, has an asymptotic regret bounded as*

$$\limsup_{T\to\infty} \frac{R(T)}{\log(T)} \leq \sum_{k>M} \frac{\mu_{(k)} - \mu_{(M)}}{\mathrm{kl}\left(\mu_{(M)}, \mu_{(k)}\right)}.$$

Wang et al. (2020) also improved the communication regret, using a leader who is the only player to explore, and tells to the other players which arms to explore. Verma et al. (2019) also proposed to adapt SIC-MMAB with a leader who is the only one to explore the arms.

Shi et al. (2020) extended the SIC-MMAB algorithm to the no sensing case using *Z-channel coding*. It yet requires the knowledge of a lower bound of the arm means $\mu_{\min}$. Indeed, while a collision is detected in a single round with collision sensing, it can be detected with high probability in $\frac{\log(T)}{\mu_{\min}}$ rounds with no sensing. The suboptimality gap Δ is also assumed to be known here, to fix the number of sent bits at each epoch (while p bits are sent after the epoch p in SIC-MMAB) .

Huang et al. (2021) overcome this issue by proposing a no sensing algorithm without additional knowledge of problem parameters. In particular, it neither requires prior knowledge of $\mu_{\min}$ nor has a regret scaling with $\frac{1}{\mu_{\min}}$. Such a result is made possible by electing a good arm before the initialization. The players indeed start the algorithm with a procedure, such that afterwards, with high probability, they have elected an arm $\overline{k}$, which is the same for all players and they have a common lower bound of $\mu_{\overline{k}}$, which is of the same order as $\mu_{(1)}$. Thanks to this, the players can then send information on this arm in $\mathcal{O}\left(\frac{\log(T)}{\mu_{(1)}}\right)$ rounds. This then makes the communication regret independent from the means μ_k, since the regret generated by a collision is at most $\mu_{(1)}$. After electing this good arm, the algorithm similar to the one by Shi et al. (2020), with a few modifications to ensure that players only communicate on the good arm $\overline{k}$.

Yet the communication cost remains large, i.e., of order $KM^2\log(T)\log\left(\frac{1}{\Delta}\right)^2$, as sending a bit requires a time of order $\log(T)$ here. Although this term is often smaller than the exploration (centralized) regret, it can be much larger for some problem parameters. Reducing this communication cost thus remains left for future work.

Heterogeneous setting. The idea of considering collision information as bits sent between players can also be used in the heterogeneous setting. Indeed, this allows the players to share their estimated arm means, and then computes the optimal matching. If the suboptimality gap Δ is known, a natural algorithm (Magesh and Veeravalli, 2019b) estimates all the arms with a precision $\Delta/(2M)$. All players then communicate their estimations, compute the optimal matching and stick to it until T.

When Δ is unknown, Tibrewal et al. (2019) proposed an ETC algorithm, with epochs of increasing sizes. Each epoch consists in an exploration phase where players pull all arms; a communication phase where players communicate their estimated means; and an exploitation phase where players pull the best estimated matching.

Chapter 5 extends SIC-MMAB to the heterogeneous setting, besides improving its communication protocol with the leader/follower scheme mentioned above. The main difficulty is that players have to explore matchings here. But exploring all matchings lead to a combinatorial regret and computation time of the algorithm. Players instead explore arm-player pairs and the SAR procedure thus accept/reject pairs that are sure to be present/absent in the optimal matching.

With a unique optimal matching, similarly to SIC-MMAB, exploration ends at some point and players start exploiting the optimal matching. In the case of several optimal matchings, we provide a $\log^{1+\delta}(T)$ regret algorithm for any $\delta > 0$, using longer exploration phases.

3.4.3 No communication

The previous section showed how the collision information can be leveraged to enable communication between players. These communication schemes are yet often unadapted to the reality, for different reasons given in Section 3.5. In particular, while the communication cost is small in T, it is large in other problem parameters such as M, K and $\frac{1}{\Delta}$. These quantities can be large in real cognitive radio networks and the communication cost of algorithms presented in Section 3.4.2 is then significant.

Some works instead focus on which level of regret is possible with no collision information at all in the homogeneous setting. Naturally, they assume a pre-agreement between players, who know beforehand M and are assigned different ranks in $[M]$.

The algorithm of Liu and Zhao (2010), presented in Section 3.4.1, provides a first algorithm using no collision information. In Chapter 6, we reach the regret bound $M \sum_{k>M} \frac{\mu_{(k)}-\mu_{(M)}}{\mathrm{kl}\left(\mu_{(M)},\mu_{(k)}\right)}$, adapting the exploitation phase of DPE1 to this setting. Especially, this instance dependent bound is optimal among the class of algorithms using no collision information (Besson and Kaufmann, 2019).

Despite being asymptotically optimal, this algorithm suffers a considerable regret when the suboptimality gap Δ is close to 0. It indeed relies on the fact that if the arm rankings of the players are the same, there is no collision, while the complementary event appears an order $\frac{1}{\Delta^2}$ of rounds.

Bubeck et al. (2020a) instead focused on reaching a $\sqrt{T\log(T)}$ minimax regret without collision information. A preliminary work (Bubeck and Budzinski, 2020) proposed a first geometric solution for two players and three arms, before being extended to general numbers of players and arms with combinatorial arguments. Their algorithm has 0 collision with high probability, using a colored partition of $[0,1]^K$, where a color gives a matching between players and arms. Thus, the estimation $\hat{\boldsymbol{\mu}}^j$ of all arms by a player gives a point in $[0,1]^K$ and consequently, an arm to pull for this player. The key of the algorithm is that for close points in $[0,1]^K$, different matchings might be assigned, but they do not overlap, i.e., if players have close estimations $\hat{\boldsymbol{\mu}}^j$ and $\hat{\boldsymbol{\mu}}^i$, they still pull different arms. Such a coloring implies that for some regions, players might deliberately pull suboptimal arms, but at a small cost, to avoid collisions with other players.

Unfortunately, the algorithm of Bubeck et al. (2020a) still suffers a dependency $MK^{11/2}$ in the regret, which grows considerably with the number of channels K.

3.5 Towards realistic considerations

Section 3.4 proposes algorithms reaching very good regret guarantees for different settings. Most of these algorithms are yet unrealistic, e.g., a large amount of communication occurs between the players, while only a very small level of communication is possible between the players in practice. The fact that good theoretical algorithms are actually bad in practice emphasizes that the model of Section 3.3.1 is not well designed. In particular, it might be too simple with respect to the real problem of cognitive radio networks.

Section 3.4.3 suggests that this discrepancy might be due to the fact that the number of secondary users and channels (M and K) is actually very large, and the dependency on these terms is as significant as the dependency in T. This kind of question even appears in the bandits literature for a single player (and a very large number of arms). Recent works showed that the greedy algorithm actually performs very well in this single player setting, confirming a behavior that might be observed in some real cases (Bayati et al., 2020; Jedor et al., 2021).

This section proposes other reasons for this discrepancy. Several simplifications are removed in the multiplayer model, hoping that good theoretical algorithms in these new settings are also reasonable in practice. First, the stochasticity of the reward X_k is questioned in Section 3.5.1 and replaced by either Markovian, abruptly changing or adversarial rewards. The current collision

model is then relaxed in Section 3.5.2. It instead considers a more realistic and difficult model where players only observe a decrease in reward when colliding. Section 3.5.3 considers non-collaborative players, which can be either adversarial or strategic. A dynamic setting, where secondary users do not enter or leave the network at the same instant, is finally considered in Section 3.5.4.

3.5.1 Non-stochastic rewards

Most existing works in multiplayer bandits assume that the rewards $X_k(t)$ are stochastic, i.e., they are drawn according to the same distribution at each round. This assumption might be too simple for the problem of cognitive radio networks, and other settings can instead be adapted from the bandits literature. It has indeed been the case for markovian rewards, abruptly changing rewards and adversarial rewards, as described in this section.

Markovian rewards.

A first more complex model is given by markovian rewards. This model is rather natural in the licensed band paradigm, where the presence probability of a primary user on a band might be conditioned on its presence in the previous step. A primary user might indeed uses the band in *blocks*, in which case the probability of occupation of a band for the next round is larger if it is already occupied. In this model introduced by Anantharam et al. (1987b), the reward X_k^j of arm k for player j follows an irreducible, aperiodic, reversible Markov chain on a finite space. Given the transition probability matrix P_k^j, if the last **observed** reward of arm k for player j is x, then player j will observe x' on this arm for the next pull with probability $P_k^j(x, x')$.

Given the stationary distribution p_k^j of the Markov chain represented by P_k^j, the expected reward of arm k for player j is then equal to

$$\mu_k^j = \sum_{x \in \mathcal{X}} x p_k^j(x),$$

where $\mathcal{X} \subset [0, 1]$ is the state space. The regret then compares the performance of the algorithm with the reward obtained by pulling the maximal matching with respect to μ at each round.

Anantharam et al. (1987b) proposed an optimal centralized algorithm for this setting, based on a UCB strategy. Kalathil et al. (2014) later proposed a first decentralized algorithm for this setting. Their algorithm follows the same lines as the algorithm described in Section 3.4.2 for the stochastic case. Recall that it uses explicit communication between players to assign the arms to pull. The only difference is that the UCB index has to be adapted to the markovian model. The uncertainty is indeed larger in this setting, and the regret is thus larger as well. Bistritz and

Leshem (2020) also showed that the GoT algorithm can be directly extended to this model, with a proper tuning of its different parameters.

In a more recent work, Gafni and Cohen (2021) instead consider a restless Markov chain, i.e., the state of an arm changes according to the Markov chain at each round, even when it is not pulled. Using an ETC approach, they were thus able to reach a stable matching in a logarithmic time. Their algorithm yet assumes the knowledge of the suboptimality gap Δ and the uniqueness of the stable (Pareto optimal) matching. The main difficulty of the restless setting is that the exploration phase has to be carefully done in order to correctly estimate the expected reward of each arm. This adds a dedicated random amount of time at the start of every exploration phase.

Abruptly changing rewards.

Although markovian rewards are closer to the reality, the resulting algorithms are very similar to the stochastic case. Indeed, the goal is still to pull the arm with the expected mean overall, while the change is just on its reward distribution.

A stronger model assumes instead that the expected rewards abruptly change over time, e.g., the mean vector μ is piecewise constant with the time, and each change is a *breakpoint*. It still illustrates the fact that primary users might occupy the bands in *blocks*, but it here uses a harder, frequentist point of view. Even in the single player case, this problem is far from being solved (see e.g. Auer et al., 2019; Besson et al., 2020).

Wei and Srivastava (2018) considered this setting for the homogeneous multiplayer bandits problem. Assuming a pre-agreement on the ranks of players, they propose an algorithm with regret of order $T^{\frac{1+\nu}{2}}\log(T)$ where the number of breakpoints is $\mathcal{O}\left(T^{\nu}\right)$. Players use UCB indices computed on sliding windows of length $\mathcal{O}\left(t^{\frac{1-\nu}{2}}\right)$, i.e., they compute the indices using only the observations of the last $t^{\frac{1-\nu}{2}}$ rounds. Based on this, player k either rotates on the top-M indices or focuses on the k-th best index to avoid collisions with other players.

Adversarial rewards.

The hardest model for rewards is the adversarial case, where the rewards are fixed by an adversary. Although this model might be less motivated by cognitive radios, it has a strong theoretical interest, as it considers the worst case sequence of generated rewards. In this case, the goal is to provide a minimax regret bound that holds under any problem instance. For the homogeneous stochastic case, we show in Chapter 4 that SIC-MMAB algorithm has a $K\sqrt{T\log(T)}$ regret.

Bubeck et al. (2020b) showed that for an *adaptive* adversary, who chooses the rewards $X_k(t)$ of the next round based on the previous decisions of the players, the lower bound is linear with

T. The literature thus focuses on an *oblivious* adversary, who chooses beforehand the sequences of adversarial rewards $X_k(t)$.

Bande and Veeravalli (2019) proposed a first algorithm based on the celebrated EXP.3 algorithm. The EXP.3 algorithm pulls the arm k with a probability proportional to $e^{-\eta S_k}$ where η is the learning rate and S_k is an estimator of $\sum_{s<t} X_k(s)$. Not all the terms of this sum are observed, justifying the use of an estimator. To avoid collisions, Bande and Veeravalli (2019) run EXP.3 in blocks of size $\sqrt{T}$. In each of these blocks, the players start by pulling with respect to the probability distribution of EXP.3 until finding a free arm, thanks to collision sensing. Afterwards, the player keeps pulling this arm until the end of the block. This algorithm yields a regret of order $T^{3/4}$. Dividing EXP.3 in blocks thus degrades the regret by a factor $T^{1/4}$ here.

Alatur et al. (2020) proposed a similar algorithm, with a leader-followers structure. At the beginning of each block, the leader communicates to the followers the arms they have to pull for this block, still using the probability distribution of EXP.3. Also, the size of each block is here of order $T^{1/3}$, leading to a better regret scaling with $T^{2/3}$.

Shi and Shen (2020) later extended this algorithm to the no sensing setting. They introduce the *attackability* of the adversary, which is the length of the longest possible sequence of $X_k = 0$ on an arm. Knowing this quantity W, a bit can indeed be correctly sent in time $W + 1$. When the attackability is of order T^α and α is known, the algorithm of Alatur et al. (2020) can then be adapted and yields a regret of order $T^{\frac{2+\alpha}{3}}$.

The problem is much harder when α is unknown. In this case, the players estimate α by starting from 0 and increasing this quantity by ε at each communication failure. To keep the players synchronized with the same estimate of α, the followers then report the communication failure to the leader. These reports are crucial and can also fail because of 0 rewards. Shi and Shen (2020) here use error detection code and randomized communication rounds to avoid such situations.

Bubeck et al. (2020b) were the first to propose a $\sqrt{T}$ regret algorithm for the collision sensing setting, but only with two players. Their algorithm works as follows: a first player follows a low switching strategy, e.g., she changes the arm to pull after a random number of times of order $\sqrt{T}$, while the second player follows a high-switching strategy, given by EXP.3, on all the arms except the one pulled by the first player. At each change of arm for the first player, a communication round then occurs so that the second player is aware of the choice of the first one.

This algorithm requires a shared randomness between the players, as the first player changes her arm at random times. Yet, the players can choose a common *seed* during the initialization, avoiding the need for this assumption.

Bubeck et al. (2020b) also proposed a $T^{1-\frac{1}{2M}}$ algorithm for the no sensing setting. For two players, the first, low-switching player runs an algorithm on the arms $\{2, \dots, K\}$ and divide the time in fixed blocks of length of order $\sqrt{T}$. Meanwhile on each block, the high-switching player runs EXP.3 on an increasing set S_t starting from $S_t = \{1\}$. At random times, this player pulls arms not in S_t and adds them in the set S_t if they get a positive reward. The arm pulled by the first player is then never added to S_t.

For more than two players, Bubeck et al. (2020b) generalize this algorithm using blocks of different size for different players.

3.5.2 Different collision models

As shown in Section 3.4.2, the collision information allows communication between the different players. The discrepancy between the theoretical and practical algorithms might then be due to the collision model, which is here too strict as a collision systematically corresponds to a 0.

Non-zero collision reward. Depending on the used transmission protocol, the presence of several users on the same channel does not necessarily lead to an absence of transmission in practice, but only in a decrease of its quality. Moreover, the number of secondary users can exceed the number of channels. This harder setting was introduced by Tekin and Liu (2012). In the heterogeneous setting, when player j pulls an arm k, the expectation of the random variable $X_k^j(t)$ also depends on the total number of players pulling this arm. The problem parameters are then given by the functions $\mu_k^j(m)$ which are the expectation of X_k^j when exactly m players are pulling the arm k. Naturally, the function μ_k^j is non-increasing in m. The regret then compares the cumulative reward with the one obtained by the best allocation of players through the different arms. Note that in this problem, there is no need to assume $M \leq K$ anymore as several players can be assigned to the same arm without leading to 0 rewards on this arm.

Tekin and Liu (2012) proposed a first ETC algorithm, when players know the suboptimality gap of the problem and always observe the number of players pulling the same arm as they do. These assumptions are pretty strong and are not considered in the more recent literature.

Bande and Veeravalli (2019) also proposed an ETC algorithm, still with the prior knowledge of the suboptimality gap. During the exploration, players pull all arms at random. The main difficulty is that when players observe a reward, they do not know how many other players are also pulling this arm. Bande and Veeravalli (2019) overcome this issue by assuming that the decrease in mean rewards with the number of players is large enough with respect to the noise in the reward. As a consequence, the observed rewards on a single arm can then be perfectly clustered, where each cluster exactly corresponds to the observations for a given number of

players pulling the arm.

In practice, this assumption is actually very strong and means that the observed rewards are almost noiseless. Magesh and Veeravalli (2019a) instead assume that all the players have different ranks. Thanks to this, they can coordinate their exploration, so that all players can explore each arm k with a known and fixed number of players m pulling it. Exploring for all arms and all numbers of players m then allows the players to know their own expectations $\mu_k^j(m)$ for each k and m. From there, the players can reach the optimal allocation using a Game of Thrones routine similar to Algorithm 3.3. This work thus extended the known results for this routine to the harder setting of non-zero rewards in case of collision.

Bande et al. (2021) recently used a similar exploration for the homogeneous setting. In this case, the allocation routine is not even needed as players can compute the optimal allocation solely based on their own arm means.

When the arm mean is exactly inversely proportional, i.e., $\mu_k^j(m) = \frac{\mu_k^j(1)}{m}$, Boyarski et al. (2021) exploit this assumption to defer a simple $\mathcal{O}\left(\log^{3+\delta}(T)\right)$ regret algorithm. During the exploration phase, all players first pull each arm k altogether and estimate $\mu_k^j(M)$. From there, they add a block where they pull the arm 1 with probability $\frac{1}{2}$, allowing to estimate M and thus the whole functions μ_k^j. The optimal matching is then assigned following a GoT subroutine.

Competing bandits. A recent stream of literature considers another collision model where only one of the pulling players gets the arm reward, based on preferences of the arm. This setting, introduced by Liu et al. (2020b), was initially not motivated by cognitive radio networks and is thus discussed later in Section 3.6.2. An asymmetric collision model is also used for decentralized queuing systems, which are discussed in Section 3.6.3 and studied in Chapter 7.

3.5.3 Non-collaborative players

Assuming perfectly collaborative players might be another oversimplification of the usual multiplayer bandits model. A short survey by Attar et al. (2012) presents the different security challenges for cognitive radio networks. Roughly, these threats are divided into two types: *jamming attacks* and *selfish players*, which both appear as soon as players are no more fully cooperative.

Jammers. Jamming attacks can happen either from agents external to the network, or directly within the network. Their goal is to deteriorate the performance of other agents as much as possible. In the first case, it can be seen as malicious manipulations of the rewards generated on each arm. Wang et al. (2015) then propose to consider the problem as an adversarial instance and use EXP.3 algorithm in the centralized setting.

Sawant et al. (2019) on the other side consider jammers directly within the network. The jammers thus aim at causing a maximal loss of the other players by either pulling the best arms or creating collisions. Without any restriction on the jammers' strategy, they can perfectly adapt to the other players' strategy and cause tremendous losses. Because of this, the jammers' strategy is restricted to pulling at random the top J-arms for any $J \in [K]$, either in a centralized (no collision between jammers) or decentralized way. The players then use an ETC algorithm, where the exploration aims at estimating the arm means, but also both the number of players and the number of jammers. Afterwards, they exploit by sequentially pulling the top J-arms where J is chosen to maximize the earned reward.

Fairness. A first attempt at preventing from selfish behaviors is to ensure *fairness* of the algorithms, as noted by Attar et al. (2012). A fair algorithm should not favor some player with respect to another. In the homogeneous setting, a first definition of fairness is to guarantee the same expected rewards to all players (Besson and Kaufmann, 2018a). Note that all symmetric algorithms (i.e., no prior ranking of the players) ensure this property. A stronger notion would be to guarantee the same asymptotic rewards to all players without expectation[4], which can still be easily reached by making the players sequentially pull all the top-M arms in the exploitation phase.

The notion of fairness becomes complex in the heterogeneous setting, since it can be antagonistic to the maximization of the collective reward. Bistritz et al. (2021) consider max-min fairness, which is broadly used in the resource allocation literature. Instead of maximizing the sum of players' rewards, the goal is to maximize the minimal reward earned by each player at each round. They propose an ETC algorithm which determines the largest possible γ such that all players can earn at least γ at each round. For the allocation, the players follow a specific Markov chain to determine whether players can all reach some given γ. If instead the objective is for each player j to earn at least γ_j for some known and feasible vector $\boldsymbol{\gamma}$, there is no need to explore which is the largest possible γ and the regret becomes constant.

Selfish players. While jammers try to cause a huge loss to other players at any cost, selfish players have a different objective: they maximize their own individual reward. In the algorithms mentioned so far, a selfish player could largely improve her earned regret at the expense of the other players. Chapter 6 proposes algorithms robust to selfish players, being a $\mathcal{O}(\log(T))$-Nash equilibrium. Without collision information, we adapt DPE1 without communication between the players. The main difficulty comes from designing a robust initialization protocol to assign

[4]This notion is defined *ex post*, as opposed to the previous one which is *ex ante*.

ranks and estimate M. With collision information, we even show that robust communication based algorithms are possible, thanks to a Grim Trigger strategy which punishes all players as soon as a deviation from the collective strategy is detected. The centralized performances are thus still possible with selfish players.

Reaching the optimal matching might not be possible in the heterogeneous case because of the strategic feature of the players. Instead, we focus on reaching the average reward when following the Random Serial Dictatorship algorithm, which has good strategic guarantees in this setting (Abdulkadiroğlu and Sönmez, 1998).

Brânzei and Peres (2019) consider a different strategic multiplayer bandits game. First, their model is collisionless and players still earn some reward when pulling the same arm. Also, they consider two players and a one armed bandit game, with a prior over the arm mean. Players observe both their obtained reward and the choice of the other player.

They then compare the different Nash equilibria when players are either collaborative (maximizing sum of two rewards), neutral (maximizing their sole reward) and competitive (maximizing the difference between their reward and the other player's reward). Players tend to explore more when cooperative and less when competitive. A similar behavior is intuitive in the classical model of multiplayer bandits as selfish players would more aggressively appropriate the best arms to keep them for a long time.

3.5.4 Dynamic case

Most of the multiplayer algorithms depend on a high level of synchronisation between the players. In particular, they assume that all players respectively start and end the game at times $t = 1$ and $t = T$. This assumption actually makes the problem much simpler because it allows a high level of synchronisation, while being unrealistic since secondary users enter and leave the network at different time steps.

The dynamic model thus proposes a weaker level of synchronisation: the time step division remains global and shared by all players, but players enter and leave the bandits instance at different (unknown) times. This is different from asynchronicity, which corresponds to a heterogeneous time division between players and has been very little studied in theory (Bonnefoi et al., 2017).

The MEGA algorithm of Avner and Mannor (2014) was the first proposed algorithm to deal with this dynamic model. The exact same algorithm as the one described in Section 3.4.1 still reaches a regret of order $NT^{\frac{2}{3}}$ in this case, where N is the total number of players entering or leaving the network.

In general, N is assumed to be sublinear in T as otherwise players would enter and leave

the network too fast to learn the different problem parameters. Rosenski et al. (2016) propose to divide the game duration into $\sqrt{NT}$ epochs of equal size and run independently the MUSICAL CHAIRS algorithm on each epoch. The number of failing epochs is at most N and their total incurred regret is thus of order $\sqrt{NT}$. Finally, the total regret by this algorithm is of order $\sqrt{NT}\frac{K^2\log(T)}{\Delta^2}$.

This technique can be used to adapt any static algorithm, but it requires the knowledge of the number of entering/leaving players N, as well as a shared clock between players, to remain synchronized on each epoch. Because it also works in time windows of size $\sqrt{T}$, the algorithm of Bande and Veeravalli (2019) in the adversarial setting still has $T^{\frac{3}{4}}$ regret guarantees in the dynamic setting.

On the other hand, Bande and Veeravalli (2019) and Bande et al. (2021) propose to adapt their static algorithms, with epochs of linearly increasing size. Players do not need to know N here, but instead need a stronger shared clock, since they also need to know in which epoch they currently are.

Besides requiring some strong assumption on either players' knowledge or synchronisation, this kind of technique also leads to large dependencies in T. Players indeed run independent algorithms on a large number of time windows and thus suffer a considerable loss when summing over all the epochs.

To avoid this kind of behavior, Chapter 4 considers a simpler dynamic setting, where players can enter at any time but all leave the game at time T. We propose a no sensing ETC algorithm, which requires no prior knowledge and no further assumption. The idea is that exploring uniformly at random is robust to the entering/committing of other players. The players then try to commit on the best known available arm. This algorithm leads to a $\frac{NK\log(T)}{\Delta^2}$ regret.

On the other hand, the algorithm by Darak and Hanawal (2019) recovers from the event of entry/leave of a player after some time depending on the problem parameters. However, if enter/leave events happen in a short time window, the algorithm has no guarantees. This algorithm is thus adapted to another simpler dynamic setting, where the events of entering or leaving of a new player are separated from a minimal duration.

3.6 Related problems

This section introduces related problems that have also been considered in the literature. All these models consider a bandits game with multiple agents with some level of interaction between the agents. Because of these similarities with multiplayer bandits, methods and techniques mentioned in this survey can be directly used or adapted to these related problems.

The widely studied problem of multi-agent bandits is first mentioned. Section 3.6.2 then introduces the problem of competing bandits, motivated by matching markets. Section 3.6.3 finally discusses the problem of queuing systems, motivated by packet routing to servers.

3.6.1 Multi-agent bandits

The multi-agent bandits problem (also called cooperative bandits and distributed bandits) introduced by Awerbuch and Kleinberg (2008) considers a bandit game played by M players. Motivated by distributed networks where agents can share their cumulated information, players here encounter no collision when pulling the same arm: their goal is to collectively determine the best arm. While running a single player algorithm such as UCB already yields regret guarantees, players can improve their performance by collectively sharing some information. The way players can communicate yet remains limited: they can only directly communicate with their neighbours in a given graph $\mathcal{G}$.

This problem has been widely studied in the past years, and we do not claim to provide an extensive review of its literature.

Many algorithms are based on a gossip procedure, which is widely used in the more general field of decentralized computation. Roughly, a player i updates its estimates $\hat{x}^i$ by averaging (potentially with different weights) the estimates $\hat{x}^j$ of her neighbors j. Mathematically, the estimated vector $\hat{\boldsymbol{x}}$ is updated as follows:

$$\hat{\boldsymbol{x}} \leftarrow P\hat{\boldsymbol{x}},$$

where P is a communication matrix. To respect the communication graph structure, $P_{i,j} > 0$ if and only if the edge (i, j) is in $\mathcal{G}$. P thus gives the weights used to average these estimates.

Szorenyi et al. (2013) propose an ε-greedy strategy with gossip based updates, while Landgren et al. (2016) propose gossip UCB algorithms. Their regret decomposes in two terms: a centralized term approaching the regret incurred by a centralized algorithm and a term, which is constant in T but depends on the spectral gap of the communication matrix P, which can be seen as the *delay* to pass a message along the graph with the gossip procedure. Improving this graph dependent term is thus the main focus of many works. Martínez-Rubio et al. (2018) propose a UCB algorithm with gossip acceleration techniques, improving upon previous work (Landgren et al., 2016).

Another common procedure is to elect a leader in the graph, who sends the arm (or distribution) to pull to the other players. In particular, Wang et al. (2020) adapt the DPE1 algorithm described in Section 3.4.2 to the multi-agent bandits problem. The leader is the only exploring

player and sends her best empirical arm to the other players. Besides having an optimal regret bound in T, the second term of the regret due to communication scales with the diameter of the graph $\mathcal{G}$. Moreover, this algorithm only requires for the players to send 1-bit messages at each time step, while most multi-agent bandits work assume that the players can send real messages with infinite precision.

In the adversarial setting, Bar-On and Mansour (2019) propose to elect *local* leaders who send the distribution to play to their followers, based on EXP.3. Instead of focusing on the collective regret as usually done, they provide good individual regret guarantees.

Another line of work assumes that a player observes the rewards of all her neighbors at each time step. Cesa-Bianchi et al. (2019b) even assume to observe rewards of all players at distance at most d, with a delay depending on the distance of the player. EXP.3 with smartly chosen weights then allows to reach a small regret in the adversarial setting.

More recent works even assume that the players are asynchronous, i.e., players are active at a given time step with some activation probability. This is for example similar to the model by Bonnefoi et al. (2017) in the multiplayer setting. Cesa-Bianchi et al. (2020) then use an Online Mirror Descent based algorithm for the adversarial setting. Della Vecchia and Cesari (2021) extended this idea in the combinatorial setting, where players can pull multiple arms.

Similarly to multiplayer bandits, the problem of multi-agent bandits is wide and many directions remain to be explored. For instance, Vial et al. (2020) recently proposed an algorithm that is robust to malicious players. While malicious players cannot create collisions on purpose here, they can still send corrupted information to their neighbors, leading to bad behaviors.

3.6.2 Competing bandits

The problem of competing bandits was first introduced by Liu et al. (2020b), motivated by decentralized learning processes in matching markets. This model is very similar to the heterogeneous multiplayer bandits: they only differ in their collision model. Here, arms also have preferences over players: $j \succ_k j'$ means that the arm k prefers being pulled by the player j over j'. When several players pull the same arm k, only the top-ranked player for arm k gets its reward, while the others receive no reward. Mathematically the collision indicator is thus defined as:

$$\eta_k^j(t) = \mathbb{1}\left(\exists j' \succ_k j \text{ such that } \pi^{j'}(t) = k\right).$$

As often in bipartite matching problems, the goal is thus to reach a stable matching between players and arms. A matching is *stable* if every unmatched pair (j, k) would prefer to be matched. Mathematically, this corresponds to the following definition.

Definition 3.2. *A matching $\pi : [M] \to [K]$ is stable if for all $j \neq j'$, either $\mu^j_{\pi(j)} > \mu^j_{\pi(j')}$ or $j' \succ_{\pi(j')} j$ and for all unmatched arms k, $\mu^j_{\pi(j)} > \mu^j_k$.*

Several stable matchings can exist. Two different definitions of individual regret then appear. First the *optimal regret* compares with the best possible arm for player j in a stable matching, noted $\overline{k}_j$:

$$\overline{R}_j(T) = \mu^j_{\overline{k}_j} T - \sum_{t=1}^{T} \mu^j_{\pi^j(t)} \cdot (1 - \eta^j_{\pi^j(t)}(t)).$$

Similarly, the pessimal regret is defined with respect to the worst possible arm for player j in a stable matching, noted $\underline{k}_j$:

$$\underline{R}_j(T) = \mu^j_{\underline{k}_j} T - \sum_{t=1}^{T} \mu^j_{\pi^j(t)} \cdot (1 - \eta^j_{\pi^j(t)}(t)).$$

Liu et al. (2020b) propose a centralized UCB algorithm, where at each time step, the players send their UCB indexes to a central agent. This agent computes the optimal stable matching based on these indexes using the celebrated Gale Shapley algorithm and the players then pull according to the output of Gale Shapley algorithm. Although being natural, this algorithm only reaches a logarithmic regret for the pessimal definition, but can still incur a linear optimal regret.

Cen and Shah (2021) showed that a logarithmic optimal regret is reachable for this algorithm, if the platform can also choose transfers between the players and arms. The idea is to smartly choose the transfers, so that the optimal matching is the only stable matching when taking into account these transfers.

Liu et al. (2020b) also propose an ETC algorithm reaching a logarithmic optimal regret. After the exploration, the central agent computes the Gale Shapley matching which is pulled until T. A decentralized version of this algorithm is even possible, as Gale Shapley can be run in times N^2 in a decentralized way when observing the collision indicators η^j_k. This decentralized algorithm yet requires prior knowledge of Δ. Basu et al. (2021) extend this algorithm without knowing Δ, but the regret is then of order $\log^{1+\varepsilon}(T)$ for a parameter ε.

Liu et al. (2020a) also propose a decentralized UCB algorithm with a collision avoidance mechanism. Yet their algorithm requires for the players to observe the actions of all other players at each time step and only incurs a pessimal regret of order $\log^2(T)$, besides having an exponential dependency in the number of players.

Because of the difficulty of the general problem, even with collision sensing, another line of work focuses on simple instances of arm preferences. For example, when players are *globally ranked*, i.e., all the arms have the same preference orders $\succ_k$, there is a unique stable matching.

Moreover, it can be computed with the Serial Dictatorship algorithm, where the first player chooses her best arm, the second player chooses her best available arm and so on. In particular for this case, the algorithm of Liu et al. (2020a) yields a $\log(T)$ regret with no exponential dependency in other parameters.

Using this simplified structure, Sankararaman et al. (2020) also propose a decentralized UCB algorithm with collision avoidance mechanism. Working in epochs of increasing size, players mark as blocked the arms declared by players of smaller ranks and only play UCB on the unblocked arms. Their algorithm yields a regret bound close to the lower bound, which is shown to be at least of order $R_j(T) = \Omega\left(\max\left(\frac{(j-1)\log(T)}{\Delta^2}, \frac{K\log(T)}{\Delta}\right)\right)$ for some instance[5]. The first term in the $\max$ is the number of collisions encountered with players of smaller ranks, while the second term is the usual regret in single player stochastic bandits.

Serial Dictatorship can lead to the unique stable matching even in more general settings than globally ranked players. In particular, this is the case when the preferences profile satisfy the uniqueness consistency. Basu et al. (2021) then adapt the aforementioned algorithm to this setting, by using a more subtle collision avoidance mechanism.

3.6.3 Queuing systems

Gaitonde and Tardos (2020a) extended the queuing systems introduced by Krishnasamy et al. (2016) to the multi-agent setting. Similarly to competing bandits, this problem might benefit from multiplayer bandits approaches.

In this model, players are queues with arrival rates λ_i. At each time step, a packet is generated within the queue i with probability λ_i and the arm (server) k has a clearing probability μ_k.

This model assumes some asynchronicity between the players as they have different arrival rates λ_i. Yet it remains different from the usual asynchronous setting (Bonnefoi et al., 2017), as players can play as long as they have remaining packets.

When several players send packets to the same arm, it only treats the oldest received packet and clears it with probability μ_k, i.e., when colliding, only the queue with the oldest packet gets to pull the arm. A queue is said *stable* when its number of packets grows almost surely as $o(t)$.

A crucial quantity of interest is the largest real η such that

$$\eta \sum_{i=1}^{k} \lambda_{(i)} \leq \sum_{i=1}^{k} \mu_{(i)} \qquad \text{for all } k \in [M].$$

In the centralized case, stability of all queues is possible if and only if $\eta > 1$.

[5] Optimal and pessimal regret coincide here as there is a unique stable matching.

Gaitonde and Tardos (2020a) study whether a similar result is possible in the decentralized case where players are strategic. They first show that if players follow *suitable* no regret strategies, stability is reached if $\eta > 2$. Yet, for smaller values of η, no regret strategies can still lead to unstable queues.

In a subsequent work (Gaitonde and Tardos, 2020b), they claim that minimizing the regret is not a good objective as it leads to myopic behaviors of the players. Players here might prefer to be patient, as there is a carryover feature over the rounds. The issue of a round indeed depends on the past as a server treats the oldest packet sent by a player. A player thus can have interest in letting the other players to clear their packets, as it guarantees her to avoid colliding with them in the future.

To illustrate this point, Gaitonde and Tardos (2020b) consider the following *patient game*: all players have perfect knowledge of λ and μ and play a fixed probability distribution $\boldsymbol{p}$. The cost incurred by a player is then the asymptotic value $\lim_{t\to+\infty} \frac{Q_t^i}{t}$, where Q_t^i is the age of the oldest remaining packet of player i at time t.

Theorem 3.3 (Gaitonde and Tardos 2020b)**.** *If $\eta > \frac{e}{e-1}$ and all players follow a Nash equilibrium of the patient game described above, the system is stable.*

When players are patient, the limit ratio η where the system is stable is thus smaller. Yet this result holds only without learning consideration. Whether such a result is valid when players follow learning strategies remained an open question.

In Chapter 7, we argue that even patient learners might be unstable for $\eta < 2$, if they selfishly minimize some (patient) form of regret. In the light of this result, assuming cooperation between the learning agents seem required for stability with small values of η. We thus propose a first decentralized learning strategy that is stable as long as $\eta > 1$, thus being comparable to centralized strategies. Moreover, this algorithm converges to a correlated Nash equilibrium of the patient game described above.

3.7 Summary table

Tables 3.3 and 3.4 below summarize the theoretical guarantees of the algorithms presented in this survey. Unfortunately, some significant algorithms such as GoT (Bistritz and Leshem, 2020) are omitted, as the explicit dependencies of their upper bounds with other problem parameters than T are unknown and not provided in the original papers.

Algorithms using baselines different from the optimal matching in the regret definition are also omitted, as they can not be easily compared with other algorithms. This includes algorithms

taking only a stable matching as baseline in the heterogeneous case, or algorithm which are robust to jammers for instance.

Here is a list of the different notations used in Tables 3.3 and 3.4.

Δ	$= \min\{U^* - U(\pi) > 0 \mid \pi \in \mathcal{M}\}$
$\Delta_{(m,k)}$	$= \min\{U^* - U(\pi) > 0 \mid \pi \in \mathcal{M} \text{ and } \pi(m) = k\}$
$\overline{\Delta}_{(M)}$	$= \min_{k \leq M} \mu_{(k)} - \mu_{(k+1)}$
δ	arbitrarily small positive constant
$\mu_{(k)}$	k-th largest mean (homogeneous case)
M	number of players simultaneously in the game
$\mathcal{M}$	set of matchings between arms and players
N	total number of players entering/leaving the game (dynamic)
attackability	length of longest time sequence with successive $X_k(t) = 0$
rank	different ranks are attributed beforehand to players
T	horizon
$U(\pi)$	$= \sum_{m=1}^{M} \mu^m_{\pi(m)}$
U^*	$= \max_{\pi \in \mathcal{M}} U(\pi)$

Model	**Reference**	**Prior knowledge**	**Extra consideration**	**Upper bound**
Centralized	CUCB [83]	M	-	$\sum_{m=1}^{M}\sum_{k=1}^{K} \frac{M \log(T)}{\Delta_{(m,k)}}$
Centralized	CTS [237]	M	Independent arms	$\sum_{m=1}^{M}\sum_{k=1}^{K} \frac{\log^2(M) \log(T)}{\Delta_{(m,k)}}$
Coll. sensing	dE^3 [176]	T, Δ, M	communicating players	$M^3 K^2 \frac{\log(T)}{\Delta^2}$
Coll. sensing	D-MUMAB[165]	T, Δ	unique optimal matching	$\frac{M \log(T)}{\Delta^2} + \frac{K M^3 \log(\frac{1}{\Delta}) \log(T)}{\log(M)}$
Coll. sensing	ELIM-ETC [61]	T	$\delta = 0$ if unique optimal matching	$\sum_{k=1}^{K}\sum_{m=1}^{M} \left(\frac{M^2 \log(T)}{\Delta_{(m,k)}}\right)^{1+\delta}$

Table 3.3: Summary of presented algorithms in the heterogeneous setting. The last column provides the asymptotic upper bound, up to some universal multiplicative constant.

Model	Reference	Prior knowledge	Extra consideration	Upper bound
Centralized	MP-TS [141]	M	-	$\sum_{k>M} \frac{\log(T)}{\mu_{(M)}-\mu_{(k)}}$
Full sensing	SIC-GT [58]	T	$\mathcal{O}(\log(T))$-Nash equilibrium	$\sum_{k>M} \frac{\log(T)}{\mu_{(M)}-\mu_{(k)}} + MK^2\log(T)$
Stat. sensing	MCTopM [44]	M	-	$M^3 \sum_{1\leq i<k\leq K} \frac{\log(T)}{(\mu_{(i)}-\mu_{(k)})^2}$
Stat. sensing	RR-SW-UCB# [238]	T, M, rank	$\mathcal{O}(T^\nu)$ changes of $\boldsymbol{\mu}$	$\frac{K^2M}{\Delta^2} T^{\frac{1+\nu}{2}} \log(T)$
Stat. sensing	Selfish-Robust MMAB [58]	T	$\mathcal{O}(\log(T))$-Nash equilibrium	$M\sum_{k>M} \frac{\log(T)}{\mu_{(M)}-\mu_{(k)}} + \frac{MK^3}{\mu_{(K)}}\log(T)$
Coll. sensing	MEGA [25]	-	-	$M^2KT^{\frac{2}{3}}$
Coll. sensing	MC [198]	$T, \mu_{(M)}-\mu_{(M+1)}$	-	$\frac{MK\log(T)}{(\mu_{(M)}-\mu_{(M+1)})^2}$
Coll. sensing	SIC-MMAB [59]	T	-	$\sum_{k>M} \frac{\log(T)}{\mu_{(M)}-\mu_{(k)}} + MK\log(T)$
Coll. sensing	DPE1 [235]	T	-	$\sum_{k>M} \frac{\log(T)}{\mu_{(M)}-\mu_{(k)}}$
Coll. sensing	C&P[7]	T	Adversarial rewards	$K^{\frac{4}{3}}M^{\frac{2}{3}}\log(M)^{\frac{1}{3}}T^{\frac{2}{3}}$
Coll. sensing	[72]	T, rank, two players	Adversarial rewards	$K^2\sqrt{T\log(K)\log(T)}$
No sensing	[162]	T, M	-	$\frac{MK\log(T)}{(\mu_{(M)}-\mu_{(M+1)})^2}$
No sensing	[162]	$T, M, \mu_{(M)}$	-	$\frac{MK^2}{\mu_{(M)}}\log^2(T) + MK\frac{\log(T)}{\Delta}$
No sensing	[211]	$T, \mu_{(K)}, \Delta$	-	$\sum_{k>M} \frac{\log(T)}{\mu_{(M)}-\mu_{(k)}} + M^2K\frac{\log(\frac{1}{\Delta})\log(T)}{\mu_{(K)}}$
No sensing	[127]	T	-	$\sum_{k>M} \frac{\log(T)}{\mu_{(M)}-\mu_{(k)}} + MK^2\log(\frac{1}{\Delta})^2\log(T)$
No sensing	A2C2 [210]	T, M, α	Adversarial rewards attackability $\mathcal{O}(T^\alpha)$	$M^{\frac{4}{3}}K^{\frac{1}{3}}\log(K)^{\frac{2}{3}}T^{\frac{2+\alpha+\delta}{3}}$
No sensing	[71]	M, rank shared randomness	No collision with high proba	$MK^{\frac{11}{2}}\sqrt{T\log(T)}$
No sensing	[72]	M, rank	Adversarial rewards	$MK^{\frac{3}{2}}T^{1-\frac{1}{2M}}\sqrt{\log(K)}$
No sensing No zero collision ($M \geq K$)	[31]	T, M, Δ	Small variance of noise	$\frac{KM}{\Delta^2}e^{\frac{M-1}{K-1}}\log(T)$
No sensing No zero collision ($M \geq K$)	[32]	M, rank	-	$\frac{M^3K}{\Delta^2}\log(T)$
Dynamic, coll. sensing	DMC [198]	$T, \overline{\Delta}_{(M)}$	-	$\frac{M\sqrt{K\log(T)T}}{\overline{\Delta}^2_{(M)}}$
Dynamic, coll. sensing	[31]	T	Adversarial rewards $N = \mathcal{O}\left(\sqrt{T}\right)$	$\frac{K^{K+2}}{\sqrt{K\log(K)}}T^{\frac{3}{4}} + NK\sqrt{T}$
Dynamic, no sensing	DYN-MMAB[59]	T	All players end at T	$\frac{MK\log(T)}{\overline{\Delta}^2_{(M)}} + \frac{M^2K\log(T)}{\mu_{(M)}}$

Table 3.4: Summary of presented algorithms in the homogeneous setting. The last column provides the asymptotic upper bound, up to some universal multiplicative constant.

Chapter 4

SIC-MMAB: Synchronisation Involves Communication in Multiplayer Multi-Armed Bandits

This chapter presents a decentralized algorithm that achieves the same performance as a centralized one for homogeneous multiplayer bandits, by "hacking" the standard model with a communication protocol between players that deliberately enforces collisions, allowing them to share their information at a negligible cost. This motivates the introduction of a more appropriate dynamic setting without sensing, where similar communication protocols are no longer possible. However, we show that the logarithmic growth of the regret is still achievable for this model with a new algorithm.

This chapter considers the homogeneous multiplayer bandits problem introduced in Section 3.3.1 and presents the following contributions.

With collision sensing, Section 4.1 introduces a new decentralized algorithm that is "hacking" the setting and induces communication between players through deliberate collisions. The regret of this algorithm, called SIC-MMAB, reaches asymptotically (up to some universal constant) the lower bound of the centralized problem, contradicting the previously believed lower bounds. SIC-MMAB relies on the unrealistic assumption that all users start transmitting at the very same time. It therefore appears that the assumption of synchronization has to be removed for practical considerations.

Without synchronization or collision observations, Section 4.2 proposes the first algorithm with a logarithmic regret. The dependencies in the gaps between arm means yet become quadratic.

We compare empirically SIC-MMAB with MCTOPM (Besson and Kaufmann, 2018a) on a toy example in Section 4.A. Especially, it nicely illustrates how SIC-MMAB scales better with the suboptimality gap and also confirms its smaller minimax regret bound.

Besson and Kaufmann (2018a) studied the SELFISH algorithm, consisting in unilaterally following UCB algorithm, and conjectured that it leads to a linear regret with positive (constant) probability. We prove this conjecture for agents with infinite calculus precision. Yet the question remains open for machines with finite precision.

4.1 Collision Sensing: achieving centralized performances by communicating through collisions

In this section, we consider the Collision Sensing model of Section 3.3.1 and prove that the decentralized problem is almost as complex, in terms of regret growth, as the centralized one. When players are synchronized, we provide an algorithm with an exploration regret similar to the known centralized lower bound (Anantharam et al., 1987a). This algorithm strongly relies on the synchronization assumption, which we leverage to allow communication between players through observed collisions. The communication protocol is detailed and explained in Section 4.1.2. This result also implies that the two lower bounds provided in the literature (Besson and Kaufmann, 2018a; Liu and Zhao, 2010) are unfortunately not correct. Indeed, the factor M that was supposed to be the cost of the decentralization in the regret should not appear.

Let us describe our algorithm SIC-MMAB. It consists of several phases.

1. The initialization phase first estimates the number of players and assigns ranks among them.

2. Players then alternate between exploration phases and communication phases.

   (a) During the p-th exploration phase, each arm is pulled 2^p times and its performance is estimated in a Successive Accepts and Rejects fashion (Perchet and Rigollet, 2013; Bubeck et al., 2013).

   (b) During the communication phases, players communicate their statistics to each other using collisions. Afterwards, the updated common statistics are known to all players.

3. The last phase, the exploitation one, is triggered for a player as soon as an arm is detected as optimal and assigned to her. This player then pulls this arm until the final horizon T.

4.1.1 Some preliminary notations

Players that are not in the exploitation phase are called **active**. We denote, with a slight abuse of notation, by $[M_p]$ the set of active players during the p-th phase of exploration-communication and by $M_p \leq M$ its cardinality. Notice that M_p is non increasing because players never leave the exploitation phase.
Each arm among the top-M ones is called **optimal** and each other arm is **sub-optimal**. Arms that still need to be explored (players cannot determine whether they are optimal or sub-optimal yet) are **active**. We denote, with the same abuse of notation, the set of active arms by $[K_p]$ of cardinality $K_p \leq K$. By construction of our algorithm, this set is common to all active players at each stage.

Our algorithm is based on a protocol called *sequential hopping* (Joshi et al., 2018). It consists of incrementing the index of the arm pulled by a specific player m: if she plays arm $\pi^m(t)$ at time t, she will play $\pi^m(t+1) = \pi^m(t)+1 \pmod{[K_p]}$ at time $t+1$ during the p-th exploration phase.

4.1.2 Description of our protocol

As mentioned above, the SIC-MMAB algorithm consists of several phases. During the communication phase, players communicate with each other. At the end of this phase, each player thus knows the statistics of all players on all arms, so that this decentralized problem becomes similar to the centralized one. After alternating enough times between exploration and communication phases, sub-optimal arms are eliminated and players are fixed to different optimal arms and will exploit them until stage T. The complete pseudocode of SIC-MMAB is given by Algorithm 4.6.

Initialization phase

The objective of the first phase is to estimate the number of players M and to assign **internal ranks** to players. First, players follow the Musical Chairs algorithm (Rosenski et al., 2016), described by Algorithm 4.1 below, during $T_0 := \lceil K \log(T) \rceil$ steps in order to reach an **orthogonal setting**, i.e., a position where they are all pulling different arms. The index of the arm pulled by a player at stage T_0 will then be her **external rank**.

Algorithm 4.1: MusicalChairs Protocol

input: $[K_p]$ (active arms), T_0 (time of procedure)
Initialize `Fixed` $\leftarrow -1$
for T_0 *time steps* **do**
  if $Fixed = -1$ **then**
    Sample k uniformly at random in $[K_p]$ and play it in round t
    if $\eta_k(t) = 0$ *($r_k(t) > 0$ for No Sensing setting)* **then** `Fixed` $\leftarrow k$ `// player stays in arm k if no collision`
  end
  else Play `Fixed`
end
return `Fixed` `// External rank`

The second procedure, given by Algorithm 4.2, determines M and assigns a unique internal rank in $[M]$ to each player. For example, if there are three players on arms 5, 7 and 2 at $t = T_0$, their external ranks are 5, 7 and 2 respectively, while their internal ranks are 2, 3 and 1. Roughly speaking, the players follow each other sequentially hopping through all the arms so that players with external ranks k and k' collide exactly after a time $k + k'$. Each player then deduces M and her internal rank from observed collisions during this procedure that lasts $2K$ steps.

Algorithm 4.2: Estimate_M Protocol

input: $k \in [K]$ (external rank)
Initialize $\hat{M} \leftarrow 1$, $j \leftarrow 1$ and $\pi \leftarrow k$ `// estimates of` M `and the internal rank`
for $2k$ *time steps* **do**
  Pull π
  if $\eta_\pi(t) = 1$ **then** $hatM \leftarrow hatM + 1$ and $j \leftarrow j + 1$ `// increases if collision`
end
for $2(K - k)$ *time steps* **do**
  $\pi \leftarrow \pi + 1 \pmod K$ and pull π `// sequential hopping`
  if $\eta_\pi(t) = 1$ **then** $hatM \leftarrow hatM + 1$ `// increases if collision`
end
return $hatM, j$

In the next phases, active players will always know the set of active players $[M_p]$. This is how the initial symmetry among players is broken and it allows the decentralized algorithm to establish communication protocols.

Exploration phase

During the p-th exploration phase, active players sequentially hop among the active arms for $K_p 2^p$ steps. Each active arm is thus pulled 2^p times by each active player. Using their internal rank, players start and remain in an orthogonal setting during the exploration phase, which is collision-free.

We denote by $B_s = 3\sqrt{\frac{\log(T)}{2s}}$ the error bound after s pulls and by $N_k(p)$ (resp. $S_k(p)$) the centralized number of pulls (resp. sum of rewards) for the arm k during the p first exploration phases, i.e., $N_k(p) = \sum_{j=1}^{M} N_k^j(p)$ where $N_k^m(p)$ is the number of pulls for the arm k by player m during the p first exploration phases. During the communication phase, quantized rewards $\widetilde{S}_k^m(p)$ will be communicated between active players as described in Section 4.1.2.

After a succession of two phases (exploration and communication), an arm k is **accepted** if

$$\#\Big\{i \in [K_p] \,\Big|\, \widetilde{\mu}_k(p) - B_{N_k(p)} \geq \widetilde{\mu}_i(p) + B_{N_i(p)}\Big\} \geq K_p - M_p,$$

where $\widetilde{\mu}_k(p) = \frac{\sum_{m=1}^{M} \widetilde{S}_k^m(p)}{N_k(p)}$ is the centralized quantized empirical mean of the arm k[1], which is an approximation of $\hat{\mu}_k(p) = \frac{S_k(p)}{N_k(p)}$. This inequality implies that k is among the top-M_p active arms with high probability. In the same way, k is **rejected** if

$$\#\Big\{i \in [K_p] \,\Big|\, \widetilde{\mu}_i(p) - B_{N_i(p)} \geq \widetilde{\mu}_k(p) + B_{N_k(p)}\Big\} \geq M_p,$$

meaning that there are at least M_p active arms better than k with high probability. Notice that each player j uses her own quantized statistics $\widetilde{S}_k^j(p)$ to accept/reject an arm instead of the exact ones $S_k^j(p)$. Otherwise, the estimations $\widetilde{\mu}_k(p)$ would indeed differ between the players as well as the sets of accepted and rejected arms. With Bernoulli distributions, the quantization becomes unnecessary and the confidence bound can be chosen as $B_s = \sqrt{2\log(T)/s}$.

Communication phase

In this phase, each active player communicates, one at a time, her statistics of the active arms to all other active players. Each player has her own communicating arm, corresponding to her internal rank. When the player j is communicating, she sends a bit at a time step to the player l by deciding which arm to pull: a 1 bit is sent by pulling the communicating arm of player

[1]For a player m already exploiting since the p^m-th phase, we instead use the last statistic $\widetilde{S}_k^m(p) = \widetilde{S}_k^m(p^m)$.

l (a collision occurs) and a 0 bit by pulling her own arm. The main originality of SIC-MMAB comes from this trick which allows implicit communication through collisions and is used in subsequent papers as explained in Section 3.4.2. In an independent work, Tibrewal et al. (2019) also proposed using similar communication protocols for the heterogeneous case.

As an arm is pulled 2^n times by a single player during the n-th exploration phase, it has been pulled $2^{p+1}-1$ times in total at the end of the p-th phase and the statistic $S_k^j(p)$ is a real number in $[0, 2^{p+1}-1]$. Players then send a quantized **integer** statistic $\widetilde{S}_k^j(p) \in [2^{p+1}-1]$ to each other in $p+1$ bits, i.e., collisions. Let $n = \lfloor S_k^j(p) \rfloor$ and $d = S_k^j(p) - n$ be the integer and decimal parts of $S_k^j(p)$, the quantized statistic is then $n+1$ with probability d and n otherwise, so that $\mathbb{E}[\widetilde{S}_k^j(p)] = S_k^j(p)$.

Algorithm 4.3: Receive Protocol

input: p (phase number), l (own internal rank), $[K_p]$ (active arms)
$s \leftarrow 0$ and $\pi \leftarrow$ index of l-th active arm
for $n = 0, \dots, p$ **do**
  Pull π
  if $\eta_\pi(t) = 1$ **then** $s \leftarrow s + 2^n$ `// other player sends 1`
end
return s (statistic sent by other player)

Algorithm 4.4: Send Protocol

input: l (player receiving), s (statistics to send), p (phase number), j (own internal rank), $[K_p]$ (active arms)
$s \leftarrow 0$ and $\pi \leftarrow$ index of the l-th active arm
$\mathbf{m} \leftarrow$ binary writing of s of length $p+1$, i.e., $s = \sum_{n=0}^{p} m_n 2^n$
for $n = 0, \dots, p$ **do**
  if $m_n = 1$ **then** Pull the l-th active arm `// send 1`
  else Pull the j-th active arm `// send 0`
end

An active player can have three possible statuses during the communication phase:

1. either she is receiving some other players' statistics about the arm k. In that case, she proceeds to **Receive Protocol** (see Algorithm 4.3).

2. Or she is sending her quantized statistics about arm k to player l (who is then receiving). In that case, she proceeds to **Send Protocol** (see Algorithm 4.4) to send them in a time $p+1$.

3. Or she is pulling her communicating arm, while waiting for other players to finish communicating statistics among them.

Communicated statistics are all of length $p+1$, even if they could be sent with shorter messages, in order to maintain synchronization among players. Using their internal ranks, the players can communicate in turn without interfering with each other. The general protocol for each communication phase is described in Algorithm 4.5 below.

Algorithm 4.5: Communication Protocol

input: $\mathbf{s}$ (personal statistics of previous phases), p (phase number), j (own internal rank), $[K_p]$ (active arms), $[M_p]$ (active players)

For all k, sample $\widetilde{s}[k] = \begin{cases} \lfloor s[k] \rfloor + 1 \text{ with probability } s[k] - \lfloor s[k] \rfloor \\ \lfloor s[k] \rfloor \text{ otherwise} \end{cases}$ `// quantize`

Define $E_p \coloneqq \{(i,l,k) \in [M_p] \times [M_p] \times [K_p] \mid i \neq l\}$ and set $\widetilde{\mathbf{S}}^{\mathbf{j}} \leftarrow \widetilde{\mathbf{s}}$

for $(i,l,k) \in E_p$ **do** `// Player i sends stats of arm k to player l`

  if $i = j$ **then** Send $(l, \widetilde{s}[k], p, j, [K_p])$ `// sending player`

  else if $l = j$ **then** $\widetilde{S}^i[k] \leftarrow$ Receive$(p, j, [K_p])$ `// receiving player`

  else

    for $p+1$ *rounds* **do** pull j-th active arm `// wait while others communicate`

  end

end

return $\widetilde{\mathbf{S}}$

At the end of the communication phase, all active players know the statistics $\widetilde{S}_k^j(p)$ and so which arms to accept or reject. Rejected arms are removed right away from the set of active arms. Thanks to the assigned ranks, accepted arms are assigned to one player each. The remaining active players then update both sets of active players and arms as described in Algorithm 4.6, Line 21.

This communication protocol uses the fact that a bit can be sent with a single collision. Without sensing, this can not be done in a single time step, but communication is still somehow possible. A bit can then be sent in $\frac{\log(T)}{\mu_{(K)}}$ steps with probability $1 - \frac{1}{T}$. Using this trick, two different algorithms relying on communication protocols were proposed No Sensing setting in the conference version of this chapter (Boursier and Perchet, 2019).

Regret bound of SIC-MMAB

Theorem 4.1 bounds the expected regret incurred by SIC-MMAB and its proof is delayed to Section 4.B.1.

Theorem 4.1. *With the choice $T_0 = \lceil K \log(T) \rceil$, for any given set of parameters K, M and $\boldsymbol{\mu}$*

such that the arm means are distinct, $\mu_{(1)} > \mu_{(2)} > \ldots > \mu_{(K)}$, the regret is bounded as

$$R(T) \leq c_1 \sum_{k>M} \min\left\{\frac{\log(T)}{\mu_{(M)} - \mu_{(k)}}, \sqrt{T\log(T)}\right\} + c_2 KM\log(T) + c_3 KM^3 \log^2\left(\min\left\{\frac{\log(T)}{(\mu_{(M)} - \mu_{(M+1)})^2}, T\right\}\right)$$

where c_1, c_2 and c_3 are universal constants.

Algorithm 4.6: SIC-MMAB algorithm

input: T (horizon)

Initialization Phase:

Initialize `Fixed` $\leftarrow -1$ and $T_0 \leftarrow \lceil K\log(T)\rceil$

$k \leftarrow$ MusicalChairs $([K], T_0)$

$(M, j) \leftarrow$ Estimate_M (k) `// estimated number of players and internal rank`

Initialize $p \leftarrow 1$; $M_p \leftarrow M$; $[K_p] \leftarrow [K]$ and $\widetilde{\mathbf{S}}, \mathbf{s}, \mathbf{N} \leftarrow$ Zeros(K) `// Zeros(K) returns a vector of length K containing only zeros`

while `Fixed`$= -1$ **do**

  Exploration Phase:

  $\pi \leftarrow j$-th active arm `// start of a new phase`

  for $K_p 2^p$ *time steps* **do**

    $\pi \leftarrow \pi + 1 \pmod{[K_p]}$ and play π in round t `// sequential hopping`

    $s[\pi] \leftarrow s[\pi] + r_\pi(t)$ `// Update individual statistics`

  end

  Communication Phase:

  $\widetilde{\mathbf{S}}_\mathbf{p} \leftarrow$ Communication($\mathbf{s}$, p, j, $[K_p]$, $[M_p]$) and $\widetilde{\mathbf{S}}^\mathbf{l} \leftarrow \widetilde{\mathbf{S}}^\mathbf{l}_\mathbf{p}$ for every active player l

  $N[k] \leftarrow N[k] + M_p 2^p$ for every active arm k

  Update Statistics: `// recall that` $B_s = 3\sqrt{\frac{\log(T)}{2s}}$ `here`

  Rej $\leftarrow$ set of active arms k verifying $\#\Big\{i \in [K_p] \mid \frac{\sum_{l=1}^{M} \widetilde{S}^l[i]}{N[i]} - B_{N[i]} \geq \frac{\sum_{l=1}^{M} \widetilde{S}^l[k]}{N[k]} + B_{N[k]}\Big\} \geq M_p$

  Acc $\leftarrow$ set of active arms k verifying $\#\Big\{i \in [K_p] \mid \frac{\sum_{l=1}^{M} \widetilde{S}^l[k]}{N[k]} - B_{N[k]} \geq \frac{\sum_{l=1}^{M} \widetilde{S}^l[i]}{N[i]} + B_{N[i]}\Big\} \geq K_p - M_p$

  if $M_p - j + 1 \leq$ `length`(Acc) **then** `Fixed` $\leftarrow$ Acc$[M_p - j + 1]$ `// exploit`

  else `// update all the statistics`

    $M_p \leftarrow M_p -$ `length(`Acc`)` and $[K_p] \leftarrow [K_p] \setminus$ (Acc $\cup$ Rej)

  end

  $p \leftarrow p + 1$

end

Exploitation Phase: Pull `Fixed` until T

The first, second and third terms respectively correspond to the regret incurred by the exploration, initialization and communication phases, which dominate the regret due to low probability events of bad initialization or incorrect estimations. Notice that the minmax regret scales with $\mathcal{O}(K\sqrt{T\log(T)})$.
Experiments on synthetic data are described in Section 4.A. They empirically confirm that SIC-MMAB scales better than MCTopM (Besson and Kaufmann, 2018a) with the gaps Δ, besides having a smaller minmax regret.

4.1.3 In contradiction with lower bounds?

Theorem 4.1 is in contradiction with the two lower bounds by Besson and Kaufmann (2018a) and Liu and Zhao (2010), however SIC-MMAB respects the conditions required for both. It was thought that the decentralized lower bound was $\Omega\left(M\sum_{k>M}\frac{\log(T)}{\mu_{(M)}-\mu_{(k)}}\right)$, while the centralized lower bound was already known to be $\Omega\left(\sum_{k>M}\frac{\log(T)}{\mu_{(M)}-\mu_{(k)}}\right)$ (Anantharam et al., 1987a). However, it appears that the asymptotic regret of the decentralized case is not that much different from the latter, at least if players are synchronized. Indeed, SIC-MMAB takes advantage of this synchronization to establish communication protocols as players are able to communicate through collisions. The subsequent paper by Proutiere and Wang (2019) later improved the communication protocols of SIC-MMAB to obtain both initialization and communication costs constant in T, confirming that the lower bound of the centralized case is also tight for the decentralized model considered so far.

Liu and Zhao (2010) proved the lower bound "by considering the best case that they do not collide". This is only true if colliding does not provide valuable information and the policies just maximize the losses at each round, disregarding the information gathered for the future. Our algorithm is built upon the idea that the value of the information provided by collisions can exceed in the long run the immediate loss in rewards (which is standard in dynamic programming or reinforcement learning for instance). The mistake of Besson and Kaufmann (2018a) is found in the proof of Lemma 12 after the sentence "We now show that second term in (25) is zero". The conditional expectation cannot be put inside/outside of the expectation as written and the considered term, which corresponds to the difference of information given by collisions for two different distributions, is therefore not zero.
These two lower bounds disregarded the amount of information that can be deduced from collisions, while SIC-MMAB obviously takes advantage from this information.

Our exploration regret reaches, up to a constant factor, the lower bound of the centralized problem (Anantharam et al., 1987a). Although it is sub-logarithmic in time, the communication cost scales with KM^3 and can thus be predominant in practice. Indeed for large networks, M^3

can easily be greater than $\log(T)$ and the communication cost would then prevail over the other terms. This highlights the importance of the parameter M in multiplayer MAB and future work should focus on the dependency in both M and T instead of only considering asymptotic results in T. The communication scheme of SIC-MMAB is improved in Chapter 5, which reduces its total cost by a factor larger than M.

Synchronization is not a reasonable assumption for practical purposes and it also leads to undesirable algorithms relying on communication protocols such as SIC-MMAB. We thus claim that this assumption should be removed in the multiplayer MAB and the *dynamic model* should be considered instead. However, this problem seems complex to model formally. Indeed, if players stay in the game only for a very short period, learning is not possible. The difficulty to formalize an interesting and nontrivial dynamic model may explain why most of the literature focused on the static model so far.

4.2 Without synchronization, the dynamic setting

In the previous section, it was crucial that all exploration/communication phases start and end at the same time for the SIC-MMAB algorithm. The synchronization assumption we leveraged was the following.

Assumption 4.1 (Synchronization)**.** *Player i enters the bandit game at the time $\tau_i = 0$ and stays until the final horizon T. This is common knowledge to all players.*

From now on, we no longer assume that players can communicate using synchronization. This assumption is clearly unrealistic and should be alleviated, as radios do not start and end transmitting simultaneously.

We instead assume in the following that players do not leave the game once they have started, as formalized by Assumption 4.2 below.

Assumption 4.2 (Quasi-Asynchronization)**.** *Players enter at different times $\tau_i \in \{0, \dots, T-1\}$ and stay until the final horizon T. The τ_i are unknown to all players (including i).*

Yet, we mention that our results can also be adapted to the cases when players can leave the game during specific intervals or share an internal synchronized clock (Rosenski et al., 2016). If the time is divided in several intervals, DYN-MMAB can be run independently on each of these intervals as suggested by Rosenski et al. (2016). In some cases, players will be leaving in the middle of these intervals, leading to a large regret. But for any other interval, every player stays until its end, thus satisfying Assumption 4.2.

With quasi-asynchronicity[2], the model is **dynamic** and several variants already exist (Rosenski et al., 2016). Denote by $\mathbf{M}(t)$ the set of players in the game at time t (unknown but not random). The total regret is then defined for the dynamic model (it is also valid for the static one) by:

$$R(T) \coloneqq \sum_{t=1}^{T} \sum_{k=1}^{\#\mathbf{M}(t)} \mu_{(k)} - \mathbb{E}_{\mu}\left[\sum_{t=1}^{T} \sum_{m \in \mathbf{M}(t)} r^{m}(t)\right].$$

In this section, Assumption 4.2 holds. At each stage $t = t_j + \tau_j$, player j does not know t but only t_j (duration since joining). We denote by $T^j = T - \tau_j$ the (known) time horizon of player j. We also consider the more difficult No Sensing setting in this section.

4.2.1 A logarithmic regret algorithm

As synchronization no longer holds, we propose the DYN-MMAB algorithm, relying on different tools than SIC-MMAB. The main ideas of DYN-MMAB are given in Section 4.2.2, while its thorough description is given in 4.2.3.

The regret incurred by DYN-MMAB in the dynamic No Sensing model is given by Theorem 4.2 and its proof is delayed to Section 4.B.2. We also mention that DYN-MMAB leads to a Pareto optimal configuration in the more general problem where users' reward distributions differ (Avner and Mannor, 2014; Avner and Mannor, 2015; Avner and Mannor, 2019; Bistritz and Leshem, 2018).

Theorem 4.2. *In the dynamic setting, the regret incurred by* DYN-MMAB *is upper bounded as follows:*

$$R(T) = \mathcal{O}\left(\frac{M^2 K \log(T)}{\mu_{(M)}} + \frac{MK\log(T)}{\overline{\Delta}^2_{(M)}}\right),$$

where $M = \#\mathbf{M}(T)$ *is the total number of players in the game and* $\overline{\Delta}_{(M)} = \min_{i=1,\dots,M}(\mu_{(i)} - \mu_{(i+1)})$.

4.2.2 A communication-less protocol

DYN-MMAB's ideas are easy to understand but the upper bound proof is quite technical. This section gives some intuitions about DYN-MMAB and its performance guarantees stated in Theorem 4.2. A more detailed description is given in Section 4.2.3 below.

A player will only follow two different sampling strategies: either she samples uniformly at random in $[K]$ during the exploration phase; or she exploits an arm and pulls it until the final

[2]We prefer not to mention asynchronicity as players still use shared discrete time slots.

horizon. In the first case, the exploration of the other players is not too disturbed by collisions as they only change the mean reward of all arms by a common multiplicative term. In the second case, the exploited arm will appear as sub-optimal to the other players, which is actually convenient for them as this arm is now exploited.

During the exploration phase, a player will update a set of arms called `Occupied` $\subset [K]$ and an ordered list of arms called `Preferences` $\subset [K]$. As soon as an arm is detected as occupied (by another player), it is then added to `Occupied` (which is the empty set at the beginning). If an arm is discovered to be the best one amongst those that are neither in `Occupied` nor in `Preferences`, it is then added to `Preferences` (at the last position). An arm is **active** for player j if it was neither added to `Occupied` nor to `Preferences` by this player yet.

To handle the fact that players can enter the game at anytime, we introduce the quantity $\gamma^j(t)$, the expected multiplicative factor of the means defined by

$$\gamma^j(t) = \frac{1}{t} \sum_{t'=1+\tau_j}^{t+\tau_j} \mathbb{E}\Big[(1 - \frac{1}{K})^{m_{t'}-1}\Big],$$

where m_t is the number of players in their exploration phase at time t. The value of $\gamma^j(t)$ is unknown to the player and random but it only affects the analysis of DYN-MMAB and not how it runs.

The objective of the algorithm is still to form estimates and confidence intervals of the performances of arms. However, it might happen that the true mean μ_k does not belong to this confidence interval. Indeed, this is only true for $\gamma^j(t)\mu_k$, if the arm k is still free (not exploited). This is the first point of Lemma 4.1 below. Notice that as soon as the confidence interval for the arm i dominates the confidence interval for the arm k, then it must hold that $\gamma^j(t)\mu_i \geq \gamma^j(t)\mu_k$ and thus arm i is better than k.

The second crucial point is to detect when an arm k is exploited by another player. This detection will happen if a player receives too many 0 rewards successively (so that it is statistically very unlikely that this arm is not occupied). The number of zero rewards needed for player j to disregard arm k is denoted by L_k^j, which is sequentially updated during the process (following the rule of Equation (4.1) in Section 4.2.3), so that $L_k^j \geq 2e\log(T^j)/\mu_k$. As the probability of observing a 0 reward on a free arm k is smaller than $1 - \mu_k/e$, no matter the current number of players, observing L_k^j successive 0 rewards on an unexploited arm happens with probability smaller than $\frac{1}{(T^j)^2}$.

The second point of Lemma 4.1 then states that an exploited arm will either be quickly detected as occupied after observing L_k^j zeros (if L_k^j is small enough) or its average reward will quickly drop because it now gives zero rewards (and it will be dominated by another arm after a

relatively small number of pulls). The proof of Lemma 4.1 is delayed to Section 4.B.2.

Lemma 4.1. *We denote by $\hat{r}_k^j(t)$ the empirical average reward of arm k for player j at stage $t+\tau_j$.*

1. *For every player j and arm k, if k is still free at stage $t+\tau_j$, then*

$$\mathbb{P}\Big[|\hat{r}_k^j(t) - \gamma^j(t)\mu_k| > 2\sqrt{\frac{6\,K\log(T^j)}{t}}\Big] \leq \frac{4}{(T^j)^2}.$$

   *We then say that the arm k is **correctly estimated** by player j if $|\hat{r}_k^j(t) - \gamma^j(t)\mu_k| \leq 2\sqrt{\frac{6\,K\log(T^j)}{t}}$ holds as long as k is free.*

2. *On the other hand, if k is exploited by some player $j' \neq j$ at stage $t^0+\tau_j$, then, conditionally on the correct estimation of all the arms by player j, with probability $1-\mathcal{O}\left(\frac{1}{T^j}\right)$:*

   - *either k is added to* `Occupied` *at a stage at most $t^0+\tau_j+\mathcal{O}\left(\frac{K\log(T)}{\mu_k}\right)$ by player j,*
   - *or k is dominated by another unoccupied arm i (for player j) at stage at most $\mathcal{O}\left(\frac{K\log(T)}{\mu_i^2}\right)+\tau_j$.*

It remains to describe how players start exploiting arms. After some time (upper-bounded by Lemma 4.10 in Section 4.B.2), an arm which is still free and such that all better arms are occupied will be detected as the best remaining one. The player will try to occupy it, and this happens as soon as she gets a positive reward from it: either she succeeds and starts exploiting it, or she fails and assumes it is occupied by another player (this only takes a few number of steps, see Lemma 4.1). In the latter case, she resumes exploring until she detects the next available best arm. With high probability, the player will necessarily end up exploiting an arm while all the better arms are already exploited by other players.

4.2.3 DYN-MMAB description

This section thoroughly describes DYN-MMAB algorithm. Its pseudocode is given in Algorithm 4.7 below.

We first describe the rules explaining when a player adds an arm to `Occupied` or `Preferences`. An arm k is added to `Occupied` (it may already be in `Preferences`) if only 0 rewards have been observed during a whole block of L_k^j pulls on arm k for player j. Such a block ends when L_k^j observations have been gathered on arm k and a new block is then restarted. L_k^j is an estimation of the required number of successive 0 to observe before considering an arm as occupied

Algorithm 4.7: DYN-MMAB algorithm

input: T^j (personal horizon)

1 $p \leftarrow 1$, `Fixed` $\leftarrow -1$ and initialize `Preferences`, `Occupied` as empty lists
2 $\mathbf{N}, \mathbf{N}^{\text{temp}}, \mathbf{S}, \mathbf{S}^{\text{temp}} \leftarrow \text{Zeros}(K)$ and define $\mathbf{L}$ as a vector of K elements equal to ∞
3 $r_{\inf}[k] \leftarrow 0$ and $r_{\sup}[k] \leftarrow 1$ for every arm k `// Initialize the confidence intervals`
4 **while** `Fixed` $= -1$ **do** `//` $B^j(t) = 2\sqrt{\frac{6\ K\log(T^j)}{t}}$ `here`
5 Pull $k \sim \mathcal{U}([K])$; $N^{\text{temp}}[k] \leftarrow N^{\text{temp}}[k] + 1$ and $N[k] \leftarrow N[k] + 1$
6 $S^{\text{temp}}[k] \leftarrow S^{\text{temp}}[k] + r_k(t)$ and $S[k] \leftarrow S[k] + r_k(t)$
7 For all arms i, $r_{\inf}[i] \leftarrow \left(\frac{S[i]}{N[i]} - B^j(t)\right)_+$ and $r_{\sup}[i] \leftarrow \min\left(\frac{S[i]}{N[i]} + B^j(t), 1\right)$
8 $L[k] \leftarrow \min\left(\frac{2e\log(T^j)}{r_{\inf}[k]},\ L[k]\right)$
9 **if** $k =$ `Preferences`$[p]$ *and* $r_k(t) > 0$ **then** `Fixed` $\leftarrow k$ `// no collision on the arm to exploit`
10 **if** `Preferences`$[p] \in$ `Occupied` **then** $p \leftarrow p + 1$ `// exploited by another player`
11 **if** $S^{temp}[k] = 0$ **then** `//` k `is occupied`
12 Add k to `Occupied` ; Reset $S^{\text{temp}}[k], N^{\text{temp}}[k] \leftarrow 0$
13 **end**
14 **if** *for some active arm* i *and all other active arms* l, $r_{\inf}[i] > r_{\sup}[l]$ **then**
15 Add i to `Preferences` (last position) `//` i `is better than all other active arms`
16 **end**
17 **if** $\exists l \notin$ `Preferences`$[1:p]$ *such that* $r_{\inf}[l] > r_{\sup}[$`Preferences`$[p]]$ **then**
18 Add `Preferences`$[p]$ to `Occupied``// the mean of the available best arm has significantly dropped`
19 **end**
20 **end**
21 Pull `Fixed` until T^j `// Exploitation phase`

with high probability. Its value at stage $t+\tau_j$, $L_k^j(t)$, is thus constantly updated using the current estimation of a lower bound of μ_k:

$$L_k^j(t+1) \leftarrow \min\left(\frac{2e\log(T^j)}{\left(\hat{r}_k^j(t+1) - B^j(t+1)\right)_+},\ L_k^j(t)\right) \quad \text{and } L_k^j(0) = +\infty, \tag{4.1}$$

where $\hat{r}_k^j(t)$ is the empirical mean reward on the arm k at stage $t+\tau_j$, $B^j(t) = 2\sqrt{\frac{6\ K\log(T^j)}{t}}$, $x_+ = \max(x, 0)$ and $\frac{2e\log(T^j)}{0} = +\infty$. This rule is described at Line 12 in Algorithm 4.7.

An active arm k is added to `Preferences` (at last position) if it is better than all other

active arms, in term of confidence interval. This rule is described at Line 14 in Algorithm 4.7.

Another rule needs to be added to handle the possible case of an arm in `Preferences` already exploited by another player. As soon as an arm k in `Preferences` becomes worse (in terms of confidence intervals) than an active arm or an arm with a higher index in `Preferences`, then k is added to `Occupied`. This rule is described at Line 18 in Algorithm 4.7.

Following these rules, as soon as there is an arm in `Preferences`, player j tries to occupy the p-th arm in `Preferences` (starting with $p = 1$), yet she still continues to explore. As soon as she encounters a positive reward on it, she occupies it and starts the exploitation phase. If she does not end up occupying an optimal arm, this arm will be added to `Occupied` at some point. The player then increments p and tries to occupy the next available best arm. This point is described at lines 9-10 in Algorithm 4.7. Notice that `Preferences` can have more than p elements, but the player must not exploit the q-th element of `Preferences` with $q > p$ yet as it can lead the player in exploiting a sub-optimal arm.

Appendix

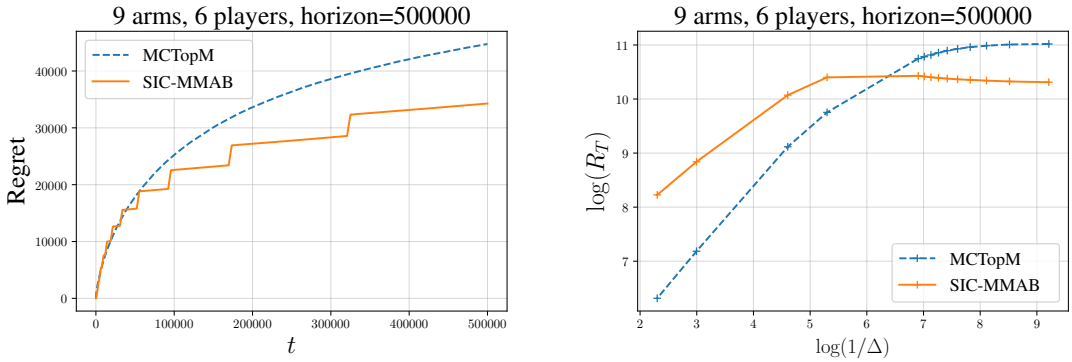
4.A Experiments

We compare in Figure 4.1 the empirical performances of SIC-MMAB with the MCTopM algorithm(Besson and Kaufmann, 2018a) on generated data[3]. We also compared with the MusicalChairs algorithm (Rosenski et al., 2016), but its performance was irrelevant and out of scale. This is mainly due to its scaling with $1/\Delta^2$, besides presenting large constant terms in its regret. Also, its main advantage comes from its scaling with M, which is here small for computational reasons. All the considered regret values are averaged over 200 runs. The experiments are run with Bernoulli distributions. Thus, there is no need to quantize the sent statistics and a tighter confidence bound $B_s = \sqrt{\frac{2\log(T)}{s}}$ is used.

Figure 4.1a represents the evolution of the regret for both algorithms with the following problem parameters: $K = 9$, $M = 6$, $T = 5 \times 10^5$. The means of the arms are linearly distributed between 0.9 and 0.89, so the gap between two consecutive arms is 1.25×10^{-3}. The switches between exploration and communication phases for SIC-MMAB are easily observable. A larger horizon (near 40 times larger) is required for SIC-MMAB to converge to a constant regret, but this alternation between the phases could not be visible for such a value of T.

Figure 4.1b represents the evolution of the final regret as a function of the gap Δ between two consecutive arms in a logarithmic scale. The problem parameters K, M and T are the same. Although MCTopM seems to provide better results with larger values of Δ, SIC-MMAB seems to have a smaller dependency in $1/\Delta$. This confirms the theoretical results claiming that MCTopM scales with Δ^{-2} while SIC-MMAB scales with Δ^{-1}. This can be observed on the left part of Figure 4.1b where the slope for MCTopM is approximately twice as large as for SIC-MMAB. Also, a different behavior of the regret appears for very low values of Δ which is certainly due to the fact that the regret only depends on T for extremely small values of Δ (minmax regret).

[3]The code is available at `https://github.com/eboursier/sic-mmab`.



(a) Evolution of regret over time.

(b) Final regret as a function of $\frac{1}{\Delta}$.

Figure 4.1: Performance comparison between SIC-MMAB and MCTopM algorithms.

4.B Omitted proofs

4.B.1 Regret analysis of SIC-MMAB

In this section, we prove the regret bound for SIC-MMAB algorithm given by Theorem 4.1. In what follows, the statement *"with probability* $1 - \mathcal{O}(\delta(T))$*, it holds that* $f(T) = \mathcal{O}(g(T))$*"* means that there is a universal constant $c \in \mathbb{R}_+$ such that $f(T) \leq cg(T)$ with probability at least $1 - c\delta(T)$. We also denote $\eta^m(t) = \eta_{\pi^m(t)}(t)$ in the following for conciseness.

We first decompose the regret as follows:

$$R(T) = \mathbb{E}[R^{\text{init}} + R^{\text{comm}} + R^{\text{explo}}], \tag{4.2}$$

where

$$R^{\text{init}} = T_{\text{init}} \sum_{k=1}^{M} \mu_{(k)} - \sum_{t=1}^{T_{\text{init}}} \sum_{m=1}^{M} \mu_{\pi^m(t)}(1 - \eta^m(t)) \text{ with } T_{\text{init}} = T_0 + 2K,$$

$$R^{\text{comm}} = \sum_{t \in \text{Comm}} \sum_{m=1}^{M} (\mu_{(m)} - \mu_{\pi^m(t)}(1 - \eta^m(t))) \text{ with Comm the set of communication steps,}$$

$$R^{\text{explo}} = \sum_{t \in \text{Explo}} \sum_{m=1}^{M} (\mu_{(m)} - \mu_{\pi^m(t)}(1 - \eta^m(t))) \text{ with Explo} = \{T_{\text{init}} + 1, \dots, T\} \setminus \text{Comm}.$$

A **communication step** is defined as a time step where a player is communicating statistics, i.e., using **Send Protocol**. These terms respectively correspond to the regret due to the initialization phase, the communication and the regret of both exploration and exploitation phases. Note that the terms R^{init}, R^{comm} and R^{explo} are here random variables.

Initialization analysis

The initialization regret is obviously bounded by $M(T_0 + 2K)$ as the initialization phase lasts $T_0 + 2K$ steps. Lemma 4.2 provides the probability to reach an orthogonal setting at time T_0. If this orthogonal setting is reached, the initialization phase is **successful**. In that case, the players then determine M and a unique internal rank using Algorithm 4.2. This is shown by observing that players with external ranks k and k' will exactly collide at round $T_0 + k + k'$.

Lemma 4.2. *After a time T_0, all players pull different arms with probability at least $1 - M\exp\left(-\frac{T_0}{K}\right)$.*

Proof. As there is at least one arm that is not played by all the other players at each time step, the probability of having no collision at time t for a single player j is lower bounded by $\frac{1}{K}$. It thus holds:

$$\mathbb{P}\left[\forall t \leq T_0, \eta^j(t) = 1\right] \leq \left(1 - \frac{1}{K}\right)^{T_0} \leq \exp\left(-\frac{T_0}{K}\right).$$

For a single player j, her probability to encounter only collisions until time T_0 is at most $\exp\left(-\frac{T_0}{K}\right)$. The union bound over the M players then yields the desired result. □

Exploration regret

This section aims at proving Lemma 4.3, which bounds the exploration regret.

Lemma 4.3. *With probability $1 - \mathcal{O}\left(\frac{K\log(T)}{T} + M\exp\left(-\frac{T_0}{K}\right)\right)$,*

$$R^{\text{explo}} = \mathcal{O}\left(\sum_{k>M} \min\left\{\frac{\log(T)}{\mu_{(M)} - \mu_{(k)}}, \sqrt{T\log(T)}\right\}\right).$$

The proof of Lemma 4.3 is divided in several auxiliary lemmas. It first relies on the correctness of the estimations before taking the decision to accept or reject an arm.

Lemma 4.4. *For each arm k and positive integer n, $\mathbb{P}[\exists p \leq n : |\widetilde{\mu}_k(p) - \mu_k| \geq B_{N_k(p)}] \leq \frac{4n}{T}$.*

Proof. For each arm k and positive integer n, Hoeffding inequality gives the following, classical inequality in MAB: $\mathbb{P}[\exists p \leq n : |\hat{\mu}_k(p) - \mu_k| \geq \sqrt{\frac{2\log(T)}{T_k(p)}}] \leq \frac{2n}{T}$. It remains to bound the estimation error due to quantization.

Notice that $\sum_{j=1}^{M}(\widetilde{S}_k^j - \lfloor S_k^j \rfloor)$ is the sum of M independent Bernoulli at each phase p. Hoeffding inequality thus also claims that $\mathbb{P}[|\sum_{j=1}^{M}(\widetilde{S}_k^j(p) - S_k^j(p))| \geq \sqrt{\frac{\log(T)M}{2}}] \leq \frac{2}{T}$. As $N_k(p) \geq M$, it then holds $\mathbb{P}[\exists p \leq n : |\widetilde{\mu}_k^j(p) - \hat{\mu}_k^j(p)| \geq \sqrt{\frac{\log(T)}{2N_k(p)}}] \leq \frac{2n}{T}$. Using the triangle inequality with this bound and the first Hoeffding inequality of the proof yields the final result. □

For both exploration and exploitation phases, we control the number of times an arm is pulled before being accepted or rejected.

Proposition 4.1. *With probability* $1 - \mathcal{O}\left(\frac{K\log(T)}{T} + M\exp\left(-\frac{T_0}{K}\right)\right)$*, every optimal arm* k *is accepted after at most* $\mathcal{O}\left(\frac{\log(T)}{(\mu_k - \mu_{(M+1)})^2}\right)$ *pulls during exploration phases, and every sub-optimal arm* k *is rejected after at most* $\mathcal{O}\left(\frac{\log(T)}{(\mu_{(M)} - \mu_k)^2}\right)$ *pulls during exploration phases.*

Proof. With probability at least $1 - M\exp\left(-\frac{T_0}{K}\right)$, the initialization is successful, i.e., all players have been assigned different ranks. The remaining of the proof is conditioned on that event.

As there are at most $\log_2(T)$ exploration-communication phases, $|\widetilde{\mu}_k(p) - \mu_k| \leq B_{N_k(p)}$ holds for all arms and phases with probability $1 - \mathcal{O}\left(\frac{K\log(T)}{T}\right)$ thanks to Lemma 4.4. The remaining of the proof is conditioned on that event.

We first consider an optimal arm k. Let $\Delta_k = \mu_k - \mu_{(M+1)}$ be the gap between the arm k and the first sub-optimal arm. We assume $\Delta_k > 0$ here, the case of equality holds considering $\frac{\log(T)}{0} = \infty$. Let s_k be the first integer such that $4B_{s_k} \leq \Delta_k$.

With $N_k(p) = \sum_{l=1}^{p} M_l 2^l$ the number of times an active arm has been pulled after the p-th exploration phase, it holds that

$$N(p+1) \leq 3N(p) \qquad \text{as } M_p \text{ is non-increasing.} \tag{4.3}$$

For some $p \in \mathbb{N}$, $T(p-1) < s_k \leq T(p)$ or the arm k is active at time T. In the second case, it is obvious that k is pulled less than $\mathcal{O}(s_k)$ times. Otherwise, the triangle inequality for such a p, for any active sub-optimal arm i, yields $\widetilde{\mu}_k(p) - B_{N_k(p)} \geq \widetilde{\mu}_i(p) + B_{N_i(p)}$.

So the arm k is accepted after at most p phases. Using the same argument as in (Perchet et al., 2015), it holds $s_k = \mathcal{O}\left(\frac{\log(T)}{(\mu_k - \mu_{(M+1)})^2}\right)$, and also for $N_k(p)$ thanks to Equation (4.3). Also, k can not be wrongly rejected conditionally on the same event, as it can not be dominated by any sub-optimal arm in term of confidence intervals.

The proof for the sub-optimal case is similar if we denote $\Delta_k = \mu_{(M)} - \mu_k$. □

In the following, we keep the notation $t_k = \min\left\{\frac{c\log(T)}{\left(\mu_k - \mu_{(M)}\right)^2},\ T\right\}$, where c is a universal constant such that with the probability considered in Proposition 4.1, the number of exploration pulls before accepting/rejecting k is at most t_k.

For both exploration and exploitation phases, the decomposition used in the centralized case (Anantharam et al., 1987a) holds because there is no collision during these two types of phases (conditionally on the success of the initialization phase):

$$R^{\text{explo}} = \sum_{k>M} (\mu_{(M)} - \mu_{(k)}) N_{(k)}^{\text{explo}} + \sum_{k \leq M} (\mu_{(k)} - \mu_{(M)})(T^{\text{explo}} - N_{(k)}^{\text{explo}}), \tag{4.4}$$

where $T^{\text{explo}} = \#\text{Explo}$ and $N_{(k)}^{\text{explo}}$ is the centralized number of time steps where the k-th best arm is pulled during exploration or exploitation phases.

Lemma 4.5. *With probability* $1 - \mathcal{O}\left(\frac{K\log(T)}{T} + M\exp\left(-\frac{T_0}{K}\right)\right)$*, the following hold simultaneously:*

i) for a sub-optimal arm k, $(\mu_{(M)} - \mu_k)N_k^{\text{explo}} = \mathcal{O}\left(\min\left\{\frac{\log(T)}{\mu_{(M)}-\mu_k}, \sqrt{T\log(T)}\right\}\right)$.

ii) $\sum_{k\leq M}(\mu_{(k)} - \mu_{(M)})(T^{\text{explo}} - N_{(k)}^{\text{explo}}) = \mathcal{O}\left(\sum_{k>M}\min\left\{\frac{\log(T)}{\mu_{(M)}-\mu_{(k)}}, \sqrt{T\log(T)}\right\}\right)$.

Proof. i) From Proposition 4.1, $N_k^{\text{explo}} \leq \mathcal{O}\left(\min\left\{\frac{\log(T)}{(\mu_{(M)}-\mu_k)^2}, T\right\}\right)$ with the considered probability, so $(\mu_{(M)} - \mu_k)N_k^{\text{explo}} = \mathcal{O}\left(\min\left\{\frac{\log(T)}{(\mu_{(M)}-\mu_k)}, (\mu_{(M)} - \mu_k)T\right\}\right)$. The function $\Delta \mapsto \min\left\{\frac{\log(T)}{\Delta}, \Delta T\right\}$ is maximized for $\Delta = \sqrt{\frac{\log(T)}{T}}$ and its maximum is $\sqrt{T\log(T)}$. Thus, the inequality $\min\left\{\frac{\log(T)}{\Delta}, \Delta T\right\} \leq \min\left\{\frac{\log(T)}{\Delta}, \sqrt{T\log(T)}\right\}$ always holds for $\Delta \geq 0$ and yields the first point.

ii) We (re)define the following: $\hat{t}_k$ the number of exploratory pulls before accepting/rejecting the arm k, M_l the number of active player during the l-th exploration phase, $N(p) = \sum_{l=1}^{p} 2^l M_l$ and $\hat{p}_T$ the total number of exploration phases.

$N(p)$ describes the total number of exploration pulls processed at the end of the p-th exploration phase on every active arm for $p < \hat{p}_T$. Since the $\hat{p}_T$-th phase may remain uncompleted, $N(\hat{p}_T)$ is then greater that the number of exploration pulls at the end of the $\hat{p}_T$-th phase.

With probability $1 - \mathcal{O}\left(\frac{K\log(T)}{T} + M\exp\left(-\frac{T_0}{K}\right)\right)$, the initialization is successful, every arm is correctly accepted or rejected and $\hat{t}_k \leq t_k$ for all k. The remaining of the proof is conditioned on that event. We now decompose the proof in two main parts given by Lemmas 4.6 and 4.7 proven below.

Lemma 4.6. *Conditionally on the success of the initialization phase and on correct estimations of all arms:*

$$\sum_{k\leq M}(\mu_{(k)} - \mu_{(M)})(T^{\textit{explo}} - N_{(k)}^{\textit{explo}}) \leq \sum_{j>M}\sum_{k\leq M}\sum_{p=1}^{\hat{p}_T} 2^p(\mu_{(k)} - \mu_{(M)})\mathbb{1}_{\min(\hat{t}_{(j)},\hat{t}_{(k)})>N(p-1)}.$$

Lemma 4.7. *Conditionally on the success of the initialization phase and on correct estimations*

of all arms:

$$\sum_{k\leq M}\sum_{p=1}^{\hat{p}_T} 2^p(\mu_{(k)}-\mu_{(M)})\mathbb{1}_{\min(\hat{t}_{(j)},\hat{t}_{(k)})>N(p-1)} \leq \mathcal{O}\left(\min\left\{\frac{\log(T)}{\mu_{(M)}-\mu_{(j)}}, \sqrt{T\log(T)}\right\}\right).$$

These two lemmas directly yield the second point in Lemma 4.5. □

Proof of Lemma 4.6. Let us consider an optimal arm k. During the p-th exploration phase, there are two possibilities:

- either k has already been accepted, i.e., $\hat{t}_k \leq N(p-1)$. Then the arm k is pulled the whole phase, i.e., $K_p 2^p$ times.
- Or k is still active. Then it is pulled 2^p times by each active player, i.e., it is pulled $M_p 2^p$ times in total. This means that it is not pulled $(K_p - M_p)2^p$ times.

From these two points, it holds that $N_k^{\text{explo}} \geq T^{\text{explo}} - \sum_{p=1}^{\hat{p}_T} 2^p(K_p - M_p)\mathbb{1}_{\hat{t}_k>N(p-1)}$.

Notice that $K_p - M_p$ is the number of active sub-optimal arms. By definition, $K_p - M_p = \sum_{j>M} \mathbb{1}_{\hat{t}_{(j)}>N(p-1)}$. We thus get that $N_k^{\text{explo}} \geq T^{\text{explo}} - \sum_{j>M}\sum_{p=1}^{\hat{p}_T} 2^p \mathbb{1}_{\min(\hat{t}_{(j)},\hat{t}_k)>N(p-1)}$.

The double sum actually is the number of times a sub-optimal arm is pulled instead of k. This yields the result when summing over all optimal arms k. □

Proof of Lemma 4.7. Let us define $A_j = \sum_{k\leq M}\sum_{p=1}^{\hat{p}_T} 2^p(\mu_{(k)}-\mu_{(M)})\mathbb{1}_{\min(\hat{t}_j,\hat{t}_{(k)})>N(p-1)}$ the cost associated to the sub-optimal arm j. Lemma 4.7 upper bounds A_j for any sub-optimal arm j.

Recall that $t_{(k)} = \min\left(\frac{c\log(T)}{(\mu_{(k)}-\mu_{(M)})^2}, T\right)$ for a universal constant c. The proof is conditioned on the event $\hat{t}_{(k)} \leq t_{(k)}$, so that if we define $\Delta(p) = \sqrt{\frac{c\log(T)}{N(p-1)}}$, the inequality $\hat{t}_{(k)} > N(p-1)$ implies $\mu_{(k)} - \mu_{(M)} < \Delta(p)$. We also write p^j the first integer such that $\hat{t}_j \leq N(p^j)$. It follows:

$$\begin{aligned}
A_j &\leq \sum_{k\leq M}\sum_{p=1}^{p^j} 2^p\Delta(p)\mathbb{1}_{\hat{t}(k)>N(p-1)} \\
&\leq \sum_{p=1}^{p^j} \Delta(p)\left(N(p)-N(p-1)\right) && \text{as } \sum_{k\leq M}\mathbb{1}_{\hat{t}(k)>N(p-1)} = M_p. \\
&= c\log(T)\sum_{p=1}^{p^j}\Delta(p)\left(\frac{1}{\Delta(p+1)}+\frac{1}{\Delta(p)}\right)\left(\frac{1}{\Delta(p+1)}-\frac{1}{\Delta(p)}\right)
\end{aligned}$$

$$\leq (1+\sqrt{3})c\log(T)\sum_{p=1}^{p^j}(\frac{1}{\Delta(p+1)}-\frac{1}{\Delta(p)}) \qquad \text{thanks to Equation (4.3).}$$

$$\leq (1+\sqrt{3})c\log(T)\frac{1}{\Delta(p^j+1)} \qquad \text{by convention, } \frac{1}{\Delta(1)}=0.$$

By definition of p^j, we have $t_j \geq N(p^j-1)$. Thus, $\Delta(p^j) \geq \sqrt{\frac{c\log(T)}{t_j}}$ and Equation (4.3) gives $\Delta(p^j+1) \geq \sqrt{\frac{c\log(T)}{3t_j}}$. It then holds $A_j \leq (3+\sqrt{3})\sqrt{c\,t_j\log(T)}$. The result follows since $t_j = \mathcal{O}\left(\min\left\{\frac{\log(T)}{(\mu_{(M)}-\mu_j)^2}, T\right\}\right)$. □

Using the two points of Lemma 4.5, along with Equation (4.4), yields Lemma 4.3.

Communication cost

We now focus on the R^{comm} term in Equation (4.2). Lemma 4.8 states it is negligible compared to $\log(T)$ and has a significant impact on the regret only for small values of T.

Lemma 4.8. *With probability* $1-\mathcal{O}\left(\frac{K\log(T)}{T}+M\exp\left(-\frac{T_0}{K}\right)\right)$*, the following holds:*

$$R^{\text{comm}} = \mathcal{O}\left(KM^3\log^2\left(\min\left\{\frac{\log(T)}{(\mu_{(M)}-\mu_{(M+1)})^2},\ T\right\}\right)\right).$$

Proof. As explained in Section 4.1.2, the length of the communication phase $p \in [P]$ is at most $KM^2(p+1)$, where P is the number of exploration phases. The cost of communication is then smaller than $KM^3\sum_{p=1}^{P}(p+1) = \mathcal{O}\left(KM^3P^2\right)$. Proposition 4.1 in Section 4.B.1, claims with the considered probability that P is at most $\mathcal{O}\Big(\log\Big(\min\Big\{\frac{\log(T)}{(\mu_{(M)}-\mu_{(M+1)})^2},\ T\Big\}\Big)\Big)$, which yields Lemma 4.8. □

Total regret

The choice $T_0 = \lceil K\log(T)\rceil$ along with Lemmas 4.2, 4.3 and 4.8 claim that a bad event occurs with probability at most $\mathcal{O}\left(\frac{K\log(T)}{T}+\frac{M}{T}\right)$. The average regret due to bad events is thus upper bounded by $\mathcal{O}(KM\log(T))$. Using these lemmas along with Equation (4.2) finally yields the bound in Theorem 4.1.

4.B.2 Regret analysis of DYN-MMAB

Auxiliary lemmas

This section is devoted to the proof of Theorem 4.2. It first proves the first point of Lemma 4.1.

Proof of Lemma 4.1.1. We first introduce $Z_t \coloneqq X_k(t+\tau_j)(1-\eta_k(t+\tau_j))\mathbb{1}_{\pi^j(t+\tau_j)=k}$ and $p_t \coloneqq \mathbb{E}[Z_t]$. Notice that $p_t \leq \frac{1}{K}$ because $\mathbb{1}_{\pi^j(t+\tau_j)=k}$ is a Bernoulli of parameter $\frac{1}{K}$ in the exploration phase. Chernoff bound states that:

$$\mathbb{P}\Big[\sum_{t'=1}^{t}(Z_{t'}-\mathbb{E}[Z_{t'}]) \geq t\delta\Big] \leq \min_{\lambda>0} e^{-\lambda t\delta}\,\mathbb{E}\big[\prod_{t'=1}^{t} e^{\lambda(Z_{t'}-\mathbb{E}[Z_{t'}])}\big].$$

By convexity, $e^{\lambda z} \leq 1+z(e^\lambda-1)$ for $z\in[0,1]$. It thus holds:

$$\begin{aligned}\mathbb{E}\Big[e^{\lambda(Z_t-\mathbb{E}[Z_t])}\Big] &\leq e^{-\lambda p_t}\Big(1+p_t(e^\lambda-1)\Big) \leq e^{-\lambda p_t}e^{p_t(e^\lambda-1)} \qquad \text{as } 1+x\leq e^x.\\ &\leq e^{p_t(e^\lambda-1-\lambda)} \leq e^{\frac{e^\lambda-1-\lambda}{K}} \qquad \text{as } p_t\leq\frac{1}{K} \text{ and } e^\lambda-1-\lambda\geq 0.\end{aligned}$$

It can then be deduced:

$$\begin{aligned}\mathbb{P}\Big[\sum_{t'=1}^{t}(Z_{t'}-\mathbb{E}[Z_{t'}]) \geq t\delta\Big] &\leq \min_{\lambda>0} e^{-\lambda t\delta}e^{t\frac{e^\lambda-1-\lambda}{K}}. \qquad \text{For } \lambda=\log(1+K\delta):\\ &\leq \exp\left(-\frac{t}{K}h(K\delta)\right) \qquad \text{with } h(u)=(1+u)\log(1+u)-u.\end{aligned}$$

Similarly, we show for the negative error: $\mathbb{P}\Big[\sum_{t'=1}^{t}(Z_{t'}-\mathbb{E}[Z_{t'}]) \leq -t\delta\Big] \leq \exp\left(-\frac{t}{K}h(-K\delta)\right)$.

Either $t \leq \frac{16}{3}K\log(T^j)$ and the desired inequality holds almost surely, or $K\delta<1$ with $\delta=\sqrt{\frac{16\log(T^j)}{3tK}}$. As $h(x)\geq\frac{3x^2}{8}$ for $|x|<1$, it then holds

$$\mathbb{P}\Big[\Big|\sum_{t'=1}^{t}(Z_{t'}-\mathbb{E}[Z_{t'}])\Big| \geq t\delta\Big] \leq 2e^{-\frac{3t(K\delta)^2}{8K}} \qquad \text{and after multiplication with } \frac{K}{t}:$$

$$\mathbb{P}\left[\Big|\frac{K}{t}\sum_{t'=1+\tau_j}^{t+\tau_j} X_k(t')(1-\eta_k(t'))\mathbb{1}_{\pi^j(t')=k}-\gamma_j(t)\mu_k\Big| \geq \sqrt{\frac{16K\log(T^j)}{3t}}\right] \leq \frac{2}{(T^j)^2}. \tag{4.5}$$

Chernoff bound also provides a confidence interval on the number of pulls on a single arm:

$$\mathbb{P}\left[\Big|N_k^j(t)-\frac{t}{K}\Big| \geq \sqrt{\frac{6t\log(T^j)}{K}}\right] \leq \frac{2}{(T^j)^2}. \tag{4.6}$$

From Equation (4.6), it can be directly deduced that $\mathbb{P}\Big[|\frac{KN_k^j(t)}{t}-1| \geq \sqrt{\frac{6K\log(T^j)}{t}}\Big] \leq \frac{2}{(T^j)^2}$. As $\hat{r}_k^j(t)\leq 1$,

$$\mathbb{P}\left[\Big|\frac{KN_k^j(t)}{t}\hat{r}_k^j(t)-\hat{r}_k^j(t)\Big| \geq \sqrt{\frac{6K\log(T^j)}{t}}\right] \leq \frac{2}{(T^j)^2}. \tag{4.7}$$

As $\frac{KN_k^j(t)}{t}\hat{r}_k^j(t) = \frac{K}{t}\sum_{t'=1+\tau_j}^{t+\tau_j} X_k(t')(1-\eta_k(t'))\mathbb{1}_{\pi^j(t')=k}$, using the triangle inequality with Equations (4.5) and (4.7) finally yields $\mathbb{P}\Big[|\hat{r}_k^j(t) - \gamma^j(t)\mu_k| \geq 2\sqrt{\frac{6\,K\log(T^j)}{t}}\Big] \leq \frac{4}{(T^j)^2}$. □

The second point of Lemma 4.1 is proved below.

Proof of Lemma 4.1.2. The previous point gives that with probability $1 - \mathcal{O}\left(\frac{K}{T^j}\right)$, player j correctly estimated all the free arms until stage T. The remaining of the proof is conditioned on this event. We also assume that t^0 is the first stage where k is occupied for the proof. The general result claimed in Lemma 4.1 directly follows.

When t^0 is small, the second case will happen, i.e., the number of pulls on the arm k is small and its average reward can quickly drop to 0. When t^0 is large, $\gamma_j(t)\mu_k$ is tightly estimated so that L_k^j is small. Then, the first case will happen, i.e., the arm k will be quickly detected as occupied.

a) We first assume $t^0 \leq 12K\log(T^j)$. The empirical reward after $N_k^j(t) \geq N_k^j(t^0)$ pulls is $\hat{r}_k^j(t) = \frac{\hat{r}_k^j(t^0)N_k^j(t^0)}{N_k^j(t)}$, because all pulls after the stage $t^0 + \tau_j$ will return 0 rewards. However, using Chernoff bound as in Equation (4.6), it appears that if $t^0 \leq 12K\log(T^j)$ then $N_k^j(t^0) \leq 18\log(T^j)$ with probability $1 - \mathcal{O}\left(\frac{1}{T^j}\right)$, so $\hat{r}_k^j(t) \leq \frac{18\log(T^j)}{N_k^j(t)}$.

Conditionally on the correct estimations of the arms, there is at least an unoccupied arm i with $\mu_i \leq \mu_k$. Therefore with $t_i = \frac{72Ke\log(T^j)}{\mu_i^2}$, as $t_i \geq 12K\log(T^j)$, Chernoff bound guarantees that the following holds, with probability at least $1 - \frac{2}{T^j}$,

$$\frac{3t_i}{2K} \geq N_k^j(t_i) \geq \frac{t_i}{2K} = \frac{36e\log(T^j)}{\mu_i^2}. \tag{4.8}$$

This gives that $\hat{r}_k^j(t_i) \leq \frac{\mu_i}{2e}$. After stage $\tau_j + \frac{d'K\log(T^j)}{\mu_i^2}$, where d' is some universal constant, the error bounds of both arms are upper bounded by $\frac{\mu_i}{8e}$. The confidence intervals would then be disjoint for the arms k and i. So k will be detected as worse than i after a time at most $\mathcal{O}\left(\frac{K\log(T)}{\mu_i^2}\right)$ as $T^j \leq T$.

b) We now assume that $12K\log(T^j) \leq t^0 \leq \frac{24\lambda K\log(T^j)}{\mu_k^2}$ with $\lambda = 16e^2$. It still holds $\hat{r}_k^j(t) = \frac{\hat{r}_k^j(t^0)N_k^j(t^0)}{N_k^j(t)}$. Correct estimations of the free arms are assumed in this proof, so in particular

$$\hat{r}_k^j(t) \leq \frac{(\mu_k + B^j(t^0))T_k^j(t^0)}{T_k^j(t)}. \tag{4.9}$$

As in Equation (4.8), it holds that $N_k^j(t^0) \leq \frac{3t^0}{2K}$ with probability $1 - \mathcal{O}\left(\frac{1}{T^j}\right)$ and thus

$B^j(t^0) \leq 6\sqrt{\frac{\log(T^j)}{N_k^j(t^0)}}$. Also, $N_k^j(t) \geq \frac{d\log(T^j)}{2\mu_i\mu_k}$ for $t = d\frac{K\log(T^j)}{\mu_i^2}$. Equation (4.9) then becomes

$$\hat{r}_k^j(t) \leq \frac{\mu_k N_k^j(t^0)}{N_k^j(t)} + \frac{B^j(t^0)N_k^j(t^0)}{N_k^j(t)} \leq \frac{36\lambda}{d}\mu_i + \frac{6\sqrt{N_k^j(t^0)\log(T^j)}}{N_k^j(t)} \leq \left(\frac{36\lambda}{d} + \frac{72\sqrt{\lambda}}{d}\right)\mu_i.$$

Thus, for a well chosen d, the empirical reward verifies $\hat{r}_k^j(t) \leq \frac{\mu_i}{2e}$. We then conclude as for the first case that the arm k would be detected as worse than the free arm i after a time $\mathcal{O}\left(\frac{K\log(T)}{\mu_i^2}\right)$.

c) The last case corresponds to $t^0 > \frac{24\lambda K\log(T^j)}{\mu_k^2}$. It then holds $B^j(t^0) \leq \frac{\mu_k}{\sqrt{\lambda}} = \frac{\mu_k}{4e}$.

By definition, $L_k^j \leq \frac{2e\log(T^j)}{\hat{r}_k^j - B^j(t)}$. Conditionally on the correct estimation of the free arms, it holds that $\gamma_j(t)\mu_k - 2B^j(t) \leq \hat{r}_k^j - B^j(t) \leq \mu_k$. So with the choice of L_k^j described by Equation (4.1), as long as k is free,

$$\begin{aligned}\frac{2e\log(T^j)}{\mu_k} \leq L_k^j \quad &\leq \frac{2e\log(T^j)}{\gamma_j(t)\mu_k - 2B^j(t)} \\ &\leq \frac{2e^2\log(T^j)}{\mu_k - 2eB^j(t)}.\end{aligned} \tag{4.10}$$

As $B^j(t^0) \leq \frac{\mu_k}{4e}$, it holds that $L_k^j(t^0) \leq \frac{4e^2\log(T^j)}{\mu_k}$. Since L_k^j is non-increasing by definition, this actually holds for all t larger than t^0.

From that point, Equation (4.8) gives that with probability $1 - \mathcal{O}\left(\frac{1}{T^j}\right)$, the arm k will be pulled at least $2L_k^j$ times between stage t^0+1 and $t^0 + 24KL_k^j$ with probability $1 - \mathcal{O}\left(\frac{1}{T^j}\right)$. Thus, a whole block of L_k^j pulls receiving only 0 rewards on k happens before stage $t^0 + 24KL_k^j$.

The arm k is then detected as occupied after a time $\mathcal{O}\left(\frac{K\log(T^j)}{\mu_k}\right)$ from t^0, leading to the result. □

Lemma 4.9. *At every stage, no free arm k is falsely detected as occupied by player j with probability* $1 - \mathcal{O}\left(\frac{K}{T^j}\right)$.

Proof. As shown above, with probability $1 - \mathcal{O}\left(\frac{K}{T^j}\right)$, player j correctly estimated the average rewards of all the free arms until stage T. The remaining of the proof is conditioned on that event. As long as k is free, it can not become dominated by some arm that was not added to `Preferences` before k, so it can not be added to `Occupied` from the rule given at lines 17-18 in Algorithm 4.7.

For the rule of Line 12, Equation (4.10) gives that

$$L_k^j(t') \geq \frac{2e\log(T^j)}{\mu_k} \qquad \text{at each stage } t' \leq t. \tag{4.11}$$

As in Section 4.B.1, the probability of detecting L successive 0 rewards on a free arm k is then smaller than $(1-\frac{\mu_k}{e})^L \leq \exp\left(-\frac{L\mu_k}{e}\right)$.

Using this along with Equation (4.11) yields that with probability $1-\mathcal{O}\left(\frac{1}{(T^j)^2}\right)$, at least one positive reward will be observed on arm k in a single block. The union bound over all blocks yields the result. □

Finally, Lemma 4.10 yields that, after some time, each player starts exploiting an arm while all the better arms are already occupied by other players.

Lemma 4.10. *We denote* $\overline{\Delta}_{(k)} = \min_{i=1,\dots,k}(\mu_{(i)} - \mu_{(i+1)})$. *With probability* $1-\mathcal{O}\left(\frac{K}{T^j}\right)$, *it holds that for a single player* j, *there exists* k_j *such that after a stage at most* $\bar{t}_{k_j}+\tau_j$, *she is exploiting the* k_j*-th best arm and all the better arms are also exploited by other players, where* $\bar{t}_{k_j} = \mathcal{O}\left(\frac{K\log(T)}{\overline{\Delta}^2_{(k_j)}} + k_j\frac{K\log(T)}{\mu_{(k_j)}}\right)$.

Proof. Player j correctly estimates all the arms until stage T, with probability $1-\mathcal{O}\left(\frac{K}{T^j}\right)$. The remaining of the proof is conditioned on that event. We define $\bar{t}_i = \frac{cK\log(T^j)}{\overline{\Delta}^2_{(i)}} + i\frac{cK\log(T^j)}{\mu_{(i)}}$ for some universal constant c and k_j (random variable) defined as

$$k_j = \min\left\{i \in [K] \mid i\text{-th best arm not exploited by another player at stage } \bar{t}_i+\tau_j\right\}. \tag{4.12}$$

k_j^* (k_j-th best arm) is the best arm not exploited by another player (than player j) after the stage $\bar{t}_{k_j}+\tau_j$. The considered set is not empty as $M \leq K$.

Lemma 4.9 gives that with probability $1-\mathcal{O}\left(\frac{K}{T^j}\right)$, k_j^* is not falsely detected as occupied until stage T. All arms below k_j^* will be detected as worse than k_j^* after a time $\frac{dK\log(T^j)}{\overline{\Delta}^2_{(k_j)}}$ for some universal constant d.

By definition of k_j, any arm i^* better than k_j^* is already occupied at stage $\bar{t}_i+\tau_j$. Lemma 4.1, gives that with probability $1-\mathcal{O}\left(\frac{1}{T^j}\right)$, either i^* is detected as occupied after stage $\bar{t}_i+\tau_j+\frac{d'K\log(T^j)}{\mu_{(i)}}$ or dominated by k_j^* after stage $\frac{d_2K\log(T^j)}{\overline{\Delta}^2_{(k_j)}}+\tau_j$ for some universal constants d' and d_2.

Thus the player detects the arm k_j^* as optimal and starts trying to occupy k_j^* at a stage at most $\widetilde{t} = \max\left(\bar{t}_{k_j-1} + \frac{d'K\log(T^j)}{\mu_{(k_j)}}, \max(d,d_2)\frac{K\log(T^j)}{\overline{\Delta}^2_{(k_j)}}\right)+\tau_j$ with probability $1-\mathcal{O}\left(\frac{K}{T^j}\right)$ (where $\bar{t}_0 = 0$).

Using similar arguments as for Lemma 4.9, player j will observe a positive reward on k_j^* with probability $1 - \mathcal{O}\left(\frac{1}{T^j}\right)$ after a stage at most $\widetilde{t} + \frac{d_2' K \log(T^j)}{\mu_{(k_j)}}$ for some constant d_2', if k_j is still free at this stage. With the choice $c = \max(d, d_2, d' + d_2')$, this stage is smaller than $\bar{t}_{k_j}$ and k_j^* is then still free. Thus, player j will start exploiting k_j^* after stage at most $\bar{t}_{k_j}$ with the considered probability. □

Regret in dynamic setting

Proof of Theorem 4.2. Lemma 4.10 states that a player only needs an exploration time bounded as $\mathcal{O}\Big(\frac{K\log(T)}{\overline{\Delta}_{(k)}^2} + k\frac{K\log(T)}{\mu_{(k)}}\Big)$ before starting exploiting, with high probability. Furthermore, the better arms are already exploited when she does so. Thus, the exploited arms are the top-M arms. The regret is then upper bounded by twice the sum of exploration times (and the low probability events of wrong estimations), as a collision between players can only happen with at most one player in her exploitation phase.

The regret incurred by low probability events mentioned in Lemma 4.10 is in $\mathcal{O}(KM^2)$ and is thus dominated by the exploration regret. □

4.C On the inefficiency of SELFISH algorithm

A linear regret for the SELFISH algorithm in the No Sensing model has been recently conjectured (Besson and Kaufmann, 2018a). This algorithm seems to have good results in practice, although rare runs with linear regret appear. This is due to the fact that with probability $p > 0$ at some point t, both independent from T, some players might have the same number of pulls and the same observed average rewards for each arm. In that case, the players would pull the exact same arms and thus collide until they reach a tie breaking point where they could choose different arms thanks to a random tie breaking rule. However, it was observed that such tie breaking points would not appear in the experiments, explaining the linear regret for some runs. Here we claim that such tie breaking points might never happen in theory for the SELFISH algorithm when the rewards follow Bernoulli distributions, if we add the constraint that the numbers of positive rewards observed for the arms are all different at some stage. This event remains possible with a probability independent from T.

Proposition 4.2. *For $s, s' \in \mathbb{N}$ with $s \neq s'$:*

$$\forall n \geq 2, t, t' \in \mathbb{N}, \qquad \frac{s}{t} + \sqrt{\frac{2\log(n)}{t}} \neq \frac{s'}{t'} + \sqrt{\frac{2\log(n)}{t'}}.$$

Proof. First, if $t = t'$, these two quantities are obviously different as $s \neq s'$.

We now assume $\frac{s}{t} + \sqrt{\frac{2\log(n)}{t}} = \frac{s'}{t'} + \sqrt{\frac{2\log(n)}{t'}}$ with $t \neq t'$.
This means that $\sqrt{\frac{2\log(n)}{t}} - \sqrt{\frac{2\log(n)}{t'}}$ is a rational, i.e., for some rational p, $\log(n)(t + t' - 2\sqrt{tt'}) = 2p$.

It then holds

$$\log(n)\sqrt{tt'} = \log(n)\frac{t+t'}{2} - p,$$

$$tt'\log^2(n) = \log^2(n)(\frac{t+t'}{2})^2 - p(t+t')\log(n) + p^2,$$

$$\log^2(n)(\frac{t-t'}{2})^2 - p(t+t')\log(n) + p^2 = 0.$$

Since $(\frac{t-t'}{2})^2 \neq 0$ and all the coefficients are in $\mathbb{Q}$ here, this would mean that $\log(n)$ is an algebraic number. However, Lindemann–Weierstrass theorem implies that $\log(n)$ is transcendental for any integer $n \geq 2$. We thus have a contradiction. □

The proof is only theoretical as computer are not precise enough to distinguish rationals from irrationals. The advanced arguments are not applicable in practice. Still, this seems to confirm the conjecture proposed by Besson and Kaufmann, 2018a: a tie breaking point is never reached, or at least not before a very long period of time.

However, if the players are not synchronised (dynamic setting or asynchronous setting) or if they are using confidence bounds of the form $\sqrt{\frac{\eta^m \log(n)}{t}}$ where η^m is some variable proper to player m, this proof does not hold anymore. It thus remains unknown, whether slightly modifying the SELFISH algorithm could lead to interesting regret guarantees.

Chapter 5

A Practical Algorithm for Multiplayer Bandits when Arm Means Vary Among Players

For the more challenging *heterogeneous* setting, arms may have different means for different players. This chapter proposes a new and efficient algorithm that combines the idea of leveraging forced collisions for implicit communication and that of performing matching eliminations. We present a finite-time analysis of our algorithm, giving the first sublinear minimax regret bound for this problem, and prove that if the optimal assignment of players to arms is unique, our algorithm attains the optimal $\mathcal{O}(\log(T))$ regret, solving an open question raised at NeurIPS 2018 by Bistritz and Leshem (2018).

This chapter studies the heterogeneous collision sensing model described in Section 3.3.1, for which each arm has a possibly different mean for each player.

Bistritz and Leshem (2018) proposed an algorithm with regret bounded by $\mathcal{O}\left(\log^{2+\kappa}(T)\right)$ (for any constant κ), proved a lower bound of $\Omega(\log T)$ *for any algorithm*, and asked if there is an algorithm matching this lower bound. We propose a new algorithm for this model, M-ETC-Elim, which depends on a hyperparameter c, and we upper bound its regret by $\mathcal{O}\left(\log^{1+1/c}(T)\right)$ for any $c > 1$. We also bound its worst-case regret by $\mathcal{O}\left(\sqrt{T \log T}\right)$, which is the first sublinear minimax bound for this problem. Moreover, if the optimal assignment of the players to the arms is unique, we prove that instantiating M-ETC-Elim with $c = 1$ yields regret at most $\mathcal{O}\left(\log(T)\right)$, which is optimal and answers affirmatively the open question mentioned above in this particular case. We present a non-asymptotic regret analysis of M-ETC-Elim leading to nearly optimal regret upper bounds, and also demonstrate the empirical efficiency of this new algorithm via simulations.

This chapter is structured as follows. In Section 5.1, we present our contributions and put them in perspective by comparison with the literature. We describe the M-ETC-Elim algorithm in Section 5.2 and upper bound its regret in Section 5.3. Finally, we report in Section 5.4 results from an experimental study demonstrating the competitive practical performance of M-ETC-Elim.

5.1 Contributions

We propose an efficient algorithm for the heterogeneous multiplayer bandit problem achieving (quasi) logarithmic regret. The algorithm, called Multiplayer Explore-Then-Commit with matching Elimination (M-ETC-Elim), is described in detail in Section 5.2. It combines the idea of exploiting collisions for implicit communication, initially proposed in Chapter 4 for the homogeneous setting (which we have improved and adapted to our setting), with an efficient way to perform "matching eliminations."

M-ETC-Elim consists of several epochs combining exploration and communication, and may end with an exploitation phase if a unique optimal matching has been found. The algorithm depends on a parameter c controlling the epoch sizes and enjoys the following regret guarantees.

Theorem 5.1. *(a) The M-ETC-Elim algorithm with parameter $c \in \{1, 2, \dots\}$ satisfies*

$$R(T) = \mathcal{O}\left(MK\left(\frac{M^2 \log(T)}{\Delta}\right)^{1+1/c}\right).$$

(b) If the maximum matching is unique, M-ETC-Elim with $c = 1$ satisfies

$$R(T) = \mathcal{O}\left(\frac{M^3 K \log(T)}{\Delta}\right).$$

(c) Regardless of whether the optimal matching is unique or not, M-ETC-Elim with $c = 1$ satisfies the minimax regret bound

$$R(T) = \mathcal{O}\left(M^{\frac{3}{2}}\sqrt{KT\log(T)}\right).$$

We emphasize that we carry out a non-asymptotic analysis of M-ETC-Elim. The regret bounds of Theorem 5.1 are stated with the $\mathcal{O}(\cdot)$ notation for the ease of presentation and the hidden constants depend on the chosen parameter c only. In Theorems 5.2, 5.3 and 5.4 we provide the counterparts of these results with explicit constants.

A consequence of part (a) is that for a fixed problem instance, for any (arbitrarily small) κ, there exists an algorithm (M-ETC-Elim with parameter $c = \lceil 1/\kappa \rceil$) with regret $R(T) = \mathcal{O}\left((\log(T))^{1+\kappa}\right)$. This quasi-logarithmic regret rate improves upon the $\mathcal{O}\left(\log^2(T)\right)$ regret rate of (Bistritz and Leshem, 2018). Moreover, we provide additional theoretical guarantees for M-ETC-Elim using the parameter $c = 1$: an improved analysis in the presence of a unique optimal matching, which yields logarithmic regret (part (b)); and a problem-independent $\mathcal{O}\left(\sqrt{T \log T}\right)$ regret bound (part (c)), which supports the use of this particular parameter tuning regardless of whether the optimal matching is unique. This is the first sublinear minimax regret bound for this problem.

To summarize, we present a unified algorithm that can be used in the presence of either a unique or multiple optimal matchings and get a nearly logarithmic regret in both cases, almost matching the known logarithmic lower bound. Moreover, our algorithm is easy to implement, performs well in practice and does not need problem-dependent hyperparameter tuning.

5.1.1 Context and related work

Our algorithm also leverages the ideas of arm elimination and communication through collisions developed in Chapter 4, with the following enhancements. In our new communication protocol, the followers only send each piece of information once, to the leader, instead of sending it to the $M - 1$ other players. Then, while we used *arm eliminations* (coordinated between players) to

reduce the regret in Chapter 4, we cannot employ the same idea for our heterogeneous problem, as an arm that is bad for one player might be good for another player, and therefore cannot be eliminated. M-ETC-Elim instead relies on *matching eliminations*.

As mentioned in Chapter 3, the fully distributed heterogeneous setting was first studied by Bistritz and Leshem (2018), who proposed the Game-of-Thrones (GoT) algorithm and proved its regret is bounded by $\mathcal{O}\left(\log^{2+\kappa}(T)\right)$ for any given constant $\kappa > 0$, if its parameters are "appropriately tuned'.' In a more recent work (Bistritz and Leshem, 2020), the same authors provide an improved analysis, showing the same algorithm (with slightly modified phase lengths) enjoys quasi-logarithmic regret $\mathcal{O}\left(\log^{1+\kappa}(T)\right)$. GoT is quite different from M-ETC-Elim: it proceeds in epochs, each consisting of an exploration phase, a so-called GoT phase and an exploitation phase. During the GoT phase, the players jointly run a Markov chain whose unique stochastically stable state corresponds to a maximum matching of the estimated means. A parameter $\varepsilon \in (0, 1)$ controls the accuracy of the estimated maximum matching obtained after a GoT phase. Letting c_1, c_2, c_3 be the constants parameterizing the lengths of the phases, the improved analysis of GoT (Bistritz and Leshem, 2020) upper bounds its regret by $Mc_3 2^{k_0+1} + 2(c_1 + c_2)M \log_2^{1+\kappa}(T/c_3 + 2)$. This upper bound is asymptotic as it holds for T *large enough*, where "how large" is not explicitly specified and *depends on* Δ.[1] Moreover, the upper bound is valid only when the parameter ε is chosen *small enough*: ε should satisfy some constraints (Equations (66)-(67)) also featuring Δ. Hence, a valid tuning of the parameter ε would require prior knowledge of arm utilities. In contrast, we provide in Theorem 5.2 a non-asymptotic regret upper bound for M-ETC-Elim, which holds for any choice of the parameter c controlling the epoch lengths. Also, we show that if the optimal assignment is unique, M-ETC-Elim has logarithmic regret. Besides, we also illustrate in Section 5.4 that M-ETC-Elim outperforms GoT in practice. Finally, GoT has several parameters to set $(\delta, \varepsilon, c_1, c_2, c_3)$, while M-ETC-Elim has only one integral parameter c, and setting $c = 1$ works very well in all our experiments.

If Δ is known, an algorithm with similar ideas to M-ETC-Elim with $\mathcal{O}(\log T)$ regret was presented independently in the work of Magesh and Veeravalli (2019b).

Finally, the independent work of Tibrewal et al. (2019) studies a slightly stronger feedback model than ours: they assume each player in each round has the option of "observing whether a given arm has been pulled by someone," without actually pulling that arm (thus avoiding collision due to this "observation"), an operation that is called "sensing." Due to the stronger feedback, communications do not need to be implicitly done through collisions and bits can be broadcast to other players via sensing. Note that it is actually possible to send a single bit of information from one player to all other players in a single round in their model, an action that

[1] (Bistritz and Leshem, 2020, Theorem 4) requires T to be larger than $c_3(2^{k_0} - 2)$, where k_0 satisfies Equation (16), which features κ and Δ.

requires $M-1$ rounds in our model. Still, the algorithms proposed by Tibrewal et al. (2019) can be modified to obtain algorithms for our setting, and M-ETC-Elim can also be adapted to their setting. The two algorithms proposed by Tibrewal et al. (2019) share similarities with M-ETC-Elim: they also have exploration, communication and exploitation phases, but they do not use eliminations. Regarding their theoretical guarantees, a first remark is that those proved in Tibrewal et al. (2019) only hold in the presence of a unique optimal matching, whereas our analysis of M-ETC-Elim applies in the general case. The second remark is that their regret bounds for the case in which Δ is unknown (Theorems 3(ii) and 4) feature exponential dependence on the gap $1/\Delta$, whereas our regret bounds have polynomial dependence. Finally, the first-order term of their Theorem 4 has a quadratic dependence in $1/\Delta$, whereas our Theorem 5.1(b) scales linearly, which is optimal and allows us to get the $\mathcal{O}\left(\sqrt{\log(T)T}\right)$ minimax regret bound for M-ETC-Elim.

The best known lower bound in the centralized heterogeneous setting is $\Omega\left(\frac{KM}{\Delta}\log(T)\right)$ as explained in Section 3.3.3 (Combes et al., 2015). Moreover, a minimax lower bound of $\Omega(M\sqrt{KT})$ was given by Audibert et al. (2014) in the same setting. These lower bounds show that the dependency in T, Δ and K obtained in Theorem 5.1(b),(c) are essentially not improvable, but that the dependency in M might be. However, finding an algorithm whose regrets attain the available lower bounds for combinatorial semi-bandits is already hard even without the extra challenge of decentralization.

5.2 The M-ETC-Elim Algorithm

Our algorithm relies on an initialization phase in which the players elect a leader in a distributed manner. Then a communication protocol is set up, in which the leader and the followers have different roles: followers explore some arms and communicate to the leader estimates of the arm means, while the leader maintains a list of "candidate optimal matchings" and communicates to the followers the list of arms that need exploration in order to refine the list, i.e. to eliminate some candidate matchings. The algorithm is called *Multiplayer Explore-Then-Commit with matching Eliminations* (M-ETC-Elim for short). Formally, each player executes Algorithm 5.1 below.

Algorithm 5.1: M-ETC-Elim with parameter c

Input: Time horizon T, number of arms K

$R, M \longleftarrow$ INIT$(K, 1/KT)$

if $R = 1$ **then** LEADERALGORITHM(M) **else** FOLLOWERALGORITHM(R,M)

M-ETC-Elim requires as input the number of arms K (as well as a shared numbering of the arms across the players) and the time horizon T (the total number of arm selections). However,

if the players know only an upper bound on T, our results hold with T replaced by that upper bound as well. If no upper bound on T is known, the players can employ a simple doubling trick (Besson and Kaufmann, 2018b): we execute the algorithm assuming $T = 1$, then we execute it assuming $T = 2 \times 1$, and so on, until the actual time horizon is reached. If the expected regret of the algorithm for a known time horizon T is $R(T)$, then the expected regret of the modified algorithm for unknown time horizon T would be $R'(T) \leq \sum_{i=0}^{\log_2(T)} R_{2^i} \leq \log_2(T) \times R(T)$.

Initialization. The initialization procedure, similar to the initialization of SIC-MMAB described in Section 4.1.2. It first outputs for each player a rank $R \in [M]$ as well as the value of M, which is initially unknown to the players. This initialization phase relies on a "musical chairs" phase after which the players end up on distinct arms, followed by a "sequential hopping" protocol that permits them to know their ordering. For the sake of completeness, the initialization procedure is described in detail in Section 5.A. It corresponds to the same initialization as SIC-MMAB and the following lemma has thus already been proven in Section 4.B.1.

Lemma 5.1. *Fix $\delta_0 > 0$. With probability at least $1 - \delta_0$, if the M players run the* $\textsc{Init}(K, \delta_0)$ *procedure, which takes $K \log(K/\delta_0) + 2K - 2 < K \log(e^2 K/\delta_0)$ many rounds, all players learn M and obtain a distinct ranking from 1 to M.*

Communication Phases. Once all players have learned their ranks, player 1 becomes the *leader* and other players become the *followers*. The leader executes additional computations, and communicates with the followers individually, while each follower communicates only with the leader.

The leader and follower algorithms, described below, rely on several *communication phases*, which start at the same time for every player. During communication phases, the default behavior of each player is to pull her *communication arm*. It is crucial that these communication arms are distinct: an optimal way to do so is for each player to use her arm in the best matching found so far. In the first communication phase, such an assignment is unknown and players simply use their ranking as communication arm. Suppose at a certain time the leader wants to send a sequence of b bits $t_1, \ldots, t_b$ to the player with ranking i and communication arm k_i. During the next b rounds, for each $j = 1, 2, \ldots, b$, if $t_j = 1$, the leader pulls arm k_i; otherwise, she pulls her own communication arm k_1, while all followers stick to their communication arms. Player i can thus reconstruct these b bits after these b rounds, by observing the collisions on arm k_i. The converse communication between follower i and the leader is similar. The rankings are also useful to know *in which order communications should be performed*, as the leader successively communicates messages to the $M - 1$ followers, and then the $M - 1$ followers successively communicate messages to the leader.

Note that in case of unreliable channels where some of the communicated bits may be lost, there are several options to make this communication protocol more robust, such as sending each bit multiple times or using the Bernoulli signaling protocol of Tibrewal et al. (2019). Robustness has not been the focus of our work.

Leader and Follower Algorithms. The leader and the followers perform distinct algorithms, explained next. Consider a bipartite graph with parts of size M and K, where the edge (m, k) has weight μ_k^m and associates player m to arm k. The weights μ_k^m are unknown to the players, but the leader maintains a set of *estimated* weights that are sent to her by the followers, and approximate the real weights. The goal of these algorithms is for the players to jointly explore the matchings in this graph, while gradually focusing on better and better matchings. For this purpose, the leader maintains a set of *candidate edges* $\mathcal{E}$, which is initially $[M] \times [K]$, that can be seen as edges that are potentially contained in optimal matchings, and gradually refines this set by performing eliminations, based on the information obtained from the exploration phases and shared during communication phases.

M-ETC-Elim proceeds in epochs whose length is parameterized by c. In epoch $p = 1, 2, \ldots$, the leader weights the edges using the estimated weights. Then for every edge $(m, k) \in \mathcal{E}$, the leader computes the associated matching $\widetilde{\pi}_p^{m,k}$ defined as the estimated maximum matching containing the edge (m, k). This computation can be done in polynomial time using, e.g., the Hungarian algorithm (Munkres, 1957). The leader then computes the utility of the maximum matching and eliminates from $\mathcal{E}$ every edge for which the weight of its associated matching is smaller by at least $4M\varepsilon_p$, where

$$\varepsilon_p \coloneqq \sqrt{\frac{\log(2/\delta)}{2^{1+p^c}}}, \text{ with } \delta \coloneqq \frac{1}{M^2KT^2}. \tag{5.1}$$

The leader then forms the set of associated candidate matchings $\mathcal{C} \coloneqq \{\widetilde{\pi}_p^{m,k}, (m, k) \in \mathcal{E}\}$ and communicates to each follower the list of arms to explore in these matchings. Then exploration begins, in which for each candidate matching every player pulls its assigned arm 2^{p^c} times and records the received reward. Then another communication phase begins, during which each follower sends her observed estimated mean for the arms to the leader. More precisely, for each explored arm, the follower truncates the estimated mean (a number in $[0, 1]$) and sends only the $\frac{p^c+1}{2}$ most significant bits of this number to the leader. The leader updates the estimated weights and everyone proceeds to the next epoch. If at some point the list of candidate matchings $\mathcal{C}$ becomes a singleton, it means that (with high probability) the actual maximum matching is unique and has been found; so all players jointly pull that matching for the rest of the game (the exploitation phase).

Possible Exploitation Phase. Note that in the presence of several optimal matchings, the players will not enter the exploitation phase but will keep exploring several optimal matchings, which still ensures small regret. On the contrary, in the presence of a unique optimal matching, they are guaranteed to eventually enter the exploitation phase.[2] Also, observe that the set $\mathcal{C}$ of candidate optimal matchings does not necessarily contain *all* potentially optimal matchings, but all the edges in those matchings remain in $\mathcal{E}$ and are guaranteed to be explored.

The pseudocode for the leader's algorithm is given below, while the corresponding follower algorithm appears in Section 5.A. In the pseudocodes, `(comm.)` refers to a call to the communication protocol.

Procedure LeaderAlgorithm(M) for the M-ETC-Elim algorithm with parameter c

Input: Number of players M
$\mathcal{E} \longleftarrow [M] \times [K]$ `// list of candidate edges`
$\widetilde{\mu}_k^m \longleftarrow 0$ for all $(m,k) \in [M] \times [K]$ `// empirical estimates for utilities`
for $p = 1, 2, \ldots$ **do**
  $\mathcal{C} \longleftarrow \emptyset$ `// list of associated matchings`
  $\pi_p^* \longleftarrow \arg\max \left\{ \sum_{n=1}^M \widetilde{\mu}_{\pi(n)}^n : \pi \in \mathcal{M} \right\}$ `// using Hungarian algorithm`
  for $(m,k) \in \mathcal{E}$ **do**
    $\widetilde{\pi}_p^{m,k} \longleftarrow \arg\max \left\{ \sum_{n=1}^M \widetilde{\mu}_{\pi(n)}^n : \pi(m) = k \right\}$ `// using Hungarian algorithm`
    if $\sum_{n=1}^M \left\{ \widetilde{\mu}_{\pi_p^*(n)}^n - \widetilde{\mu}_{\widetilde{\pi}_p^{m,k}(n)}^n \right\} \leq 4M\epsilon_p$ **then** add $\widetilde{\pi}_p^{m,k}$ to $\mathcal{C}$
    else remove (m,k) from $\mathcal{E}$
  end
  for *each player* $m = 2, \ldots, M$ **do**
    Send to player m the value of $\mathrm{size}(\mathcal{C})$ `// (comm.)`
    for $i = 1, 2, \ldots, \mathrm{size}(\mathcal{C})$ **do**
      Send to player m the arm associated to player m in $\mathcal{C}[i]$ `// (comm.)`
    end
    Send to player m communication arms of the leader and player m, namely $\widetilde{\pi}_p^*(1)$ and $\widetilde{\pi}_p^*(m)$
  end
  if $size(\mathcal{C}) = 1$ **then** pull for the rest of the game the arm assigned to player 1 in the unique matching in $\mathcal{C}$ `// enter the exploitation phase`
  for $i = 1, 2, \ldots, \mathrm{size}(\mathcal{C})$ **do**
    pull 2^{p^c} times the arm assigned to player 1 in the matching $\mathcal{C}[i]$ `// exploration`
  end
  for $k = 1, 2, \ldots, K$ **do**
    $\widetilde{\mu}_k^1 \longleftarrow$ empirically estimated utility of arm k if it was pulled in this epoch, 0 otherwise
  end
  Receive the values $\widetilde{\mu}_1^m, \widetilde{\mu}_2^m, \ldots, \widetilde{\mu}_K^m$ from each player m `// (comm.)`
end

[2] This different behavior is the main reason for the improved regret upper bound obtained when the optimal matching is unique.

5.3 Analysis of M-ETC-Elim

We may assume that $K \leq T$, otherwise all parts of Theorem 5.1 would be trivial, since $R(T) \leq MT$ always. Theorem 5.2 provides a non-asymptotic upper bound on the regret of M-ETC-Elim.

Theorem 5.2. *Let $\pi^{m,k}$ be the best suboptimal matching assigning arm k to player m, and $\Delta_{(k,m)}$ its associated suboptimality gap, namely,*

$$\pi^{m,k} \in \arg\max \{U(\pi) \mid \pi(m) = k \text{ and } U(\pi) < U^\star\}$$
$$\text{and } \Delta_{(m,k)} \coloneqq \Delta(\pi^{m,k}) = U^* - U(\pi^{m,k}).$$

For all $c \geq 1$, let $T_0(c) := \exp\left(2^{\frac{c^c}{\log^c(1+\frac{1}{2c})}}\right)$. For all $T \geq T_0(c)$, the regret of M-ETC-Elim with parameter c is upper bounded as[3]

$$\begin{aligned} R(T) \leq\ & 2 + MK\log(e^2K^2T) + 6M^2K\log_2(K)(\log_2 T)^{1/c} \\ & + e^2MK(\log_2 T)^{1+1/c} + \frac{2M^3K\log_2(K)}{\sqrt{2}-1}\sqrt{\log(2M^2KT^2)} \\ & + \frac{2\sqrt{2}}{3-2\sqrt{2}}M^2K\sqrt{\log(2M^2KT^2)}\log_2(\log(T)) \\ & + \frac{2\sqrt{2}-1}{\sqrt{2}-1}\sum_{(m,k)\in[M]\times[K]}\left(\frac{32M^2\log(2M^2KT^2)}{\Delta_{(m,k)}}\right)^{1+1/c}, \end{aligned}$$

i.e.,

$$R(T) = \mathcal{O}\left(\sum_{m,k}\left(\frac{M^2\log(T)}{\Delta_{(m,k)}}\right)^{1+1/c} + MK(\log_2 T)^{1+1/c}\right).$$

The first statement of Theorem 5.1(a) easily follows by lower bounding $\Delta_{(m,k)} \geq \Delta$ for all m, k. Parts (b) and (c) of Theorem 5.1 similarly follow respectively from Theorems 5.3 and 5.4 in Sections 5.C.1 and 5.C.2, with proofs similar to that of Theorem 5.2 presented below.

The constant $T_0(c)$ in Theorem 5.2 equals 252 for $c = 1$ but becomes large when c increases. Still, the condition on T is explicit and independent of the problem parameters. In the case of multiple optimal matchings, our contribution is mostly theoretical, as we would need a large enough value of c and a long time $T_0(c)$ for reaching a prescribed $\log^{1+o(1)}(T)$ regret. However, in the case of a unique optimal matching (common in practice, and sometimes assumed in other papers), for the choice $c = 1$, the logarithmic regret upper bound stated in Theorem 5.3 is valid for all $T \geq 1$. Even if there are several optimal matchings, the minimax bound of Theorem 5.4 gives an $\mathcal{O}\left(\sqrt{T\log T}\right)$ regret bound that is a best-possible worst-case bound (also known as the

[3] $\log(\cdot)$ and $\log_2(\cdot)$ here denote the natural logarithm and the logarithm in base 2, respectively.

minimax rate), up to the $\sqrt{\log T}$ factor. Hence M-ETC-Elim with $c = 1$ is particularly good, both in theory and in practice. Our experiments also confirm that for $c = 1, 2$ the algorithm performs well (i.e., beats our competitors) even in the presence of multiple optimal matchings.

5.3.1 Sketch of Proof of Theorem 5.2

The analysis relies on several lemmas with proofs delayed to Section 5.C.3. Let $\mathcal{C}_p$ denote the set of candidate matchings used in epoch p, and for each matching π let $\widetilde{U}_p(\pi)$ be the utility of π that the leader can estimate based on the information received by the end of epoch p. Let $\hat{p}_T$ be the total number of epochs before the (possible) start of the exploitation phase. As $2^{\hat{p}_T^c} \leq T$, we have $\hat{p}_T \leq \log_2(T)$. Recall that a successful initialization means all players identify M and their ranks are distinct. Define the *good event*

$$\mathcal{G}_T \coloneqq \Big\{ \textsc{Init}(K, 1/KT) \text{ is successful and} \\ \forall p \leq \hat{p}_T, \forall \pi \in \mathcal{C}_{p+1}, |\widetilde{U}_p(\pi) - U(\pi)| \leq 2M\epsilon_p \Big\}. \tag{5.2}$$

During epoch p, for each candidate edge (m, k), player m has pulled arm k at least 2^{p^c} times and the quantization error is smaller than ϵ_p. Hoeffding's inequality and a union bound over at most $\log_2(T)$ epochs (see Section 5.C.3) together with Lemma 5.1 yield that $\mathcal{G}_T$ holds with large probability.

Lemma 5.2. $\mathbb{P}(\mathcal{G}_T) \geq 1 - \frac{2}{MT}$.

If $\mathcal{G}_T$ does not hold, we may upper bound the regret by MT. Hence it suffices to bound the expected regret conditional on $\mathcal{G}_T$, and the unconditional expected regret is bounded by this value plus 2.

Suppose that $\mathcal{G}_T$ happens. First, the regret incurred during the initialization phase is upper bounded by $MK\log(e^2K^2T)$ by Lemma 5.1. Moreover, the gap between the best estimated matching of the previous phase and the best matching is at most $2M\epsilon_{p-1}$ during epoch p. Each single communication round then incurs regret at most $2 + 2M\epsilon_{p-1}$, the first term being due to the collision between the leader and a follower, the second to the gap between the optimal matching and the matching used for communication. Summing over all communication rounds and epochs leads to Lemma 5.3 below.

Lemma 5.3. *The regret due to communication is bounded by*

$$3M^2K\log_2(K)\hat{p}_T + \frac{2^c\sqrt{2}}{3-2\sqrt{2}}M^2K\sqrt{\log(2/\delta)} \\ + MK(\hat{p}_T)^{c+1} + \frac{2M^3K\log_2(K)}{\sqrt{2}-1}\sqrt{\log(2/\delta)}.$$

For large horizons, Lemma 5.4 bounds some terms such as $\hat{p}_T$ and $(\hat{p}_T)^c$. When $c = 1$, tighter bounds that are valid for any T are used to prove Theorems 5.1(b) and 5.1(c).

Lemma 5.4. *For every suboptimal matching π, let $P(\pi) := \inf\{p \in \mathbb{N} : 8M\varepsilon_p < \Delta(\pi)\}$. The assumption $T \geq T_0(c)$ implies that for every matching π, $\Delta(\pi)2^{P(\pi)^c} \leq \left(\frac{32M^2\log(2M^2KT^2)}{\Delta(\pi)}\right)^{1+\frac{1}{c}}$. Also, $2^c \leq 2\log_2(\log(T))$, $\hat{p}_T \leq 2(\log_2 T)^{1/c}$ and $(\hat{p}_T)^c \leq e\log_2 T$.*

Hence for $T \geq T_0(c)$, we can further upper bound the first three terms of the sum in Lemma 5.3 by

$$6M^2K\log_2(K)(\log_2 T)^{1/c} + e^2MK(\log_2 T)^{1+1/c} + \frac{2\sqrt{2}}{3-2\sqrt{2}}M^2K\sqrt{\log(2/\delta)}\log_2(\log(T)). \tag{5.3}$$

It then remains to upper bound the regret incurred during exploration and exploitation phases. On $\mathcal{G}_T$, during the exploitation phase the players are jointly pulling an optimal matching and no regret is incurred. For an edge (m,k), let $\widetilde{\Delta}_p^{m,k} := U^\star - U(\widetilde{\pi}_p^{m,k})$ be the gap of its associated matching at epoch p. During epoch p, the incurred regret is then $\sum_{\pi\in\mathcal{C}_p}\Delta(\pi)2^{p^c} = \sum_{(m,k)\in\mathcal{E}}\widetilde{\Delta}_p^{m,k}2^{p^c}$.

Recall that $\pi^{m,k}$ is the best suboptimal matching assigning arm k to player m. Observe that for each epoch $p > P(\pi^{m,k})$, since $\mathcal{G}_T$ happens, $\pi^{m,k}$ (and any worse matching) is not added to $\mathcal{C}_p$; thus during each epoch $p > P(\pi^{m,k})$, the edge (m,k) is either eliminated from the set of candidate edges, or it is contained in some optimal matching and satisfies $\widetilde{\Delta}_p^{m,k} = 0$. Hence, the total regret incurred during exploration phases is bounded by

$$\sum_{(m,k)\in[M]\times[K]}\sum_{p=1}^{P(\pi^{m,k})}\widetilde{\Delta}_p^{m,k}2^{p^c}. \tag{5.4}$$

The difficulty for bounding this sum is that $\widetilde{\Delta}_p^{m,k}$ is random since $\widetilde{\pi}_p^{m,k}$ is random. However, $\widetilde{\Delta}_p^{m,k}$ can be related to $\Delta(\pi^{m,k})$ by $\widetilde{\Delta}_p^{m,k} \leq \frac{\epsilon_{p-1}}{\epsilon_{P(\pi^{m,k})}}\Delta(\pi^{m,k})$. A convexity argument then allows us to bound the ratio $\frac{\epsilon_{p-1}}{\epsilon_{P(\pi^{m,k})}}$, which yields Lemma 5.5, proved in Section 5.C.3.

Lemma 5.5. *For every edge (m,k), if $p < P(\pi^{m,k})$ then $\widetilde{\Delta}_p^{m,k}2^{p^c} \leq \Delta(\pi^{m,k})\frac{2^{P(\pi^{m,k})^c}}{\sqrt{2}^{P(\pi^{m,k})-(p+1)}}$.*

By Lemma 5.5, $\sum_{p=1}^{P(\pi^{m,k})}\widetilde{\Delta}_p^{m,k}2^{p^c}$ is upper bounded by $\left(\sum_{p=0}^{\infty}1/\sqrt{2}^p\right)\Delta(\pi^{m,k})2^{P(\pi^{m,k})^c} + \widetilde{\Delta}_{P(\pi^{m,k})}^{m,k}2^{P(\pi^{m,k})^c}$. As $\widetilde{\pi}_{P(\pi^{m,k})}^{m,k}$ is either optimal or its gap is larger than $\Delta(\pi^{m,k})$, Lemma 5.4 yields

$$\widetilde{\Delta}_{P(\pi^{m,k})}^{m,k}2^{P(\pi^{m,k})^c} \leq \left(\frac{32M^2\log(2M^2KT^2)}{\Delta(\pi^{m,k})}\right)^{1+1/c}$$

in both cases. Therefore, we find that

$$\sum_{p=1}^{P(\pi^{m,k})} \widetilde{\Delta}_p^{m,k} 2^{p^c} \leq \frac{2\sqrt{2}-1}{\sqrt{2}-1} \left(\frac{32M^2 \log(2M^2KT^2)}{\Delta(\pi^{m,k})} \right)^{1+1/c}.$$

Plugging this bound in (5.4), the bound (5.3) in Lemma 5.3 and summing up all terms yields Theorem 5.2.

5.3.2 Proof of Theorem 5.1(b), Unique Optimal Matching

The reader may wonder why can we obtain a better (logarithmic) bound if the maximum matching is unique. The intuition is as follows: in the presence of a unique optimal matching, M-ETC-Elim eventually enters the exploitation phase (which does not happen with multiple optimal matchings), and we can therefore provide a tighter bound on the number of epochs before exploitation phase compared with the one provided by Lemma 5.4. More precisely, in that case we have $\hat{p}_T \leq \log_2 \left(64M^2\Delta^{-2}\log(2M^2KT^2)\right)$. Moreover, another bound given by Lemma 5.4 can be tightened when $c = 1$ regardless of whether the optimal matching is unique or not: $\Delta(\pi)2^{P(\pi)} \leq 64M^2\log(2M^2KT^2)/\Delta(\pi)$.These two inequalities lead to Theorem 5.1(b), proved in Section 5.C.1.

5.3.3 Proof of Theorem 5.1(c), Minimax Regret Bound

Using the definition of the elimination rule, on $\mathcal{G}_T$ we have $\widetilde{\Delta}_p^{m,k} \leq 8M\epsilon_{p-1}$. Directly summing over these terms for all epochs yields an exploration regret scaling with $\sum_{m,k}\sqrt{t_{m,k}}$, where $t_{m,k}$ roughly corresponds to the number of exploration rounds associated with edge (m,k). This regret is maximized when all $t_{m,k}$ are equal, which leads to the sublinear regret bound of Theorem 5.1(c). See Section 5.C.2 for the rigorous statement and proof.

5.4 Numerical Experiments

We executed the following algorithms:M-ETC-Elim with $c = 1$ and $c = 2$, GoT (the latest version Bistritz and Leshem, 2020) with parameters[4] $\delta = 0, \varepsilon = 0.01, c_1 = 500, c_2 = c_3 = 6000$ and Selfish-UCB, a heuristic studied by Besson and Kaufmann (2018a) in the homogeneous setting which often performs surprisingly well despite the lack of theoretical evidence. In Selfish-UCB, each player runs the UCB1 algorithm of Auer et al. (2002a) on the reward sequence

[4]These parameters and the reward matrix U_1 are taken from the simulations section of (Bistritz and Leshem, 2020).

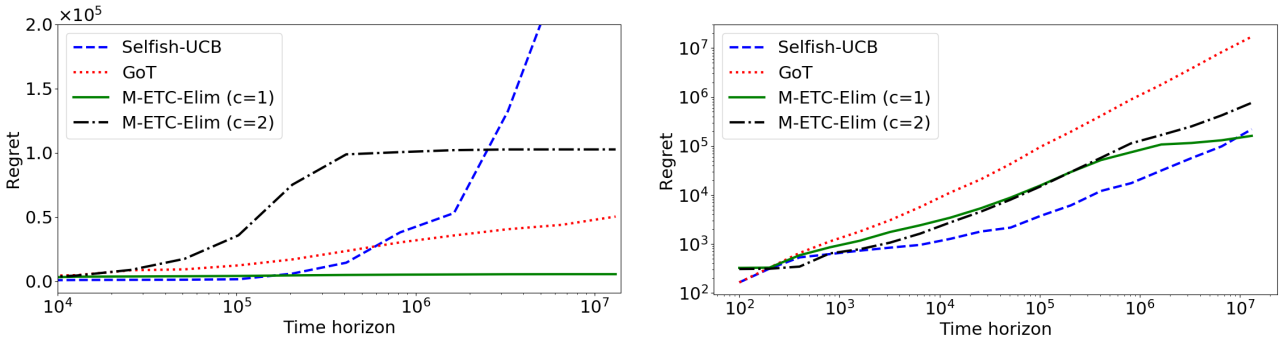


Figure 5.1: $R(T)$ as a function of T with reward matrices U_1 (left) and U_2 (right) and Bernoulli rewards.

$(r^m(t))_{t=1}^{\infty}$.[5] We experiment with Bernoulli rewards and the following reward matrices, whose entry (m, k) gives the value of μ_k^m:

$$U_1 = \begin{bmatrix} 0.1 & 0.05 & 0.9 \\ 0.1 & 0.25 & 0.3 \\ 0.4 & 0.2 & 0.8 \end{bmatrix}, U_2 = \begin{bmatrix} 0.5 & 0.49 & 0.39 & 0.29 & 0.5 \\ 0.5 & 0.49 & 0.39 & 0.29 & 0.19 \\ 0.29 & 0.19 & 0.5 & 0.499 & 0.39 \\ 0.29 & 0.49 & 0.5 & 0.5 & 0.39 \\ 0.49 & 0.49 & 0.49 & 0.49 & 0.5 \end{bmatrix}$$

Figure 5.1 reports the algorithms' regrets for various time horizons T, averaged over 100 independent replications. The first instance (matrix U_1, left plot) has a unique optimal matching and we observe that M-ETC-Elim has logarithmic regret (as promised by Theorem 5.1) and largely outperforms all competitors. The second instance (matrix U_2, right plot) is more challenging, with more arms and players, two optimal matchings and several near-optimal matchings. M-ETC-Elim with $c = 1$ performs the best for large T as well, though Selfish-UCB is also competitive. Yet there is very little theoretical understanding of Selfish-UCB, and it fails badly on the other instance. Section 5.B contains additional experiments corroborating our findings, where we also discuss practical aspects of implementing M-ETC-Elim.

[5] Note that this sequence is *not* i.i.d. due to some observed zeros that are due to collisions.

Appendix

5.A Description of the Initialization Procedure and Followers' Pseudocode

The pseudocode of the INIT(K, δ_0) procedure, already presented in Chapter 4, is presented in Algorithm 5.2 for the sake of completeness.

Next, we present the pseudocode that the followers execute in M-ETC-Elim. Recall that `(comm.)` refers to a call to the communication protocol.

5.B Practical Considerations and Additional Experiments

5.B.1 Implementation Enhancements for M-ETC-Elim

In the implementation of M-ETC-Elim, the following enhancements significantly improve the regret in practice (and have been used for the reported numerical experiments), but only by constant factors in theory, hence we have not included them in the analysis for the sake of brevity.

First, to estimate the means, the players are better off taking into account all pulls of the arms, rather than just the last epoch. Note that after the exploration phase of epoch p, each candidate edge has been pulled $N_p \coloneqq \sum_{i=1}^{p} 2^{ic}$ times. Thus, with probability at least $1-2\log_2(T)/(MT)$, each edge has been estimated within additive error $\leq \varepsilon'_p = \sqrt{\log(M^2TK)/2N_p}$ by Hoeffding's inequality. The players then truncate these estimates using $b \coloneqq \lceil -\log_2(0.1\varepsilon'_p) \rceil$ bits, adding up to $0.1\varepsilon'_p$ additive error due to quantization. They then send these b bits to the leader. Now, the threshold for eliminating a matching would be $2.2M\varepsilon'_p$ rather than $4M \times \sqrt{\log(2M^2KT^2)/2^{1+p^c}}$ (compare with line 8 of the Leaderalgorithm presented on page 103).

The second enhancement is to choose the set $\mathcal{C}$ of matchings to explore more carefully. Say that a matching is *good* if its estimated gap is at most $2.2M\varepsilon'_p$, and say an edge is *candidate* (lies in $\mathcal{E}$) if it is part of some good matching. There are at most MK candidate edges, and we

Algorithm 5.2: INIT, the initialization algorithm

Input: number of arms K, failure probability δ_0
Output: Ranking R, number of players M

```
// first, occupy a distinct arm using the musical chairs algorithm
```
$k \longleftarrow 0$
for $T_0 \coloneqq K\log(K/\delta_0)$ *rounds* **do** // rounds $1, \ldots, T_0$
  if $k = 0$ **then**
    pull a uniformly random arm $i \in [K]$
    if *no collision occurred* **then** $k \longleftarrow i$ // arm k is occupied
  else
    pull arm k
  end
end
```
// next, learn M and identify your ranking
```
$R \longleftarrow 1$
$M \longleftarrow 1$
for $2k-2$ *rounds* **do** // rounds $T_0+1, \ldots, T_0+2k-2$
  pull arm k
  if *collision occurred* **then**
    $R \longleftarrow R+1$
    $M \longleftarrow M+1$
  end
end
for $i = 1, 2, \ldots, K-k$ **do** // rounds $T_0+2k-1, \ldots, T_0+K+k-2$
  pull arm $k+i$
  if *collision occurred* **then**
    $M \longleftarrow M+1$
  end
end
for $K-k$ *rounds* **do** // rounds $T_0+K+k-1, \ldots, T_0+2K-2$
  pull arm 1
end

need only estimate those in the next epoch. Now, for each candidate edge, we can choose any good matching containing it, and add that to $\mathcal{C}$. This guarantees that $|\mathcal{C}| \leq MK$, which gives the bound in Theorem 5.1. But to reduce the size of $\mathcal{C}$ in practice, we do the following: initially, all edges are candidate. After each exploration phase, we do the following: we mark all edges as *uncovered*. For each candidate uncovered edge e, we compute the maximum matching π' containing e (using estimated means). If this matching π' has gap larger than $2.2M\varepsilon'_p$, then it is not good hence we remove e from the set of candidate edges. Otherwise, we add π' to $\mathcal{C}$, and moreover, we mark all of its edges as *covered*. We then look at the next uncovered candidate

Procedure Followeralgorithm(R,M) for the M-ETC-Elim algorithm with parameter c

Input: Ranking R, number of players M

for $p = 1, 2, \ldots$ **do**
  Receive the value of $\text{size}(\mathcal{C})$ // (comm.)
  for $i = 1, 2, \ldots, \text{size}(\mathcal{C})$ **do**
    Receive the arm assigned to this player in $\mathcal{C}[i]$ // (comm.)
  end
  Receive the communication arm of the leader and of this player
  if $size(\mathcal{C}) = 1$ // (enter exploitation phase)
  then
    pull for the rest of the game the arm assigned to this player in the unique matching in $\mathcal{C}$
  end
  for $i = 1, 2, \ldots, \text{size}(\mathcal{C})$ **do**
    pull 2^{p^c} times the arm assigned to this player in the matching $\mathcal{C}[i]$
  end
  for $k = 1, 2, \ldots, K$ **do**
    $\widehat{\mu}_k^R \longleftarrow$ empirically estimated utility of arm k if arm k has been pulled in this epoch, 0 otherwise
    Truncate $\widehat{\mu}_k^R$ to $\widetilde{\mu}_k^R$ using the $\frac{p^c+1}{2}$ most significant bits
  end
  Send the values $\widetilde{\mu}_1^R, \widetilde{\mu}_2^R, \ldots, \widetilde{\mu}_K^R$ to the leader // (comm.)
end

edge, and continue similarly, until all candidate edges are covered. This guarantees that all the candidate edges are explored, while the number of explored matchings could be much smaller than the number of candidate edges, which results in faster exploration and a smaller regret in practice.

To reduce the size of $\mathcal{C}$ even further, we do the following after each exploration phase: first, find the maximum matching (using estimated means), add it to $\mathcal{C}$, mark all its edges as covered, and only then start looking for uncovered candidate edges as explained above.

5.B.2 Other Reward Distributions

In our model and analysis, we have assumed $X_k^m(t) \in [0, 1]$ for simplicity (this is a standard assumption in online learning), but it is immediate to generalize the algorithm and its analysis to reward distributions bounded in any known interval via a linear transformation. Also, we can adapt our algorithm and analysis to subgaussian distributions with mean lying in a known interval. A random variable X is σ-subgaussian if for all $\lambda \in \mathbb{R}$ we have $\mathbb{E}[e^{\lambda(X-\mathbb{E}X)}] \leq e^{\sigma^2\lambda^2/2}$. This includes Gaussian distributions and distributions with bounded support. Suppose

for simplicity that the means lie in $[0, 1]$. Then the algorithm need only change in two places: first, when the followers are sending the estimated means to the leader, they must send 0 and 1 if the empirically estimated mean is < 0 and > 1, respectively. Second, the definition of ε_p must be changed to $\varepsilon_p := \sqrt{\sigma^2 \log(2/\delta)/2^{p^c-1}}$. The only change in the analysis is that instead of using Hoeffding's inequality which requires a bounded distribution, one has to use a concentration inequality for sums of subgaussian distributions(see e.g., Wainwright, 2019, Proposition 2.5).

We executed the same algorithms as in Section 5.4 with the same reward matrices but with Gaussian rewards with variance 0.05. The results are somewhat similar to the Bernoulli case and can be found in Figure 5.2.

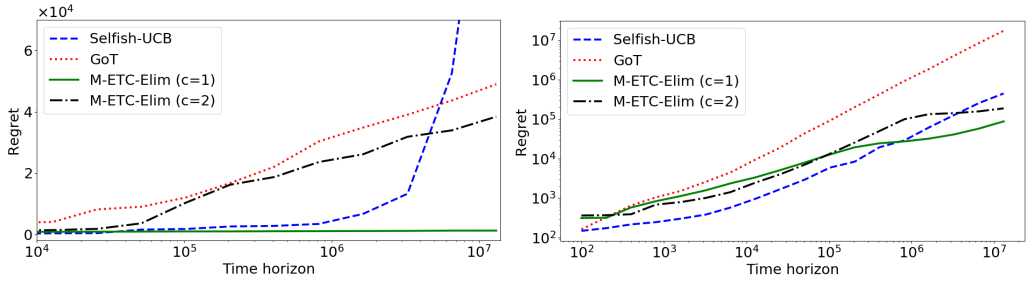


Figure 5.2: Numerical comparison of M-ETC-Elim, GoT and Selfish-UCB on reward matrices U_1 (left) and U_2 (right) with Gaussian rewards and variance 0.05. The x-axis has logarithmic scale in both plots. The y-axis has logarithmic scale in the right plot.

The reason we performed these Gaussian experiments is to have a more fair comparison against GoT. Indeed, the numerical experiments of Bistritz and Leshem (2020) rely on the same reward matrix U_1 and Gaussian rewards.

5.C Omitted proofs

5.C.1 Regret Analysis in the Presence of a Unique Maximum Matching

In Theorem 5.3 below we provide a refined analysis of M-ETC-Elim with parameter $c = 1$ if the maximum matching is unique, justifying the $\mathcal{O}\left(\frac{KM^3}{\Delta}\log(T)\right)$ regret upper bound stated in Theorem 5.1(b). Its proof, given below, follows essentially the same line as the finite-time analysis given in Section 5.3, except for the last part. Recall that $\log(\cdot)$ denotes the natural logarithm and $\log_2(\cdot)$ denotes logarithm in base 2.

Theorem 5.3. *If the maximum matching is unique, for any $T > 0$ the regret of the M-ETC-Elim*

algorithm with parameter $c = 1$ is upper bounded by

$$2 + MK\log(e^2K^2T) + 3M^2K\log_2(K)\log_2\left(\frac{64M^2\log(2M^2KT^2)}{\Delta^2}\right) + MK\log_2^2\left(\frac{64M^2\log(2M^2KT^2)}{\Delta^2}\right)$$
$$+ \frac{4\sqrt{2}-2}{3-2\sqrt{2}}M^3K\log_2(K)\sqrt{\log(2M^2KT^2)} + \frac{2\sqrt{2}-1}{\sqrt{2}-1}\sum_{(m,k)\in[M]\times[K]}\frac{64M^2\log(2M^2KT^2)}{\Delta(\pi^{m,k})}.$$

Proof. The good event and the regret incurred during the initialization phase are the same as in the finite-time analysis given in Section 5.3. Recall the definition of P, which is $P(\pi) = \inf\{p \in \mathbb{N} : 8M\varepsilon_p < \Delta(\pi)\}$. When there is a unique optimal matching, if the good event happens, the M-ETC-Elim algorithm will eventually enter the exploitation phase, so $\hat{p}_T$ can be much smaller than the crude upper bound given by Lemma 5.4. Specifically, introducing π' as the second maximum matching so that $\Delta(\pi') = \Delta$, we have, on the event $\mathcal{G}_T$,

$$\hat{p}_T \le P(\pi') \le \log_2\left(\frac{64M^2\log(2M^2KT^2)}{\Delta^2}\right).$$

Plugging this bound in Lemma 5.3 yields that the regret incurred during communications is bounded by

$$3M^2K\log_2(K)\log_2\left(\frac{64M^2\log(2M^2KT^2)}{\Delta^2}\right) + MK\log_2^2\left(\frac{64M^2\log(2M^2KT^2)}{\Delta^2}\right)$$
$$+\frac{2M^3K\log_2 K}{\sqrt{2}-1}\sqrt{\log(2/\delta)} + \frac{2\sqrt{2}}{3-2\sqrt{2}}M^2K\sqrt{\log(2/\delta)}.$$

Also, for $c = 1$ and ever matching π, the definition of ε_p in (5.1) gives

$$P(\pi) \le 1 + \log_2\left(\frac{32M^2\log(2M^2KT^2)}{\Delta(\pi)^2}\right).$$

In particular, $\Delta(\pi)2^{P(\pi)} \le \frac{64M^2\log(2M^2KT^2)}{\Delta(\pi)}$. Using the same argument as in Section 5.3, the regret incurred during the exploration phases is bounded by

$$\frac{2\sqrt{2}-1}{\sqrt{2}-1}\sum_{(m,k)\in[M]\times[K]}\frac{64M^2\log(2M^2KT^2)}{\Delta_{(m,k)}}.$$

Summing up the regret bounds for all phases proves Theorem 5.3. □

5.C.2 Minimax Regret Analysis

In Theorem 5.4 below we provide a minimax regret bound for M-ETC-Elim with parameter $c = 1$, justifying the $\mathcal{O}\left(M^{\frac{3}{2}}\sqrt{KT\log(T)}\right)$ regret upper bound stated in Theorem 5.1(c).

Theorem 5.4. *For all T, the regret of the M-ETC-Elim algorithm with parameter $c = 1$ is upper bounded by*

$$2 + MK\log(e^2K^2T) + 3M^2K\log_2(K)\log_2(T) + MK\log_2^2(T) \\ + \frac{4\sqrt{2}-2}{3-2\sqrt{2}}M^3K\log_2(K)\sqrt{\log(2M^2KT^2)} + \frac{8}{\sqrt{2}-1}K^{\frac{1}{2}}M^{\frac{3}{2}}\sqrt{T\log(2M^2KT^2)}.$$

Note that the above regret bound is independent of the suboptimality gaps.

Proof. The good event and the regret incurred during the initialization phase are the same as in the finite-time analysis given in Section 5.3. Furthermore, using Lemma 5.3 stated therein and since $\hat{p}_T \leq \log_2(T)$, the regret incurred during the communication phases is bounded by

$$3M^2K\log_2(K)\log_2(T) + MK\log_2^2(T) + \frac{4\sqrt{2}-2}{3-2\sqrt{2}}M^3K\log_2(K)\sqrt{\log(2M^2KT^2)}.$$

We next bound the exploration regret. Fix the edge (m,k), and let $\widetilde{P}^{m,k}$ be the last epoch in which this edge is explored. If this edge belongs to an optimal matching, i.e., if $\pi^{m,k}$ is optimal, we instead define $\widetilde{P}^{m,k}$ as the last epoch in which the pulled matching $\widetilde{\pi}_p^{m,k}$ associated with (m,k) is suboptimal. In either case, the contribution of the edge (m,k) to the exploration regret can be bounded by $\sum_{p=1}^{\widetilde{P}^{m,k}} \widetilde{\Delta}_p^{m,k} 2^p$.

Fix an epoch $p \leq \widetilde{P}^{m,k}$. Recall that $\mathcal{C}_p$ contains at least one actual maximum matching, which we denote by $\pi^\star$. Also, let $\widetilde{\pi}_p^\star$ denote the maximum empirical matching right before the start of epoch p. Since (m,k) is candidate in epoch p, we have

$$\begin{aligned}
\widetilde{\Delta}_p^{m,k} &= U^* - \widetilde{U}_{p-1}(\pi_p^\star) + \widetilde{U}_{p-1}(\pi_p^\star) - \widetilde{U}_{p-1}(\widetilde{\pi}_p^{m,k}) + \widetilde{U}_{p-1}(\widetilde{\pi}_p^{m,k}) - U(\widetilde{\pi}_p^{m,k}) \\
&\leq (U^\star - \widetilde{U}_{p-1}(\pi^\star)) + (\widetilde{U}_{p-1}(\widetilde{\pi}_p^\star) - \widetilde{U}_{p-1}(\widetilde{\pi}_p^{m,k}) + (\widetilde{U}_{p-1}(\widetilde{\pi}_p^{m,k}) - U(\widetilde{\pi}_p^{m,k})) \\
&\leq 2M\epsilon_{p-1} + 4M\epsilon_p + 2M\epsilon_{p-1} \\
&\leq 8M\epsilon_{p-1} = 8M\sqrt{\frac{\log(2/\delta)}{2^p}},
\end{aligned}$$

so, the contribution of the edge (m,k) to the exploration regret can further be bounded by

$$\sum_{p=1}^{\widetilde{P}^{m,k}} \widetilde{\Delta}_p^{m,k} 2^p \leq 8M\sqrt{\log(2/\delta)}\left(\sum_{p=1}^{\widetilde{P}^{m,k}} \sqrt{2}^p\right) < \frac{8\sqrt{2}M\sqrt{\log(2/\delta)}}{\sqrt{2}-1}\sqrt{2}^{\widetilde{P}^{m,k}}.$$

To bound the total exploration regret, we need to sum this over all edges (m,k).

Note that during each epoch $p = 1, 2, \ldots, \widetilde{P}_{m,k}$, there are exactly 2^p exploration rounds

associated with the edge (m, k). Since the total number of rounds is T, we find that

$$\sum_{(m,k)\in[M]\times[K]} \sum_{p=1}^{\widetilde{P}_{m,k}} 2^p \leq T,$$

and in particular,

$$\sum_{(m,k)\in[M]\times[K]} 2^{\widetilde{P}_{m,k}} \leq T,$$

hence by the Cauchy-Schwarz inequality,

$$\sum_{(m,k)\in[M]\times[K]} \sqrt{2}^{\widetilde{P}_{m,k}} = \sum_{(m,k)\in[M]\times[K]} \sqrt{2^{\widetilde{P}_{m,k}}} \leq \sqrt{MKT},$$

so the total exploration regret can be bounded by

$$\frac{8\sqrt{2}M\sqrt{\log(2/\delta)}}{\sqrt{2}-1} \sum_{(m,k)\in[M]\times[K]} \sqrt{2}^{\widetilde{P}^{m,k}} \leq \frac{8\sqrt{2}M\sqrt{\log(2/\delta)}}{\sqrt{2}-1}\sqrt{MKT},$$

completing the proof of Theorem 5.4. □

5.C.3 Proofs of Auxiliary Lemmas for Theorems 5.2 and 5.3

Proof of Lemma 5.2

We recall Hoeffding's inequality.

Proposition 5.1 (Hoeffding's inequality Hoeffding, 1963, Theorem 2)**.** *Let $X_1, \ldots, X_n$ be independent random variables taking values in $[0, 1]$. Then for $t \geq 0$ we have*

$$\mathbb{P}\left(\left|\frac{1}{n}\sum X_i - \mathbb{E}\left[\frac{1}{n}\sum X_i\right]\right| > t\right) < 2\exp(-2nt^2).$$

Recall the definition of the good event

$$\mathcal{G}_T = \Big\{\textsc{Init}(K, 1/KT) \text{ is successful and } \forall p \leq \hat{p}_T, \forall \pi \in \mathcal{C}_{p+1}, |\widetilde{U}_p(\pi) - U(\pi)| \leq 2M\epsilon_p\Big\}.$$

and recall that $\varepsilon_p := \sqrt{\log(2/\delta)/2^{p^c+1}}$ and $\delta = 1/M2KT^2$. Let $\mathcal{H}$ be the event that $\textsc{Init}(K, 1/KT)$ is successful for all players. Then,

$$\begin{aligned}\mathbb{P}\left(\mathcal{G}_T^c\right) &\leq \mathbb{P}\left(\mathcal{H}^c\right) + \mathbb{P}\left(\mathcal{H} \text{ happens and } \exists p \leq \hat{p}_T, \exists \pi \in \mathcal{M} \text{ with candidate edges such that } |\widetilde{U}_p(\pi) - U(\pi)| > 2M\epsilon_p\right) \\ &\leq \frac{1}{KT} + \mathbb{P}\left(\mathcal{H} \text{ happens and } \exists p \leq \log_2(T), \exists \pi \in \mathcal{M} \text{ with candidate edges such that} |\widetilde{U}_p(\pi) - U(\pi)| > 2M\epsilon_p\right),\end{aligned}$$

where we have used that $\hat{p}_T \leq \log_2(T)$ deterministically.

Fix an epoch p and a candidate edge (m,k). We denote by $\widehat{\mu}_k^m(p)$ the estimated mean of arm k for player m at the end of epoch p and by $\widetilde{\mu}_k^m(p)$ the truncated estimated mean sent to the leader by this player at the end of epoch p.

By Hoeffding's inequality and since this estimated mean is based on at least 2^{p^c} pulls, we have

$$\mathbb{P}\left(|\widehat{\mu}_k^m(p) - \mu_k^m| > \varepsilon_p\right) < \delta.$$

The value $\widetilde{\mu}_k^m(p) \in [0,1]$ which is sent to the leader uses the $(p^c+1)/2$ most significant bits. The truncation error is thus at most $2^{-(p^c+1)/2} < \varepsilon_p$, hence we have

$$\mathbb{P}\left(|\widetilde{\mu}_k^m(p) - \mu_k^m| > 2\varepsilon_p\right) < \delta.$$

Given the event $\mathcal{H}$ that the initialization is successful, the quantity $\widetilde{U}_p(\pi)$ is a sum of M values $\widetilde{\mu}_k^m(p)$ for M different edges $(m,k) \in [M] \times [K]$. Hence, we have

$$\begin{aligned}\mathbb{P}\Big(&\mathcal{H} \text{ happens and } \exists \pi \in \mathcal{M} \text{ with candidate edges such that } |\widetilde{U}_p(\pi) - U(\pi)| > 2M\varepsilon_p|\Big) \\ &\leq \mathbb{P}\left(\exists \text{ candidate edge } (m,k) \text{ such that } |\widetilde{\mu}_k^m(p) - \mu_k^m| > 2\varepsilon_p\right) \leq KM\delta.\end{aligned}$$

Finally, a union bound on p yields

$$\mathbb{P}\left(\mathcal{G}_T^c\right) \quad \leq \quad \frac{1}{KT} + \log_2(T)KM\delta \leq \frac{1}{MT} + \frac{1}{MT},$$

completing the proof of Lemma 5.2

Proof of Lemma 5.3

For each epoch p, the leader first communicates to each player the list of candidate matchings. There can be up to MK candidate matchings, and for each of them the leader communicates to the player the arm she has to pull (there is no need to communicate to her the whole matching) which requires $\log_2 K$ bits, and there are a total of M players, so this takes at most $M^2K\log_2(K)$ many rounds.[6]

At the end of the epoch, each player sends the leader the empirical estimates for the arms she has pulled, which requires at most $MK(1+p^c)/2$ many rounds. As players use the best estimated matching as communication arms for the communication phases, a single communication round incurs regret at most $2 + 2M\epsilon_{p-1}$, since the gap between the best estimated matching of the previous phase and the best matching is at most $2M\epsilon_{p-1}$ conditionally to $\mathcal{G}_T$ (we define

[6] Strictly speaking, the leader also sends her communication arm and the size of the list she is sending, but there are at most $MK - M + 1$ candidate matchings, as the best one is repeated M times. So, this communication still takes at most $M^2K\log_2 K$ many rounds.

$\epsilon_0 \coloneqq \sqrt{\frac{\log(2/\delta)}{2}} \geq \frac{1}{2}$). The first term is for the two players colliding, while the term $2M\epsilon_{p-1}$ is due to the other players who are pulling the best estimated matching instead of the real best one. With $\hat{p}_T$ denoting the number of epochs before the (possible) start of the exploitation, the total regret due to communication phases can be bounded by

$$\begin{aligned} R_c &\leq \sum_{p=1}^{\hat{p}_T} \left(2M^2 K \log_2(K) + MK(1+p^c)\right)(1+M\epsilon_{p-1}) \\ &\leq 3M^2 K \log_2(K)\hat{p}_T + MK(\hat{p}_T)^{c+1} + M^2 K \sum_{p=1}^{\hat{p}_T} \left(2M\log_2(K) + (1+p^c)\right)\epsilon_{p-1}. \end{aligned}$$

We now bound the sum as:

$$\begin{aligned} \sum_{p=1}^{\hat{p}_T} \left(2M\log_2(K) + (1+p^c)\right)\epsilon_{p-1} &= 2M\log_2(K)\sqrt{\log(2/\delta)} \sum_{p=0}^{\hat{p}_T-1} \frac{1}{\sqrt{2}^{1+p^c}} + \sqrt{\log(2/\delta)} \sum_{p=0}^{\hat{p}_T-1} \frac{1+(p+1)^c}{\sqrt{2}^{1+p^c}} \\ &\leq 2M\log_2(K)\sqrt{\log(2/\delta)} \sum_{n=1}^{\infty} \frac{1}{\sqrt{2}^n} + \sqrt{\log(2/\delta)} \sum_{n=1}^{\infty} \frac{n2^c}{\sqrt{2}^n} \\ &\leq 2M\log_2(K)\sqrt{\log(2/\delta)} \frac{1}{\sqrt{2}-1} + \sqrt{\log(2/\delta)} \frac{2^c\sqrt{2}}{(\sqrt{2}-1)^2}, \end{aligned}$$

completing the proof of Lemma 5.3.

Proof of Lemma 5.4

The assumption $T \geq \exp(2^{\frac{c^c}{\log^c(1+\frac{1}{2c})}})$ gives $\log_2(\log T)^{1/c} \geq \frac{c}{\log(1+1/2c)}$. In particular, $(\log_2 T)^{1/c} \geq c$. We will also use the inequality

$$(x+1)^c \leq e^{c/x} x^c, \tag{5.5}$$

which holds for all positive x, since $(x+1)^c/x^c = (1+1/x)^c \leq \exp(1/x)^c = \exp(c/x)$.

Using a crude upper bound on the number of epochs that can fit within T rounds, we get $\hat{p}_T \leq 1 + (\log_2 T)^{1/c}$. As $(\log_2 T)^{1/c} \geq c \geq 1$ we have $\hat{p}_T \leq 2(\log_2 T)^{1/c}$. Also (5.5) gives $(\hat{p}_T)^c \leq e \log_2 T$.

Also, $2\log_2(\log(T)) \geq 2c^c \geq 2^c$. It remains to show the first inequality of Lemma 5.4.

Straightforward calculations using the definition of ε_p in (5.1) give

$$P(\pi) \leq 1 + L(\pi)^{1/c}, \text{ where } L(\pi) \coloneqq \log_2\left(\frac{32M^2\log(2M^2KT^2)}{\Delta(\pi)^2}\right).$$

We claim that we have

$$P(\pi)^c \leq \left(1+\frac{1}{2c}\right) L(\pi). \tag{5.6}$$

Indeed, since $\Delta(\pi) \leq M$, we have $L(\pi)^{1/c} > (\log_2 \log T)^{1/c} \geq \frac{c}{\log(1+1/2c)}$ and so (5.5) with $x = L(\pi)^{1/c}$ gives (5.6). Hence,

$$\Delta(\pi)2^{P(\pi)^c} \leq \Delta(\pi)\left(\frac{32M^2\log(2M^2KT^2)}{\Delta(\pi)^2}\right)^{1+1/2c} \leq \left(\frac{32M^2\log(2M^2KT^2)}{\Delta(\pi)}\right)^{1+1/c}, \tag{5.7}$$

completing the proof of Lemma 5.4.

Proof of Lemma 5.5

For brevity we define, for this proof only, $\Delta \coloneqq \Delta(\pi^{m,k})$, $P \coloneqq P(\pi^{m,k})$ and $\Delta_p \coloneqq \widetilde{\Delta}_p^{m,k}$. First, $\Delta > 8M\epsilon_P$ by definition of P. Also, $\Delta_p \leq 8M\epsilon_{p-1}$ for every $p \leq P-1$, otherwise the edge (m,k) would have been eliminated before epoch p. It then holds

$$\Delta_p \leq \frac{\epsilon_{p-1}}{\epsilon_P}\Delta = \sqrt{2}^{P^c-(p-1)^c}\Delta. \tag{5.8}$$

It comes from the convexity of $x \mapsto x^c$ that $(p+1)^c + (p-1)^c - 2p^c \geq 0$, and thus

$$P^c + (p-1)^c - 2p^c \geq P^c - (p+1)^c \geq P - (p+1).$$

It then follows

$$p^c + \frac{P^c - (p-1)^c}{2} \leq P^c + \frac{p+1-P}{2}.$$

Plugging this in (5.8) gives

$$2^{p^c}\Delta_p \leq \frac{2^{P^c}}{\sqrt{2}^{P-(p+1)}}\Delta,$$

completing the proof of Lemma 5.5.

Chapter 6

Selfish Robustness and Equilibria in Multi-Player Bandits

While the cooperative case where players maximize the collective reward (obediently following some fixed protocol) has been mostly considered, robustness to malicious players is a crucial and challenging concern of multiplayer bandits. Existing approaches consider only the case of adversarial *jammers* whose objective is to blindly minimize the collective reward.

We shall consider instead the more natural class of selfish players whose incentives are to maximize their individual rewards, potentially at the expense of the social welfare. We provide the first algorithm robust to selfish players (a.k.a. Nash equilibrium) with a logarithmic regret, when the arm performance is observed. When collisions are also observed, *Grim Trigger* type of strategies enable some implicit communication-based algorithms and we construct robust algorithms in two different settings: the homogeneous (with a regret comparable to the centralized optimal one) and heterogeneous cases (for an adapted and relevant notion of regret). We also provide impossibility results when only the reward is observed or when arm means vary arbitrarily among players.

In most of the multiplayer bandits literature, as well as in Chapters 4 and 5, a crucial (yet sometimes only implicitly stated) assumption is that all players follow cautiously and meticulously some designed protocols and that none of them tries to free-ride the others by acting greedily, selfishly or maliciously. The concern of designing multiplayer bandit algorithms robust to such players has been raised (Attar et al., 2012), but only addressed under the quite restrictive assumption of adversarial players called *jammers*. Those try to perturb as much as possible the cooperative players (Wang et al., 2015; Sawant et al., 2018; Sawant et al., 2019), even if this is extremely costly to them as well. Because of this specific objective, they end up using tailored strategies such as only attacking the top channels.

We focus instead on the construction of algorithms with "good" regret guarantees even if one (or actually more) selfish player does not follow the common protocol but acts strategically in order to manipulate the other players in the sole purpose of increasing her own payoff – maybe at the cost of other players. This concept appeared quite early in the cognitive radio literature (Attar et al., 2012), yet it is still not understood as robustness to selfish player is intrinsically different (and even non-compatible) with robustness to jammers, as shown in Section 6.1.1. In terms of game theory, we aim at constructing (ε-Nash) equilibria in this repeated game with partial observations.

This chapter is organized as follows. Section 6.1 introduces notions and concepts of selfishness-

robust multiplayer bandits and showcases reasons for the design of robust algorithms. Besides its state of the art regret guarantees when collisions are not directly observed, `Selfish-Robust MMAB`, presented in Section 6.2, is also robust to selfish players. In the more complex settings where only the reward is observed or the arm means vary among players, Section 6.3 shows that no algorithm can guarantee both a sublinear regret and selfish-robustness. The latter case is due to a more general result for random assignments. Instead of comparing the cumulated reward with the best collective assignment in the heterogeneous case, it is then necessary to compare it with a *good* and appropriate suboptimal assignment, leading to the new notion of *RSD-regret*.

When collisions are always observed, Section 6.4 proposes selfish-robust communication protocols. Thanks to this, an adaptation of the work of Boursier and Perchet (2019) is possible to provide a robust algorithm with a collective regret almost scaling as in the centralized case. In the heterogeneous case, this communication – along with other new deviation control and punishment protocols – is also used to provide a robust algorithm with a logarithmic RSD-regret.

Our contributions are thus diverse: on top of introducing notions of selfish-robustness, we provide robust algorithms with state of the art regret bounds (w.r.t. non-robust algorithms) in several settings. This is especially surprising when collisions are observed, since it leads to a near centralized regret. Moreover, we show that such algorithms can not be designed in harder settings. This leads to the new, adapted notion of RSD-regret in the heterogeneous case with selfish players and we also provide a *good* algorithm in this case. These results of robustness are even more intricate knowing they hold against any possible selfish strategy, in contrast to the known results for jammer robust algorithms.

6.1 Problem statement

In this section, we introduce concepts and notions of robustness to selfish players (or equilibria concepts) in the problem of multiplayer bandits introduced in Section 3.3.1.

6.1.1 Considering selfish players

As mentioned above, the literature focused on adversarial malicious players, a.k.a. *jammers*, while considering selfish players instead of adversarial ones is as (if not more) crucial. These two concepts of malicious players are fundamentally different. Jamming-robust algorithms must stop pulling the best arm if it is being jammed. Against this algorithm, a selfish player could therefore pose as a jammer, always pull the best arm and be left alone on it most of the time. On the contrary, an algorithm robust to selfish players has to actually pull this best arm if jammed

by some player in order to "punish" her so that she does not benefit from deviating from the collective strategy.

We first introduce some game theoretic concepts before defining notions of robustness. Each player j follows an individual strategy (or algorithm) $s^j \in \mathcal{S}$ which determines her action at each round given her past observations. As in Section 2.1, we denote by $(s^1, \dots, s^M) = s \in \mathcal{S}^M$ the strategy profile of all players and by $(s', \boldsymbol{s^{-j}})$ the strategy profile given by s except for the j-th player whose strategy is replaced by s'. Let $\mathrm{Rew}_T^j(\boldsymbol{s})$ be the cumulative reward of player j when players play the profile $\boldsymbol{s}$. As usual in game theory, we consider a single selfish player – even if the algorithms we propose are robust to several selfish players assuming M is known beforehand (its initial estimation can easily be tricked by several players).

Definition 6.1. *A strategy profile $\boldsymbol{s} \in \mathcal{S}^M$ is an ε-Nash equilibrium if for all $s' \in \mathcal{S}$ and $j \in [M]$:*

$$\mathbb{E}[\mathrm{Rew}_T^j(s', \boldsymbol{s^{-j}})] \leq \mathbb{E}[\mathrm{Rew}_T^j(\boldsymbol{s})] + \varepsilon.$$

This simply states that a selfish player wins at most ε by deviating from s^j. We now introduce a more restrictive property of stability that involves two points: if a selfish player still were to deviate, this would only incur a small loss to other players. Moreover, if the selfish player wants to incur some considerable loss to the collective players (e.g., she is adversarial), then she also has to incur a comparable loss to herself. Obviously, an ε-Nash equilibrium is $(0, \varepsilon)$-stable.

Definition 6.2. *A strategy profile $s \in \mathcal{S}^M$ is (α, ε)-stable if for all $s' \in \mathcal{S}$, $l \in \mathbb{R}_+$ and $i, j \in [M]$:*

$$\mathbb{E}[\mathrm{Rew}_T^i(s', \boldsymbol{s^{-j}})] \leq \mathbb{E}[\mathrm{Rew}_T^i(s)] - l \implies \mathbb{E}[\mathrm{Rew}_T^j(s', \boldsymbol{s^{-j}})] \leq \mathbb{E}[\mathrm{Rew}_T^j(\boldsymbol{s})] + \varepsilon - \alpha l.$$

6.1.2 Limits of existing algorithms.

This section explains why existing algorithms are not robust to selfish players, i.e., are not even $o(T)$-Nash equilibria. Besides justifying the design of new appropriate algorithms, this provides some first insights on the way to achieve robustness.

Communication between players. Many recent algorithms rely on communication protocols between players to gather their statistics. Facing an algorithm of this kind, a selfish player would communicate fake statistics to the other players in order to keep the best arm for herself. In case of collision, the colliding player(s) remains unidentified, so a selfish player could modify *incognito* the statistics sent by other players, making them untrustworthy. A way to make such protocols robust to malicious players is proposed in Section 6.4. Algorithms relying on communication can then be adapted in the Full Sensing setting.

Necessity of fairness An algorithm is fair if all players asymptotically earn the same reward *a posteriori* (or *ex post*) and not only in expectation (*ex ante*). As already noticed (Attar et al., 2012), fairness seems to be a significant criterion in the design of selfish-robust algorithms. Indeed, without fairness, a selfish player tries to always be the one with the largest reward .

For example, against algorithms attributing an arm among the top-M ones to each player (Rosenski et al., 2016; Besson and Kaufmann, 2018a; Boursier and Perchet, 2019), a selfish player could easily rig the attribution to end with the best arm, largely increasing her individual reward. Other algorithms work on the basis of first come-first served (Boursier and Perchet, 2019). Players first explore and when they detect an arm as both optimal and available, they pull it forever. Such an algorithm is unfair and a selfish player could play more aggressively to end her exploration before the others and to commit on an arm, maybe at the risk of committing on a suboptimal one (but with high probability on the best arm). The risk taken by the early commit is small compared to the benefit of being the first committing player. As a consequence, these algorithms are not $o\left(T\right)$-Nash equilibria.

6.2 Statistic sensing setting

In the statistic sensing setting where X_k and r_k are observed at each round, the `Selfish-Robust MMAB` algorithm provides satisfying theoretical guarantees.

6.2.1 Description of `Selfish-Robust MMAB`

Algorithm 6.1: `Selfish-Robust MMAB`

Input: $T, \gamma_1 \coloneqq \frac{13}{14}, \gamma_2 \coloneqq \frac{16}{15}$

$\beta \leftarrow 39$; $\hat{M}, t_m \leftarrow$ `EstimateM` (β, T)

Pull $k \sim \mathcal{U}(K)$ until round $\frac{\gamma_2}{\gamma_1} t_m$ `// first waiting room`

$j \leftarrow$ `GetRank` $(\hat{M}, t_m, \beta, T)$

Pull j until round $\left(\frac{\gamma_2}{\gamma_1^2 \beta^2 K^2} + \frac{\gamma_2^2}{\gamma_1^2}\right) t_m$ `// second waiting room`

Run `Alternate Exploration` $(\hat{M}, j)$ until T

A global description of `Selfish-Robust MMAB` is given by Algorithm 6.1. The pseudocodes of `EstimateM`, `GetRank` and `Alternate Exploration` are given by Protocols 6.1, 6.2 and Algorithm 6.2 in Section 6.A for completeness.

`EstimateM` and `GetRank` respectively estimate the number of players M and attribute ranks in $[M]$ among the players. They form the initialization phase, while `Alternate Exploration` optimally balances between exploration and exploitation.

Initialization phase

Let us first introduce the following quantities:

- $T_k^j(t) = \{t' \leq t \mid \pi^j(t') = k \text{ and } X_k(t') > 0\}$ are rounds when player j observed η_k.
- $C_k^j(t) = \{t' \in T_k^j(t) \mid \eta_k(t') = 1\}$ are rounds when player j observed a collision.
- $\hat{p}_k^j(t) = \frac{\#C_k^j(t)}{\#T_k^j(t)}$ is the empirical probability to collide on the arm k for player j.

During the initialization, the players estimate M with large probability as given by Lemma 6.1 in Section 6.A.1. Players first pull uniformly at random in $[K]$. As soon as $\#T_k^j \geq n$ for all $k \in [K]$ and some fixed n, player j ends the `EstimateM` protocol and estimates $\hat{M}$ as the closest integer to $1 + \log(1 - \frac{\sum_k \hat{p}_k^j(t_M)}{K}) / \log(1 - \frac{1}{K})$. This estimation procedure is the same as the one of Rosenski et al. (2016), except for the following features:

i) Collisions indicators are not always observed, as we consider statistic sensing here. For this reason, the number of observations of η_k is random. The stopping criterion $\min_k \#T_k^j(t) \geq n$ ensures that players don't need to know $\mu_{(K)}$ beforehand, but they also do not end `EstimateM` simultaneously. This is why a *waiting room* is needed, during which a player continues to pull uniformly at random to ensure that all players are still pulling uniformly at random if some player is still estimating M.

ii) The collision probability is not averaged over all arms, but estimated for each arm individually, then averaged. This is necessary for robustness as explained in Section 6.A, despite making the estimation longer.

Attribute ranks. After this first procedure, players then proceed to a *Musical Chairs* (Rosenski et al., 2016) phase to attribute ranks among them as given by Lemma 6.2 in Section 6.A.1. Players sample uniformly at random in $[M]$ and stop on an arm j as soon as they observe a positive reward. The player's rank is then j and only attributed to her. Here again, a *waiting room* is required to ensure that all players are either pulling uniformly at random or only pulling a specific arm (corresponding to their rank) during this procedure. During this second waiting room, a player thus pulls the arm corresponding to her rank.

Exploration/exploitation

After the initialization, players know M and have different ranks. They enter the second phase, where they follow `Alternate Exploration`, inspired by Proutiere and Wang (2019). Player j sequentially pulls arms in $\mathcal{M}^j(t)$, which is the ordered list of her M best empirical arms, unless she has to pull her M-th best empirical arm. In that case, she instead chooses at random

between actually pulling it or pulling an arm to explore (any arm not in $\mathcal{M}^j(t)$ with an upper confidence bound larger than the M-th best empirical mean, if there is any).

Since players proceed in a shifted fashion, they never collide when $\mathcal{M}^j(t)$ are the same for all j. Having different $\mathcal{M}^j(t)$ happens in expectation a constant (in T) amount of times, so that the contribution of collisions to the regret is negligible.

6.2.2 Theoretical results

This section provides theoretical guarantees of `Selfish-Robust MMAB`. Theorem 6.1 first presents guarantees in terms of regret. Its proof is given in Section 6.A.2.

Theorem 6.1. *The collective regret of `Selfish-Robust MMAB` is bounded as*

$$R(T) \leq M \sum_{k>M} \frac{\mu_{(M)} - \mu_{(k)}}{\mathrm{kl}(\mu_{(k)}, \mu_{(M)})} \log(T) + \mathcal{O}\left(\frac{MK^3}{\mu_{(K)}} \log(T)\right).$$

It can also be noted from Lemma 6.3 in Section 6.A.2 that the regret due to `Alternate Exploration` is $M \sum_{k>M} \frac{\mu_{(M)}-\mu_{(k)}}{\mathrm{kl}(\mu_{(k)},\mu_{(M)})} \log(T) + o(\log(T))$, which is known to be optimal for algorithms using no collision information (Besson and Kaufmann, 2019). `Alternate Exploration` thus gives an optimal algorithm under this constraint, if M is already known and ranks already attributed (as the $\mathcal{O}(\cdot)$ term in the regret is the consequence of their estimation).

On top of good regret guarantees, `Selfish-Robust MMAB` is robust to selfish behaviors as highlighted by Theorem 6.2 (whose proof is deterred to Section 6.A.2).

Theorem 6.2. *There exist α and ε satisfying*

$$\varepsilon = \textstyle\sum_{k>M} \frac{\mu_{(M)}-\mu_{(k)}}{\mathrm{kl}(\mu_{(k)},\mu_{(M)})} \log(T) + \mathcal{O}\left(\frac{\mu_{(1)}}{\mu_{(K)}} K^3 \log(T)\right), \qquad \alpha = \frac{\mu_{(M)}}{\mu_{(1)}}$$

such that playing `Selfish-Robust MMAB` is an ε-Nash equilibrium and is (α, ε)-stable

These points are proved for an *omniscient* selfish player (knowing all the parameters beforehand). This is a very strong assumption and a real player would not be able to win as much by deviating from the collective strategy. Intuitively, a selfish player would need to explore suboptimal arms as given by the known individual lower bounds. However, a selfish player can actually decide to not explore but deduce the exploration of other players from collisions.

6.3 On harder problems

Following the positive results of the previous section (existence of robust algorithms) in the homogeneous case with statistical sensing, we now provide in this section impossibility results

for both no sensing and heterogeneous cases. By showing its limitations, it also suggests a proper way to consider the heterogeneous problem in the presence of selfish players.

6.3.1 Hardness of no sensing setting

Theorem 6.3. *In the no sensing setting, there is no individual strategy s such that for all problem parameters $(M, \boldsymbol{\mu})$, if all players follow the strategy s, $R(T) = o(T)$ and $(s, s, \dots, s)$ is an $\varepsilon(T)$-Nash equilibrium with $\varepsilon(T) = o(T)$.*

Proof. Consider a strategy $\boldsymbol{s}$ verifying the first property and a problem instance $(M, \boldsymbol{\mu})$ where the selfish player only pulls the best arm. Let $\boldsymbol{\mu'}$ be the mean vector $\boldsymbol{\mu}$ where $\mu_{(1)}$ is replaced by 0. Then, because of the considered observation model, the cooperative players can not distinguish the two worlds $(M, \boldsymbol{\mu})$ and $(M-1, \boldsymbol{\mu'})$. Having a sublinear regret in the second world implies $o(T)$ pulls on the arm 1 for the cooperative players. So in the first world, the selfish player will have a reward in $\mu_{(1)}T - o(T)$, which is thus a linear improvement in comparison with following $\boldsymbol{s}$ if $\mu_{(1)} > \mu_{(2)}$. □

Theorem 6.3 is proved for a selfish players who knows the means $\boldsymbol{\mu}$ beforehand, as the notion of Nash equilibrium prevents against all possible strategies, which includes committing to an arm for the whole game. The knowledge of $\boldsymbol{\mu}$ is actually not needed, as a similar result holds for a selfish player committing to an arm chosen at random when the best arm is K times better than the second one. The question of existence of robust algorithms remains yet open if we restrict selfish strategies to more *reasonable* algorithms.

6.3.2 Heterogeneous model

We consider the full sensing heterogeneous model described in Section 3.3.1 in this section.

A first impossibility result

Theorem 6.4. *If the regret is compared with the optimal assignment, there is no strategy $\boldsymbol{s}$ such that, for all problem parameters $\boldsymbol{\mu}$, $R(T) = o(T)$ and s is an $\varepsilon(T)$-Nash equilibrium with $\varepsilon(T) = o(T)$.*

Proof. Assume $\boldsymbol{s}$ satisfies these properties and consider a problem instance $\boldsymbol{\mu}$ such that the selfish player unique best arm j_1 has mean $\mu^j_{(1)} = 1/2$ and the difference between the optimal assignment utility and the utility of the best one assigning arm j_1 to j is $1/3$.

Such an instance is of course possible. Consider a selfish player j playing exactly the strategy s^j but as if her reward vector $\boldsymbol{\mu^j}$ was actually $\boldsymbol{\mu'^j}$ where $\mu^j_{(1)}$ is replaced by 1 and all other μ^j_k by

0, i.e., she fakes a second world $\boldsymbol{\mu'}$ in which the optimal assignment gives her the arm j_1. In this case, the sublinear regret assumption of $\boldsymbol{s}$ implies that player j pulls j_1 a time $T - o(T)$, while in the true world, she would have pulled it $o(T)$ times. She thus earns an improvement at least $(\mu^j_{(1)} - \mu^j_{(2)})T - o(T)$ w.r.t. playing s^j, contradicting the Nash equilibrium assumption. □

Random assignments

We now take a step back and describe "relevant" allocation procedures for the heterogeneous case, when the vector of means $\boldsymbol{\mu^j}$ is already known by player j.

An assignment is *symmetric* if, when $\boldsymbol{\mu^j} = \boldsymbol{\mu^i}$, players i and j get the same **expected** utility, i.e., no player is *a priori* favored[1]. It is *strategyproof* if being truthful is a dominant strategy for each player and *Pareto optimal* if no player can improve her own reward without decreasing the reward of any other player. Theorem 6.4 is a consequence of Theorem 6.5 below.

Theorem 6.5 (Zhou 1990)**.** *For $M \geq 3$, there is no symmetric, Pareto optimal and strategyproof random assignment algorithm.*

Liu et al. (2020b) circumvent this assignment problem with player-preferences for arms. Instead of assigning a player to a contested arm, the latter decides who gets to pull it, following its preferences.

In the case of random assignment, Abdulkadiroğlu and Sönmez (1998) proposed the Random Serial Dictatorship (RSD) algorithm, which is symmetric and strategyproof. The algorithm is rather simple: pick uniformly at random an ordering of the M players. Following this order, the first player picks her preferred arm, the second one her preferred remaining arm and so on. Svensson (1999) justified the choice of RSD for symmetric strategyproof assignment algorithms. Adamczyk et al. (2014) recently studied efficiency ratios of such assignments: if $U_{\max}$ denotes the expected social welfare of the optimal assignment, the expected social welfare of RSD is greater than $U^2_{\max}/eM$ while no strategyproof algorithm can guarantee more than $U^2_{\max}/M$. As a consequence, RSD is optimal up to a (multiplicative) constant and will serve as a benchmark in the remaining.

Instead of defining the regret in comparison with the optimal assignment as done in the classical heterogeneous multiplayer bandits, we are indeed going to define it with respect to RSD to incorporate strategy-proofness constraints. Formally, the RSD-regret is defined as:

$$R^{\text{RSD}}(T) \coloneqq T\mathbb{E}_{\sigma\sim\mathcal{U}(\mathfrak{S}_M)}\left[\sum_{k=1}^{M}\mu^{\sigma(k)}_{\pi_\sigma(k)}\right] - \sum_{t=1}^{T}\sum_{j=1}^{M}\mathbb{E}[r^j_{\pi^j(t)}(t)],$$

[1]The concept of fairness introduced above is stronger, as no player should be *a posteriori* favored.

with $\mathfrak{S}_M$ the set of permutations over $[M]$ and $\pi_\sigma(k)$ the arm attributed by RSD to player $\sigma(k)$ when the order of dictators is $(\sigma(1), \dots, \sigma(M))$. Mathematically, π_σ is defined by:

$$\pi_\sigma(1) = \underset{l\in[M]}{\arg\max}\, \mu_l^{\sigma(1)} \quad \text{and} \quad \pi_\sigma(k+1) = \underset{\substack{l\in[M] \\ l\notin\{\pi_\sigma(l') \mid l'\leq k\}}}{\arg\max}\, \mu_l^{\sigma(k+1)}.$$

6.4 Full sensing setting

This section focuses on the full sensing setting, where both $\eta_k(t)$ and $X_k(t)$ are always observed as we proved impossibility results for more complex settings. As seen in the previous chapters, near optimal algorithms leverage the observation of collisions to enable some communication between players by forcing them. Some of these communication protocols can be modified to allow robust communication. This section is structured as follows. First, insights on two new protocols are given for robust communications. Second, a robust adaptation of SIC-MMAB is given, based on these two protocols. Third, they can also be used to reach a logarithmic RSD-regret in the heterogeneous case.

6.4.1 Making communication robust

To have robust communication, two new complementary protocols are needed. The first one allows to send messages between players and to detect when they have been corrupted by a malicious player. If this has been the case, the players then use the second protocol to proceed to a collective punishment, which forces every player to suffer a considerable loss for the remaining of the game. Such punitive strategies are called "Grim Trigger" in game theory and are used to deter defection in repeated games (Friedman, 1971; Axelrod and Hamilton, 1981; Fudenberg and Maskin, 2009).

Back and forth messaging

Communication protocols in the collision sensing setting usually rely on the fact that collision indicators can be seen as bits sent from a player to another one as follows. If player i sends a binary message $m_{i\to j} = (1, 0, \dots, 0, 1)$ to player j during a predefined time window, she proceeds to the sequence of pulls $(j, i, \dots, i, j)$, meaning she purposely collides with j to send a 1 bit (reciprocally, not colliding corresponds to a 0 bit). A malicious player trying to corrupt a message can only create new collisions, i.e., replace zeros by ones. The key point is that the inverse operation is not possible.

If player j receives the (potentially corrupted) message $\hat{m}_{i\to j}$, she repeats it to player i. This second message can also be corrupted by the malicious player and player i receives $\widetilde{m}_{i\to j}$. However, since the only possible operation is to replace zeros by ones, there is no way to transform back $\hat{m}_{i\to j}$ to $m_{i\to j}$ if the first message had been corrupted. The player i then just has to compare $\widetilde{m}_{i\to j}$ with $m_{i\to j}$ to know whether or not at least one of the two messages has been corrupted. We call this protocol *back and forth* communication.

In the following, other malicious communications are possible. Besides sending false information (which is managed differently), a malicious player can send different statistics to the others, while they need to have the exact same statistics. To overcome this issue, players will send to each other statistics sent to them by every player. If two players have received different statistics by the same player, at least one of them automatically realizes it.

Collective punishment

The back and forth protocol detects if a malicious player interfered in a communication and, in that case, a collective punishment is triggered (to deter defection). The malicious player is yet unidentified and can not be specifically targeted. The punishment thus guarantees that the average reward earned by each player is smaller than the average reward of the algorithm, $\overline{\mu}_M \coloneqq \frac{1}{M}\sum_{k=1}^{M}\mu_{(k)}$.

A naive way to *punish* is to pull all arms uniformly at random. The selfish player then gets the reward $(1-1/K)^{M-1}\mu_{(1)}$ by pulling the best arm, which can be larger than $\overline{\mu}_M$. A good punishment should therefore pull arms more often the better they are.

During the punishment, players pull each arm k with probability $1-\big(\gamma\frac{\sum_{l=1}^{M}\hat{\mu}^j_{(l)}(t)}{M\hat{\mu}^j_k(t)}\big)^{\frac{1}{M-1}}$ at least, where $\gamma=(1-1/K)^{M-1}$. Such a strategy is possible as shown by Lemma 6.13 in Section 6.B. Assuming the arms are correctly estimated, i.e., the expected reward a selfish player gets by pulling k is approximately $\mu_k(1-p_k)^{M-1}$, with $p_k=\max\Big(1-\big(\gamma\frac{\overline{\mu}_M}{\mu_k}\big)^{\frac{1}{M-1}},0\Big)$.

If $p_k=0$, then μ_k is smaller than $\gamma\overline{\mu}_M$ by definition; otherwise, it necessarily holds that $\mu_k(1-p_k)^{M-1}=\gamma\overline{\mu}_M$. As a consequence, in both cases, the selfish player earns at most $\gamma\overline{\mu}_M$, which involves a relative positive decrease of $1-\gamma$ in reward w.r.t. following the cooperative strategy. More details on this protocol are given by Lemma 6.21 in Section 6.C.3.

6.4.2 Homogeneous case: `SIC-GT`

In the homogeneous case, these two protocols can be incorporated in the SIC-MMAB algorithm of Chapter 4 to provide `SIC-GT`, which is robust to selfish behaviors and still ensures a regret comparable to the centralized lower bound.

The communication protocol of SIC-MMAB was improved by choosing a leader and communicating all the information only to this leader. A malicious player would do anything to be the leader. `SIC-GT` avoids such a behavior by choosing two leaders who either agree or trigger the punishment. More generally with $n+1$ leaders, this protocol is robust to n selfish players. The detailed algorithm is given by Algorithm 6.3 in Section 6.C.1.

Initialization. The original initialization phase of SIC-MMAB has a small regret term, but it is not robust. During the initialization, the players here pull uniformly at random to estimate M as in `Selfish-Robust MMAB` and then attribute ranks the same way. The players with ranks 1 and 2 are then leaders. Since the collision indicator is always observed here, this estimation can be done in an easier and better way. The observation of η_k also enables players to remain synchronized after this phase as its length does not depend on unknown parameters and is deterministic.

Exploration and Communication. Players alternate between exploration and communication once the initialization is over. During the p-th exploration phase, each arm still requiring exploration is pulled 2^p times by every player in a collisionless fashion. Players then communicate to each leader their empirical means in binary after every exploration phase, using the back and forth trick explained in Section 6.4.1. Leaders then check that their information match. If some undesired behavior is detected, a collective punishment is triggered.

Otherwise, the leaders determine the sets of optimal/suboptimal arms and send them to everyone. To prevent the selfish player from sending fake statistics, the leaders gather the empirical means of all players, except the extreme ones (largest and smallest) for every arm. If the selfish player sent outliers, they are thus cut out from the collective estimator, which is thus the average of $M-2$ individual estimates. This estimator can be biased by the selfish player, but a concentration bound given by Lemma 6.17 in Section 6.C.2 still holds.

Exploitation. As soon as an arm is detected as optimal, it is pulled until the end. To ensure fairness of `SIC-GT`, players will actually rotate over all the optimal arms so that none of them is favored. This point is thoroughly described in Section 6.C.1. Theorem 6.6, proved in Section 6.C, gives theoretical results for `SIC-GT`.

Theorem 6.6. *Define* $\alpha = \frac{1-(1-1/K)^{M-1}}{2}$ *and assume* $M \geq 3$.

1. *The collective regret of* `SIC-GT` *is bounded as*

$$R(T) = \mathcal{O}\Big(\sum_{k>M} \frac{\log(T)}{\mu_{(M)} - \mu_{(k)}} + MK^2 \log(T) + M^2 K \log^2\Big(\frac{\log(T)}{(\mu_{(M)} - \mu_{(M+1)})^2}\Big)\Big).$$

2. *There exists ε satisfying*

$$\varepsilon = \mathcal{O}\Big(\sum_{k>M} \frac{\log(T)}{\mu_{(M)} - \mu_k} + K^2 \log(T) + MK \log^2 \Big(\frac{\log(T)}{(\mu_{(M)} - \mu_{(M+1)})^2} \Big) + \frac{K \log(T)}{\alpha^2 \mu_{(K)}} \Big)$$

such that playing `SIC-GT` *is an ε-Nash equilibrium and is (α, ε)-stable.*

6.4.3 Semi-heterogeneous case: `RSD-GT`

The punishment strategies described above can not be extended to the heterogeneous case, as the relevant probability of choosing each arm would depend on the preferences of the malicious player which are unknown (even her identity might not be discovered). Moreover, as already explained in the homogeneous case, pulling each arm uniformly at random is not an appropriate punishment strategy[2]. We therefore consider the δ-heterogeneous setting, which allows punishments for small values of δ as given by Lemma 6.24 in Section 6.D.3. The heterogeneous model was justified by the fact that transmission quality depends on individual factors such as localization. The δ-heterogeneous assumption relies on the idea that such individual factors are of a different order of magnitude than global factors (as the availability of a channel). As a consequence, even if arm means differ from player to player, these variations remain relatively small.

Definition 6.3. *The setting is δ-heterogeneous if there exists $\{\mu_k; k \in [K]\}$ such that for all j and k, $\mu_k^j \in [(1-\delta)\mu_k, (1+\delta)\mu_k]$.*

In the semi-heterogeneous full sensing setting, `RSD-GT` provides a robust, logarithmic RSD-regret algorithm. Its complete description is given by Algorithm 6.4 in Section 6.D.1.

Algorithm description

`RSD-GT` starts with the exact same initialization as `SIC-GT` to estimate M and attribute ranks among the players. The time is then divided into superblocks which are divided into M blocks. During the j-th block of a superblock, the dictators ordering[3] is $(j, \dots, M, 1, \dots, j-1)$. Moreover, only the j-th player can send messages during this block.

Exploration. The exploring players pull sequentially all the arms. Once player j knows her M best arms and their ordering, she waits for a block j to initiate communication.

[2]Unless in the specific case where $\mu_{(1)}^j (1-1/K)^{M-1} < \frac{1}{M} \sum_{k=1}^M \mu_{(k)}^j$.

[3]The ordering is actually $(\sigma(j), \dots, \sigma(j-1))$ where $\sigma(j)$ is the player with rank j after the initialization. For sake of clarity, this consideration is omitted here.

Communication. Once a player starts a communication block, she proceeds in three successive steps as follows:

1. she first collides with all players to signal the beginning of a communication block. The other players then enter a listening state, ready to receive messages.
2. She then sends to every player her ordered list of M best arms. Each player then repeats this list to detect the potential intervention of a malicious player.
3. Finally, players who detected the intervention of a malicious player signal to everyone the beginning of a collective punishment.

After a communication block j, every one knows the preferences order of player j, who is now in her exploitation phase, unless a punishment protocol has been started.

Exploitation. While exploiting, player j knows the preferences of all other exploiting players. Thanks to this, she can easily compute the arms attributed by the RSD algorithm between the exploiting players, given the dictators ordering of the block.

Moreover, as soon as she collides in the beginning of a block while not intended (by her), this means an exploring player is starting a communication block. The exploiting player then starts listening to the arm preferences of the communicating player.

Theoretical guarantees

Here are some insights to understand how `RSD-GT` reaches the utility of the RSD algorithm, which are rigorously detailed by Lemma 6.25 in Section 6.D.3. With no malicious player, the players ranks given by the initialization provide a random permutation $\sigma \in \mathfrak{S}_M$ of the players and always considering the dictators ordering $(1, \dots, M)$ would lead to the expected reward of the RSD algorithm. However, a malicious player can easily rig the initialization to end with rank 1. In that case, she largely improves her individual reward w.r.t. following the cooperative strategy.

To avoid such a behavior, the dictators ordering should rotate over all permutations of $\mathfrak{S}_M$, so that the rank of the player has no influence. However, this leads to an undesirable combinatorial $M!$ dependency of the regret. `RSD-GT` instead rotates over the dictators ordering $(j, \dots, M, 1, \dots, j-1)$ for all $j \in [M]$. If we note σ_0 the M-cycle $(1 \dots M)$, the considered permutations during a superblock are of the form $\sigma \circ \sigma_0^{-m}$ for $m \in [M]$. The malicious player j can only influence the distribution of $\sigma^{-1}(j)$: assume w.l.o.g. that $\sigma(1) = j$. The permutation σ given by the initialization then follows the uniform distribution over $\mathfrak{S}_M^{j \to 1} = \{\sigma \in$

$\mathfrak{S}_M \mid \sigma(1) = j\}$. But then, for $m \in [M]$, $\sigma \circ \sigma_0^{-m}$ has a uniform distribution over $\mathfrak{S}_M^{j \to 1+m}$. In average over a superblock, the induced permutation still has a uniform distribution over $\mathfrak{S}_M$. So the malicious player has no interest in choosing a particular rank during the initialization, making the algorithm robust.

Thanks to this remark and robust communication protocols, `RSD-GT` possesses theoretical guarantees given by Theorem 6.7 (whose proof is deterred to Section 6.D).

Theorem 6.7. *Consider the δ-heterogeneous setting and define $r = \frac{1-\left(\frac{1+\delta}{1-\delta}\right)^2(1-1/K)^{M-1}}{2}$ and $\Delta = \min_{(j,k)\in[M]^2} \mu^j_{(k)} - \mu^j_{(k+1)}$.*

1. *The RSD-regret of `RSD-GT` is bounded as: $R^{\mathrm{RSD}}(T) = \mathcal{O}\big(MK\Delta^{-2}\log(T) + MK^2\log(T)\big)$.*
2. *If $r > 0$, there exist ε and α satisfying*
   - $\varepsilon = \mathcal{O}\Big(\frac{K\log(T)}{\Delta^2} + K^2\log(T) + \frac{K\log(T)}{(1-\delta)r^2\mu_{(K)}}\Big)$,
   - $\alpha = \min\Big(r\left(\frac{1+\delta}{1-\delta}\right)^3 \frac{\sqrt{\log(T)}-4M}{\sqrt{\log(T)}+4M}, \quad \frac{\Delta}{(1+\delta)\mu_{(1)}}, \quad \frac{(1-\delta)\mu_{(M)}}{(1+\delta)\mu_{(1)}}\Big)$

   such that playing `RSD-GT` is an ε-Nash equilibrium and is (α, ε)-stable.

Appendix

6.A Missing elements for `Selfish-Robust MMAB`

This section provides a complete description of `Selfish-Robust MMAB` and the proofs of Theorems 6.1 and 6.2.

6.A.1 Thorough description of `Selfish-Robust MMAB`

In addition to Section 6.2, the pseudocodes of `EstimateM`, `GetRank` and `Alternate Exploration` are given here. The following Protocol 6.1 describes the estimation of M using the notations introduced in Section 5.

Protocol 6.1: `EstimateM`

Input: β, T

$t_m \leftarrow 0$

while $\min_k \#T_k^j(t) < \beta^2 K^2 \log(T)$ **do**

  Pull $k \sim \mathcal{U}(K)$; Update $\#T_k^j(t)$ and $\#C_k^j(t)$; $t_m \leftarrow t_m + 1$

end

$\hat{M} \leftarrow 1 + \mathrm{round}\Big(\frac{\log\left(1-\frac{1}{K}\sum_k \hat{p}_k^j(t_M)\right)}{\log\left(1-\frac{1}{K}\right)}\Big)$ `// round(x) = closest integer to x`

return $\hat{M}, t_m$

Since the duration t_m^j of `EstimateM` for player j is random and differs between players, each player continues sampling uniformly at random until $\frac{\gamma_2}{\gamma_1} t_m^j$, with $\gamma_1 = \frac{13}{14}$ and $\gamma_2 = \frac{16}{15}$. Thanks to this additional *waiting room*, Lemma 6.1 below guarantees that all players are sampling uniformly at random until at least t_m^j for each j.

The estimation of M here tightly estimates the probability to collide individually for each arm. This restriction provides an additional M factor in the length of this phase in comparison with (Rosenski et al., 2016), where the probability to collide is globally estimated. This is however required because of the Statistic Sensing, but if η_k was always observed, then the protocol from Rosenski et al. (2016) would be robust.

Indeed, if we directly estimated the global probability to collide, the selfish player could pull only the best arm. The number of observations of η_k is larger on this arm, and the estimated probability to collide would thus be positively biased because of the selfish player.

Afterwards, ranks in $[M]$ are attributed to players by sampling uniformly at random in $[M]$ until observing no collision, as described in Protocol 6.2. For the same reason, a waiting room is added to guarantee that all players end this protocol with different ranks.

Protocol 6.2: `GetRank`

Input: $\hat{M}, t_m^j, \beta, T$
$n \leftarrow \beta^2 K^2 \log(T)$ and $j \leftarrow -1$
for $t_m^j \log(T)/(\gamma_1 n)$ *rounds* **do**
  if $j = -1$ **then**
    Pull $k \sim \mathcal{U}(\hat{M})$; **if** $r_k(t) > 0$ **then** $j \leftarrow k$ `// no collision`
  else Pull j
end
return j

The following quantities are used to describe `Alternate Exploration` in Algorithm 6.2:

- $\mathcal{M}^j(t) = \left(l_1^j(t), \ldots, l_M^j(t)\right)$ is the list of the empirical M best arms for player j at round t. It is updated only each M rounds and ordered according to the index of the arms, i.e., $l_1^j(t) < \ldots < l_M^j(t)$.
- $\hat{m}^j(t)$ is the empirical M-th best arm for player j at round t.
- $b_k^j(t) = \sup\{q \geq 0 \mid N_k^j(t)\mathrm{kl}(\hat{\mu}_k^j(t), q) \leq f(t)\}$ is the kl-UCB index of the arm k for player j at round t, where $f(t) = \log(t) + 4\log(\log(t))$, $N_k^j(t)$ is the number of times player j pulled k and $\hat{\mu}_k^j$ is the empirical mean.

6.A.2 Proofs of Section 6.2

Let us define $\alpha_k \coloneqq \mathbb{P}(X_k(t) > 0) \geq \mu_k$, $\gamma_1 = \frac{13}{14}$ and $\gamma_2 = \frac{16}{15}$.

Regret analysis

This section aims at proving Theorem 6.1. This proof is divided in several auxiliary lemmas given below. First, the regret can be decomposed as follows:

$$R(T) = \mathbb{E}[R^{\text{init}} + R^{\text{explo}}], \tag{6.1}$$

Algorithm 6.2: `Alternate Exploration`

Input: M, j

if $t = 0 \pmod M$ **then** Update $\hat{\mu}^j(t), b^j(t), \hat{m}^j(t)$ and $\mathcal{M}^j(t) = (l_1, \dots, l_M)$

$\pi \leftarrow t + j \pmod M + 1$

if $l_\pi \neq \hat{m}^j(t)$ **then** Pull l_π `// exploit the` $M-1$ `best empirical arms`

else

  $\mathcal{B}^j(t) = \{k \notin \mathcal{M}^j(t) \mid b^j_k(t) \geq \hat{\mu}^j_{\hat{m}^j(t)}(t)\}$ `// arms to explore`

  if $\mathcal{B}^j(t) = \emptyset$ **then** Pull l_π

  else Pull $\begin{cases} l_\pi \text{ with proba } 1/2 \\ k \text{ chosen uniformly at random in } \mathcal{B}^j(t) \text{ otherwise} \end{cases}$ `// explore`

end

where

$$R^{\text{init}} = T_0 \sum_{k=1}^{M} \mu_{(k)} - \sum_{t=1}^{T_0} \sum_{j=1}^{M} \mu_{\pi^j(t)}(1 - \eta^j(t)) \text{ with } T_0 = \left(\frac{\gamma_2}{\gamma_1^2 \beta^2 K^2} + \frac{\gamma_2^2}{\gamma_1^2} \right) \max_j t^j_m,$$

$$R^{\text{explo}} = (T - T_0) \sum_{k=1}^{M} \mu_{(k)} - \sum_{t=T_0+1}^{T} \sum_{j=1}^{M} \mu_{\pi^j(t)}(1 - \eta^j(t)).$$

Lemma 6.1 first gives guarantees on the `EstimateM` protocol. Its proof is given in Section 6.A.2.

Lemma 6.1. *If $M-1$ players run `EstimateM` with $\beta \geq 39$, followed by a waiting room until $\frac{\gamma_2}{\gamma_1} t^j_m$, then regardless of the strategy of the remaining player, with probability larger than $1 - \frac{6KM}{T}$, for any player:*

$$\hat{M}^j = M \text{ and } \frac{t^j_m \alpha_{(K)}}{K} \in [\gamma_1 n, \gamma_2 n],$$

where $n = \beta^2 K^2 \log(T)$.

When $\hat{M}^j = M$ and $\frac{t^j_m \alpha_{(K)}}{K} \in [\gamma_1 n, \gamma_2 n]$ for all cooperative players j, we say that the estimation phase is **successful**.

Lemma 6.2. *Conditioned on the success of the estimation phase, with probability $1 - \frac{M}{T}$, all the cooperative players end `GetRank` with different ranks $j \in [M]$, regardless of the behavior of other players.*

The proof of Lemma 6.2 is given in Section 6.A.2. If the estimation is successful and all players end `GetRank` with different ranks $j \in [M]$, the initialization is said successful.

Using the same arguments as Proutiere and Wang (2019), the collective regret of the `Alternate Exploration` phase can be shown to be $M\sum_{k>M}\frac{\mu_{(M)}-\mu_{(k)}}{\mathrm{kl}(\mu_{(M)},\mu_{(k)})}\log(T)+o\left(\log(T)\right)$. This result is given by Lemma 6.3, whose proof is given in Section 6.A.2.

Lemma 6.3. *If all players follow* `Selfish-Robust MMAB`*:*

$$\mathbb{E}[R^{explo}]\leq M\sum_{k>M}\frac{\mu_{(M)}-\mu_{(k)}}{\mathrm{kl}(\mu_{(M)},\mu_{(k)})}\log(T)+o\left(\log(T)\right).$$

Proof of Theorem 6.1. Thanks to Lemma 6.3, the total regret is bounded by

$$M\sum_{k>M}\frac{\mu_{(M)}-\mu_{(k)}}{\mathrm{kl}(\mu_{(M)},\mu_{(k)})}\log(T)+\mathbb{E}[T_0]M+o\left(\log(T)\right).$$

Thanks to Lemmas 6.1 and 6.2, $\mathbb{E}[T_0]=\mathcal{O}\left(\frac{K^3\log(T)}{\mu_{(K)}}\right)$, yielding Theorem 6.1. □

Proof of Lemma 6.1

Proof. Let j be a cooperative player and $q_k(t)$ be the probability at round t that the remaining player pulls k. Define $p_k^j(t)=\mathbb{P}[t\in C_k^j(t)\mid t\in T_k^j(t)]$. By definition, $p_k^j(t)=1-(1-1/K)^{M-2}(1-q_k(t))$ when all cooperative players are pulling uniformly at random. Two auxiliary Lemmas using classical concentration inequalities are used to prove Lemma 6.1. The proofs of Lemmas 6.4 and 6.5 are given in Section 6.A.2.

Lemma 6.4. *For any* $\delta>0$*,*

1. $\mathbb{P}\left[\left|\frac{\#C_k^j(T_M)}{\#T_k^j(T_M)}-\frac{1}{\#T_k^j(T_M)}\sum_{t\in T_k^j(T_M)}p_k^j(t)\right|\geq\delta\;\middle|\;T_k^j(T_M)\right]\leq 2\exp(-\frac{\#T_k^j(T_M)\delta^2}{2})$.

For any $\delta\in(0,1)$ *and fixed* T_M*,*

2. $\mathbb{P}\left[\left|\#T_k^j-\frac{\alpha_k T_M}{K}\right|\geq\delta\frac{\alpha_k T_M}{K}\right]\leq 2\exp(-\frac{T_M\alpha_k\delta^2}{3K})$.

3. $\mathbb{P}\left[\left|\sum_{t=1}^{T_M}(\mathbb{1}\left(t\in T_k^j\right)-\frac{\alpha_k}{K})p_k^j(t)\right|\geq\delta\frac{\alpha_k T_M}{K}\right]\leq 2\exp\left(-\frac{T_M\alpha_k\delta^2}{3K}\right)$.

Lemma 6.5. *For all* k*,* j *and* $\delta\in(0,\frac{\alpha_k}{K})$*, with probability larger than* $1-\frac{6KM}{T}$*,*

$$\left|\hat{p}_k^j(t_m^j)-\frac{1}{t_m^j}\sum_{t=1}^{t_m^j}p_k^j(t)\right|\leq 2\sqrt{\frac{6\log(T)}{n\left(1-2\sqrt{\frac{3}{2\beta^2}(1+\frac{3}{2\beta^2})}\right)}}+2\sqrt{\frac{\log(T)}{n}}.$$

And for $\beta\geq 39$*:*

$$\frac{t_m^j\alpha_{(k)}}{K}\in\left[\frac{13}{14}n,\;\frac{16}{15}n\right].$$

Let $\varepsilon = 2\sqrt{\frac{6\log(T)}{n\left(1-2\sqrt{\frac{3}{2\beta^2}(1+\frac{3}{2\beta^2})}\right)}} + 2\sqrt{\frac{\log(T)}{n}}$ and $p_k^j = \frac{1}{t_m^j}\sum_{t=1}^{t_m^j} p_k^j(t)$ such that with probability at least $1-\frac{6KM}{T}$, $|\hat{p}_k^j - p_k^j| \leq \varepsilon$. The remaining of the proof is conditioned on this event.

By definition of n, $\varepsilon = \frac{1}{K}f(\beta)$ where $f(x) = \frac{2}{x}\sqrt{\frac{6}{1-2\sqrt{\frac{3}{2x^2}(1+\frac{3}{2x^2})}}} + 2/x$. Note that $f(x) \leq \frac{1}{2e}$ for $x \geq 39$ and thus $\varepsilon \leq \frac{1}{2Ke}$ for the considered β.

The last point of Lemma 6.5 yields that $t_m^j \leq \frac{\gamma_2}{\gamma_1} t_m^{j'}$ for any pair j, j'. All the cooperative players are thus pulling uniformly at random until at least t_m^j, thanks to the additional waiting room. Then,

$$\frac{1}{K}\sum_k (1-p_k^j(t)) \;=\; (1-1/K)^{M-2}\left(1-\frac{1}{K}\sum_k q_k(t)\right) \;=\; (1-1/K)^{M-1}.$$

When summing over k, it follows:

$$\begin{aligned}
\frac{1}{K}\sum_k (1-p_k^j) - \varepsilon \leq \frac{1}{K}\sum_k (1-\hat{p}_k^j) &\leq \frac{1}{K}\sum_k (1-p_k^j) + \varepsilon \\
(1-1/K)^{M-1} - \varepsilon \leq \frac{1}{K}\sum_k (1-\hat{p}_k^j) &\leq (1-1/K)^{M-1} + \varepsilon \\
M-1+\frac{\log(1+\frac{\varepsilon}{(1-1/K)^{M-1}})}{\log(1-1/K)} \leq \frac{\log\left(\frac{1}{K}\sum_k (1-\hat{p}_k^j)\right)}{\log(1-1/K)} &\leq M-1+\frac{\log(1-\frac{\varepsilon}{(1-1/K)^{M-1}})}{\log(1-1/K)} \\
M-1+\frac{\log(1+\frac{1}{2K})}{\log(1-1/K)} \leq \frac{\log\left(\frac{1}{K}\sum_k (1-\hat{p}_k^j)\right)}{\log(1-1/K)} &\leq M-1+\frac{\log(1-\frac{1}{2K})}{\log(1-1/K)}
\end{aligned}$$

The last line is obtained by observing that $\frac{\varepsilon}{(1-1/K)^{M-1}}$ is smaller than $\frac{1}{2K}$.
Observing that $\max\left(\frac{\log(1-x/2)}{\log(1-x)}, -\frac{\log(1+x/2)}{\log(1-x)}\right) < 1/2$ for any $x > 0$, the last line implies:

$$1+\frac{\log\left(\frac{1}{K}\sum_k (1-\hat{p}_k^j)\right)}{\log(1-1/K)} \in (M-1/2, M+1/2).$$

When rounding this quantity to the closest integer, we thus obtain M, which yields the first part of Lemma 6.1. The second part is directly given by Lemma 6.5. □

Proof of Lemma 6.2

The proof of Lemma 6.2 relies on two lemmas given below.

Lemma 6.6. *Conditionally on the success of the estimation phase, when a cooperative player j*

proceeds to `GetRank`*, all other cooperative players are either running* `GetRank` *or in a waiting room*[4]*, i.e., they are not proceeding to* `Alternate Exploration` *yet.*

Proof. Recall that $\gamma_1 = 13/14$ and $\gamma_2 = 16/15$. Conditionally on the success of the estimation phase, for any pair (j, j'), $\frac{\gamma_2}{\gamma_1} t_m^j \geq t_m^{j'}$. Let $t_r^j = \frac{t_m^j}{\gamma_1 K^2 \beta^2}$ be the duration time of `GetRank` for player j. For the same reason, $\frac{\gamma_2}{\gamma_1} t_r^j \geq t_r^{j'}$. Player j ends `GetRank` at round $t^j = \frac{\gamma_2}{\gamma_1} t_m^j + t_r^j$ and the second waiting room at round $\frac{\gamma_2}{\gamma_1} t^j$.

As $\frac{\gamma_2}{\gamma_1} t^j \geq t^{j'}$, this yields that when a player ends `GetRank`, all other players are not running `Selfish-Robust MMAB` yet. Because $\frac{\gamma_2}{\gamma_1} t_m^j \geq t_m^{j'}$, when a player starts `GetRank`, all other players also have already ended `EstimateM`. This yields Lemma 6.6. □

Lemma 6.7. *Conditionally on the success of the estimation phase, with probability larger than* $1 - \frac{1}{T}$*, cooperative player* j *ends* `GetRank` *with a rank in* $[M]$.

Proof. Conditionally on the success of the estimation phase and thanks to Lemma 6.5, $t_r^j = \frac{t_m^j}{\gamma_1 K^2 \beta^2} \geq \frac{K \log(T)}{\alpha_{(K)}}$. Moreover, at any round of `GetRank`, the probability of observing $\eta_k(t) = 0$ is larger than $\frac{\alpha_{(K)}}{M}$. Indeed, the probability of observing $\eta_k(t)$ is larger than $\alpha_{(K)}$ with Statistic sensing. Independently, the probability of having $\eta_k = 0$ is larger than $1/M$ since there is at least an arm among $[M]$ not pulled by any other player. These two points yield, as $M \leq K$:

$$\begin{aligned}
\mathbb{P}[\text{player does not observe } \eta_k(t) = 0 \text{ for } t_r^j \text{ successive rounds}] &\leq \left(1 - \frac{\alpha_{(K)}}{M}\right)^{t_r^j} \\
&\leq \exp\left(-\frac{\alpha_{(K)} t_r^j}{M}\right) \\
&\leq \frac{1}{T}
\end{aligned}$$

Thus, with probability larger than $1 - \frac{1}{T}$, player j observes $\eta_k(t) = 0$ at least once during `GetRank`, i.e., she ends the procedure with a rank in $[M]$. □

Proof of Lemma 6.2. Combining Lemmas 6.6 and 6.7 yields that the cooperative player j ends `GetRank` with a rank in $[M]$ and no other cooperative player ends with the same rank. Indeed, when a player gets the rank j, any other cooperative player has either no attributed rank (still running `GetRank` or the first waiting room), or an attributed rank j'. In the latter case, thanks to Lemma 6.6, this other player is either running `GetRank` or in the second waiting room, meaning she is still pulling j'. Since the first player ends with the rank j, this means that she did not encounter a collision when pulling j and especially, $j \neq j'$.

Considering a union bound among all cooperative players now yields Lemma 6.2. □

[4]Note that there is a waiting room before **and** after `GetRank`.

Proof of Lemma 6.3

Let us denote $T_0^j = \left(\frac{\gamma_2}{\gamma_1^2\beta^2K^2} + \frac{\gamma_2^2}{\gamma_1^2}\right) t_m^j$ such that player j starts running `Alternate Exploration` at time T_0^j. This section aims at proving Lemma 6.3. In this section, the initialization is assumed to be successful. The regret due to an unsuccessful initialization is constant in T and thus $o(\log(T))$. We prove in this section, in case of a successful initialization, the following:

$$\mathbb{E}[R^{\text{explo}}] \leq M \sum_{k>M} \frac{\mu_{(M)} - \mu_{(k)}}{\text{kl}(\mu_{(M)}, \mu_{(k)})} \log(T) + o(\log(T)). \tag{6.2}$$

This proof follows the same scheme as the regret proof from Proutiere and Wang (2019), except that there is no leader here. Every *bad event* then happens independently for each individual player. This adds a M factor in the regret compared to the follower/leader algorithm[5] used by Proutiere and Wang (2019). For conciseness, we only give the main steps and refer to the original Lemmas in (Proutiere and Wang, 2019) for their detailed proof.

We first recall useful concentration Lemmas which correspond to Lemmas 1 and 2 in (Proutiere and Wang, 2019). They are respectively simplified versions of Lemma 5 in (Combes et al., 2015) and Theorem 10 in (Garivier and Cappé, 2011).

Lemma 6.8. *Let $k \in [K]$, $c > 0$ and H be a (random) set such that for all t, $\{t \in H\}$ is $\mathcal{F}_{t-1}$ measurable. Assume that there exists a sequence $(Z_t)_{t\geq 0}$ of binary random variables, independent of all $\mathcal{F}_t$, such that for $t \in H$, $\pi^j(t) = k$ if $Z_t = 1$. Furthermore, if $\mathbb{E}[Z_t] \geq c$ for all t, then:*

$$\sum_{t\geq 1} \mathbb{P}[t \in H \mid |\hat{\mu}_k^j(t) - \mu_k| \geq \delta] \leq \frac{4 + 2c/\delta^2}{c^2}.$$

Lemma 6.9. *If player j starts following `Alternate Exploration` at round $T_0^j + 1$:*

$$\sum_{t>T_0^j} \mathbb{P}[b_k^j(t) < \mu_k] \leq 15.$$

Let $0 < \delta < \delta_0 \coloneqq \min_k \frac{\mu_{(k)} - \mu_{(k+1)}}{2}$. Besides the definitions given in Section 6.A.1, define the following:

- $\mathcal{M}^*$ the list of the M-best arms, ordered according to their indices.
- $\mathcal{A}^j = \{t > T_0^j \mid \mathcal{M}^j(t) \neq \mathcal{M}^*\}$.
- $\mathcal{D}^j = \{t > T_0^j \mid \exists k \in \mathcal{M}^j(t),\ |\hat{\mu}_k^j(t) - \mu_k| \geq \delta\}$.
- $\mathcal{E}^j = \{t > T_0^j \mid \exists k \in \mathcal{M}^*,\ b_k^j(t) < \mu_k\}$.

[5] Which is not selfish-robust.

- $\mathcal{G}^j = \{t \in \mathcal{A}^j \setminus \mathcal{D}^j \mid \exists k \in \mathcal{M}^* \setminus \mathcal{M}^j(t),\ |\hat{\mu}_k^j(t) - \mu_k| \geq \delta\}$.

Lemma 6.10. *If player j starts following* `Alternate Exploration` *at round $T_0^j + 1$:*

$$\mathbb{E}[\#(\mathcal{A}^j \cup \mathcal{D}^j)] \leq 8MK^2(6K + \delta^{-2}).$$

Proof. Similarly to Proutiere and Wang (2019), we have $(\mathcal{A}^j \cup \mathcal{D}^j) \subset (\mathcal{D}^j \cup \mathcal{E}^j \cup \mathcal{G}^j)$. We can then individually bound $\mathbb{E}[\#\mathcal{D}^j]$, $\mathbb{E}[\#\mathcal{E}^j]$ and $\mathbb{E}[\#\mathcal{G}^j]$, leading to Lemma 6.10. The detailed proof is omitted here as it exactly corresponds to Lemmas 3 and 4 in (Proutiere and Wang, 2019). □

Lemma 6.11. *Consider a suboptimal arm k and define $\mathcal{H}_k^j = \{t \in \{T_0^j + 1, \dots, T\} \setminus (\mathcal{A}^j \cup \mathcal{D}^j) \mid \pi^j(t) = k\}$. It holds*

$$\mathbb{E}\left[\#\mathcal{H}_k^j\right] \leq \frac{\log T + 4\log(\log T)}{\mathrm{kl}(\mu_k + \delta, \mu_{(M)} - \delta)} + 4 + 2\delta^{-2}.$$

Lemma 6.11 can be proved using the arguments of Lemma 5 in (Proutiere and Wang, 2019).

Proof of Lemma 6.3. If $t \in \mathcal{A}^j \cup \mathcal{D}^j$, player j collides with at most one player j' such that $t \notin \mathcal{A}^{j'} \cup \mathcal{D}^{j'}$.

Otherwise, $t \notin \mathcal{A}^j \cup \mathcal{D}^j$ and player j collides with a player j' only if $t \in \mathcal{A}^{j'} \cup \mathcal{D}^{j'}$. Also, she pulls a suboptimal arm k only on an exploration slot, i.e., instead of pulling the M-th best arm. Thus, the regret caused by pulling a suboptimal arm k when $t \notin \mathcal{A}^j \cup \mathcal{D}^j$ is $(\mu_{(M)} - \mu_k)$ and this actually happens when $t \in \mathcal{H}_k^j$.

This discussion provides the following inequality, which concludes the proof of Lemma 6.3 when using Lemmas 6.10 and 6.11 and taking $\delta \to 0$.

$$\mathbb{E}\left[R^{\text{explo}}\right] \leq \underbrace{2\sum_{j=1}^{M} \mathbb{E}\left[\#(\mathcal{A}^j \cup \mathcal{D}^j)\right]}_{\text{collisions}} + \underbrace{\sum_{j \leq M} \sum_{k > M} (\mu_{(M)} - \mu_{(k)}) \mathbb{E}\left[\#\mathcal{H}_k^j\right]}_{\text{pulls of suboptimal arms}}.$$

□

Proof of Theorem 6.2

Proof. 1. Let us first prove the Nash equilibrium property. Assume that the player j is deviating from `Selfish-Robust MMAB` and define $\mathcal{E} = [T_0] \cup \Big(\bigcup_{m \in [M] \setminus \{j\}} (\mathcal{A}^m \cup \mathcal{D}^m) \Big)$ with the definitions of T_0, $\mathcal{A}^m$ and $\mathcal{D}^m$ given in Section 6.A.2[6]. Thanks to Lemmas 6.1 and 6.2, regardless of

[6]The max of T_0 is here defined over all $m \in [M] \setminus \{j\}$.

the strategy of the selfish player, all other players successfully end the initialization after a time T_0 with probability $1 - \mathcal{O}(KM/T)$. The remaining of the proof is conditioned on this event.

The selfish player earns at most $\mu_{(1)}T_0$ during the initialization. Note that `Alternate Exploration` never uses collision information, meaning that the behavior of the strategic player during this phase does not change the behaviors of the cooperative players. Thus, the optimal strategy during this phase for the strategic player is to pull the best available arm. Let j be the rank of the strategic player[7]. For $t \notin \mathcal{E}$, this arm is the k-th arm of $\mathcal{M}^*$ with $k = t + j \pmod M + 1$. In a whole block of length M in $[T] \setminus \mathcal{E}$, the selfish player then earns at most $\sum_{k=1}^{M} \mu_{(k)}$.

Over all, when a strategic player deviates from `Alternate Exploration`, she earns at most:

$$\mathbb{E}[\mathrm{Rew}_T^j(s', \boldsymbol{s}^{-\boldsymbol{j}})] \leq \mu_{(1)}(\mathbb{E}\left[\#\mathcal{E} + M\right)] + \frac{T}{M}\sum_{k=1}^{M}\mu_{(k)}.$$

Note that we here add a factor $\mu_{(1)}$ in the initialization regret. This is only because the true loss of colliding is not 1 but $\mu_{(1)}$. Also, the additional $\mu_{(1)}M$ term is due to the fact that the last block of length M of `Alternate Exploration` is not totally completed.

Thanks to Theorem 6.1, it also comes:

$$\mathbb{E}[\mathrm{Rew}_T^j(\boldsymbol{s})] \geq \frac{T}{M}\sum_{k=1}^{M}\mu_{(k)} - \sum_{k>M}\frac{\mu_{(M)} - \mu_{(k)}}{\mathrm{kl}(\mu_{(k)}, \mu_{(M)})}\log(T) - \mathcal{O}\left(\mu_{(1)}\frac{K^3}{\mu_{(K)}}\log(T)\right).$$

Lemmas 6.2 and 6.10 yield that $\mathbb{E}[\#\mathcal{E}] = \mathcal{O}\left(\frac{K^3\log(T)}{\mu_{(K)}}\right)$, which concludes the proof.

2. We now prove the (α, ε)-stability of `Selfish-Robust MMAB`. Let $\varepsilon' = \mathbb{E}[\#\mathcal{E}] + M$. Note that this value is independent from the strategy of the deviating player j, since the sets $\mathcal{A}^m$ and $\mathcal{D}^m$ are independent from the actions of the player j. This is a consequence of the statistic sensing assumption.

Consider that player j is playing a deviation strategy $s' \in \mathcal{S}$ such that for some other player i and $l > 0$:

$$\mathbb{E}[\mathrm{Rew}_T^i(s', \boldsymbol{s}^{-\boldsymbol{j}})] \leq \mathbb{E}[\mathrm{Rew}_T^i(\boldsymbol{s})] - l - (\varepsilon' + M).$$

We will first compare the reward of player j with her optimal possible reward. The only way for the selfish player to influence the sampling strategy of another player is in modifying the rank attributed to this other player. The total rewards of cooperative players with ranks j and j' only differ by at most $\varepsilon' + M$ in expectation, without considering the loss due to collisions with the selfish player.

[7] If the strategic player has no attributed rank, it is the only non-attributed rank in $[M]$.

The only other way to cause regret to another player i is then to pull $\pi^i(t)$ at time t. This incurs a loss at most $\mu_{(1)}$ for player i, while this incurs a loss at least $\mu_{(M)}$ for player j, in comparison with her optimal strategy. This means that for incurring the additional loss l to the player i, player j must suffer herself from a loss $\frac{\mu_{(M)}}{\mu_{(1)}}$ compared to her optimal strategy s^*. Thus, for $\alpha = \frac{\mu_{(M)}}{\mu_{(1)}}$:

$$\mathbb{E}[\mathrm{Rew}_T^i(s', \boldsymbol{s^{-j}})] \leq \mathbb{E}[\mathrm{Rew}_T^i(\boldsymbol{s})] - l - (\varepsilon' + M) \implies \mathbb{E}[\mathrm{Rew}_T^j(s', \boldsymbol{s^{-j}})] \leq \mathbb{E}[\mathrm{Rew}_T^j(s^*, \boldsymbol{s^{-j}})] - \alpha l$$

The first point of Theorem 6.2 yields for its given ε: $\mathbb{E}[\mathrm{Rew}_T^j(s^*, \boldsymbol{s^{-j}})] \leq \mathbb{E}[\mathrm{Rew}_T^j(\boldsymbol{s})] + \varepsilon$.

Noting $l_1 = l + \varepsilon' + M$ and $\varepsilon_1 = \varepsilon + \alpha(\varepsilon' + M) = \mathcal{O}(\varepsilon)$, we have shown:

$$\mathbb{E}[\mathrm{Rew}_T^i(s', \boldsymbol{s^{-j}})] \leq \mathbb{E}[\mathrm{Rew}_T^i(\boldsymbol{s})] - l_1 \implies \mathbb{E}[\mathrm{Rew}_T^j(s', \boldsymbol{s^{-j}})] \leq \mathbb{E}[\mathrm{Rew}_T^j(\boldsymbol{s})] + \varepsilon_1 - \alpha l_1.$$

□

Auxiliary lemmas

This section provides useful Lemmas for the proof of Lemma 6.1. We first recall a useful version of Chernoff bound.

Lemma 6.12. *For any independent variables $X_1, \ldots, X_n$ in $[0,1]$ and $\delta \in (0,1)$:*

$$\mathbb{P}\left(\left|\sum_{i=1}^n X_i - \mathbb{E}[X_i]\right| \geq \delta \sum_{i=1}^n \mathbb{E}[X_i]\right) \leq 2e^{-\frac{\delta^2 \sum_{i=1}^n \mathbb{E}[X_i]}{3}}.$$

Proof of Lemma 6.4. 1. This is an application of Azuma-Hoeffding inequality on the variables $\mathbb{1}\left(t \in C_k^j(T_M)\right) \mid t \in T_k^j(T_M)$.

2. This is a consequence of Lemma 6.12 on the variables $\mathbb{1}\left(t \in T_k^j\right)$.

3. This is the same result on the variables $\mathbb{1}\left(t \in T_k^j\right) p_k^j(t) \mid \mathcal{F}_{t-1}$ where $\mathcal{F}_{t-1}$ is the filtration associated to the past events, using $\sum_{t=1}^{T_M} \mathbb{E}[\mathbb{1}\left(t \in T_k^j\right) p_k^j(t) \mid \mathcal{F}_{t-1}] \leq \frac{T_M \alpha_k}{K}$.

□

Proof of Lemma 6.5. From Lemma 6.4, it comes:

- $\mathbb{P}\left[\exists t \leq T, \left|\hat{p}_k^j(t) - \frac{1}{\#T_k^j} \sum_{t' \in T_k^j} p_k^j(t')\right| \geq 2\sqrt{\frac{\log(T)}{\#T_k^j}}\right] \leq \frac{2}{T}$,
- $\mathbb{P}\left[\exists t \leq T, \left|\frac{K \#T_k^j}{\alpha_k t} - 1\right| \geq \sqrt{\frac{6\log(T)K}{\alpha_k t}}\right] \leq \frac{2}{T}$, (6.3)

- $\mathbb{P}\left[\exists t \leq T, \left|\frac{K}{\alpha_k t}\sum_{t'\in T_k^j} p_k^j(t') - \frac{1}{t}\sum_{t'\leq t} p_k^j(t')\right| \geq \sqrt{\frac{6\log(T)K}{\alpha_k t}}\right] \leq \frac{2}{T}$.

Noting that $\sum_{t'\in T_k^j} p_k^j(t') \leq \#T_k^j$, Equation (6.3) implies:

$$\mathbb{P}\left[\exists t \leq T, \left|\frac{K}{\alpha_k t}\sum_{t'\in T_k^j} p_k^j(t') - \frac{1}{\#T_k^j}\sum_{t'\in T_k^j} p_k^j(t')\right| \geq \sqrt{\frac{6\log(T)K}{\alpha_k t}}\right] \leq \frac{2}{T}.$$

Combining these three inequalities and making the union bound over all the players and arms yield that with probability larger than $1 - \frac{6KM}{T}$:

$$\left|\hat{p}_k^j(t_m^j) - \frac{1}{t_m^j}\sum_{t\leq t_m^j} p_k^j(t)\right| \leq 2\sqrt{\frac{6\log(T)K}{\alpha_k t_m^j}} + 2\sqrt{\frac{\log(T)}{\#T_k^j(t_m^j)}}. \tag{6.4}$$

Moreover, under the same event, Equation (6.3) also gives that

$$T_k^j(t_m^j) \in \left[\frac{\alpha_k t_m^j}{K} - \sqrt{\frac{6\alpha_k t_m^j \log(T)}{K}},\ \frac{\alpha_k t_m^j}{K} + \sqrt{\frac{6\alpha_k t_m^j \log(T)}{K}}\right].$$

Specifically, this yields $n \leq \frac{\alpha_k t_m^j}{K} + \sqrt{\frac{6\alpha_k t_m^j \log(T)}{K}}$, or equivalently $\frac{t_m^j \alpha_k}{K} \geq n - 2\sqrt{\frac{3\log(T)}{2}}\sqrt{n + \frac{3\log(T)}{2}}$. Since $n = \beta^2 K^2 \log(T)$, this becomes $\frac{t_m^j \alpha_k}{K} \geq n(1 - 2\sqrt{\frac{3}{2\beta^2K^2}}\sqrt{1 + \frac{3}{2\beta^2K^2}})$ and Equation (6.4) now rewrites into:

$$\left|\hat{p}_k^j(t_m^j) - \frac{1}{t_m^j}\sum_{t\leq t_m^j} p_k^j(t)\right| \leq 2\sqrt{\frac{6\log(T)}{n\left(1 - 2\sqrt{\frac{3}{2\beta^2K^2}(1 + \frac{3}{2\beta^2K^2})}\right)}} + 2\sqrt{\frac{\log(T)}{n}}$$

Also, $n \geq \frac{\alpha_k t_m^j}{K} - \sqrt{\frac{6\log(T)\alpha_k t_m^j}{K}}$ for some k, which yields $\frac{t_m^j \alpha_k}{K} \leq n(1 + \frac{3}{\beta^2K^2} + 2\sqrt{\frac{3}{2\beta^2K^2}}\sqrt{1 + \frac{3}{2\beta^2K^2}})$. This relation then also holds for $\frac{t_m^j \alpha_{(K)}}{K}$. We have therefore proved that:

$$n\left(1 - 2\sqrt{\frac{3}{2\beta^2}}\sqrt{1 + \frac{3}{2\beta^2}}\right) \leq \frac{t_m^j \alpha_{(k)}}{K} \leq n\left(1 + \frac{3}{\beta^2} + 2\sqrt{\frac{3}{2\beta^2}}\sqrt{1 + \frac{3}{2\beta^2}}\right).$$

For $\beta \geq 39$, this gives the bound in Lemma 6.5. □

6.B Collective punishment proof

Recall that the punishment protocol consists in pulling each arm k with probability at least $p_k^j = \max\left(1 - \left(\gamma\frac{\sum_{l=1}^M \hat{\mu}_{(l)}^j}{M\hat{\mu}_k^j}\right)^{\frac{1}{M-1}}, 0\right)$. Lemma 6.13 below guarantees that such a sampling strategy is possible.

Lemma 6.13. *For* $p_k = \max\left(1-\left(\frac{\gamma\sum_{l=1}^{M}\hat{\mu}^j_{(l)}}{M\hat{\mu}^j_k}\right)^{\frac{1}{M-1}}, 0\right)$ *with* $\gamma = \left(1-\frac{1}{K}\right)^{M-1}$*:* $\sum_{k=1}^{K} p_k \leq 1$.

Proof. For ease of notation, define $x_k \coloneqq \hat{\mu}^j_k$, $\overline{x}_M \coloneqq \frac{\sum_{l=1}^{M} x_{(l)}}{M}$ and $S \coloneqq \{k \in [K] \mid x_k > \gamma\overline{x}_M\} = \{k \in [K] \mid p_k > 0\}$. We then get by concavity of $x \mapsto -x^{-\frac{1}{M-1}}$,

$$\sum_{k\in S} p_k = \#S \times \left(1 - (\gamma\overline{x}_M)^{\frac{1}{M-1}} \sum_{k\in S} \frac{(x_k)^{-\frac{1}{M-1}}}{\#S}\right), \tag{6.5}$$

$$\leq \#S \times \left(1 - \left(\frac{\gamma\overline{x}_M}{\overline{x}_S}\right)^{\frac{1}{M-1}}\right) \qquad \text{with } \overline{x}_S = \frac{1}{\#S}\sum_{k\in S} x_k. \tag{6.6}$$

We distinguish two cases.

First, if $\#S \leq M$, we then get $M\overline{x}_M \geq \#S\overline{x}_S$ because S is a subset of the M best empirical arms. The last inequality then becomes

$$\sum_{k\in S} p_k \leq \#S\left(1 - \left(\gamma\frac{\#S}{M}\right)^{\frac{1}{M-1}}\right).$$

Define $g(x) = \frac{\gamma}{M} - x(1-x)^{M-1}$. For $x \in (0,1]$:

$$\begin{aligned} g(x) \geq 0 &\iff \frac{\gamma}{xM} \geq (1-x)^{M-1}, \\ &\iff 1 - \left(\frac{\gamma}{xM}\right)^{\frac{1}{M-1}} \leq x, \\ &\iff \frac{1}{x}\left(1 - \left(\frac{\gamma}{xM}\right)^{\frac{1}{M-1}}\right) \leq 1. \end{aligned}$$

Thus, $g(\frac{1}{\#S}) \geq 0$ implies $\sum_{k\in S} p_k \leq 1$. We now show that g is indeed non negative on $[0,1]$. $x(1-x)^{M-1}$ is maximized at $\frac{1}{M}$ and is thus smaller than $\frac{1}{M}(1-1/M)^{M-1}$, and using the fact that $\frac{1}{M}(1-1/M)^{M-1} \leq \frac{\gamma}{M}$ for our choice of γ, we get the result for the first case.

The other case corresponds to $\#S > M$. In this case, the M best empirical arms are all in S and thus $\overline{x}_M \geq \overline{x}_S$. Equation (6.6) becomes:

$$\sum_{k\in S} p_k \leq \#S\left(1 - \gamma^{\frac{1}{M-1}}\right) \leq K(1-(1-1/K)) = 1.$$

□

6.C Missing elements for `SIC-GT`

In this whole section, M is assumed to be at least 3.

6.C.1 Description of the algorithm

This section provides a complete description of `SIC-GT`. The pseudocode of `SIC-GT` is given in Algorithm 6.3 and relies on several auxiliary protocols, which are described by Algorithms 6.3 to 6.9.

Algorithm 6.3: `SIC-GT`

Input: T, δ

$M, j \leftarrow$ `Initialize` (T, K) and punish $\leftarrow$ False
OptArms $\leftarrow \emptyset$, $M_p \leftarrow M$, $[K_p] \leftarrow [K]$ and $p \leftarrow 1$
while *not punish and* $\#$*OptArms* $< M$ **do**
  for $m = 0, \dots, \left\lceil \frac{K_p 2^p}{M_p} \right\rceil - 1$ **do**
    ArmstoPull $\leftarrow$ OptArms $\cup \left\{ i \in [K_p] \,\middle|\, i - mM_p \pmod{K_p} \in [M_p] \right\}$
    for M *rounds* **do**
      $k \leftarrow j + t \pmod{M} + 1$ and pull i the k-th element of ArmstoPull
      if $N_i^j(p) \leq 2^p$ **then** Update $\hat{\mu}_i^j$ `// N_i^j pulls on i by j this phase`
      if $\eta_i = 1$ **then** punish $\leftarrow$ True `// collisionless exploration`
    end
  end
  (punish, OptArms, $[K_p], M_p$) $\leftarrow$ `CommPhase` $(\hat{\mu}^j, j, p,$ OptArms $, [K_p], M_p)$
  $p \leftarrow p + 1$
end
if *punish* **then** `PunishHomogeneous` (p)
else `// exploitation phase`
  $k \leftarrow j + t \pmod{M} + 1$ and pull i, the k-th arm of OptArms
  if $\eta_i = 1$ **then** punish $\leftarrow$ True
end

Protocol 6.5: `ReceiveMean`

Input: j, p

$\widetilde{\mu} \leftarrow 0$
for $n = 0, \dots, p$ **do**
  Pull j
  if $\eta_j(t) = 1$ **then** $\widetilde{\mu} \leftarrow \widetilde{\mu} + 2^{-n}$
end
return $\widetilde{\mu}$ `// sent mean`

Protocol 6.6: `SendMean`

Input: $j, l, p, \widetilde{\mu}$

$\mathbf{m} \leftarrow$ dyadic writing of $\widetilde{\mu}$ of length $p+1$, i.e., $\widetilde{\mu} = \sum_{n=0}^{p} m_n 2^{-n}$
for $n = 0, \dots, p$ **do**
  if $m_n = 1$ **then** Pull l `// send 1`
  else Pull j `// send 0`
end

Initialization phase. The purpose of the initialization phase is to estimate M and attribute ranks in $[M]$ to all the players. This is done by `Initialize`, which is given in Algorithm 6.3. It simply consists in pulling uniformly at random for a long time to infer M from the probability

Protocol 6.3: `Initialize`

Input: T, K

$n_{\text{coll}} \leftarrow 0$ and $j \leftarrow -1$

for $12eK^2\log(T)$ *rounds* **do** Pull $k \sim \mathcal{U}(K)$ and $n_{\text{coll}} \leftarrow n_{\text{coll}} + \eta_k$ `// estim. M`

$\hat{M} \leftarrow 1 + \text{round}\left(\log\left(1 - \frac{n_{\text{coll}}}{12eK^2\log(T)}\right) / \log\left(1 - \frac{1}{K}\right)\right)$

for $K\log(T)$ *rounds* **do** `// get rank`

  if $j = -1$ **then**

    Pull $k \sim \mathcal{U}(\hat{M})$; **if** $\eta_k = 0$ **then** $j \leftarrow k$

  else Pull j

end

return $(\hat{M}, j)$

of collision. Then it proceeds to a Musical Chairs procedure so that each player ends with a different arm in $[M]$, corresponding to her rank.

Exploration phase. As explained in Section 6.4.2, each arm that still needs to be explored (those in $[K_p]$, with Algorithm 6.3 notations) is pulled at least $M2^p$ times during the p-th exploration phase. Moreover, as soon as an arm is found optimal, it is pulled for each remaining round of the exploration. The last point is that each arm is pulled the exact same amount of time by every player, in order to ensure fairness of the algorithm, while still avoiding collisions. This is the interest of the ArmstoPull set in Algorithm 6.3. At each time step, the pulled arms are the optimal ones and M_p arms that still need to be explored. The players proceed to a sliding window over these arms to explore, so that the difference in pulls for two arms in $[K_p]$ is at most 1 for any player and phase.

Communication phase. The pseudocode for a whole communication phase is given by `CommPhase` in Protocol 6.4. Players first quantize their empirical means before sending them in p bits to each leader. The protocol to send a message is given by Protocol 6.6, while Protocol 6.5 describes how to receive the message. The messages are sent using back and forth procedures to detect corrupted messages.

After this, leaders communicate the received statistics to each other, to ensure that no player sent differing ones to them.

They can then determine which arms are optimal/suboptimal using `RobustUpdate` given by Protocol 6.7. As explained in Section 6.4.2, it cuts out the extreme estimates and decides based on the $M - 2$ remaining ones.

Afterwards, the leaders signal to the remaining players the sets of optimal and suboptimal arms as described by Protocol 6.8. If the leaders send differing information, it is detected by at

Protocol 6.4: `CommPhase`

Input: $\hat{\mu}^j, j, p, \text{OptArms}, [K_p], M_p$

punish $\leftarrow$ False
for K *rounds* **do** `// receive punishment signal`
  Pull $k = t + j \pmod K + 1$; **if** $\eta_k = 1$ **then** punish $\leftarrow$ True
end
$\widetilde{\mu}_k^j \leftarrow \begin{cases} 2^{-p}\left(\lfloor 2^p \hat{\mu}_k^j \rfloor + 1\right) \text{ with proba } 2^p \hat{\mu}_k^j - \lfloor 2^p \hat{\mu}_k^j \rfloor \\ 2^{-p}\lfloor 2^p \hat{\mu}_k^j \rfloor \text{ otherwise} \end{cases}$ `// quantization`
for $(i, l, k) \in [M] \times \{1, 2\} \times [K]$ *such that* $i \neq l$ **do** `// i sends` $\widetilde{\mu}_k^i$ `to l`
  if $j = i$ **then** `// sending player`
    `SendMean` $(j, l, p, \widetilde{\mu}_k^j)$ and $q \leftarrow$ `ReceiveMean` (j, p) `// back and forth`
    if $q \neq \widetilde{\mu}_k^j$ **then** punish $\leftarrow$ True `// corrupted message`
  else if $j = l$ **then** $\widetilde{\mu}_k^i \leftarrow$ `ReceiveMean` (j, p) and `SendMean` $(j, i, p, \widetilde{\mu}_k^i)$
  else Pull j `// waiting for others`
end
for $(i, l, m, k) \in \{(1, 2), (2, 1)\} \times [M] \times [K]$ **do** `// leaders check info match`
  if $j = i$ **then** `SendMean` $(j, l, p, \widetilde{\mu}_k^m)$
  else if $j = l$ **then**
    $q \leftarrow$ `ReceiveMean` (j, p); **if** $q \neq \widetilde{\mu}_k^m$ **then** punish $\leftarrow$ True `// info differ`
  else Pull j `// waiting for leaders`
end
if $j \in \{1, 2\}$ **then** (Acc, Rej) $\leftarrow$ `RobustUpdate` $(\widetilde{\mu}, p, \text{OptArms}, [K_p], M_p)$
else Acc, Rej $\leftarrow \emptyset$ `// arms to accept/reject`
(punish, Acc) $\leftarrow$ `SignalSet` (Acc, j, punish)
(punish, Rej) $\leftarrow$ `SignalSet` (Rej, j, punish)
return $(\text{punish}, \text{OptArms} \cup \text{Acc}, [K_p] \setminus (\text{Acc} \cup \text{Rej}), M_p - \#\text{Acc})$

least one player.

If the presence of a malicious player is detected at some point of this communication phase, then players signal to each other to trigger the punishment protocol described by Protocol 6.9.

Exploitation phase. If no malicious player perturbed the communication, players end up having detected the M optimal arms. As soon as it is the case, they only pull these M arms in a collisionless way until the end.

6.C.2 Regret analysis

This section aims at proving the first point of Theorem 6.6, using similar techniques as in (Boursier and Perchet, 2019). The regret is first divided into three parts:

Protocol 6.7: `RobustUpdate`

Input: $\widetilde{\mu}, p, \text{OptArms}, [K_p], M_p$

Define for all k, $i^k \leftarrow \arg\max_{j\in[M]} \widetilde{\mu}_k^j$ and $i_k \leftarrow \arg\min_{j\in[M]} \widetilde{\mu}_k^j$

$\widetilde{\mu}_k \leftarrow \sum_{j\in[M]\setminus\{i^k,i_k\}} \widetilde{\mu}_k^j$ and $b \leftarrow 4\sqrt{\frac{\log(T)}{(M-2)2^{p+1}}}$

Rej $\leftarrow$ set of arms k verifying $\#\{i \in [K_p] \mid \widetilde{\mu}_i - b \geq \widetilde{\mu}_k + b\} \geq M_p$

Acc $\leftarrow$ set of arms k verifying $\#\{i \in [K_p] \mid \widetilde{\mu}_k - b \geq \widetilde{\mu}_i + b\} \geq K_p - M_p$

return (Acc, Rej)

Protocol 6.8: `SignalSet`

Input: S, j, punish

length_S $\leftarrow \#S$ `// length of` S `for leaders, 0 for others`

for K *rounds* **do** `// leaders send` $\#S$

  if $j \in \{1, 2\}$ **then** Pull length_S

  else

    Pull $k = t + j \pmod K + 1$

    if $\eta_k = 1$ *and length_S* $\neq 0$ **then** punish $\leftarrow$ True `// receive different info`

    if $\eta_k = 1$ *and length_S* $= 0$ **then** length_S $\leftarrow k$

end

for $n = 1, \ldots, \text{length\_S}$ **do** `// send/receive` S

  for K *rounds* **do**

    if $j \in \{1, 2\}$ **then** Pull n-th arm of S

    else

      Pull $k = t + j \pmod K + 1$; **if** $\eta_k = 1$ **then** Add k to S

  end

end

if $\#S \neq$ length_S **then** punish $\leftarrow$ True `// corrupted info`

return (punish, S)

$$R(T) = \mathbb{E}[R^{\text{init}} + R^{\text{comm}} + R^{\text{explo}}], \tag{6.7}$$

where

$$R^{\text{init}} = T_{\text{init}} \sum_{k=1}^{M} \mu_{(k)} - \sum_{t=1}^{T_{\text{init}}} \sum_{j=1}^{M} \mu\pi^j(t)(1 - \eta^j(t)) \text{ with } T_{\text{init}} = (12eK^2 + K)\log(T),$$

$$R^{\text{comm}} = \sum_{t\in\text{Comm}} \sum_{j=1}^{M} (\mu_{(j)} - \mu\pi^j(t)(1 - \eta^j(t))) \text{ with Comm the set of communication steps,}$$

$$R^{\text{explo}} = \sum_{t\in\text{Explo}} \sum_{j=1}^{M} (\mu_{(j)} - \mu\pi^j(t)(1 - \eta^j(t))) \text{ with Explo} = \{T_{\text{init}} + 1, \ldots, T\} \setminus \text{Comm.}$$

Protocol 6.9: `PunishHomogeneous`

Input: p

if *communication phase p starts in less than M rounds* **then**
  for $M + K$ *rounds* **do** Pull j `// signal punish to everyone`
else for M *rounds* **do** Pull the first arm of ArmstoPull as defined in Algorithm 6.3

$\gamma \leftarrow (1 - 1/K)^{M-1}$ and $\delta = \frac{1-\gamma}{1+3\gamma}$; Set $\hat{\mu}_k^j, S_k^j, s_k^j, n_k^j \leftarrow 0$
while $\exists k \in [K], \delta\hat{\mu}_k^j < 2s_k^j(\log(T)/n_k^j)^{1/2} + \frac{14\log(T)}{3(n_k^j-1)}$ **do** `// estimate` μ_k
  Pull $k = t + j \pmod K + 1$
  if $\delta\hat{\mu}_k^j < 2s_k^j(\log(T)/n_k^j)^{1/2} + \frac{14\log(T)}{3(n_k^j-1)}$ **then**
    Update $\hat{\mu}_k^j \leftarrow \frac{n_k^j}{n_k^j+1}\hat{\mu}_k^j + X_k(t)$ and $n_k^j \leftarrow n_k^j + 1$
    Update $S_k^j \leftarrow S_k^j + (X_k)^2$ and $s_k^j \leftarrow \sqrt{\frac{S_k^j - (\hat{\mu}_k^j)^2}{n_k^j - 1}}$
end
$p_k \leftarrow \left(1 - \left(\gamma \frac{\sum_{l=1}^M \hat{\mu}_{(l)}^j(t)}{M\hat{\mu}_k^j(t)}\right)^{\frac{1}{M-1}}\right)_+$; $\widetilde{p}_k \leftarrow p_k / \sum_{l=1}^K p_l$ `// renormalize`
while $t \leq T$ **do** Pull k with probability p_k `// punish`

A communication step is defined as a round where any player is using the `CommPhase` protocol. Lemma 6.14 provides guarantees about the initialization phase. When all players correctly estimate M and have different ranks after the protocol `Initialize`, the initialization phase is said successful.

Lemma 6.14. *Independently of the sampling strategy of the selfish player, if all other players follow* `Initialize`*, with probability at least* $1 - \frac{3M}{T}$*: $\hat{M}^j = M$ and all cooperative players end with different ranks in $[M]$.*

Proof. Let $q_k(t) = \mathbb{P}[\text{selfish player pulls } k \text{ at time } t]$. Then, for each cooperative player j during the initialization phase:

$$\begin{aligned}
\mathbb{P}[\text{player } j \text{ observes a collision at time } t] &= \sum_{k=1}^{K} \frac{1}{K}(1-1/K)^{M-2}(1-q_k(t)) \\
&= (1-1/K)^{M-2}(1 - \frac{\sum_{k=1}^K q_k(t)}{K}) \\
&= (1-1/K)^{M-1}
\end{aligned}$$

Define $p = (1-1/K)^{M-1}$ the probability to collide and $\hat{p}^j = \frac{\sum_{t=1}^{12eK^2\log(T)} \mathbb{1}\left(\eta_{\pi^j(t)}=1\right)}{12eK^2\log(T)}$ its

estimation by player j. The Chernoff bound given by Lemma 6.12 gives:

$$\begin{aligned}\mathbb{P}\left[\left|\hat{p}^j - p\right| \geq \frac{p}{2K}\right] &\leq 2e^{-\frac{p\log(T)}{e}} \\ &\leq 2/T\end{aligned}$$

If $|\hat{p}^j - p| < \frac{p}{2K}$, using the same reasoning as in the proof of Lemma 6.1 leads to $1+\frac{\log(1-\hat{p}^j)}{\log(1-1/K)} \in (M-1/2, M+1/2)$ and then $\hat{M}^j = M$. With probability at least $1-2M/T$, all cooperative players correctly estimate M.

Afterwards, the players sample uniformly in $[M]$ until observing no collision. As at least an arm in $[M]$ is not pulled by any other player, at each time step of this phase, when pulling uniformly at random:

$$\mathbb{P}[\eta_{\pi^j(t)} = 0] \geq 1/M.$$

A player gets a rank as soon as she observes no collision. With probability at least $1-(1-1/M)^n$, she thus gets a rank after at most n pulls during this phase. Since this phase lasts $K\log(T)$ pulls, she ends the phase with a rank with probability at least $1-1/T$. Using a union bound finally yields that every player ends with a rank and a correct estimation of M. Moreover, these ranks are different between all the players, because a player fixes to the arm j as soon as she gets attributed the rank j. □

Lemma 6.15 bounds the exploration regret of `SIC-GT` and is proved in Section 6.C.2. Note that a minimax bound can also be proved as done in Chapter 4.

Lemma 6.15. *If all players follow* `SIC-GT`*, with probability* $1-\mathcal{O}\left(\frac{KM\log(T)}{T}\right)$*,*

$$R^{\textit{explo}} = \mathcal{O}\left(\sum_{k>M}\frac{\log(T)}{\mu_{(M)}-\mu_{(k)}}\right).$$

Lemma 6.16 finally bounds the communication regret.

Lemma 6.16. *If all players follow* `SIC-GT`*, with probability* $1-\mathcal{O}\left(\frac{KM\log(T)}{T}+\frac{M}{T}\right)$*:*

$$R^{\textit{comm}} = \mathcal{O}\left(M^2K\log^2\left(\frac{\log(T)}{(\mu_{(M)}-\mu_{(M+1)})^2}\right)\right).$$

Proof. The proof is conditioned on the success of the initialization phase, which happens with probability $1-\mathcal{O}\left(\frac{M}{T}\right)$. Proposition 6.1 given in Section 6.C.2 yields that with probability $1-\mathcal{O}\left(\frac{KM\log(T)}{T}\right)$, the number of communication phases is bounded by $N = \mathcal{O}\left(\log\left(\frac{\log(T)}{(\mu_{(M)}-\mu_{(M+1)})^2}\right)\right)$. The p-th communication phase lasts $8MK(p+1)+3K+K\#\text{Acc}(p)+K\#\text{Rej}(p)$, where

Acc and Rej respectively are the accepted and rejected arms at the p-th phase. Their exact definitions are given in Algorithm 6.7. An arm is either accepted or rejected only once, so that $\sum_{p=1}^{N} \#\text{Acc}(p) + \#\text{Rej}(p) = K$. The total length of Comm is thus bounded by:

$$\begin{aligned}\#\text{Comm} &\leq \sum_{p=1}^{N} 8MK(p+1) + 3K + K\#\text{Acc}(p) + K\#\text{Rej}(p) \\ &\leq 8MK\frac{(N+2)(N+1)}{2} + 3KN + K^2\end{aligned}$$

Which leads to $R^{\text{comm}} = \mathcal{O}\left(M^2 K \log^2\left(\frac{\log(T)}{(\mu_{(M)}-\mu_{(M+1)})^2}\right)\right)$ using the given bound for N. □

Proof of Theorem 6.6. Using Lemmas 6.14 to 6.16 and Equation (6.7) it comes that with probability $1 - \mathcal{O}\left(\frac{KM\log(T)}{T}\right)$:

$$R_T \leq \mathcal{O}\left(\sum_{k>M} \frac{\log(T)}{\mu_{(M)} - \mu_{(k)}} + M^2 K \log^2\left(\frac{\log(T)}{(\mu_{(M)} - \mu_{(M+1)})^2}\right) + MK^2\log(T)\right).$$

The regret incurred by the low probability event is $\mathcal{O}(KM^2\log(T))$, leading to Theorem 6.6. □

Proof of Lemma 6.15

Lemma 6.15 relies on the following concentration inequality.

Lemma 6.17. *Conditioned on the success of the initialization and independently of the means sent by the selfish player, if all other players play cooperatively and send uncorrupted messages, for any $k \in [K]$:*

$$\mathbb{P}[\exists p \leq n, |\widetilde{\mu}_k(p) - \mu_k| \geq B(p)] \leq \frac{6nM}{T}$$

where $B(p) = 4\sqrt{\frac{\log(T)}{(M-2)2^{p+1}}}$ and $\widetilde{\mu}_k(p)$ is the centralized mean of arm k at the end of phase p, once the extremes have been cut out. It exactly corresponds to the $\widetilde{\mu}_k$ of Protocol 6.7.

Proof. At the end of phase p, $(2^{p+1}-1)$ observations are used for each player j and arm k. Hoeffding bound then gives: $\mathbb{P}\left[\left|\widehat{\mu}_k^j(p) - \mu_k\right| \geq \sqrt{\frac{\log(T)}{2^{p+1}}}\right] \leq \frac{2}{T}$. The quantization only adds an error of at most 2^{-p}, yielding for every cooperative player:

$$\mathbb{P}\left[\left|\widetilde{\mu}_k^j(p) - \mu_k\right| \geq 2\sqrt{\frac{\log(T)}{2^{p+1}}}\right] \leq \frac{2}{T} \tag{6.8}$$

Assume w.l.o.g. that the selfish player has rank M. Hoeffding inequality also yields:

$$\mathbb{P}\left[\left|\frac{1}{M-1}\sum_{j=1}^{M-1}\hat{\mu}_k^j(p)-\mu_k\right|\geq\sqrt{\frac{\log(T)}{(M-1)2^{p+1}}}\right]\leq\frac{2}{T}.$$

Since $\sum_{j=1}^{M-1}2^p(\widetilde{\mu}_k^j(p)-\hat{\mu}_k^j(p))$ is the difference between $M-1$ Bernoulli variables and their expectation, Hoeffding inequality yields $\mathbb{P}\left[\left|\frac{1}{M-1}\sum_{j=1}^{M-1}(\widetilde{\mu}_k^j-\hat{\mu}_k^j(p))\right|\geq\sqrt{\frac{\log(T)}{(M-1)2^{p+1}}}\right]\leq\frac{2}{T}$ and:

$$\mathbb{P}\left[\left|\frac{1}{M-1}\sum_{j=1}^{M-1}\widetilde{\mu}_k^j(p)-\mu_k\right|\geq 2\sqrt{\frac{\log(T)}{(M-1)2^{p+1}}}\right]\leq\frac{4}{T}. \tag{6.9}$$

Using the triangle inequality combining Equations (6.8) and (6.9) yields for any $j\in[M-1]$:

$$\begin{aligned}\mathbb{P}\left[\left|\frac{1}{M-2}\sum_{\substack{j'\in[M-1]\\j'\neq j}}\widetilde{\mu}_k^j(p)-\mu_k\right|\geq 4\sqrt{\frac{\log(T)}{(M-2)2^{p+1}}}\right]&\leq\mathbb{P}\left[\frac{M-1}{M-2}\left|\frac{1}{M-1}\sum_{j'\in[M-1]}\widetilde{\mu}_k^j(p)-\mu_k\right|\right.\\&\quad\left.+\frac{1}{M-2}\left|\widetilde{\mu}_k^j(p)-\mu_k\right|\geq 4\sqrt{\frac{\log(T)}{(M-2)2^{p+1}}}\right]\\&\leq\mathbb{P}\left[\left|\frac{1}{M-1}\sum_{j=1}^{M-1}\widetilde{\mu}_k^j(p)-\mu_k\right|\geq 2\sqrt{\frac{\log(T)}{(M-1)2^{p+1}}}\right]\\&\quad+\mathbb{P}\left[\left|\widetilde{\mu}_k^j(p)-\mu_k\right|\geq 2\sqrt{\frac{\log(T)}{2^{p+1}}}\right]\\&\leq\frac{6}{T}.\end{aligned} \tag{6.10}$$

Moreover by construction, no matter what mean sent the selfish player,

$$\min_{j\in[M-1]}\frac{1}{M-2}\sum_{\substack{j'\in[M-1]\\j'\neq j}}\widetilde{\mu}_k^j(p)\leq\widetilde{\mu}_k(p)\leq\max_{j\in[M-1]}\frac{1}{M-2}\sum_{\substack{j'\in[M-1]\\j'\neq j}}\widetilde{\mu}_k^j(p).$$

Indeed, assume that the selfish player sends a mean larger than all other players. Then her mean as well as the minimal sent mean are cut out and $\widetilde{\mu}_k(p)$ is then equal to the right term. Conversely if she sends the smallest mean, $\widetilde{\mu}_k(p)$ corresponds to the left term. Since $\widetilde{\mu}_k(p)$ is non-decreasing in $\widetilde{\mu}_k^M(p)$, the inequality also holds in the case where the selfish player sends neither the smallest nor the largest mean.

Finally, using a union bound over all $j\in[M-1]$ with Equation (6.10) yields Lemma 6.17.

□

Using classical MAB techniques then yields Proposition 6.1.

Proposition 6.1. *Independently of the selfish player behavior, as long as the* `PunishHomogeneous` *protocol is not used, with probability* $1 - \mathcal{O}\left(\frac{KM\log(T)}{T}\right)$*, every optimal arm* k *is accepted after at most* $\mathcal{O}\left(\frac{\log(T)}{(\mu_k - \mu_{(M+1)})^2}\right)$ *pulls and every sub-optimal arm* k *is rejected after at most* $\mathcal{O}\left(\frac{\log(T)}{(\mu_{(M)} - \mu_k)^2}\right)$*pulls during exploration phases.*

Proof. The fact that the `PunishHomogeneous` protocol is not started just means that no corrupted message is sent between cooperative players. The proof is conditioned on the success of the initialization phase, which happens with probability $1 - \mathcal{O}\left(\frac{M}{T}\right)$. Note that there are at most $\log_2(T)$ exploration phases. Thanks to Lemma 6.17, with probability $1 - \mathcal{O}\left(\frac{KM\log(T)}{T}\right)$, the inequality $|\widetilde{\mu}_k(p) - \mu_k| \leq B(p)$ thus holds for any p. The remaining of the proof is conditioned on this event. Especially, an optimal arm is never rejected and a suboptimal one never accepted.

First consider an optimal arm k and note $\Delta_k = \mu_k - \mu_{(M+1)}$ the optimality gap. Let p_k be the smallest integer p such that $(M-2)2^{p+1} \geq \frac{16^2\log(T)}{\Delta_k^2}$. In particular, $4B(p_k) \leq \Delta_k$, which implies that the arm k is accepted at the end of the communication phase p_k or before.

Necessarily, $(M-2)2^{p_k+1} \leq \frac{2\cdot 16^2\log(T)}{\Delta_k^2}$ and especially, $M2^{p_k+1} = \mathcal{O}\left(\frac{\log(T)}{\Delta_k^2}\right)$. Note that the number of exploratory pulls on arm k during the p first phases is bounded by $M(2^{p+1}+p)$[8], leading to Proposition 6.1. The same holds for the sub-optimal arms with $\Delta_k = \mu_{(M)} - \mu_k$. □

In the following, we keep the notation $t_k = \frac{c\log(T)}{\left(\mu_k - \mu_{(M)}\right)^2}$, where c is a universal constant, such that with probability $1 - \mathcal{O}\left(\frac{KM}{T}\right)$, every arm k is correctly accepted or rejected after a time at most t_k. All players are now assumed to play `SIC-GT`, e.g., there is no selfish player. Since there is no collision during exploration/exploitation (conditionally on the success of the initialization phase), the following decomposition holds (Anantharam et al., 1987a):

$$R^{\text{explo}} = \sum_{k>M}(\mu_{(M)} - \mu_{(k)})T_{(k)}^{\text{explo}} + \sum_{k\leq M}(\mu_{(k)} - \mu_{(M)})(T^{\text{explo}} - T_{(k)}^{\text{explo}}), \tag{6.11}$$

where $T^{\text{explo}} = \#\text{Explo}$ and $T_{(k)}^{\text{explo}}$ is the centralized number of pulls on the k-th best arm during exploration or exploitation.

Lemma 6.18. *If all players follow* `SIC-GT`*, with probability* $1 - \mathcal{O}\left(\frac{KM\log(T)}{T}\right)$*, it holds:*

- *for* $k > M$, $(\mu_{(M)} - \mu_{(k)})T_{(k)}^{\text{explo}} = \mathcal{O}\left(\frac{\log(T)}{\mu_{(M)} - \mu_{(k)}}\right)$.

[8] During the exploration phase p, each explored arm is pulled between $M2^p$ and $M(2^p+1)$ times.

- $\sum_{k\leq M}(\mu_{(k)}-\mu_{(M)})(T^{explo}-T^{explo}_{(k)}) = \mathcal{O}\left(\sum_{k>M}\frac{\log(T)}{\mu_{(M)}-\mu_k}\right)$.

Proof. With probability $1-\mathcal{O}\left(\frac{KM\log(T)}{T}\right)$, Proposition 6.1 yields that every arm k is correctly accepted or rejected at time at most t_k. The remaining of the proof is conditioned on this event and the success of the initialization phase. The first point of Lemma 6.18 is a direct consequence of Proposition 6.1. It remains to prove the second point.

Let $\hat{p}_k$ be the number of the phase at which the arm k is either accepted or rejected and let K_p be the number of arms that still need to be explored at the beginning of phase p and M_p be the number of optimal arms that still need to be explored. The following two key Lemmas are crucial to obtain the second point.

Lemma 6.19. *Under the assumptions of Lemma 6.18:*

$$\sum_{k\leq M}(\mu_{(k)}-\mu_{(M)})(T^{explo}-T^{explo}_{(k)}) \leq \sum_{j>M}\sum_{k\leq M}\sum_{p=1}^{\min(\hat{p}_{(k)},\hat{p}_{(j)})}(\mu_{(k)}-\mu_{(M)})2^p\frac{M}{M_p} + o\left(\log(T)\right).$$

Lemma 6.20. *Under the assumptions of Lemma 6.18, for any $j>M$:*

$$\sum_{k\leq M}\sum_{p=1}^{\min(\hat{p}_{(k)},\hat{p}_{(j)})}(\mu_{(k)}-\mu_{(M)})2^p\frac{M}{M_p} \leq \mathcal{O}\left(\frac{\log(T)}{\mu_{(M)}-\mu_{(j)}}\right).$$

Combining these two Lemmas with Equation (6.11) finally yields Lemma 6.15. □

Proof of Lemma 6.19. Consider an optimal arm k. During the p-th exploration phase, either k has already been accepted and is pulled $M\left\lceil\frac{K_p2^p}{M_p}\right\rceil$ times; or k has not been accepted yet and is pulled at least 2^pM, i.e., is not pulled at most $M\left(\left\lceil\frac{K_p2^p}{M_p}\right\rceil-2^p\right)$ times. This gives:

$$\begin{aligned}
(\mu_{(k)}-\mu_{(M)})(T^{\text{explo}}-T^{\text{explo}}_{(k)}) &\leq \sum_{p=1}^{\hat{p}_k}(\mu_{(k)}-\mu_{(M)})M\left(\left\lceil\frac{K_p2^p}{M_p}\right\rceil-2^p\right),\\
&\leq \sum_{p=1}^{\hat{p}_k}(\mu_{(k)}-\mu_{(M)})M\left(\frac{K_p2^p}{M_p}-2^p+1\right),\\
&\leq \hat{p}_k(\mu_{(k)}-\mu_{(M)})M + \sum_{p=1}^{\hat{p}_k}(\mu_{(k)}-\mu_{(M)})(K_p-M_p)\frac{M}{M_p}2^p.
\end{aligned}$$

We assumed that every arm k is correctly accepted or rejected after a time at most t_k. This implies that $\hat{p}_k = o\left(\log(T)\right)$. Moreover, K_p-M_p is the number of suboptimal arms not rejected at phase p, i.e., $K_p-M_p=\sum_{j>M}\mathbb{1}\left(p\leq\hat{p}_{(j)}\right)$ and this proves Lemma 6.19. □

Proof of Lemma 6.20. For $j > M$, define $A_j = \sum_{k\leq M} \sum_{p=1}^{\min(\hat{p}_{(k)},\hat{p}_{(j)})} (\mu_{(k)} - \mu_{(M)}) 2^p \frac{M}{M_p}$. We want to show $A_j \leq \mathcal{O}\left(\frac{\log(T)}{\mu_{(M)}-\mu_{(j)}}\right)$ with the considered conditions. Note $N(p) = M(2^{p+1}-1)$ and $\Delta(p) = \sqrt{\frac{c\log(T)}{N(p)}}$. The inequality $\hat{p}_{(k)} \geq p$ then implies $\mu_{(k)} - \mu_{(M)} < \Delta(p)$, i.e.,

$$\begin{aligned} A_j &\leq \sum_{k\leq M} \sum_{p=1}^{\hat{p}_{(j)}} 2^p \Delta(p) \mathbb{1}\left(p \leq \hat{p}_{(k)}\right) \frac{M}{M_p} = \sum_{p=1}^{\hat{p}_{(j)}} 2^p \Delta(p) M \\ &\leq \sum_{p=1}^{\hat{p}_{(j)}} \Delta(p)(N(p) - N(p-1)) \end{aligned}$$

The equality comes because $\sum_{k\leq M} \mathbb{1}\left(p \leq \hat{p}_{(k)}\right)$ is exactly M_p. Then from the definition of $\Delta(p)$:

$$\begin{aligned} A_j &\leq c\log(T) \sum_{p=1}^{\hat{p}_{(j)}} \Delta(p) \left(\frac{1}{\Delta(p)} + \frac{1}{\Delta(p-1)}\right)\left(\frac{1}{\Delta(p)} - \frac{1}{\Delta(p-1)}\right) \\ &\leq (1+\sqrt{2}) c\log(T) \sum_{p=1}^{\hat{p}_{(j)}} \left(\frac{1}{\Delta(p)} - \frac{1}{\Delta(p-1)}\right) \\ &\leq (1+\sqrt{2}) c\log(T)/\Delta(\hat{p}_{(j)}) \\ &\leq (1+\sqrt{2})\sqrt{c\log(T) N(\hat{p}_{(j)})} \end{aligned}$$

By definition, $N(\hat{p}_{(j)})$ is smaller than the number of exploratory pulls on the j-th best arm and is thus bounded by $\frac{c\log(T)}{(\mu_{(M)}-\mu_{(j)})^2}$, leading to Lemma 6.20. □

6.C.3 Selfish robustness of `SIC-GT`

In this section, the second point of Theorem 6.6 is proven. First Lemma 6.21 gives guarantees for the punishment protocol. Its proof is given in Section 6.C.3.

Lemma 6.21. *If the* `PunishHomogeneous` *protocol is started at time T_{punish} by $M-1$ players, then for the remaining player j, independently of her sampling strategy:*

$$\mathbb{E}[\text{Rew}_T^j | \text{punish}] \leq \mathbb{E}[\text{Rew}^j_{T_{punish}+t_p}] + \widetilde{\alpha} \frac{T - T_{\text{punish}} - t_p}{M} \sum_{k=1}^{M} \mu_{(k)},$$

with $t_p = \mathcal{O}\left(\frac{K}{(1-\widetilde{\alpha})^2 \mu_{(K)}} \log(T)\right)$ *and* $\widetilde{\alpha} = \frac{1+(1-1/K)^{M-1}}{2}$.

Proof of the second point of Theorem 6.6 (Nash equilibrium). First fix T_{punish} the time at which the punishment protocol starts if it happens (and T if it does not). Before this time, the selfish

player can not perturb the initialization phase, except by changing the ranks distribution. Moreover, the exploration/exploitation phase is not perturbed as well, as claimed by Proposition 6.1. The optimal strategy then earns at most T_{init} during the initialization and $\#\text{Comm}$ during the communication. With probability $1 - \mathcal{O}\left(\frac{KM\log(T)}{T}\right)$, the initialization is successful and the concentration bound of Lemma 6.5 holds for each arm and player all the time. The following is conditioned on this event.

Note that during the exploration, the cooperative players pull every arm the exact same amount of times. Since the upper bound time t_k to accept or reject an arm does not depend on the strategy of the selfish player, Lemma 6.18 actually holds for any cooperative player j:

$$\sum_{k\leq M}\left(\mu_{(k)}-\mu_{(M)}\right)\left(\frac{T^{\text{explo}}}{M}-T^j_{(k)}\right)=\mathcal{O}\left(\frac{1}{M}\sum_{k>M}\frac{\log(T)}{\mu_{(M)}-\mu_k}\right), \tag{6.12}$$

where $T^j_{(k)}$ is the number of pulls by player j on the k-th best arm during the exploration/exploitation. The same kind of regret decomposition as in Equation (6.11) is possible for the regret of the selfish player j and especially:

$$R_j^{\text{explo}}\geq\sum_{k\leq M}\left(\mu_{(k)}-\mu_{(M)}\right)\left(\frac{T^{\text{explo}}}{M}-T^j_{(k)}\right).$$

However, the optimal strategy for the selfish player is to pull the best available arm during the exploration and especially to avoid collisions. This implies the constraint $T^j_{(k)}\leq T^{\text{explo}}-\sum_{j\neq j'}T^{j'}_{(k)}$. Using this constraint with Equation (6.12) yields $\frac{T^{\text{explo}}}{M}-T^j_{(k)}\geq-\sum_{j\neq j'}\frac{T^{\text{explo}}}{M}-T^{j'}_{(k)}$ and then

$$R_j^{\text{explo}}\geq-\mathcal{O}\left(\sum_{k>M}\frac{\log(T)}{\mu_{(M)}-\mu_k}\right),$$

which can be rewritten as

$$\text{Rew}_j^{\text{explo}}\leq\frac{T^{\text{explo}}}{M}\sum_{k=1}^{M}\mu_{(k)}+\mathcal{O}\left(\sum_{k>M}\frac{\log(T)}{\mu_{(M)}-\mu_k}\right).$$

Thus, for any strategy s' when adding the low probability event of a failed exploration or initialization,

$$\begin{aligned}\mathbb{E}[\text{Rew}^j_{t_p+T_{\text{punish}}}(s',\boldsymbol{s^{-j}})]\leq{}&(T_{\text{init}}+\#\text{Comm}+t_p+\mathcal{O}(KM\log(T)))\\&+\frac{\mathbb{E}[T_{\text{punish}}]-T_{\text{init}}-\#\text{Comm}}{M}\sum_{k\leq M}\mu_{(k)}+\mathcal{O}\left(\sum_{k>M}\frac{\log(T)}{\mu_{(M)}-\mu_k}\right).\end{aligned}$$

Using Lemma 6.21, this yields:

$$\begin{aligned}\mathbb{E}[\mathrm{Rew}_T^j(s', \boldsymbol{s}^{-\boldsymbol{j}})] \leq (T_{\text{init}} + \#\text{Comm} + t_p + \mathcal{O}(KM\log(T))) \\ + \frac{\mathbb{E}[T_{\text{punish}}] - T_{\text{init}} - \#\text{Comm}}{M} \sum_{k\leq M} \mu_{(k)} + \mathcal{O}\left(\sum_{k>M} \frac{\log(T)}{\mu_{(M)} - \mu_k}\right) \\ + \widetilde{\alpha}\frac{T - \mathbb{E}[T_{\text{punish}}]}{M} \sum_{k=1}^{M} \mu_{(k)}.\end{aligned}$$

The right term is maximized when $\mathbb{E}[T_{\text{punish}}]$ is maximized, i.e., when it is T. We then get:

$$\mathbb{E}[\mathrm{Rew}_T^j(s', \boldsymbol{s}^{-\boldsymbol{j}})] \leq \frac{T}{M} \sum_{k\leq M} \mu_{(k)} + \varepsilon,$$

where $\varepsilon = \mathcal{O}\Big(\sum_{k>M} \frac{\log(T)}{\mu_{(M)}-\mu_k} + K^2\log(T) + MK\log^2\left(\frac{\log(T)}{(\mu_{(M)}-\mu_{(M+1)})^2}\right) + \frac{K\log(T)}{(1-\widetilde{\alpha})^2\mu_{(K)}}\Big)$. □

Proof of the second point of Theorem 6.6 (stability). Define $\mathcal{E}$ the *bad event* that the initialization is not successful or that an arm is poorly estimated at some time. Let $\varepsilon' = T\mathbb{P}[\mathcal{E}] + \mathbb{E}[\#\text{Comm} \mid \neg\mathcal{E}] + K\log(T)$. Then $\varepsilon' = \mathcal{O}\left(KM\log(T) + KM\log^2\left(\frac{\log(T)}{(\mu_{(M)}-\mu_{(M+1)})^2}\right)\right)$.

Assume that the player j is playing a deviation strategy s' such that for some other player i and $l > 0$:

$$\mathbb{E}[\mathrm{Rew}_T^i(s', \boldsymbol{s}^{-\boldsymbol{j}})] \leq \mathbb{E}[\mathrm{Rew}_T^i(\boldsymbol{s})] - l - \varepsilon'$$

First fix T_{punish} the time at which the punishment protocol starts. Let us now compare s' with the individual optimal strategy for player j, s^*. Let ε' take account of the communication phases, the initialization and the low probability events.

The number of pulls by each player during exploration/exploitation is given by Equation (6.12) unless the punishment protocol is started. Moreover, the selfish player causes at most a collision during exploration/exploitation before initiating the punishment protocol, so the loss of player i before punishment is at most $1 + \varepsilon'$.

After T_{punish}, Lemma 6.21 yields that the selfish player suffers a loss at least $(1-\widetilde{\alpha})\frac{T-T_{\text{punish}}-t_p}{M}\sum_{k=1}^{M}\mu_{(k)}$, while any cooperative player suffers at most $\frac{T-T_{\text{punish}}}{M}\sum_{k=1}^{M}\mu_{(k)}$.

The selfish player then suffers after T_{punish} a loss at least $(1-\widetilde{\alpha})((l-1)-t_p)$. Define $\beta = 1-\widetilde{\alpha}$. We just showed:

$$\mathbb{E}[\mathrm{Rew}_T^i(s', \boldsymbol{s}^{-\boldsymbol{j}})] \leq \mathbb{E}[\mathrm{Rew}_T^i(\boldsymbol{s})] - l - \varepsilon' \implies \mathbb{E}[\mathrm{Rew}_T^j(s', \boldsymbol{s}^{-\boldsymbol{j}})] \leq \mathbb{E}[\mathrm{Rew}_T^j(s^*, \boldsymbol{s}^{-\boldsymbol{j}})] - \beta(l-1) + \beta t_p$$

Moreover, thanks to the second part of Theorem 6.6, $\mathbb{E}[\mathrm{Rew}_T^j(s^*, \boldsymbol{s}^{-\boldsymbol{j}})] \leq \mathbb{E}[\mathrm{Rew}_T^j(\boldsymbol{s})] + \varepsilon$

with $\varepsilon = \mathcal{O}\Big(\sum_{k>M} \frac{\log(T)}{\mu_{(M)}-\mu_k} + K^2\log(T) + MK\log^2\left(\frac{\log(T)}{(\mu_{(M)}-\mu_{(M+1)})^2}\right) + \frac{K\log(T)}{(1-\widetilde{\alpha})^2\mu_{(K)}}\Big)$. Then by defining $l_1 = l + \varepsilon'$, $\varepsilon_1 = \varepsilon + \beta t_p + \beta\varepsilon' + 1 = \mathcal{O}(\varepsilon)$, we get:

$$\mathbb{E}[\mathrm{Rew}_T^i(s', \boldsymbol{s^{-j}})] \leq \mathbb{E}[\mathrm{Rew}_T^i(\boldsymbol{s})] - l_1 \implies \mathbb{E}[\mathrm{Rew}_T^j(s', \boldsymbol{s^{-j}})] \leq \mathbb{E}[\mathrm{Rew}_T^j(\boldsymbol{s})] + \varepsilon_1 - \beta l_1.$$

□

Proof of Lemma 6.21.

The punishment protocol starts by estimating all means μ_k with a multiplicative precision of δ. This is possible thanks to Lemma 6.22, which corresponds to Theorem 9 in (Cesa-Bianchi et al., 2019a) and Lemma 13 in (Berthet and Perchet, 2017).

Lemma 6.22. *Let $X_1, \dots, X_n$ be n-i.i.d. random variables in $[0,1]$ with expectation μ and define $S_t^2 = \frac{1}{t-1}\sum_{s=1}^t (X_s - \overline{X}_t)^2$. For all $\delta \in (0,1)$, if $n \geq n_0$, where*

$$n_0 = \left\lceil \frac{2}{3\delta\mu}\log(T)\left(\sqrt{9\frac{1}{\delta^2} + 96\frac{1}{\delta} + 85} + \frac{3}{\delta} + 1\right)\right\rceil + 2 = \mathcal{O}\left(\frac{1}{\delta^2\mu}\log(T)\right)$$

and τ is the smallest time $t \in \{2, \dots, n\}$ such that

$$\delta\overline{X}_t \geq 2S_t\left(\log(T)/t\right)^{1/2} + \frac{14\log(T)}{3(t-1)},$$

then, with probability at least $1 - \frac{3}{T}$:

1. $\tau \leq n_0$,
2. $(1-\delta)\overline{X}_\tau < \mu < (1+\delta)\overline{X}_\tau$.

Proof of Lemma 6.21. The punishment protocol starts for all cooperative players at T_{punish}. For $\delta = \frac{1-\gamma}{1+3\gamma}$, each player then estimates each arm. Lemma 6.22 gives that with probability at least $1 - 3/T$:

- the estimation ends after a time at most $t_p = \mathcal{O}\left(\frac{K}{\delta^2\mu_{(K)}}\log(T)\right)$,
- $(1-\delta)\hat{\mu}_k^j \leq \mu_k \leq (1+\delta)\hat{\mu}_k^j$.

The following is conditioned on this event. The last inequality can be reversed as $\frac{\mu_k}{1+\delta} \leq \hat{\mu}_k^j \leq \frac{\mu_k}{1-\delta}$. Then, this implies for every cooperative player j

$$1 - p_k^j \leq \left(\gamma\frac{(1+\delta)\sum_{m=1}^M \mu_{(m)}}{(1-\delta)M\mu_k}\right)^{\frac{1}{M-1}}.$$

The expected reward that gets the selfish player j by pulling k after the time $T_{\text{punish}} + t_p$ is thus smaller than $\gamma \frac{1+\delta}{1-\delta} \frac{\sum_{m=1}^{M} \mu_{(m)}}{M}$.

Note that $\gamma \frac{1+\delta}{1-\delta} = \frac{1+\gamma}{2} = \widetilde{\alpha}$. Considering the low probability event given by Lemma 6.22 adds a constant term that can be counted in t_p. This finally yields the result of Lemma 6.21. □

6.D Missing elements for RSD-GT

6.D.1 Description of the algorithm

This section provides a complete description of RSD-GT. Its pseudocode is given in Algorithm 6.4. It relies on auxiliary protocols described by Algorithms 6.3 and 6.10 to 6.14.

Initialization phase. RSD-GT starts with the exact same initialization as SIC-GT, which is given by Algorithm 6.3, to estimate M and attribute ranks among the players. Afterwards, they start the exploration.

In the remaining of the algorithm, as already explained in Section 6.4.3, the time is divided into superblocks, which are divided into M blocks of length $5K + MK + M^2K$. During the j-th block of a superblock, the dictators ordering for RSD is $(j, \dots, M, 1, \dots, j-1)$. Moreover, only the j-th player can send messages during this block if she is still exploring.

Exploration. The exploiting players sequentially pull all the arms in $[K]$ to avoid collisions with any other exploring player. Yet, they still collide with exploiting players.

RSD-GT is designed so that all players know at each round the M preferred arms of all exploiting players and their order. The players thus know which arms are occupied by the exploiting players during a block j. The communication arm is thus a common arm unoccupied by any exploiting player. When an exploring player encounters a collision on this arm at the beginning of the block, this means that another player signaled the start of a communication block. In that case, the exploring player starts Listen, described by Algorithm 6.11, to receive the messages of the communicating player.

On the other hand, when an exploring player j knows her M preferred arms and their order, she waits for the next block j to initiate communication. She then proceeds to SignalPreferences, given by Algorithm 6.13.

Communication block. In a communication block, the communicating player first collides with each exploiting and exploring player to signal them the start of a communication block as described by Algorithm 6.12. These collisions need to be done in a particular way given by

Algorithm 6.4: `RSD-GT`

Input: T, δ

$\hat{M}, j \leftarrow$ `Initialize` (T, K); state $\leftarrow$ "exploring" and blocknumber $\leftarrow 1$
Let $\boldsymbol{\pi}$ be a $M \times M$ matrix with only 0 `// $\pi^j_k$ is the $k$-th preferred arm by $j$`
while $t < T$ **do**
  blocktime $\leftarrow t \pmod{5K + MK + M^2K} + 1$
  if *blocktime* $= 1$ **then** `// new block`
    blocknumber $\leftarrow$ blocknumber $\pmod{M} + 1$; $b^j_k(t) \leftarrow \sqrt{2\log(T)/N^j_k(t)}$
    Let λ^j be the ordering of the empirical means: $\hat{\mu}^j_{\lambda^j_k}(t) \geq \hat{\mu}^j_{\lambda^j_{k+1}}(t)$ for each k
    if *(blocknumber, state)* $=$ *(*j, *"exploring") and* $\forall k \in [M], \hat{\mu}^j_{\lambda^j_k} - b^j_{\lambda^j_k} \geq \hat{\mu}^j_{\lambda^j_{k+1}} + b^j_{\lambda^j_{k+1}}$ **then** $\pi^j \leftarrow \lambda^j$; state $\leftarrow$ `SignalPreferences` $(\boldsymbol{\pi}, j)$ `// send Top-M arms`
  end
  $(l,$ comm_arm$) \leftarrow$ `ComputeRSD` $(\boldsymbol{\pi},$ blocknumber$)$ `// $j$ pulls $l^j$`
  if *state* $=$ *"exploring"* **then**
    Pull l^j and update $\hat{\mu}^j_{l^j}$
    if $l^j =$ *comm_arm and* $\eta_{l^j} = 1$ **then** `// received signal`
      if *blocktime* $> 4K$ **then** state $\leftarrow$ "punishing"
      else (state, $\pi^{\text{blocknumber}}$) $\leftarrow$ `Listen` (blocknumber, state, $\boldsymbol{\pi}$, comm_arm)
  end
  if *state* $=$ *"exploiting" and* $\exists i, k$ *such that* $\pi^i_k = 0$ **then**
    Pull l^j `// arm attributed by RSD algo`
    if $l^j \notin \{l^i | i \in [M] \setminus \{j\}\}$ *and* $\eta_{l^j}(t) = 1$ **then** `// received signal`
      if *blocktime* $> 4K$ **then** state $\leftarrow$ "punishing"
      else (state, $\pi^{\text{blocknumber}}$) $\leftarrow$ `Listen` (blocknumber, state, $\boldsymbol{\pi}$, comm_arm)
  end
  if *state* $=$ *"exploiting" and* $\forall i, k, \pi^i_k \neq 0$ **then** `// all players are exploiting`
    Draw inspect $\sim$ Bernoulli$(\sqrt{\log(T)}/T)$
    if *inspect* $= 1$ **then** `// random inspection`
      Pull l^i with i chosen uniformly at random among the other players
      if $\eta_{l^i} = 0$ **then** state $\leftarrow$ "punishing" `// lying player`
    else
      Pull l^j; **if** *observed two collisions in a row* **then** state $\leftarrow$ "punishing"
  end
  if *state* $=$ *"punishing"* **then** `PunishSemiHetero` (δ)
end

`SendBit` so that all players correctly detect the start of a communication block. These players then repeat this signal to ensure that every player is listening.

Protocol 6.10: `ComputeRSD`

Input: $\boldsymbol{\pi}$, blocknumber
taken_arms $\leftarrow \emptyset$
for $s = 0, \dots, M-1$ **do**
  dict $\leftarrow s +$ blocknumber $- 1 (\text{mod } M) + 1$ `// current dictator`
  $p \leftarrow \min\{p' \in [M] \mid \pi_{p'}^{\text{dict}} \notin$ taken_arms$\}$ `// best available choice`
  if $\pi_p^{dict} \neq 0$ **then** $l^{\text{dict}} \leftarrow \pi_p^{\text{dict}}$ and add π_p^{dict} to taken_arms
  else $l^{\text{dict}} \leftarrow t +$ dict $(\text{mod } K) + 1$ `// explore`
end
comm_arm $\leftarrow \min[K] \setminus$ taken_arms
return $(l,$ comm_arm$)$

The communicating player then sends to all players her M preferred arms in order of preferences. Afterwards, each player repeats this list to ensure that no malicious player interfered during communication. As soon as some malicious behavior is observed, the start of `PunishSemiHetero`, given by Protocol 6.14, is signaled to all players.

Exploitation. An exploiting player starts each block j by computing the attribution of the RSD algorithm between the exploiting players given their known preferences and the dictatorship ordering $(j, \dots, j-1)$. She then pulls her attributed arm for the whole block, unless she receives a signal.

A signal is received when she collides with an exploring player, while unintended[9]. If it is at the beginning of a block, it means that a communication block starts. Otherwise, she just enters the punishment protocol. Note that the punishment protocol starts by signaling the start of `PunishSemiHetero` to ensure that every cooperative player starts punishing.

Another security is required to ensure that the selfish player truthfully reports her preferences. She could otherwise report fake preferences to decrease another player's utility while her best arm remains uncontested and thus available. To avoid this, `RSD-GT` uses *random inspections* when all players are exploiting. With probability $\sqrt{\log(T)}/T$ at each round, each player checks that some other player is indeed exploiting the arm she is attributed by the RSD algorithm. If it is not the case, the inspecting player signals the start of `PunishSemiHetero` to everyone by colliding twice with everybody, since a single collision could be a random inspection. Because of this, the selfish player can not pull another arm than the attributed one too often without starting a punishment scheme. Thus, if she did not report her preferences truthfully, this also has a cost for her.

[9] She normally collides with exploring players. Yet as she knows the set of exploring players, she exactly knows when this happens.

Protocol 6.11: `Listen`

Input: blocknumber, state, $\boldsymbol{\pi}$, arm_comm

ExploitPlayers $= \{i \in [M] \mid \pi_1^i \neq 0\}$; $\lambda \leftarrow \pi^{\text{blocknumber}}$
if $\lambda_1 \neq 0$ **then** state $\leftarrow$ "punishing" `// this player already sent`
while *blocktime* $\leq 2K$ **do** Pull $t + j(\text{mod } K) + 1$
if *blocktime* $= 2K$ **then** `SendBit` (comm_arm, ExploitPlayers, j) `// repeat signal`
else while *blocktime* $\leq 4K$ **do** Pull $t + j(\text{mod } K) + 1$

for K *rounds* **do**
  if *state* $=$ *"punishing"* **then** Pull j `// signal punishment`
  else
    Pull $k = t + j(\text{mod } K) + 1$; **if** $\eta_k = 1$ **then** state $\leftarrow$ "punishing"
end

for $n = 1, \dots, MK$ **do** `// receive preferences`
  Pull $k = t + j(\text{mod } K) + 1$
  $m \leftarrow \lceil n/K \rceil$ `// communicating player sends her` m`-th pref. arm`
  if $\eta_k = 1$ **then**
    if $\lambda_m \neq 0$ **then** state $\leftarrow$ "punishing" `// received two signals`
    else $\lambda_m \leftarrow k$
end

for $n = 1, \dots, M^2K$ **do** `// repetition block`
  $m \leftarrow \left\lceil \frac{n \ (\text{mod } MK)}{K} \right\rceil$ and $l \leftarrow \lceil \frac{n}{MK} \rceil$ `//` l `repeats the` m`-th pref.`
  if $j = l$ **then** Pull λ_m
  else
    Pull $k = t + j \ (\text{mod } K) + 1$
    if $\eta_k = 1$ *and* $\lambda_m \neq k$ **then** state $\leftarrow$ "punishing" `// info differs`
end
if $\#\{\lambda_m \neq 0 \mid m \in [M]\} \neq M$ **then** state $\leftarrow$ "punishing" `// did not send all`
return (state, λ)

6.D.2 Regret analysis

This section aims at proving the first point of Theorem 6.7. `RSD-GT` uses the exact same initialization phase as `SIC-GT`, and its guarantees are thus given by Lemma 6.14. Here again, the regret is decomposed into three parts:

$$R^{\text{RSD}}(T) = \mathbb{E}[R^{\text{init}} + R^{\text{comm}} + R^{\text{explo}}], \tag{6.13}$$

Protocol 6.12: `SendBit`

Input: comm_arm, ExploitPlayers, j

if *ExploitPlayers* $= \emptyset$ **then** $\widetilde{j} \leftarrow j$
else $\widetilde{j} \leftarrow \min$ ExploitPlayers
for K *rounds* **do** Pull $t + \widetilde{j} (\text{mod } K) + 1$ `// send bit to exploiting players`
for K *rounds* **do** Pull comm_arm `// send bit to exploring players`

Protocol 6.13: `SignalPreferences`

Input: $\boldsymbol{\pi}, j,$ comm_arm

ExploitPlayers $= \{i \in [M] \setminus \{j\} \mid \pi^i_1 \neq 0\}$; $\lambda \leftarrow \pi^j$ `// λ is signal to send`
state $\leftarrow$ "exploiting" `// state after the protocol`
`SendBit` (comm_arm, ExploitPlayers, j) `// initiate communication block`
for $2K$ *rounds* **do** Pull $t + j (\text{mod } K) + 1$ `// wait for repetition`
for K *rounds* **do** `// receive punish signal`
  Pull $t + j (\text{mod } K) + 1$; **if** $\eta_k = 1$ **then** state $\leftarrow$ "punishing"
end
for $n = 1, \dots, MK$ **do** pull $\lambda_{\lceil \frac{n}{K} \rceil}$ `// send k-th preferred arm`
for $n = 1, \dots, M^2 K$ **do** `// repetition block`
  $m \leftarrow \left\lceil \frac{n \ (\text{mod } MK)}{K} \right\rceil$ and $l \leftarrow \lceil \frac{n}{MK} \rceil$ `// l repeats the m-th pref.`
  if $j = l$ **then** Pull λ_m
  else
    Pull $k = t + j \ (\text{mod } K) + 1$
    if $\eta_k = 1$ *and* $\lambda_m \neq k$ **then** state $\leftarrow$ "punishing" `// info differs`
end
return state

where

$$R^{\text{init}} = T_{\text{init}} \mathbb{E}_{\sigma \sim \mathcal{U}(\mathfrak{S}_M)} \left[\sum_{k=1}^{M} \mu^{\sigma(k)}_{\pi_\sigma(k)} \right] - \sum_{t=1}^{T_{\text{init}}} \sum_{j=1}^{M} \mu^j_{\pi^j(t)} (1 - \eta^j(t)) \text{ with } T_{\text{init}} = (12eK^2 + K) \log(T),$$

$$R^{\text{comm}} = \#\text{Comm} \mathbb{E}_{\sigma \sim \mathcal{U}(\mathfrak{S}_M)} \left[\sum_{k=1}^{M} \mu^{\sigma(k)}_{\pi_\sigma(k)} \right] - \sum_{t \in \text{Comm}} \sum_{j=1}^{M} \mu^j_{\pi^j(t)} (1 - \eta^j(t)),$$

$$R^{\text{explo}} = \#\text{Explo} \mathbb{E}_{\sigma \sim \mathcal{U}(\mathfrak{S}_M)} \left[\sum_{k=1}^{M} \mu^{\sigma(k)}_{\pi_\sigma(k)} \right] - \sum_{t \in \text{Explo}} \sum_{j=1}^{M} \mu^j_{\pi^j(t)} (1 - \eta^j(t))$$

with Comm defined as all the rounds of a block where at least a cooperative player uses `Listen` protocol and Explo $= \{T_{\text{init}} + 1, \dots, T\} \setminus$ Comm. In case of a successful initialization, a single player can only initiate a communication block once without starting a punishment protocol.

Protocol 6.14: `PunishSemiHetero`

Input: δ

if *ExploitPlayers* $= [M]$ **then** collide with each player twice
else `// signal punishment during rounds` $3K+1,\dots,5K$ `of a block`
  for $3K$ *rounds* **do** Pull $t + j \pmod K + 1$
  `SendBit` (comm_arm, ExploitPlayers, j)
end
$\alpha \leftarrow \left(\frac{1+\delta}{1-\delta}\right)^2 (1-1/K)^{M-1}$ and $\delta' = \frac{1-\alpha}{1+3\alpha}$
Set $\hat{\mu}_k^j, S_k^j, v_k^j, n_k^j \leftarrow 0$
while $\exists k \in [K], \delta' \hat{\mu}_k^j < 2s_k^j (\log(T)/n_k^j)^{1/2} + \frac{14\log(T)}{3(n_k^j - 1)}$ **do** `// estimate` μ_k^j
  Pull $k = t + j \pmod K + 1$
  if $\delta' \hat{\mu}_k^j < 2s_k^j (\log(T)/n_k^j)^{1/2} + \frac{14\log(T)}{3(n_k^j - 1)}$ **then**
    Update $\hat{\mu}_k^j \leftarrow \frac{n_k^j}{n_k^j+1}\hat{\mu}_k^j + X_k(t)$ and $n_k^j \leftarrow n_k^j + 1$
    Update $S_k^j \leftarrow S_k^j + (X_k)^2$ and $s_k^j \leftarrow \sqrt{\frac{S_k^j - (\hat{\mu}_k^j)^2}{n_k^j - 1}}$
end
$p_k \leftarrow \left(1 - \left(\alpha \frac{\sum_{l=1}^M \hat{\mu}_{(l)}^j(t)}{M\hat{\mu}_k^j(t)}\right)^{\frac{1}{M-1}}\right)_+$; $\widetilde{p}_k \leftarrow p_k / \sum_{l=1}^K p_l$ `// renormalize`
while $t \leq T$ **do** Pull k with probability p_k `// punish`

Thus, as long as no punishment protocol is started: $\#\text{Comm} \leq M(5K + MK + M^2K) = \mathcal{O}(M^3K)$.

Denote by $\Delta^j = \min_{k\in[M]} \mu_{(k)}^j - \mu_{(k+1)}^j$ the level of precision required for player j to know her M preferred arms and their order. Proposition 6.2 gives the exploration time required for every player j:

Proposition 6.2. *With probability* $1 - \mathcal{O}\left(\frac{K}{T}\right)$ *and as long as no punishment protocol is started, the player* j *starts exploiting after at most* $\mathcal{O}\left(\frac{K\log(T)}{(\Delta^j)^2} + M^3K\right)$ *exploration pulls.*

Proof. In the following, the initialization is assumed to be successful, which happens with probability $1 - \mathcal{O}\left(\frac{M}{T}\right)$. Moreover, Hoeffding inequality yields:

$$\mathbb{P}\left[\forall t \leq T, \left|\hat{\mu}_k^j(t) - \mu_k^j(t)\right| \geq \sqrt{\frac{2\log(T)}{N_k^j(t)}}\right] \leq \frac{2}{T}$$

where $N_k^j(t)$ is the number of exploratory pulls on arm k by player j. With probability $1 - \mathcal{O}\left(\frac{K}{T}\right)$, player j then correctly estimates all arms at each round. The remaining of the proof is conditioned on this event.

During the exploration, player j sequentially pulls the arms in $[K]$. Denote by n the smallest integer such that $\sqrt{\frac{2\log(T)}{n}} \leq 4\Delta^j$. It directly comes that $n = \mathcal{O}\left(\frac{\log(T)}{(\Delta^j)^2}\right)$. Under the considered events, player j then has determined her M preferred arms and their order after Kn exploratory pulls. Moreover, she needs at most M blocks before being able to initiate her communication block and starts exploiting. Thus, she needs at most $\mathcal{O}\left(\frac{K\log(T)}{(\Delta^j)^2} + M^3K\right)$ exploratory pulls, leading to Proposition 6.2. □

Proof of the first point of Theorem 6.7. Assume all players play `RSD-GT`. Simply by bounding the size of the initialization and the communication phases, it comes:

$$R^{\text{init}} + R^{\text{comm}} \leq \mathcal{O}\left(MK^2\log(T)\right).$$

Proposition 6.2 yields that with probability $1 - \mathcal{O}\left(\frac{KM}{T}\right)$, all players start exploitation after at most $\mathcal{O}\left(\frac{K\log(T)}{\Delta^2}\right)$ exploratory pulls.

For $p = \sqrt{\log(T)}/T$, with probability $\mathcal{O}(p^2M)$ at any round t, a player is inspecting another player who is also inspecting or a player receives two consecutive inspections. These are the only ways to start punishing when all players are cooperative. As a consequence, when all players follow `RSD-GT`, they initiate the punishment protocol with probability $\mathcal{O}\left(p^2MT\right)$. Finally, the total regret due to this event grows as $\mathcal{O}\left(M^2\log(T)\right)$.

If the punishment protocol is not initiated, players cycle through the RSD matchings of $\sigma\circ\sigma_0^{-1}, \dots, \sigma\circ\sigma_0^{-M}$ where σ_0 is the classical M-cycle and σ is the players permutation returned by the initialization. Define $U(\sigma) = \sum_{k=1}^{M} \mu_{\pi_\sigma(k)}^{\sigma(k)}$, where $\pi_\sigma(k)$ is the arm attributed to the k-th dictator, $\sigma(k)$, as defined in Section 6.3.2. $U(\sigma)$ is the social welfare of RSD algorithm when the dictatorships order is given by the permutation σ. As players all follow `RSD-GT` here, σ is chosen uniformly at random in $\mathfrak{S}_M$ and any $\sigma\circ\sigma_0^{-k}$ as well. Then

$$\mathbb{E}_{\sigma\sim\mathcal{U}(\mathfrak{S}_M)}\left[\frac{1}{M}\sum_{k=1}^{M} U(\sigma\circ\sigma_0^{-M})\right] = \mathbb{E}_{\sigma\sim\mathcal{U}(\mathfrak{S}_M)}\left[U(\sigma)\right].$$

This means that in expectation, the utility given by the exploitation phase is the same as the utility of the RSD algorithm when choosing a permutation uniformly at random. Considering the low probability event of a punishment protocol, an unsuccesful initialization or a bad estimation of an arm finally yields:

$$R^{\text{explo}} \leq \mathcal{O}\left(\frac{MK\log(T)}{\Delta^2}\right).$$

Equation (6.13) concludes the proof. □

6.D.3 Selfish-robustness of `RSD-GT`

In this section, we prove the two last points of Theorem 6.7. Three auxiliary Lemmas are first needed. They are proved in Section 6.D.3.

1. Lemma 6.23 compares the utility received by player j from the RSD algorithm with the utility given by sequentially pulling her M best arms in the δ-heterogeneous setting.

2. Lemma 6.24 gives an equivalent version of Lemma 6.21, but for the δ-heterogeneous setting.

3. Lemma 6.25 states that the expected utility of the assignment of any player during the exploitation phase does not depend on the strategy of the selfish player. The intuition behind this result is already given in Section 6.4.3.

   In the case of several selfish players, they could actually fix the joint distribution of $(\sigma^{-1}(j), \sigma^{-1}(j'))$. A simple rotation with a M-cycle is then not enough to recover a uniform distribution over $\mathfrak{S}_M$ in average. A more complex rotation is then required and the dependence in M would blow up with the number of selfish players.

Lemma 6.23. *In the δ-heterogeneous case for every player j and permutation σ:*

$$\frac{1}{M}\sum_{k=1}^{M}\mu^j_{(k)} \leq \widetilde{U}_j(\sigma) \leq \frac{(1+\delta)^2}{(1-\delta)^2 M}\sum_{k=1}^{M}\mu^j_{(k)},$$

where $\widetilde{U}_j(\sigma) \coloneqq \frac{1}{M}\sum_{k=1}^{M}\mu^j_{\pi_{\sigma\circ\sigma_0^{-k}}\left(\sigma_0^k\circ\sigma^{-1}(j)\right)}$.

Following the notation of Section 6.3.2, $\pi_\sigma(\sigma^{-1}(j))$ is the arm attributed to player j by RSD when the dictatorship order is given by σ. $\widetilde{U}_j(\sigma)$ is then the average utility of the exploitation when σ is the permutation given by the initialization.

Lemma 6.24. *Recall that $\gamma = (1-1/K)^{M-1}$. In the δ-heterogeneous setting with $\delta < \frac{1-\sqrt{\gamma}}{1+\sqrt{\gamma}}$, if the punish protocol is started at time $T_{\textit{punish}}$ by $M-1$ players, then for the remaining player j, independently of her sampling strategy:*

$$\mathbb{E}[\mathrm{Rew}^j_T|\textit{punishment}] \leq \mathbb{E}[\mathrm{Rew}^j_{T_{\textit{punish}}+t_p}] + \widetilde{\alpha}\frac{T - T_{\textit{punish}} - t_p}{M}\sum_{k=1}^{M}\mu^j_{(k)},$$

with $t_p = \mathcal{O}\left(\frac{K\log(T)}{(1-\delta)(1-\widetilde{\alpha})^2\mu_{(K)}}\right)$ *and* $\widetilde{\alpha} = \frac{1+\left(\frac{1+\delta}{1-\delta}\right)^2\gamma}{2}$.

Lemma 6.25. *The initialization phase is successful when all players end with different ranks in* $[M]$*. For each player* j*, independently of the behavior of the selfish player:*

$$\mathbb{E}_{\sigma\sim\textit{successful initialization}}\left[\widetilde{U}_j(\sigma)\right] = \mathbb{E}_{\sigma\sim\mathcal{U}(\mathfrak{S}_M)}\left[\mu^j_{\pi_\sigma(\sigma^{-1}(j))}\right].$$

where $\widetilde{U}_j(\sigma)$ *is defined as in Lemma 6.23 above.*

Proof of the second point of Theorem 6.7 (Nash equilibrium). First fix T_{punish} the beginning of the punishment protocol. Note s the profile where all players follow `RSD-GT` and s' the individual strategy of the selfish player j.

As in the homogeneous case, the player earns at most $T_{\text{init}} + \#\text{Comm}$ during both initialization and communication. She can indeed choose her rank at the end of the initialization, but this has no impact on the remaining of the algorithm (except for a M^3K term due to the length of the last uncompleted superblock), thanks to Lemma 6.25.

With probability $1-\mathcal{O}\left(\frac{KM+M\log(T)}{T}\right)$, the initialization is successful, the arms are correctly estimated and no punishment protocol is due to unfortunate inspections (as already explained in Section 6.D.2). The following is conditioned on this event.

Proposition 6.2 holds independently of the strategy of the selfish player. Moreover, the exploiting players run the RSD algorithm only between the exploiters. This means that when all cooperative players are exploiting, if the selfish player did not signal her preferences, she would always be the last dictator in the RSD algorithm. Because of this, it is in her interest to report as soon as possible her preferences.

Moreover, reporting truthfully is a dominant strategy for the RSD algorithm, meaning that when all players are exploiting, the expected utility received by the selfish player is at most the utility she would get by reporting truthfully. As a consequence, the selfish player can improve her expected reward by at most the length of a superblock during the exploitation phase. Wrapping up all of this and defining t_0 the time at which all other players start exploiting:

$$\mathbb{E}\left[\mathrm{Rew}^j_{T_{\text{punish}}+t_p}(s', \boldsymbol{s^{-j}})\right] \leq t_0 + (T_{\text{punish}} + t_p - t_0)\mathbb{E}_{\sigma\sim\mathcal{U}(\mathfrak{S}_M)}\left[\mu^j_{\pi_\sigma(\sigma^{-1}(j))}\right] + \mathcal{O}(M^3K).$$

with $t_0 = \mathcal{O}\left(\frac{K\log(T)}{\Delta^2} + K^2\log(T)\right)$. Lemma 6.24 then yields for $\widetilde{\alpha} = \frac{1+\left(\frac{1+\delta}{1-\delta}\right)^2\alpha}{2}$:

$$\mathbb{E}\left[\mathrm{Rew}^j_T(s', \boldsymbol{s^{-j}})\right] \leq t_0 + (T_{\text{punish}}+t_p-t_0)\mathbb{E}_{\sigma\sim\mathcal{U}(\mathfrak{S}_M)}\left[\mu^j_{\pi_\sigma(\sigma^{-1}(j))}\right] + \widetilde{\alpha}\frac{T - T_{\text{punish}} - t_p}{M}\sum_{k=1}^{M}\mu^j_{(k)} + \mathcal{O}(M^3K).$$

Thanks to Lemma 6.23, $\mathbb{E}_{\sigma\sim\mathcal{U}(\mathfrak{S}_M)}\left[\mu^j_{\pi_\sigma(\sigma^{-1}(j))}\right] \geq \frac{\sum_{k=1}^{M}\mu^j_{(k)}}{M}$. We assume $\delta < \frac{1-(1-1/K)^{\frac{M-1}{2}}}{1+(1-1/K)^{\frac{M-1}{2}}}$ here, so that $\widetilde{\alpha} < 1$. Because of this, the right term is maximized when T_{punish} is maximized, i.e., equal to T. Then:

$$\mathbb{E}\left[\mathrm{Rew}_T^j(s', \boldsymbol{s}^{-\boldsymbol{j}})\right] \leq T\mathbb{E}_{\sigma\sim\mathcal{U}(\mathfrak{S}_M)}\left[\mu^j_{\pi_\sigma(\sigma^{-1}(j))}\right] + t_0 + t_p + \mathcal{O}(M^3K).$$

Using the first point of Theorem 6.7 to compare $T\mathbb{E}_{\sigma\sim\mathcal{U}(\mathfrak{S}_M)}\left[\mu^j_{\pi_\sigma(\sigma^{-1}(j))}\right]$ with $\mathrm{Rew}_T^j(\boldsymbol{s})$ and adding the low probability event then yields the first point of Theorem 6.7. □

Proof of the second point of Theorem 6.7 (stability). For $p_0 = \mathcal{O}\left(\frac{KM+M\log(T)}{T}\right)$, with probability at least $1-p_0$, the initialization is successful, the cooperative players start exploiting with correct estimated preferences after a time at most $t_0 = \mathcal{O}\left(K^2\log(T) + \frac{K\log(T)}{\Delta^2}\right)$ and no punishment protocol is started due to unfortunate inspections. Define $\varepsilon' = t_0 + Tp_0 + 7M^3K$. Assume that the player j is playing a deviation strategy s' such that for some i and $l > 0$:

$$\mathbb{E}\left[\mathrm{Rew}_T^i(s', \boldsymbol{s}^{-\boldsymbol{j}})\right] \leq \mathbb{E}\left[\mathrm{Rew}_T^i(\boldsymbol{s})\right] - l - \varepsilon'$$

First, let us fix σ the permutation returned by the initialization, T_{punish} the time at which the punishment protocol starts and divide $l = l_{\text{before punishment}} + l_{\text{after punishment}}$ in two terms: the regret incurred before the punishment protocol and the regret after. Let us now compare s' with s^*, the optimal strategy for player j. Let ε take account of the low probability event of a bad initialization/exploration, the last superblock that remains uncompleted, the time before all cooperative players start the exploitation and the event that a punishment accidentally starts. Thus the only way for player i to suffer some additional regret before punishment is to lose it during a completed superblock of the exploitation. Three cases are possible:

1. The selfish player truthfully reports her preferences. The average utility of player i during the exploitation is then $\widetilde{U}_i(\sigma)$ as defined in Lemma 6.25. The only way to incur some additional loss to player i before the punishment is then to collide with her, in which case her loss is at most $(1+\delta)\mu_{(1)}$ while the selfish player's loss is at least $(1-\delta)\mu_{(M)}$.

After T_{punish}, Lemma 6.24 yields that the selfish player suffers a loss at least $(1-\widetilde{\alpha})\frac{T-T_{\text{punish}}-t_p}{M}\sum_{k=1}^M \mu^j_{(k)}$, while any cooperative player i suffers a loss at most $(T-T_{\text{punish}})\widetilde{U}_i(\sigma)$. Thanks to Lemma 6.23 and the δ-heterogeneity assumption, this term is smaller than $\frac{T-T_{punish}}{M}\left(\frac{1+\delta}{1-\delta}\right)^3\sum_{k=1}^M \mu^j_{(k)}$.

Then, the selfish player after T_{punish} suffers a loss at least $\frac{(1-\widetilde{\alpha})(1-\delta)^3}{(1+\delta)^3}l_{\text{after punish}} - t_p$.

In the first case, we thus have for $\beta = \min(\frac{(1-\widetilde{\alpha})(1-\delta)^3}{(1+\delta)^3}, \frac{(1-\delta)\mu_{(M)}}{(1+\delta)\mu_{(1)}})$:

$$\mathbb{E}[\mathrm{Rew}_T^j(s', \boldsymbol{s}^{-\boldsymbol{j}})|\sigma] \leq \mathbb{E}[\mathrm{Rew}_T^j(s^*, \boldsymbol{s}^{-\boldsymbol{j}})|\sigma] - \beta l + t_p.$$

2. The selfish player never reports her preferences. In this case, it is obvious that the utility returned by the assignments to any other player is better than if the selfish player reports truthfully.

Then the only way to incur some additional loss to player i before punishment is to collide with her, still leading to a ratio of loss at most $\frac{\mu^j_{(M)}}{\mu^i_{(1)}}$.

From there, it can be concluded as in the first case that for $\beta = \min(\frac{(1-\widetilde{\alpha})(1-\delta)^3}{(1+\delta)^3}, \frac{(1-\delta)\mu_{(M)}}{(1+\delta)\mu_{(1)}})$:

$$\mathbb{E}[\mathrm{Rew}^j_T(s', s_{-j})|\sigma] \leq \mathbb{E}[\mathrm{Rew}^j_T(s^*, s_{-j})|\sigma] - \beta l + t_p.$$

3. The selfish player reported fake preferences. If these fake preferences never change the issue of the `ComputeRSD` protocol, this does not change from the first case. Otherwise, for any block where the final assignment is changed, the selfish player does not receive the arm she would get if she reported truthfully. Denote by n the number of such blocks, by N_{lie} the number of times player j did not pull the arm attributed by `ComputeRSD` during such a block before T_{punish} and by l_b the loss incurred to player i on the other blocks.

As for the previous cases, the loss incurred by the selfish player during the blocks where the assignment of `ComputeRSD` is unchanged is at least $\frac{(1-\delta)\mu_{(M)}}{(1+\delta)\mu_{(1)}} l_b$.

Each time the selfish player pulls the attributed arm by `ComputeRSD` in a block where the assignment is changed, she suffers a loss at least Δ. The total loss for the selfish player is then (w.r.t. the optimal strategy s^*) at least:

$$(1-\widetilde{\alpha})\frac{T - T_{\text{punish}} - t_p}{M}\sum_{k=1}^{M}\mu^j_{(k)} + \left(\frac{n}{M}\left(T_{\text{punish}} - t_0\right) - N_{\text{lie}}\right)\Delta + \frac{(1-\delta)\mu_{(M)}}{(1+\delta)\mu_{(1)}} l_b.$$

On the other hand, the loss for a cooperative player is at most:

$$\frac{T - T_{\text{punish}}}{M}\left(\frac{1+\delta}{1-\delta}\right)^3\sum_{k=1}^{M}\mu^j_{(k)} + \frac{n}{M}(T_{\text{punish}} - t_0)(1+\delta)\mu_{(1)} + l_b.$$

Moreover, each time the selfish player does not pull the attributed arm by `ComputeRSD`, she has a probability $\widetilde{p} = 1 - (1 - \frac{p}{M-1})^{M-1} \geq \frac{p}{2}$ for $p = \frac{\sqrt{\log(T)}}{T}$, to receive a random inspection and thus to trigger the punishment protocol. Because of this, N_{lie} follows a geometric distribution of parameter $\widetilde{p}$ and $\mathbb{E}[N_{\text{lie}}] \leq \frac{2}{p}$.

When taking the expectations over T_{punish} and N_{lie}, but still fixing σ and n, we get:

$$l_{\text{selfish}} \geq (1-\widetilde{\alpha})\frac{T - \mathbb{E}[T_{\text{punish}}] - t_p}{M}\sum_{k=1}^{M}\mu^j_{(k)} + \left(\frac{n}{M}\left(\mathbb{E}[T_{\text{punish}}] - t_0\right) - 2/p\right)\Delta + \frac{(1-\delta)\mu_{(M)}}{(1+\delta)\mu_{(1)}} l_b,$$

$$l \leq \frac{T - \mathbb{E}[T_{\text{punish}}]}{M} \left(\frac{1+\delta}{1-\delta}\right)^3 \sum_{k=1}^{M} \mu^j_{(k)} + \frac{n}{M}(\mathbb{E}[T_{\text{punish}}] - t_0)(1+\delta)\mu_{(1)} + l_b.$$

First assume that $\frac{n}{M}(\mathbb{E}[T_{\text{punish}}] - t_0) \geq \frac{4}{p}$. In that case, we get:

$$l_{\text{selfish}} \geq (1-\widetilde{\alpha})\frac{T - \mathbb{E}[T_{\text{punish}}] - t_p}{M} \sum_{k=1}^{M} \mu^j_{(k)} + \frac{n}{2M}(\mathbb{E}[T_{\text{punish}}] - t_0)\Delta + \frac{(1-\delta)\mu_{(M)}}{(1+\delta)\mu_{(1)}} l_b,$$
$$l \leq \frac{T - \mathbb{E}[T_{\text{punish}}]}{M} \left(\frac{1+\delta}{1-\delta}\right)^3 \sum_{k=1}^{M} \mu^j_{(k)} + \frac{n}{M}(\mathbb{E}[T_{\text{punish}}] - t_0)(1+\delta)\mu_{(1)} + l_b.$$

In the other case, we have by noting that $(1+\delta)\mu_{(1)} \leq \frac{1+\delta}{1-\delta}\sum_{k=1}^{M} \mu^j_{(k)}$:

$$l_{\text{selfish}} \geq (1-\widetilde{\alpha})T\left(1 - \frac{4M}{\sqrt{\log(T)}} - t_p\right)\frac{1}{M}\sum_{k=1}^{M} \mu^j_{(k)} + \frac{(1-\delta)\mu_{(M)}}{(1+\delta)\mu_{(1)}} l_b,$$
$$l \leq T\left(1 + \frac{4M}{\sqrt{\log(T)}}\right)\frac{1}{M}\left(\frac{1+\delta}{1-\delta}\right)^3 \sum_{k=1}^{M} \mu^j_{(k)} + l_b.$$

In both of these two cases, for $\widetilde{\beta} = \min\left((1-\widetilde{\alpha})\left(\frac{1+\delta}{1-\delta}\right)^3 \frac{\sqrt{\log(T)}-4M}{\sqrt{\log(T)}+4M}; \frac{\Delta}{(1+\delta)\mu_{(1)}}; \frac{(1-\delta)\mu_{(M)}}{(1+\delta)\mu_{(1)}}\right)$:

$$l_{\text{selfish}} \geq \widetilde{\beta} l - t_p$$

Let us now gather all the cases. When taking the previous results in expectation over σ, this yields for the previous definition of $\widetilde{\beta}$:

$$\mathbb{E}[\text{Rew}^i_T(s', \boldsymbol{s^{-j}})] \leq \mathbb{E}[\text{Rew}^i_T(\boldsymbol{s})] - l - \varepsilon' \implies \mathbb{E}[\text{Rew}^j_T(s', \boldsymbol{s^{-j}})] \leq \mathbb{E}[\text{Rew}^j_T(s^*, \boldsymbol{s^{-j}})] - \widetilde{\beta} l + t_p + t_0.$$

Moreover, thanks to the second part of Theorem 6.7, $\mathbb{E}[\text{Rew}^j_T(s^*, \boldsymbol{s^{-j}})] \leq \mathbb{E}[\text{Rew}^j_T(\boldsymbol{s})] + \varepsilon$, with $\varepsilon = \mathcal{O}\left(\frac{K\log(T)}{\Delta^2} + K^2\log(T) + \frac{K\log(T)}{(1-\delta)r^2\mu_{(K)}}\right)$. Then by defining $l_1 = l + \varepsilon'$, $\varepsilon_1 = \varepsilon + t_p + t_0 + \widetilde{\beta}\varepsilon' = \mathcal{O}(\varepsilon)$, we get:

$$\mathbb{E}[\text{Rew}^i_T(s', \boldsymbol{s^{-j}})] \leq \mathbb{E}[\text{Rew}^i_T(\boldsymbol{s})] - l_1 \implies \mathbb{E}[\text{Rew}^j_T(s', \boldsymbol{s^{-j}})] \leq \mathbb{E}[\text{Rew}^j_T(\boldsymbol{s})] - \widetilde{\beta} l_1 + \varepsilon_1.$$

□

Auxiliary lemmas

Proof of Lemma 6.23. Assume that player j is the k-th dictator for an RSD assignment. Since only $k-1$ arms are reserved before she chooses, she earns at least $\mu^j_{(k)}$ after this assignment. This yields the first inequality:

$$\widetilde{U}_j(\sigma) \geq \frac{\sum_{k=1}^{M} \mu^j_{(k)}}{M}$$

Still assuming that player j is the k-th dictator, let us prove that she earns at most $\left(\frac{1+\delta}{1-\delta}\right)^2 \mu^j_{(k)}$. Assume w.l.o.g. that she ends up with the arm l such that $\mu^j_l > \mu^j_{(k)}$. This means that a dictator j' before her preferred an arm i to the arm l with $\mu^j_l > \mu^j_{(k)} \geq \mu^j_i$.

Since j' preferred i to l, $\mu^{j'}_i \geq \mu^{j'}_l$. Using the δ-heterogeneity assumption, it comes:

$$\mu^j_l \leq \frac{1+\delta}{1-\delta}\mu^{j'}_l \leq \frac{1+\delta}{1-\delta}\mu^{j'}_i \leq \left(\frac{1+\delta}{1-\delta}\right)^2 \mu^j_i \leq \left(\frac{1+\delta}{1-\delta}\right)^2 \mu^j_{(k)}$$

Thus, player j earns at most $\left(\frac{1+\delta}{1-\delta}\right)^2 \mu^j_{(k)}$ after this assignment, which yields the second inequality of Lemma 6.23.

□

Proof of Lemma 6.24. The punishment protocol starts for all cooperative players at T_{punish}. Define $\alpha' = \left(\frac{1+\delta}{1-\delta}\right)^2 \gamma$ and $\delta' = \frac{1-\alpha'}{1+3\alpha'}$. The condition $r > 0$ is equivalent to $\delta' > 0$.

As in the homogeneous case, each player then estimates each arm such that after $t_p = \mathcal{O}\left(\frac{K\log(T)}{(1-\delta)\cdot(\delta')^2\mu_{(K)}}\right)$[10] rounds, $(1-\delta')\hat{\mu}^j_k \leq \mu^j_k \leq (1+\delta)\hat{\mu}^j_k$ with probability $1-\mathcal{O}(KM/T)$, thanks to Lemma 6.22. This implies that for any cooperative player j':

$$\begin{aligned} 1 - p^{j'}_k &\leq \left(\gamma \frac{(1+\delta')\sum_{m=1}^{M} \mu^{j'}_{(m)}}{(1-\delta')M\mu^{j'}_k}\right)^{\frac{1}{M-1}} \\ &\leq \left(\gamma \frac{1+\delta'}{1-\delta'}\left(\frac{1+\delta}{1-\delta}\right)^2 \frac{\sum_{m=1}^{M} \mu^j_{(m)}}{M\mu^j_k}\right)^{\frac{1}{M-1}} \end{aligned}$$

The last inequality is due to the fact that in the δ-heterogeneous setting, $\frac{\mu^j_k}{\mu^{j'}_k} \in [\left(\frac{1-\delta}{1+\delta}\right)^2, \left(\frac{1+\delta}{1-\delta}\right)^2]$. Thus, the expected reward that gets the selfish player j by pulling k after the time $T_{\text{punish}} + t_p$ is smaller than $\gamma\frac{1+\delta'}{1-\delta'}\left(\frac{1+\delta}{1-\delta}\right)^2 \frac{\sum_{m=1}^{M} \mu^j_{(m)}}{M}$.

[10] The δ-heterogeneous assumption is here used to say that $\frac{1}{\mu^j_{(K)}} \leq \frac{1}{(1-\delta)\mu_{(K)}}$.

Note that $\gamma\frac{1+\delta'}{1-\delta'}\left(\frac{1+\delta}{1-\delta}\right)^2 = \widetilde{\alpha}$. Considering the low probability event of bad estimations of the arms adds a constant term that can be counted in t_p, leading to Lemma 6.24. □

Proof of Lemma 6.25. Consider the selfish player j and denote σ the permutation given by the initialization. The rank of player j' is then $\sigma^{-1}(j')$. All other players j pull uniformly at random until having an attributed rank. Moreover, player j does not know the players with which she collides. This implies that she can not correlate her rank with the rank of a specific player, i.e., $\mathbb{P}_\sigma\left[\sigma(k') = j' | \sigma(k) = j\right]$ does not depend on j' as long as $j' \neq j$.

This directly implies that the distribution of $\sigma|\sigma(k) = j$ is uniform over $\mathfrak{S}_M^{j\to k}$. Thus, the distribution of $\sigma \circ \sigma_0^{-l}|\sigma(k) = j$ is uniform over $\mathfrak{S}_M^{j\to k+l \pmod M}$ and finally for any $j' \in [M]$:

$$\begin{aligned}
\mathbb{E}_{\sigma\sim\text{successful initialization}}\left[\frac{1}{M}\sum_{l=1}^{M}\mu^j_{\pi_{\sigma\circ\sigma_0^{-l}\left(\sigma_0^l\circ\sigma^{-1}(j)\right)}}\,\middle|\,\sigma(k)=j\right] &= \frac{1}{M}\sum_{l=1}^{M}\mathbb{E}_{\sigma\sim\mathcal{U}\left(\mathfrak{S}_M^{j\to l}\right)}\left[\mu^{j'}_{\pi_\sigma(\sigma^{-1}(j'))}\right], \\
&= \frac{1}{M}\sum_{l=1}^{M}\frac{1}{(M-1)!}\sum_{\sigma\in\mathfrak{S}_M^{j\to l}}\mu^{j'}_{\pi_\sigma(\sigma^{-1}(j'))}, \\
&= \frac{1}{M!}\sum_{\sigma\in\mathfrak{S}_M}\mu^{j'}_{\pi_\sigma(\sigma^{-1}(j'))}.
\end{aligned}$$

Taking the expectation of the left term then yields Lemma 6.25. □

Part II

Other learning instances

Chapter 7

Decentralized Learning in Online Queuing Systems

Motivated by packet routing in computer networks and resource allocation in radio networks, online queuing systems are composed of queues receiving packets at different rates. Repeatedly, they send packets to servers, each of them treating only at most one packet at a time. In the centralized case, the number of accumulated packets remains bounded (i.e., the system is *stable*) as long as the ratio between service rates and arrival rates is larger than 1. In the decentralized case, individual no-regret strategies ensures stability when this ratio is larger than 2. Yet, myopically minimizing regret disregards the long term effects due to the carryover of packets to further rounds. On the other hand, minimizing long term costs leads to stable Nash equilibria as soon as the ratio exceeds $\frac{e}{e-1}$. Stability with decentralized learning strategies with a ratio below 2 was a major remaining question. We first argue that for ratios up to 2, cooperation is required for stability of learning strategies, as selfish minimization of policy regret, a *patient* notion of regret, might indeed still be unstable in this case. We therefore consider cooperative queues and propose the first learning decentralized algorithm guaranteeing stability of the system as long as the ratio of rates is larger than 1, thus reaching performances comparable to centralized strategies.

7.1 Introduction

As explained in Chapter 2, inefficient decisions in repeated games can stem from both strategic and learning considerations. First, strategic agents selfishly maximize their own individual reward at others' expense, which is measured by the price of anarchy in the pessimistic case and the price of stability in the optimistic one.

Many related results are known in classical repeated games (see e.g., Cesa-Bianchi and Lugosi, 2006; Roughgarden, 2010), where a single game is repeated over independent rounds (but the agents strategies might evolve and depend on the history). Motivated by packet routing in computer networks, Gaitonde and Tardos (2020a) introduced a repeated game with a *carryover* feature: the outcome of a round does not only depend on the actions of the agents, but also on the previous rounds. They consider heterogeneous queues sending packets to servers. If several queues simultaneously send packets to the same server, only the oldest packet is treated by the server.

Because of this carryover effect, little is known about this type of game. In a first paper, Gaitonde and Tardos (2020a) proved that if queues follow suitable no-regret strategies, a ratio of 2 between server and arrival rates leads to stability of the system, meaning that the number of packets accumulated by each queue remains bounded. However, the assumption of regret minimization sort of reflects a myopic behavior and is not adapted to games with carryover. Gaitonde and Tardos (2020b) subsequently consider a patient game, where queues instead minimize their asymptotic number of accumulated packets. A ratio only larger than $\frac{e}{e-1}$ then guarantees the stability of the system, while a smaller ratio leads to inefficient Nash equilibria. As a consequence, going below the $\frac{e}{e-1}$ factor requires some level of cooperation between the queues. This result actually holds with perfect knowledge of the problem parameters and it remained even unknown whether decentralized learning strategies can be stable with a ratio below 2.

We first argue that decentralized queues need some level of cooperation to ensure stability with a ratio of rates below 2. Policy regret can indeed be seen as a patient alternative to the regret notion. Yet even minimizing the policy regret might lead to instability when this ratio is below 2.

An explicit decentralized cooperative algorithm called ADEQUA (A DEcentralized QUeuing Algorithm) is thus proposed. It is the first decentralized learning algorithm guaranteeing stability when this ratio is only larger than 1. ADEQUA does not require communication between the queues, but uses synchronisation between them to accurately estimate the problem parameters and avoid interference when sending packets. Our main result is given by Theorem 7.1 below, whose formal version, Theorem 7.5 in Section 7.3, also provides bounds on the number of accumulated packets.

Theorem 7.1. *If the ratio between server rates and arrival rates is larger than* 1 *and all queues follow* ADEQUA*, the system is strongly stable.*

The remaining of the chapter is organised as follows. The model and existing results are recalled in Section 7.2. Section 7.3 argues that cooperation is required to guarantee stability of learning strategies when the ratio of rates is below 2. ADEQUA is then presented in Section 7.4, along with insights for the proof of Theorem 7.1. Section 7.5 finally compares the behavior of ADEQUA with no-regret strategies on toy examples and empirically confirms the different known theoretical results.

7.1.1 Additional related work

Queuing theory includes applications in diverse areas such as computer science, engineering, operation research (Shortle et al., 2018). Borodin et al. (1996) for example use the stability theorem of Pemantle and Rosenthal (1999), which was also used by Gaitonde and Tardos (2020a), to study the problem of packet routing through a network. Our setting is the single-hop particular instance of throughput maximization in wireless networks. Motivated by resource allocation in multihop radio problem, packets can be sent through more general routing paths in the original problem. Tassiulas and Ephremides (1990) proposed a first stable centralized algorithm, when the service rates are known *a priori*. Stable decentralized algorithms were later introduced in specific cases (Neely et al., 2008; Jiang and Walrand, 2009; Shah and Shin, 2012), when the rewards $X_k(t)$ are observed before deciding which server to send the packet. The main challenge is then of coordination, where queues avoid collisions with each other. The proposed algorithms are thus not adapted to our setting, where both coordination between queues and learning the service rates are required. We refer the reader to (Georgiadis et al., 2006) for an extended survey on resource allocation in wireless networks.

Krishnasamy et al. (2016) first considered online learning for such queuing systems model, in the simple case of a single queue. It is a particular instance of stochastic multi-armed bandits, a celebrated online learning model, where the agent repeatedly takes an action within a finite

set and observes its associated reward. This model becomes intricate when considering multiple queues, as they interfere when choosing the same server. It is then related to the multiplayer bandits problem studied in Part I.

The collision model is here different as one of the players still gets a reward. It is thus even more closely related to competing bandits (Liu et al., 2020b; Liu et al., 2020a), where arms have preferences over the players and only the most preferred player pulling the arm actually gets the reward. Arm preferences are here not fixed and instead depend on the packets' ages. While collisions can be used as communication tools between players in multiplayer bandits, this becomes harder with an asymmetric collision model as in competing bandits. However, some level of communication remains possible (Sankararaman et al., 2020; Basu et al., 2021). In queuing systems, collisions are not only asymmetric, but depend on the age of the sent packets, making such solutions unsuited.

While multiplayer bandits literature considers cooperative players, Chapter 6 showed that cooperative algorithms could be made robust to selfish players. On the other hand, competing bandits consider strategic players and arms as the goal is to reach a bipartite stable matching between them. Despite being cooperative, ADEQUA also has strategic considerations as the queues' strategy converges to a correlated equilibrium of the patient game described in Section 7.2.

An additional difficulty here appears as queues are asynchronous: they are not active at each round, but only when having packets left. This is different from the classical notion of asynchronicity (Bonnefoi et al., 2017), where players are active at each round with some fixed probability. Communication schemes in multiplayer bandits rely on this synchronisation assumption. While such a level of synchronisation is not available here, some lower level is still used to avoid collisions between queues and to allow a limited exchange of information between them.

7.2 Queuing Model

We consider a queuing system composed of N queues and K servers, associated with vectors of arrival and service rates $\boldsymbol{\lambda}, \boldsymbol{\mu}$, where at each time step $t = 1, 2, \dots$, the following happens:

- each queue $i \in [N]$ receives a new packet with probability $\lambda_i \in [0, 1]$, that is marked with the timestamp of its arrival time. If the queue currently has packet(s) on hold, it sends one of them to a chosen server j based on its past observations.
- Each server $j \in [K]$ attempts to clear the oldest packet it has received, breaking ties uniformly at random. It succeeds with probability $\mu_j \in [0, 1]$ and otherwise sends it back

to its original queue, as well as all other unprocessed packets.

At each time step, a queue only observes whether or not the packet sent (if any) is cleared by the server. We note Q_t^i the number of packets in queue i at time t. Given a packet-sending dynamics, the system is **stable** if, for each i in $[N]$, Q_t^i/t converges to 0 almost surely. It is **strongly stable**, if for any $r, t \geq 0$ and $i \in [N]$, $\mathbb{E}[(Q_t^i)^r] \leq C_r$, where C_r is an arbitrarily large constant, depending on r but not t. Without ambiguity, we also say the policy or the queues are (strongly) stable. Naturally, a strongly stable system is also stable (Gaitonde and Tardos, 2020a).

Without loss of generality, we assume $K \geq N$ (otherwise, we simply add fictitious servers with 0 service rate). The key quantity of a system is its **slack**, defined as the largest real number such that:

$$\sum_{i=1}^{k} \mu_{(i)} \geq \eta \sum_{i=1}^{k} \lambda_{(i)}, \ \forall\, k \leq N.$$

We also denote by $\mathcal{P}([K])$ the set of probability distributions on $[K]$ and by Δ the **margin** of the system defined by

$$\Delta \coloneqq \min_{k \in [N]} \frac{1}{k} \sum_{i=1}^{k} (\mu_{(i)} - \lambda_{(i)}). \tag{7.1}$$

Notice that the alternative system where $\widetilde{\lambda}_i = \lambda_i + \Delta$ and $\widetilde{\mu}_k = \mu_k$ has a slack 1. In that sense, Δ is the largest *margin* between service and arrival rates that all queues can individually get in the system. Note that if $\eta > 1$, then $\Delta > 0$. We now recall existing results for this problem, summarized in Figure 7.1 below.

Theorem 7.2 (Marshall et al. 1979)**.** *For any instance, there exists a strongly stable centralized policy if and only if $\eta > 1$.*

Theorem 7.3 (Gaitonde and Tardos 2020a, informal)**.** *If $\eta > 2$, queues following appropriate no regret strategies are strongly stable.*
For each $N > 0$, there exists a system and a dynamic s.t. $2 > \eta > 2 - o(1/N)$, all queues follow appropriate no-regret strategies, but they are not strongly stable.

In the above theorem, an *appropriate no regret strategy* is a strategy such that there exists a partitioning of the time into successive windows, for which the incurred regret is $o(w)$ with high probability on each window of length w. This for example includes the EXP3.P.1 algorithm (Auer et al., 2002b) where the k-th window has length 2^k.

The patient queuing game $\mathcal{G} = ([N], (c_i)_{i=1}^n, \boldsymbol{\mu}, \boldsymbol{\lambda})$ is defined as follows. The strategy space for each queue is $\mathcal{P}([K])$. Let $\boldsymbol{p}_{-i} \in (\mathcal{P}([K]))^{N-1}$ denote the vector of fixed distributions for all queues over servers, except for queue i. The cost function for queue i is defined as:

$$c_i(p_i, \boldsymbol{p}_{-i}) = \lim_{t \to +\infty} T_t^i/t,$$

where T_t^i is the age of the oldest packet in queue i at time t. Bounding T_t^i is equivalent to bounding Q_t^i.

Theorem 7.4 (Gaitonde and Tardos 2020b, informal)**.** *If $\eta > \frac{e}{e-1}$, any Nash equilibrium of the patient game $\mathcal{G}$ is stable.*

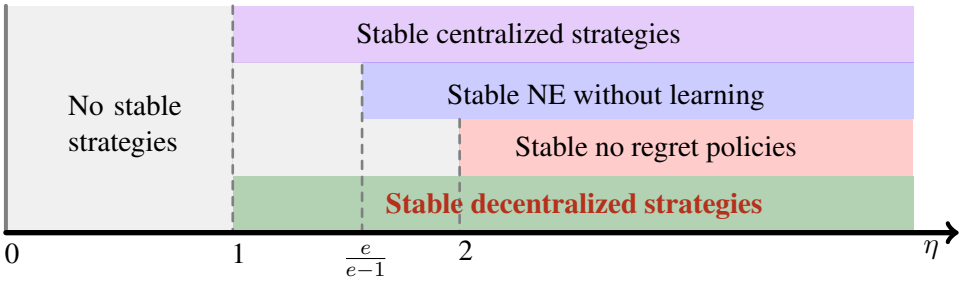


Figure 7.1: Existing results depending on the slack η. Our result is highlighted in red.

7.3 The case for a cooperative algorithm

According to Theorems 7.3 and 7.4, queues that are patient enough and select a fixed randomization over the servers are stable over a larger range of slack η than queues optimizing their individual regret. A key difference between the two settings is that when minimizing their regret, queues are myopic, which is formalized as follows. Let $\pi_{1:t}^i = (\pi_1^i, ..., \pi_t^i)$ be the vector of actions played by the queue i up to time t and let $\nu_t^i(\pi_{1:t}^i)$ be the indicator that it cleared a packet at iteration t, if it played the actions $\pi_{1:t}^i$ until t. Classical (external) regret of queue i over horizon T is then defined as:

$$R_i^{\text{ext}}(T) := \max_{p \in \mathcal{P}([K])} \sum_{t=1}^{T} \mathbb{E}_{\tilde{\pi}_t \sim p}[\nu_t^i(\pi_{1:t-1}^i, \tilde{\pi}_t)] - \sum_{t=1}^{T} \nu_t^i(\pi_{1:t}^i).$$

Thus minimizing the external regret is equivalent to maximizing the instant rewards at each iteration, ignoring the consequences of the played action on the state of the system. However, in the context of queuing systems, the actions played by the queues change the state of the system. Notably, letting other queues clear packets can be in the best interest of a queue, as it may give it priority in the subsequent iterations where it holds older packets. Since the objective is to maximize the total number of packets cleared, it seems adapted to minimize a *patient* version of the regret, namely the policy regret (Arora et al., 2012), rather than the external regret, which is defined by

$$R_i^{\text{pol}}(T) := \max_{p \in \mathcal{P}([K])} \sum_{t=1}^{T} \mathbb{E}_{\tilde{\pi}_{1:t} \sim \otimes_{i=1}^{t} p}[\nu_t^i(\tilde{\pi}_{1:t})] - \sum_{t=1}^{T} \nu_t^i(\pi_{1:t}^i).$$

That is, $R_i^{\text{pol}}(T)$ is the expected difference between the number of packets queue i cleared and the number of packets it would have cleared over the whole period by playing a fixed (possibly random) action, taking into account how this change of policy would affect the state of the system.

However, as stated in Proposition 7.1, optimizing this patient version of the regret rather than the myopic one could not guarantee stability on a wider range of slack value. This suggests that adding only patience to the learning strategy of the queues is not enough to go beyond a slack of 2, and that any strategy beating that factor 2 must somewhat include synchronisation between the queues.

Proposition 7.1. *Consider the partition of the time $t = 1, 2, \dots$ into successive windows, where $w_k = k^2$ is the length of the k-th one. For any $N \geq 2$, there exists an instance with $2N$ queues and servers, with slack $\eta = 2 - \mathcal{O}\left(\frac{1}{N}\right)$, s.t., almost surely, each queue's policy regret is $o(w_k)$ on all but finitely many of the windows, but the system is not strongly stable.*

Sketch of proof. Consider a system with $2N$ queues and servers with $\lambda_i = 1/2N$ and $\mu_i = 1/N - 1/4N^2$ for all $i \in [2N]$. The considered strategy profile is the following. For each $k \geq 0$, the k^{th} time window is split into two stages. During the first stage, of length $\lceil \alpha w_k \rceil$, queues $2i$ and $2i+1$ both play server $2i + t \pmod{2N}$ at iteration t, for all $i \in [N]$. During the second stage of the time window, queue i plays server $i + t \pmod{2N}$ at iteration t. This counter example, albeit very specific, illustrates well how when the queues are highly synchronised, it is better to remain synchronized rather than deviate, even if the synchronisation is suboptimal in terms of stability. The complete proof is provided in Section 7.C.1.

Queues following this strategy accumulate packets during the first stage, and clear more packets than they receive during the second stage. The value of α is tuned so that the queues still accumulate a linear portion of packets during each time window. For those appropriate α, the system is unstable.

Now suppose that queue i deviates from the strategy and plays a fixed action $p \in \mathcal{P}([K])$. In the first stage of each time window, queue i can clear a bit more packets than it would by not deviating. However, during the second stage, it is no longer synchronised with the other queues and collides with them a large number of times. Because of those collisions, it will accumulate many packets. In the detailed analysis, we demonstrate that, in the end, for appropriate values of α, queue i accumulates more packets than it would have without deviating. □

According to Theorem 7.4, the factor $\frac{e}{e-1}$ can be seen as the price of anarchy of the problem, as for slacks below, the worst Nash equilibria might be unstable. On the other hand, it is known that for any slack above 1, there exists a centralized stable strategy. This centralized strategy actually consists in queues playing the same joint probability at each time step, independently

from the number of accumulated packets. As a consequence, it is also a correlated equilibrium of the patient game and 1 can be seen as the correlated price of stability.

All these arguments make the case for cooperative decentralized learning strategies when η is small.

7.4 A decentralized algorithm

This section describes the decentralized algorithm ADEQUA, whose pseudocode is given in Algorithm 7.1. Due to space constraints, all the proofs are postponed to Section 7.C.2. ADEQUA assumes all queues *a priori* know the number N of queues in the game and have a unique rank or *id* in $[N]$. Moreover, the existence of a shared randomness between all queues is assumed. The *id* assumption is required to break the symmetry between queues and is classical in multiplayer bandits without collision information. On the other side, the shared randomness assumption is equivalent to the knowledge of a common seed for all queues, which then use this common seed for their random generators. A similar assumption is used in multiplayer bandits (Bubeck et al., 2020a).

ADEQUA is inspired by the celebrated ε-greedy strategy. With probability $\varepsilon_t = (N + K)t^{-\frac{1}{5}}$, at each time step, queues explore the different problem parameters as described below. Otherwise with probability $1 - \varepsilon_t$, they exploit the servers. Each queue i then sends a packet to a server following a policy solely computed from its local estimates $\hat{\lambda}^i, \hat{\mu}^i$ of the problem parameters λ and μ. The shared randomness is here used so that exploration simultaneously happens for all queues. If exploration/exploitation was not synchronized between the queues, an exploiting queue could collide with an exploring queue, biasing the estimates $\hat{\lambda}^i, \hat{\mu}^i$ of the latter.

Algorithm 7.1: ADEQUA

input: $i \in [N]$ (queue *id*), functions ϕ, ψ

for $t = 1, \dots, \infty$ **do**
  $\hat{P} \leftarrow \phi(\hat{\lambda}, \hat{\mu})$ and $\hat{A} \leftarrow \psi(\hat{P})$
  Draw $\omega_1 \sim \text{Bernoulli}((N + K)t^{-\frac{1}{5}})$ and $\omega_2 \sim \mathcal{U}(0, 1)$ `// shared randomness`
  if $\omega_1 = 1$ **then** EXPLORE(i) `// exploration`
  else Pull $\hat{A}(\omega_2)(i)$ `// exploitation`
end

Exploration. When exploring, queues choose either to explore the servers' parameters μ_k or the other queues' parameters λ_i as described in Algorithm 7.2 below. In the former case, all queues choose different servers at random (if they have packets to send). These rounds are used

to estimate the servers means: $\hat{\mu}_k^i$ is the empirical mean of server k observed by the queue i for such rounds. Thanks to the shared randomness, queues pull different servers here, making the estimates unbiased.

In the latter case, queues explore each other in a pairwise fashion. When queues i and j explore each other at round t, each of them sends their **most recent** packet to some server k, chosen uniformly at random, if and only if a packet appeared during round t. In that case, we say that *the queue i explores λ_j* (and vice versa). To make sure that i and j are the only queues choosing the server k during this step, we proceed as follows:

- queues sample a matching π between queues at random. To do so, the queues use the same method to plan an all-meet-all (or round robin) tournament, for instance Berger tables (Berger, 1899), and choose uniformly at random which round of the tournament to play. If the number of queues N is odd, in each round of the tournament, one queue remains alone and does nothing.

- the queues draw the same number $l \sim \mathcal{U}([K])$ with their shared randomness. For each pair of queues (i, j) matched in π, associate $k_{(i,j)} = l + \min(i, j) \pmod K + 1$ to this pair. The queues i and j then send to the server $k_{(i,j)}$.

As we assumed that the server breaks ties in the packets' age uniformly at random, the queue i clears with probability $(1 - \frac{\lambda_j}{2})\overline{\mu}$, where $\overline{\mu} = \frac{1}{K}\sum_{k=1}^K \mu_k$. Thanks to this, λ_j is estimated by queue i as:

$$\hat{\lambda}_j^i = 2 - 2\hat{S}_j^i/\widetilde{\mu}^i, \tag{7.2}$$

where $\widetilde{\mu}^i = \frac{\sum_{k=1}^K N_k^i \hat{\mu}_k^i}{\sum_{k=1}^K N_k^i}$, N_k^i is the number of *exploration* pulls of server k by queue i and $\hat{S}_j^i$ is the empirical probability of clearing a packet observed by queue i when exploring λ_j.

Remark 7.1. *The packet manipulation when exploring λ_j strongly relies on the servers tie breaking rules (uniformly at random). If this rule was unknown or not explicit, the algorithm can be adapted: when queue i explores λ_j, queue j instead sends the packet generated at time $t-1$ (if it exists), while queue i still sends the packet generated at time t. In that case, the clearing probability for queue i is exactly $(1-\lambda_j)\overline{\mu}$, allowing to estimate λ_j. Anticipating the nature of the round t (exploration vs. exploitation) can be done by drawing $\omega_1 \sim \mathrm{Bernoulli}(\varepsilon_t)$ at time $t-1$. If $\omega_1 = 1$, the round t is exploratory and the packet generated at time $t-1$ is then kept apart by the queue j.*

To describe the exploitation phase, we need a few more notations. We denote by $\mathfrak{B}_K$ the set of bistochastic matrices (non-negative matrices such that each of its rows and columns sums to 1) and by $\mathfrak{S}_K$ the set of permutation matrices in $[K]$ (a permutation matrix will be identified with its associated permutation for the sake of cumbersomeness).

Algorithm 7.2: EXPLORE

input: $i \in [N]$ `// queue id`
$k \leftarrow 0$
Draw $n \sim \mathcal{U}([N+K])$ `// shared randomness`
if $n \leq K$ **then** `// explore` μ
  $k \leftarrow n + i \pmod K + 1$
  Pull k ; Update N_k and $\hat{\mu}_k$
else `// explore` λ
  Draw $r \sim \mathcal{U}([N])$ and $l \sim \mathcal{U}([K])$ `// shared randomness`
  $j \leftarrow r^{\text{th}}$ opponent in the all-meet-all tournament planned according to Berger tables
  $k \leftarrow l + \min(i,j) \pmod K + 1$
  if $k \neq 0$ *and packet appeared at current round* **then** `// explore` λ_j `on server` k
    Pull k with most recent packet ; Update $\hat{S}_j$ and $\hat{\lambda}_j$ according to Equation (7.2)
  end
end

A **dominant mapping** is a function $\phi : \mathbb{R}^N \times \mathbb{R}^K \to \mathfrak{B}_K$ which, from (λ, μ), returns a bistochastic matrix P such that $\lambda_i < (P\mu)_i$ for every $i \in [N]$ if it exists (and the identity matrix otherwise).

A **BvN** (Birkhoff von Neumann) **decomposition** is a function $\psi : \mathfrak{B}_K \to \mathcal{P}(\mathfrak{S}_K)$ that associates to any bistochastic matrix P a random variable $\psi(P)$ such that $\mathbb{E}[\psi(P)] = P$; stated otherwise, it expresses P as a convex combination of permutation matrices. For convenience, we will represent this random variable as a function from $[0,1]$ (equipped with the uniform distribution) to $\mathfrak{S}_K$.

Informally speaking, those functions describe the strategies queues would follow in the centralized case: a dominant mapping gives adequate marginals ensuring stability (since the queue i clears in expectation $(P\mu)_i$ packets at each step, which is larger than λ_i by definition), while a BvN decomposition describes the associated coupling to avoid collisions. Explicitly, the joint strategy is for each queue to draw a shared random variable $\omega_2 \sim \mathcal{U}(0,1)$ and to choose servers according to the permutation $\psi(\phi(\lambda,\mu))(\omega_2)$

Exploitation. In a decentralized system, each queue i computes a mapping $\hat{A}^i := \psi(\phi(\hat{\lambda}^i, \hat{\mu}^i))$ solely based on its own estimates $\hat{\lambda}^i, \hat{\mu}^i$. A shared variable $\omega_2 \in [0,1]$ is then generated uniformly at random and queue i sends a packet to the server $\hat{A}^i(\omega_2)(i)$. If all queues knew exactly the parameters λ, μ, the computed strategies $\hat{A}^i$ would be identical and they would follow the centralized policy described above.

However, the estimates $(\hat{\lambda}^i, \hat{\mu}^i)$ are different between queues. The usual dominant mappings and BvN decompositions in the literature are non-continuous. Using those, even queues

with close estimates could have totally different $\hat{A}^i$, and thus collide a large number of times, which would impede the stability of the system. Regular enough dominant mappings and BvN decompositions are required, to avoid this phenomenon. The design of ϕ and ψ is thus crucial and appropriate choices are given in the following Sections 7.4.1 and 7.4.2. Nonetheless, they can be used in some black-box fashion, so we provide for the sake of completeness sufficient conditions for stability, as well as a general result depending on the properties of ϕ and ψ, in Section 7.A.

Remark 7.2. *The exploration probability $t^{-\frac{1}{5}}$ gives the smallest theoretical dependency in Δ in our bound. A trade-off between the proportion of exploration rounds and the speed of learning indeed appears in the proof of Theorem 7.1. Exploration rounds have to represent a small proportion of the rounds, as the queues accumulate packets when exploring. On the other hand, if queues explore more often, the regime where their number of packets decreases starts earlier. A general stability result depending on the choice of this probability is given by Theorem 7.6 in Section 7.A.*
Yet in Section 7.5, taking a probability $t^{-\frac{1}{4}}$ empirically performs better as it speeds up the exploration.

7.4.1 Choice of a dominant mapping

Recall that a dominant mapping takes as inputs (λ, μ) and returns, if possible, a bistochastic matrix P such that

$$\lambda_i < \sum_{k=1}^{K} P_{i,k}\mu_k \text{ for all } i \in [N].$$

The usual dominant mappings sort the vector λ and μ in descending orders (Marshall et al., 1979). Because of this operation, they are non-continuous and we thus need to design a regular dominant mapping satisfying the above property. Inspired by the $\log$-barrier method, it is done by taking the minimizer of a strongly convex program as follows

$$\phi(\lambda, \mu) = \operatorname*{arg\,min}_{P \in \mathfrak{B}_K} \max_{i \in [N]} -\ln\Big(\sum_{j=1}^{K} P_{i,j}\mu_j - \lambda_i\Big) + \frac{1}{2K}\|P\|_2^2. \tag{7.3}$$

Although the objective function is non-smooth because of the $\max$ operator, it enforces fairness between queues and leads to a better regularity of the $\arg\min$.

Remark 7.3. *Computing ϕ requires solving a non-smooth strongly convex minimization problem. This cannot be computed exactly, but a good approximation can be quickly obtained using the scheme described in Section 7.B. If this approximation error is small enough, it has no impact on the stability bound of Theorem 7.5. It is thus ignored for simplicity, i.e., we assume in the following that $\phi(\lambda, \mu)$ is exactly computed at each step.*

As required, ϕ always returns a matrix P satisfying that $\lambda < P\mu$ if possible, since otherwise the objective is infinite (and in that case we assume that ϕ returns the identity matrix). Moreover, the objective function is $\frac{1}{K}$-strongly convex, which guarantees some regularity of the $\arg\min$, namely local-Lipschitzness, leading to Lemma 7.1 below .

Lemma 7.1. *For any (λ, μ) with positive margin Δ (defined in Equation (7.1)), if $\|(\hat{\lambda} - \lambda, \hat{\mu} - \mu)\|_\infty \leq c_1\Delta$, for any $c_1 < \frac{1}{2\sqrt{e}+2}$, then*

$$\|\phi(\hat{\lambda}, \hat{\mu}) - \phi(\lambda, \mu)\|_2 \leq \frac{c_2 K}{\Delta}\|(\hat{\lambda} - \lambda, \hat{\mu} - \mu)\|_\infty,$$

where $c_2 = \frac{4}{(1-2c_1)/\sqrt{e}-2c_1}$. Moreover, denoting $\hat{P} = \phi(\hat{\lambda}, \hat{\mu})$, it holds for any $i \in [N]$,

$$\lambda_i \leq \textstyle\sum_{k=1}^{K} \hat{P}_{i,k}\mu_k - \left(\frac{1-2c_1}{\sqrt{e}} - 2c_1\right)\Delta.$$

The first property guarantees that if the queues have close estimates, they also have close bistochastic matrices $\hat{P}$. Moreover, the second property guarantees that each queue should clear its packets with a margin of order Δ, in absence of collisions.

7.4.2 Choice of a Birkhoff von Neumann decomposition

Given a bistochastic matrix $\hat{P}$, Birkhoff algorithm returns a convex combination of permutation matrices $P[j]$ such that $\hat{P} = \sum_j z[j]P[j]$. The classical version of Birkhoff algorithm is non-continuous in its inputs. Yet it can be smartly modified as in ORDERED BIRKHOFF, described in Algorithm 7.3, to get a regular BvN decomposition defined as follows for any $\omega \in (0, 1)$:

$$\psi(P)(\omega) = P[j_\omega] \tag{7.4}$$

where $P = \sum_j z[j]P[j]$ is the decomposition returned by ORDERED BIRKHOFF algorithm and j_ω verifies $\displaystyle\sum_{j \leq j_\omega} z[j] \leq \omega < \sum_{j \leq j_\omega + 1} z[j]$.

For a matrix P in the following, its support is defined as $\mathrm{supp}(P) = \{(i, j) \mid P_{i,j} \neq 0\}$. Obviously $\mathbb{E}_{\omega \sim \mathcal{U}(0,1)}[\psi(P)(\omega)] = P$ and permutations avoid collisions between queues. The difference with the usual Birkhoff algorithm happens at Line 4. Birkhoff algorithm usually computes any perfect matching in the graph induced by the support of $\hat{P}$ at the current iteration. This is often done with the Hopcroft-Karp algorithm, while it is here done with the Hungarian algorithm with respect to some cost matrix C. Although using the Hungarian algorithm slightly increases the computational complexity of this step (K^3 instead of $K^{2.5}$), it ensures to output the permutation matrices $P[j]$ according to a fixed order defined below.

Algorithm 7.3: ORDERED BIRKHOFF

input: $\hat{P} \in \mathfrak{B}_K$ (bistochastic matrix), $C \in \mathbb{R}^{K\times K}$ (cost matrix)

$j \leftarrow 1$
while $\hat{P} \neq \mathbf{0}$ **do**
  $C_{i,k} \leftarrow +\infty$ for all $(i,k) \notin \mathrm{supp}(\hat{P})$ `// remove edge (i,k) in induced graph`
  $P[j] \leftarrow \text{HUNGARIAN}(C)$ `// matching with minimal cost w.r.t. C`
  $z[j] \leftarrow \min_{(i,k)\in\mathrm{supp}(P[j])} \hat{P}_{i,k}$
  $\hat{P} \leftarrow \hat{P} - z[j]P[j]$ and $j \leftarrow j+1$
end
return $(z[j], P[j])_j$

Definition 7.1. *A cost matrix C induces an order $\prec_C$ on the permutation matrices defined, for any $P, P' \in \mathfrak{S}_K$ by*

$$P \prec_C P' \quad \textit{iff} \quad \textstyle\sum_{i,j} C_{i,j}P_{i,j} < \sum_{i,j} C_{i,j}P'_{i,j}.$$

This order might be non-total as different permutations can have the same cost. However, if C is drawn at random according to some continuous distribution, this order is total with probability 1. The order $\prec_C$ has to be the same for all queues and is thus determined beforehand for all queues.

Lemma 7.2. *Given matrices $C \in \mathbb{R}^{K\times K}$ and $P \in \mathfrak{B}_K$,* ORDERED BIRKHOFF *outputs a sequence $(z[j], P[j])_j$ of length at most K^2, such that*

$$P = \textstyle\sum_j z[j]P[j], \text{ where for all } j,\ z[j] > 0 \text{ and } P[j] \in \mathfrak{S}_K.$$

Moreover if the induced order $\prec_C$ is total, $z[j]$ is the j-th non-zero element of the sequence $(z_l(P))_{1\le l\le K!}$ defined by

$$z_j(P) = \min_{(i,k)\in\mathrm{supp}(P_j)} \left(P - \sum_{l=1}^{j-1} z_l(P)P_l\right)_{i,k} \tag{7.5}$$

where $(P_j)_{1\le j\le K!}$ is a $\prec_C$-increasing sequence of permutation matrices, i.e., $P_j \prec_C P_{j+1}$ for all j.

Lemma 7.2 is crucial to guarantee the regularity of ψ, given by Lemma 7.3.

Lemma 7.3. *Consider ψ defined as in Equation* (7.4) *with a cost matrix C inducing a total order $\prec_C$, then for any bistochastic matrices P, P'*

$$\int_0^1 \mathbb{1}\left(\psi(P)(\omega) \neq \psi(P')(\omega)\right) \mathrm{d}\omega \le 2^{2K^2}\|P - P'\|_\infty.$$

Lemma 7.3 indeed ensures that the probability of collision between two queues remains small when they have close estimates. Unfortunately, the regularity constant is exponential in K^2, which yields a similar dependency in the stability bound of Theorem 7.5. The existence of a BvN decomposition with polynomial regularity constants remains unknown, even without computational considerations. The design of a better BvN decomposition is left open for future work and would directly improve the stability bounds, using the general result given by Theorem 7.6 in Section 7.A.

7.4.3 Stability guarantees

This section finally provides theoretical guarantees on the stability of the system when all queues follow ADEQUA. The success of ADEQUA relies on the accurate estimation of all problem parameters by the queues, given by Lemma 7.9 in Section 7.C.2. After some time τ, the queues have tight estimations of the problem parameters. Afterwards, they clear their packets with a margin of order Δ, thanks to Lemmas 7.1 and 7.3. This finally ensures the stability of the system, as given by Theorem 7.5.

Theorem 7.5. *For any $\eta > 1$, the system where all queues follow* ADEQUA*, for every queue i and any $r \in \mathbb{N}$, there exists a constant C_r depending only on r such that*

$$\mathbb{E}[(Q_t^i)^r] \leq C_r K N \left(\frac{N^{\frac{5}{2}} K^{\frac{5}{2}} 2^{5K^2}}{(\min(1, K\overline{\mu})\underline{\lambda})^{\frac{5}{4}} \Delta^5} \right)^r, \quad \textit{for all } t \in \mathbb{N}.$$

As a consequence, for any $\eta > 1$, this decentralized system is strongly stable.

Despite yielding an exponential dependency in K^2, this anytime bound leads to a first decentralized stability result when $\eta \in (1, \frac{e}{e-1})$, which closes the stability gap left by previous works. Moreover it can be seen in the proof that the asymptotic number of packets is much smaller. It actually converges, in expectation, to the number of packets the queues would accumulate if they were following a stable centralized strategy from the beginning. As already noted by Krishnasamy et al. (2016) for a single queue, the number of packets first increases during the learning phase and then decreases once the queues have tight enough estimations, until reaching the same state as in the perfect knowledge centralized case. This is empirically confirmed in Section 7.5.

7.5 Simulations

Figures 7.2 and 7.3 compare on toy examples the stability of queues, when either each of them follows the no-regret strategy EXP3.P.1, or each queue follows ADEQUA. For practical con-

siderations, we choose the exploration probability $\varepsilon_t = (N+K)t^{-\frac{1}{4}}$ for ADEQUA, as the exploration is too slow with ε_t of order $t^{-\frac{1}{5}}$.

These figures illustrate the evolution of the average queue length on two different instances with $N = K = 4$.

In the first instance shown in Figure 7.2, for all $i \in [N]$, $\lambda_i = (N+1)/N^2$. Moreover $\mu_1 = 1$ and for all $i \geq 2$, $\mu_i = (N-1)/N^2$. Here $\eta < 2$ and no-regret strategies are known to be unstable (Gaitonde and Tardos, 2020a). It is empirically confirmed as the number of packets in each queue diverges when they follow EXP3.P.1. Conversely, when the queues follow ADEQUA, after a learning phase, the queues reach equilibrium and all succeed in clearing their packets.

In the second instance shown in Figure 7.3, for all $i \in [N]$, $\lambda_i = 0.55 - 0.1 \cdot i$ and $\mu_i = 2.1\lambda_i$. Here $\eta > 2$ and both strategies are known to be stable, which is again empirically confirmed. However, ADEQUA requires more time to learn the different parameters, suggesting that individual no-regret strategies might be better on easy instances where $\eta > 2$.

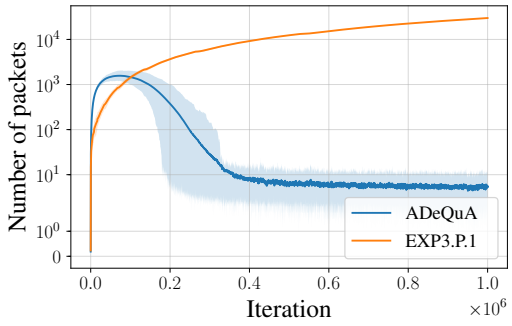


Figure 7.2: Hard instance, $\eta < 2$.

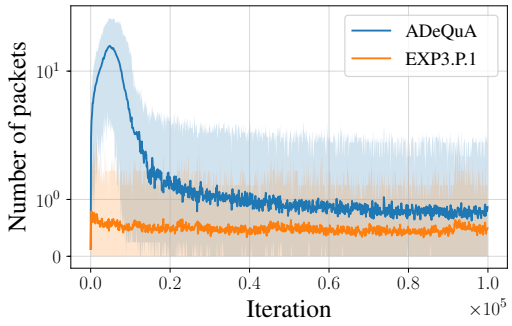


Figure 7.3: Easy instance, $\eta > 2$.

Appendix

7.A General version of Theorem 7.5

ADEQUA is described for specific choices of the functions ϕ and ψ given by Sections 7.4.1 and 7.4.2. It yet uses them in a black box fashion and different functions can be used, as long as they verify some key properties. This section provides a general version of Theorem 7.5, when the used dominant mapping and BvN decomposition respect the properties given by Assumptions 7.1 and 7.2.

Assumption 7.1 (regular dominant mapping)**.** *There are constants $c_1, c_2 > 0$ and a norm $\|\cdot\|$ on $\mathbb{R}^{K\times K}$, such that if $\|(\hat{\lambda}-\lambda, \hat{\mu}-\mu)\|_\infty \leq c_1\Delta$, then*

$$\|\phi(\hat{\lambda},\hat{\mu}) - \phi(\lambda,\mu)\| \leq L_\phi \cdot \|(\hat{\lambda}-\lambda, \hat{\mu}-\mu)\|_\infty.$$

Moreover, $\hat{P} = \phi(\hat{\lambda},\hat{\mu})$ is bistochastic and for any $i \in [N]$,

$$\lambda_i \leq \textstyle\sum_{k=1}^{K} \hat{P}_{i,k}\mu_k - c_2\Delta.$$

Assumption 7.2 (regular BvN decomposition)**.** *Consider the same norm $\|\cdot\|$ as Assumption 7.1 on $\mathbb{R}^{K\times K}$. For any bistochastic matrices P, P'*

$$\int_0^1 \psi(P)(\omega)\mathrm{d}\omega = P$$
$$\text{and } \int_0^1 \mathbb{1}\left(\psi(P)(\omega) \neq \psi(P')(\omega)\right)\mathrm{d}\omega \leq L_\psi \cdot \|P-P'\|.$$

Lemmas 7.1 and 7.3 show that the functions described in Sections 7.4.1 and 7.4.2 verify Assumptions 7.1 and 7.2 with the constants L_ϕ and L_ψ respectively of order $\frac{K}{\Delta}$ and 2^{2K^2} with the norm $\|\cdot\|_\infty$. Designing a dominant mapping and a BvN decomposition with smaller constants L_ϕ and L_ψ is left open for future work. It would lead to a direct improvement of the stability bound, as shown by Theorem 7.6.

Theorem 7.6. *Assume all queues follow* ADEQUA*, using an exploration probability $\varepsilon_t = xt^{-\alpha}$ with $x > 0, \alpha \in (0,1)$ and functions ϕ and ψ verifying Assumptions 7.1 and 7.2 with the constants L_ϕ, L_ψ. The system is then strongly stable and for any $r \in \mathbb{N}$, there exists a constant C_r such that:*

$$\mathbb{E}[(Q_t^i)^r] \leq C_r \left(\frac{x^{r/\alpha}}{\Delta^{r/\alpha}} + KN \left(\frac{N^2 K L_\phi^2 L_\psi^2}{\min(1, K\overline{\mu})\underline{\lambda}\Delta^2 x} \right)^{\frac{r}{1-\alpha}} \right), \quad \textit{for all } t \in \mathbb{N}$$

The proof directly follows the lines of the proof of Theorem 7.5 in Section 7.C.2 and is thus omitted here. From this version, it can be directly deduced that $\alpha = \frac{1}{5}$ gives the best dependency in Δ for ADEQUA. Moreover the best choice for x varies with r. When $r \to \infty$, it actually is $x = N^{\frac{2}{5}} K^{\frac{3}{5}} 2^{\frac{4}{5}K^2}$ for ADEQUA. The choice $x = N + K$ is preferred for simplicity and still yields quite similar problem dependent bounds.

7.B Efficient computation of ϕ

As mentioned in Section 7.4.1, computing exactly $\phi(\hat{\lambda}, \hat{\mu})$ is not possible. Even efficiently approximating it is not obvious, as the function to minimize is neither smooth nor Lipschitz. We here describe how an approximation of ϕ can be efficiently computed with guarantees on the approximation error.

First define the empirical estimate of the margin Δ:

$$\hat{\Delta} \coloneqq \min_{k \in [N]} \frac{1}{k} \left(\sum_{i=1}^{k} \hat{\mu}_{(i)} - \hat{\lambda}_{(i)} \right).$$

It can be computed in time $\mathcal{O}(N \log(N))$ as it only requires to sort the vectors $\hat{\lambda}$ and $\hat{\mu}$. If $\hat{\Delta} \leq 0$, then the value of the optimization problem is $+\infty$ and any matrix can be returned. Assume in the following $\hat{\Delta} > 0$. Similarly to the proof of Lemma 7.1, it can be shown that the value of the optimization problem is smaller than $-\ln(\hat{\Delta}/\sqrt{e})$. Noting by $\mathfrak{B}_K$ the set of $K \times K$ bistochastic matrices, the optimization problem given by Equation (7.3) is then equivalent to

$$\arg\min_{P \in \mathcal{X}} g(P), \tag{7.6}$$

where

$$\mathcal{X} = \left\{ P \in \mathfrak{B}_K \mid \forall i \in [N], \textstyle\sum_{j=1}^{K} P_{i,j}\mu_j - \lambda_i \geq \frac{\hat{\Delta}}{\sqrt{e}} \right\},$$

$$\text{and } g(P) = \max_{i \in [N]} -\ln\left(\textstyle\sum_{j=1}^{K} P_{i,j}\mu_j - \lambda_i\right) + \frac{1}{2K}\|P\|_2^2.$$

Thanks to this new constraint set, the objective function of Equation (7.6) is now $(\frac{\sqrt{e}}{\Delta}+1)$-Lipschitz. We can now use classical results for Lipschitz strongly convex minimization to obtain convergence rates of order $\frac{1}{t}$ for the projected gradient descent algorithm (see e.g., Bubeck, 2014, Theorem 3.9). These results yet assume that the projection on the constraint set can be exactly computed in a short time. This is not the case here, but it yet can be efficiently approximated using interior point methods (see e.g., Bubeck, 2014, Section 5.3), which has a linear convergence rate. If this approximation is good enough, similar convergence guarantees than with exact projection can be shown similarly to the original proof.

Algorithm 7.4 then describes how to quickly estimate $\phi(\hat{\lambda}, \hat{\mu})$, where $\hat{\Pi}_{\mathcal{X}}$ returns an approximation of the orthogonal projection on the set $\mathcal{X}$ and ∂g is a sub-gradient of g. It uses an averaged value of the different iterates, as the last iterate does not have good convergence guarantees.

Algorithm 7.4: Compute ϕ

input: function g, constraint set $\mathcal{X}$, $P^0 \in \mathcal{X}$

$P, \hat{P} \leftarrow P^0$

for $t = 1, \dots, n$ **do**

  $P \leftarrow \hat{\Pi}_{\mathcal{X}}\left(P - \frac{2N}{(t+1)}\partial g(P)\right)$ `// approximated projection`

  $\hat{P} \leftarrow \frac{t}{t+2}\hat{P} + \frac{2}{t+2}P$

end

return $\hat{P}$

In practice, the approximation can even be computed faster by initializing P^0 in Algorithm 7.4 with the solution of the previous round $t-1$.

7.C Omitted Proofs

7.C.1 Unstable No-Policy regret system example

Lemma 7.4. *Consider the system where the queues play according to the policy described in Algorithm 7.5 over successive windows of length $w_k = k^2$. If $\alpha > 1 - \frac{d}{N-d}$, the system is not stable.*

Proof. Note that the system is equivalent to a system where each queue or pair of queue would always pick the same server. For simplicity, the analysis deals with that equivalent system. Also, wlog, we analyse the subsystem with the two first queues and the two first servers. Let $\{B_t^i\}_{i\in[n],t\geq 1}$ be the independent random variables indicating the arrival of a packet on queue i at time t, $\{S_t^i\}_{i\in[n],t\geq 1}$ be the indicators that server j would clear a packet at iteration ℓ if one

Algorithm 7.5: Unstable No-policy regret system example

input: w_k, N, α, $\lambda = (1/N, \ldots, 1/N)$, $\mu = (2(N-d)/N^2, \ldots, 2(N-d)/N^2)$

for $k = 1, \ldots, \infty$ **do**
  for $t = 1, \ldots, \lceil \alpha w_k \rceil$ **do**
    Queues $2i$ and $2i+1$ play server $2i + t \pmod N$ `// stage 1`
  end
  for $t = \lceil \alpha w_k \rceil + 1, \ldots, w_k$ **do**
    Queue i plays server $i + t \pmod N$ `// stage 2`
  end
end

were sent to it. For each queue $i \in [N]$ and $t \geq 0$, we have by Chernoff bound

$$\Pr\left(\left|\sum_{t=1}^{\ell} B_t^i - \lambda_i \ell\right| \geq \sqrt{\ell \ln(\ell)}\right) \leq \frac{2}{\ell^2}.$$

The same holds for each queue, thus the probability that this event happens for queue 1 or queue 2 is at most, $\frac{4}{\ell^2}$. As it is summable in ℓ, The Borel-Cantelli Lemma implies that, for large enough ℓ, almost surely, for any $i \in [2]$:

$$\sum_{\ell=1}^{\ell} B_t^i = \lambda_i \ell \pm \widetilde{\mathcal{O}}\left(\sqrt{\ell}\right), \tag{7.7}$$

where $\widetilde{\mathcal{O}}$ hides poly-log factors in ℓ

Let $W_k = \sum_{i=1}^{k} w_i$. Note that $W_k = \Theta\left(k^3\right) = \Theta\left(w_k^{3/2}\right)$. Again by Chernoff bound and Borel-Cantelli, for large enough k, almost surely, for any $i \in \{1, 2\}$:

$$\sum_{t=W_{k-1}}^{W_{k-1}+\lceil \alpha w_k \rceil} S_t^i = \mu_i \alpha w_k \pm \widetilde{\mathcal{O}}\left(\sqrt{w_k}\right), \quad \sum_{t=W_{k-1}+\lceil \alpha w_k \rceil}^{W_k} S_t^i = \mu_i (1-\alpha) w_k \pm \widetilde{\mathcal{O}}\left(\sqrt{w_k}\right). \tag{7.8}$$

Thus, for any large enough k, the total number of packets in both queues at time W_k is almost surely lower bounded as:

$$\begin{aligned} Q_{W_k}^1 + Q_{W_k}^2 \geq & \sum_{t=1}^{W_k} (B_t^1 + B_t^2) - \sum_{t=1}^{W_k} S_t^1 - \sum_{l=1}^{k} \left(\sum_{t=W_{l-1}+\lceil \alpha w_l \rceil}^{W_l} S_t^2 \right) && (7.9) \\ \geq & \left[\frac{2}{N} - \frac{2(N-d)}{N^2} - (1-\alpha) \frac{2(N-d)}{N^2} \right] W_k - \widetilde{\mathcal{O}}\left(W_k^{2/3}\right) && (7.10) \\ \geq & \frac{2\left[\alpha(N-d) - (N-2d)\right]}{N^2} W_k - \widetilde{\mathcal{O}}\left(W_k^{2/3}\right) && (7.11) \end{aligned}$$

which is a diverging function of W_k. Note that this result also holds for any pair of queues

$(2i-1, 2i)$, with $i \in [N/2]$. □

Lemma 7.5. *Consider the same setting as in Lemma 7.4. For any $i \in [N]$ and large enough k, queue i clears*

$$\left(\frac{N-d}{N^2} + (1-\alpha)\frac{N-d}{N^2} + o(1)\right) w_k$$

packets almost surely over window w_k.

Proof. The proof starts by showing that for any large enough t, all the queues hold roughly the same number of packets. Then, as they receive roughly the same number of packets over a time window and we can compute the approximate total number of packets cleared, the results follows.

Let T_i^t be the age of the oldest packet in queue i at time t. By Chernoff bound,

$$\mathrm{P}(|T_i^t - NQ_i^t| \geq N\sqrt{t\ln(t)}) \leq \frac{2}{t^2}.$$

Thus, using the Borel-Cantelli lemma, for any queue i, almost surely, for all large enough k and all $t \in [W_{k-1}+1, W_k]$,

$$|T_i^t - NQ_i^t| \leq N\sqrt{t\ln(t)} = \widetilde{\mathcal{O}}(w_k^{3/4}). \tag{7.12}$$

For any $(i,j) \in [N]^2$, define

$$\phi_t^+(i,j) := \left(Q_t^i - Q_t^j - 2N\sqrt{t\ln(t)}\right)_+ \text{ and } \phi_t^-(i,j) := \left(Q_t^i - Q_t^j + 2N\sqrt{t\ln(t)}\right)_-.$$

Let C_t^i be the indicator function that queue i clears a packet at iteration t. Note that for any large enough t, $\phi_t^+(i,j)$ is a supermartingale. Indeed,

$$\begin{aligned}\mathbb{E}[\phi_{t+1}^+(i,j)|\phi_{1:t}^+(i,j)] \leq& \phi_t^+(i,j) + \mathbb{E}[B_t^i - B_t^j|\phi_{1:t}^+(i,j)] - \mathbb{E}[C_t^i - C_t^j|\phi_{1:t}^+(i,j)]\\ \leq& \phi_t^+(i,j).\end{aligned}$$

The second inequality comes from Equation (7.12), that implies that for any large enough t, if $\phi_t^+(i,j)$ is strictly positive, queue i holds the oldest packet and thus clears one with higher probability than queue j. By the same arguments, $\phi_t^-(i,j)$ a submartingale. Also, $|\phi_{t+1}^+(i,j) - \phi_t^+(i,j)| \leq 2(N+1)$ for any $t \geq 0$, and the same holds for $\phi_t^-(i,j)$. Let τ_{ij} be the stopping time of the smallest iteration after which Equation (7.12) always holds for queues i and j. By Azuma-Hoeffding's inequality,

$$\Pr\left(\phi_\ell^+(i,j) - \phi_{\tau_{ij}}^+(i,j) \geq 3(N+1)\sqrt{\ell\ln(\ell)}\right) \leq \frac{2}{\ell^2}$$

and

$$\Pr\left(\phi_\ell^-(i,j) - \phi_{\tau_{ij}}^+(i,j) \leq -3(N+1)\sqrt{\ell \ln(\ell)}\right) \leq \frac{2}{\ell^2}.$$

This, together with a union bound and Borel-Cantelli's Lemma implies that almost surely, for any large enough t, for any $(i,j) \in [N]^2$

$$Q_t^i - Q_t^j = \widetilde{\mathcal{O}}\left(\sqrt{t}\right). \tag{7.13}$$

This with Equation (7.9) implies that for any large enough k, for any $i \in [N]$, almost surely,

$$Q_{W_k}^i \geq \frac{[\alpha(N-d) - (N-2d)]}{N^2} W_k - \widetilde{\mathcal{O}}\left(W_k^{2/3}\right).$$

This means that for any large enough k, every queue holds at least one packet over the whole window w_k. This and Equation (7.8) is already enough to show that for any time-window w_k, for any large enough k, the total number of packets cleared by every couple of queues $(2i-1, 2i)$, $i \in [N/2]$ is:

$$2\left(\frac{N-d}{N^2} + (1-\alpha)\frac{N-d}{N^2}\right) w_k + \widetilde{\mathcal{O}}\left(\sqrt{w_k}\right).$$

During time window w_k, according to Equation (7.7), both every queue receives $\alpha w_k/N + \widetilde{\mathcal{O}}\left(w_k^{3/4}\right)$ packets almost surely for any large enough k. Equation (7.13) implies that for any $i \in [N/2]$

$$Q_{W_k}^{2i-1} - Q_{W_k}^{2i} = \widetilde{\mathcal{O}}\left(w_k^{3/4}\right) \text{ and } Q_{W_{k-1}}^{2i-1} - Q_{W_{k-1}}^{2i} = \widetilde{\mathcal{O}}\left(w_k^{3/4}\right).$$

Therefore, over each time-window w_k, for any large enough k, each queue clears

$$\left(\frac{N-d}{N^2} + (1-\alpha)\frac{N-d}{N^2} + o(1)\right) w_k$$

packets almost surely. □

Lemma 7.6. *Consider again the system where the queues play according to the policy described in Algorithm 7.5 over successive windows of length $w_k = k^2$. If $\alpha < 1 - \frac{1}{N-1}$, the queues have no policy regret in all but finitely many of the windows.*

Wlog, let us consider that queue 1 deviates, and plays at every iteration a server chosen from the probability distribution $\mathbf{p} = (p_1, ..., p_N)$, with p_i the probability to play server i. To upper bound the number of packets queue 1 clears over each time window, we can assume it always has priority over queue 2 and ignore it in the analysis.

Before proving Lemma 7.6, we prove the following technical one.

Lemma 7.7. *Consider that a queue deviates from the strategy considered in Lemma 7.6 and plays at every iteration a server chosen from the probability distribution $\boldsymbol{p} = (p_1, ..., p_N)$, with*

p_i the probability to play server i. For any large enough k, almost surely, the number of packets the deviating queue clears of the first stage of the k^{th} window is

$$\left(\frac{1}{2}+\frac{1}{N}\right)\frac{2(N-d)}{N^2}\alpha w_k+\widetilde{\mathcal{O}}\left(w_k^{3/4}\right).$$

Proof. The proof starts by showing that for any large enough t, every non-deviating queue holds approximately the same number of packets.

First note that for any large enough t, Equation (7.12) still holds surely for any queue i. For any $(i,j)\in\{3,\dots,N\}^2$, define

$$\phi_\ell^+(i,j):=\left(Q^i_{\lceil \ell N\rceil}-Q^j_{\lceil \ell N\rceil}-4N\sqrt{\lceil \ell N\rceil \ln(\lceil \ell N\rceil)}\right)_+$$

and

$$\phi_\ell^-(i,j):=\left(Q^i_{\lceil \ell N\rceil}-Q^j_{\lceil \ell N\rceil}+4N\sqrt{\lceil \ell N\rceil \ln(\lceil \ell N\rceil)}\right)_-.$$

For any interval $[\lceil \ell N\rceil,\lceil(\ell+1)N\rceil]$ where Equation (7.12) holds for queues $1,i$ and j, if $\phi_\ell^+(i,j)$ is strictly positive, then

$$\mathbb{E}\left[\sum_{t=\lceil \ell N\rceil}^{\lceil(\ell+1)N\rceil} C_t^j-C_t^i\middle|\phi^+_{1:t}(i,j)\right]\le 0.$$

Indeed, if $\phi_\ell^+(i,j)$ is strictly positive and Equation (7.12) holds, queue i holds the oldest packets throughout the interval. Also, queue i and queue j collide with queue 1 the same number of times over the interval in expectation, and if at one iteration of the interval, queue 1 holds an older packet than queue i, it holds an older packet than queue j over the whole interval. Thus $\phi_\ell^+(i,j)$ is a submartingale. By the same arguments, $\phi_\ell^+(i,j)$ is a supermartingale. Also, $|\phi^+_{\ell+1}(i,j)-\phi^+_\ell(i,j)|\le 4(N+1)^2$ and the same holds for $\phi_\ell^-(i,j)$. Finishing with the same arguments used to prove Equation (7.13), almost surely, for any $(i,j)\in\{3,\dots,N\}^2$,

$$Q_t^i-Q_t^j=\widetilde{\mathcal{O}}\left(\sqrt{t}\right). \tag{7.14}$$

We now show that for any large enough t, queue 1 can not hold many more packets than the non-deviating queues. Define

$$\phi_t^+:=\left(Q_t^1-\max_{i\ge 3}Q_t^i-2N\sqrt{t\ln(t)}\right)_+.$$

Once again, at every iteration where ϕ_t^+ is strictly positive and Equation (7.12) holds, queue 1 holds the oldest packet and thus has priority on whichever server it chooses. This implies that for any large enough t, ϕ_t^+ is a supermartingale. It also holds that for any $t\ge 0$, $|\phi^+_{t+1}-\phi^+_t|\le$

$2(N+1)$. Thus, with the same arguments used to prove Equation (7.13), almost surely,

$$\left(Q_t^1 - \max_{i\geq 3} Q_t^i\right)_+ = \widetilde{\mathcal{O}}\left(\sqrt{t}\right). \tag{7.15}$$

With that at hand, we prove that for any large enough k, queue 1 doesn't get priority often over the other queues during the first stage of the k^{th} window. For any $i \in \{2,\dots,N/2\}$, pose:

$$\psi_\ell^i = \frac{1}{2}\left(Q^{2i-1}_{\lceil \ell N\rceil} + Q^{2i}_{\lceil \ell N\rceil}\right) - Q^1_{\lceil \ell N\rceil} - \frac{2(N-d)}{N^3}(\lceil \ell N\rceil - W_{k-1})$$

For any ℓ s.t. $\{\lceil \ell N\rceil; \lceil (\ell+1)N-1\rceil\}$ is included in the first phase of a window, we have

$$\begin{aligned}\sum_{t=\lceil \ell N\rceil}^{\lceil (\ell+1)N\rceil-1} \mathbb{E}\left[C_t^1 \middle| \psi^+_{1:\ell}(i,j)\right] &\geq \sum_{t=\lceil \ell N\rceil}^{\lceil (\ell+1)N\rceil-1} \mathbb{E}\left[S_i^t \mathbb{1}_{\{\text{queue 1 and only queue 1 picks server } i\}} \middle| \psi^+_{1:\ell}(i,j)\right] \\ &\geq \frac{N-d}{N} + \frac{2(N-d)}{N^2}\end{aligned}$$

as well as

$$\begin{aligned}\sum_{t=\lceil \ell N\rceil}^{\lceil (\ell+1)N\rceil-1} \mathbb{E}\left[\frac{1}{2}\left(C_t^{2i} + C_t^{2i-1}\right) \middle| \psi^+_{1:\ell}(i,j)\right] &\leq \sum_{t=\lceil \ell N\rceil}^{\lceil (\ell+1)N\rceil-1} \mathbb{E}\left[\frac{1}{2} S^t_{i+t \pmod N} \middle| \psi^+_{1:\ell}(i,j)\right] \\ &\leq \frac{N-d}{N}.\end{aligned}$$

Those two inequalities imply:

$$\begin{aligned}\mathbb{E}[\psi^i_{\ell+1} | \psi^+_{1:\ell}(i,j)] =& \psi^+_\ell(i,j) + \sum_{t=\lceil \ell N\rceil}^{\lceil (\ell+1)N\rceil-1} \mathbb{E}\left[\frac{1}{2}(B_t^{2i} - B_t^{2i-1}) - B_t^1 \middle| \psi^+_{1:\ell}(i,j)\right] \\ &- \sum_{t=\lceil \ell N\rceil}^{\lceil (\ell+1)N\rceil-1} \mathbb{E}\left[\frac{1}{2}(C_t^{2i} - C_t^{2i-1}) - C_t^1 \middle| \psi^+_{1:\ell}(i,j)\right] - \frac{2(N-d)}{N^2} \\ \geq& \psi^+_\ell(i,j).\end{aligned}$$

Thus, for any ℓ s.t. $\{\lceil \ell N\rceil; \lceil (\ell+1)N-1\rceil\}$ is included in the first phase of a window, ψ^i_ℓ is a submartingale. Moreover, for any $\ell \geq 0$, $|\psi^i_{\ell+1} - \psi^i_\ell| \leq 3N$. Thus, by Azuma-Hoeffding's inequality, for any ℓ s.t.$\{\lceil \ell N\rceil; \lceil (\ell+1)N-1\rceil\} \subset [W_{k-1}, W_{k-1} + \alpha w_k]$,

$$\Pr\left(\psi^i_\ell - \psi^i_{W_k} \leq -6N\sqrt{\ell N \ln(\ell N)}\right) \leq \frac{1}{(\ell N)^2}.$$

Borel-Cantelli's lemma implies, that for any large enough ℓ s.t.$\{\lceil \ell N\rceil; \lceil (\ell+1)N-1\rceil\} \subset$

$[W_{k-1}, W_{k-1} + \alpha w_k]$, almost surely:

$$\psi_\ell^i \geq \psi_{W_k}^i - 6N\sqrt{\ell N \ln(\ell N)}.$$

This and Equation (7.15) applied at $t = W_k$, imply that for any large enough k, for any $t \in [W_{k-1}, W_{k-1} + \alpha w_k]$,

$$\begin{aligned}\frac{1}{2}\left(Q_t^{2i-1} + Q_t^{2i}\right) \geq & Q_{\lceil \ell N \rceil}^1 + \frac{2(N-d)}{N^3}(t - W_{k-1}) + \psi_{W_k}^i - \widetilde{O}(\sqrt{t}) \\ \geq & Q_{\lceil \ell N \rceil}^1 + \frac{2(N-d)}{N^3}(t - W_{k-1}) - \widetilde{O}(w_k^{3/4}).\end{aligned}$$

This and Equation (7.12) imply that during the first stage of the time window, queue 1 holds younger packets than any other queues $i \geq 3$ after at most $\widetilde{\mathcal{O}}(w_k^{3/4})$ iterations.

By Chernoff bound and the Borel-Cantelli lemma again, for any large enough k, almost surely, the number of packets queue 1 clears during the first stage of the k^{th} window on servers where it does not collide with other queues is:

$$\sum_{t=W_{k-1}+1}^{W_{k-1}+\alpha w_k} \sum_{i=1}^{N} S_i^t \mathbb{1}_{\{\text{queue 1 and only queue 1 picks server } i\}} = (\frac{1}{2} + \frac{1}{N})\frac{2(N-d)}{N^2}\alpha w_k + \widetilde{\mathcal{O}}\left(\sqrt{w_k}\right).$$

Since we have shown that for any large enough k, almost surely, queue 1 does not have priority over the other queues after at most $\widetilde{\mathcal{O}}(w_k^{3/4})$ iterations, for any large enough k, almost surely, the number of packets queue 1 clears of the first stage of the k^{th} window is

$$\left(\frac{1}{2} + \frac{1}{N}\right)\frac{2(N-d)}{N^2}\alpha w_k + \widetilde{\mathcal{O}}\left(w_k^{3/4}\right).$$

□

We are now ready to prove Lemma 7.6.

Proof. By Chernoff bound and the Borel-Cantelli lemma, almost surely for any large enough k, the number of packets queue 1 clears during the second stage of the window on servers where it does not collide with other queues is:

$$\sum_{t=W_{k-1}+\alpha w_k}^{W_k - 1} \sum_{i=1}^{N} S_i^t \mathbb{1}_{\{\text{queue 1 and only queue 1 picks server } i\}} = \frac{4(N-d)}{N^3}(1-\alpha)w_k + \widetilde{\mathcal{O}}\left(\sqrt{w_k}\right). \tag{7.16}$$

Suppose that during the second stage of the window, queue 1 never gets priority over another queue. In that case, according to equation Equation (7.16) and Lemma 7.7, for any large enough k, almost surely, the total number of packets cleared by queue 1 during the time window is

$$\left(\frac{\alpha}{2}+\frac{2-\alpha}{N}\right)\frac{2(N-d)}{N^2}w_k+\widetilde{\mathcal{O}}(w_k^{3/4}).$$

For any large enough k, if $\alpha \leq 1-\frac{1}{N-1}$ this is smaller than the number of packets queue 1 would have cleared had it not deviated, according to Lemma 7.5.

On the other hand, suppose that queue gets priority over some other queue i at some iteration τ of the second stage of the window. In that case, at that iteration, queue 1 holds the oldest packets, which, according to Equation (7.12), implies

$$Q_1^\tau > Q_i^\tau - \widetilde{\mathcal{O}}(w_k^{3/4})$$

During the second stage of the window, for any $i \geq 3$, $\gamma_t^i := \left(Q_i^t - Q_1^t - 2N\sqrt{t\ln(t)}\right)_+$ is a supermartingale with bounded increments for any t where Equation (7.12) holds for queues 1 and i. Indeed, in that case, if γ_t^i is strictly positive, queue i holds an older packet than queue 1, and thus, whether they collide or not, it has a higher probability to clear a packet than queue 1. Thus, by Azuma-Hoeffding and the Borel-cantelli lemma again, for any large enough k, almost surely,

$$Q_i^{W_k} - Q_1^{W_k} \leq Q_i^\tau - Q_1^\tau + \widetilde{\mathcal{O}}(w_k^{3/4}).$$

Thus it holds that $Q_1^{W_k} \geq Q_i^{W_k} - \widetilde{\mathcal{O}}(w_k^{3/4})$ for any $i \geq 2$. This and Equation (7.15) imply that all the queues clear approximately the same number of packets over those time windows for any large enough k almost surely. Thus queue 1 clears

$$\left[(2-\alpha)(N-2)+(\alpha+\frac{4-2\alpha}{N})\right]\frac{(N-d)}{(N-1)N^2}w_k+\widetilde{\mathcal{O}}\left(w_k^{3/4}\right)$$

packets almost surely, which again is smaller than the number of packets it would have cleared had it not deviated.

Thus, the deviating queue clears almost surely less packets by time window than it would have had it not deviated on all but finitely many of the time windows, which implies that it has no policy regret on all but finitely many of the time windows. □

7.C.2 Proofs of Section 7.4

Proof of Lemma 7.1

We want to show that if $\|(\hat{\lambda} - \lambda, \hat{\mu} - \mu)\|_\infty \leq c_1\Delta$, then

$$\|\phi(\hat{\lambda}, \hat{\mu}) - \phi(\lambda, \mu)\|_2 \leq \frac{c_2 K}{\Delta}\|(\hat{\lambda} - \lambda, \hat{\mu} - \mu)\|_\infty, \tag{7.17}$$

with the constants c_1, c_2 given in Lemma 7.1.

Recall that ϕ is defined as

$$\phi(\lambda, \mu) = \arg\min_{P \in \mathfrak{B}_K} f(P, \lambda, \mu),$$

where $\mathfrak{B}_K$ is the set of $K \times K$ bistochastic matrices and f is defined as:

$$f(P, \lambda, \mu) \coloneqq \max_{i \in [N]} -\ln(\sum_{j=1}^{K} P_{i,j}\mu_k - \lambda_i) + \frac{1}{2K}\|P\|_2^2$$

Let P^* and $\hat{P}^*$ be the minimizers of f with the respective parameters (λ, μ) and $(\hat{\lambda}, \hat{\mu})$. They are uniquely defined as f is $\frac{1}{K}$ strongly convex.

As the property of Lemma 7.1 is symmetric, we can assume without loss of generality that $f(P^*, \lambda, \mu) \geq f(\hat{P}^*, \hat{\lambda}, \hat{\mu})$.

Given the definition of Δ, we actually have the bound

$$-\ln(\Delta) + \frac{1}{2} \geq f(P^*, \lambda, \mu) \geq -\ln(\Delta).$$

The lower bound holds because the term in the $\ln$ is at most Δ for at least one i. For the upper bound, some matrix P ensures that the term in the $\ln$ is at least Δ for all i and $\|P\|_2^2 \leq K$.

Similarly for $\hat{P}^*$, it comes:

$$-\ln((1 - 2c_1)\Delta) + \frac{1}{2} \geq f(\hat{P}^*, \hat{\lambda}, \hat{\mu}) \geq -\ln((1 + 2c_1)\Delta).$$

As a consequence, it holds for any $i \in [N]$:

$$\begin{aligned} -\ln\left(\sum_{j=1}^{K} \hat{P}^*_{i,j}\hat{\mu}_j - \hat{\lambda}_i\right) &\leq f(\hat{P}^*, \hat{\lambda}, \hat{\mu}) \\ &\leq -\ln((1 - 2c_1)\Delta/\sqrt{e}) \\ \sum_{j=1}^{K} \hat{P}^*_{i,j}\hat{\mu}_j - \hat{\lambda}_i &\geq (1 - 2c_1)\Delta/\sqrt{e}. \end{aligned}$$

Note that for any $i \in [N]$,

$$\sum_{j=1}^{K} \hat{P}^*_{i,j}\hat{\mu}_j - \hat{\lambda}_i \leq \sum_{j=1}^{K} \hat{P}^*_{i,j}\mu_j - \lambda_i + 2\|(\hat{\lambda}-\lambda, \hat{\mu}-\mu)\|_\infty.$$

It then yields the second point of Lemma 7.1:

$$\sum_{j=1}^{K} \hat{P}^*_{i,j}\mu_j - \lambda_i \geq \left((1-2c_1)/\sqrt{e} - 2c_1\right)\Delta$$

Moreover, it comes

$$\begin{aligned}
-\ln\left(\sum_{j=1}^{K} \hat{P}^*_{i,j}\hat{\mu}_j - \hat{\lambda}_i\right) &\geq -\ln\left(\sum_{j=1}^{K} \hat{P}^*_{i,j}\mu_j - \lambda_i\right) - \ln\left(1 + \frac{2\|(\hat{\lambda}-\lambda, \hat{\mu}-\mu)\|_\infty}{\sum_{j=1}^{K} \hat{P}^*_{i,j}\mu_j - \lambda_i}\right) \\
&\geq -\ln\left(\sum_{j=1}^{K} \hat{P}^*_{i,j}\mu_j - \lambda_i\right) - \frac{2\|(\hat{\lambda}-\lambda, \hat{\mu}-\mu)\|_\infty}{\left((1-2c_1)/\sqrt{e} - 2c_1\right)\Delta}
\end{aligned}$$

Recall that for a $\frac{1}{K}$-strongly convex function g of global minimum x^* and any x:

$$\|x - x^*\|_2 \leq 2K(g(x) - g(x^*))$$

As a consequence, it comes:

$$\begin{aligned}
f(\hat{P}^*, \hat{\lambda}, \hat{\mu}) &\geq f(\hat{P}^*, \lambda, \mu) - \frac{2\|(\hat{\lambda}-\lambda, \hat{\mu}-\mu)\|_\infty}{\left((1-2c_1)/\sqrt{e} - 2c_1\right)\Delta} \\
&\geq f(P^*, \lambda, \mu) - \frac{2\|(\hat{\lambda}-\lambda, \hat{\mu}-\mu)\|_\infty}{\left((1-2c_1)/\sqrt{e} - 2c_1\right)\Delta} + \frac{1}{2K}\|P^* - \hat{P}^*\|_2.
\end{aligned}$$

Equation (7.17) then follows.

Proof of Lemma 7.2

The coefficient $C_{i,j}$ is replaced by $+\infty$ as soon as the whole weight $P_{i,j}$ is exhausted. Thanks to this, the HUNGARIAN algorithm does return a perfect matching with respect to the bipartite graph with edges (i, j) where there remains some weight for $P_{i,j}$. Because of this, it can be shown following the usual proof of Birkhoff algorithm (Birkhoff, 1946) that the sequence $(z[j], P[j])$ is indeed of length at most K^2 and is a valid decomposition of P.

Now assume that $\prec_C$ is a total order. At each iteration j of HUNGARIAN algorithm, denote

$\widetilde{P}^j = P - \sum_{s=1}^{j-1} z[s]P[s]$ the remaining weights to attribute.

Let l_j be such that $P[j] = P_{l_j}$ for any iteration j of HUNGARIAN algorithm.

It can now be shown by induction that

$$\widetilde{P}^j = P - \sum_{l=1}^{l_j} z_l(P)P_l.$$

where $z_l(P)$ are defined by Equation (7.5). Indeed, by definition

$$\begin{aligned}\widetilde{P}^{j+1} &= \widetilde{P}^j - z[j+1]P[j+1] \\ &= \widetilde{P}^j - z[j+1]P_{l_{j+1}}\end{aligned}$$

The HUNGARIAN algorithm returns the minimal cost matching with respect to the modified cost matrix C where the coefficients i, k such that $\widetilde{P}^j_{i,k} = 0$ are replaced by $+\infty$. Thanks to this, $P_{l_{j+1}}$ is the minimal cost permutation matrix P_l (for $\prec_C$) such that $\widetilde{P}^j_{i,k} > 0$ for all $(i,k) \in \mathrm{supp}(P_l)$.

This means that for any $l < l_{j+1}$

$$\min_{(i,k)\in\mathrm{supp}(P_l)} (\widetilde{P}^j)_{i,k} = 0.$$

Using the induction hypothesis, this implies that $z_l(P) = 0$ for any $l_j < l < l_{j+1}$. And finally, this also implies that $z_{l_{j+1}}(P) = z[j+1]$.

This finally concludes the proof as $\widetilde{P}^j = 0$ after the last iteration.

Proof of Lemma 7.3

For z and z' the respective decompositions of P and P' defined in Lemma 7.2, then

$$\int_0^1 \mathbb{1}\left(\psi(P)(\omega) \neq \psi(P')(\omega)\right) \mathrm{d}\omega = \mathbb{P}_{\omega\sim U(0,1)}\left(\psi(P)(\omega) \neq \psi(P')(\omega)\right).$$

In the following, note $A = \psi(P)$ and $A' = \psi(P')$. It comes

$$\begin{aligned}\int_0^1 \mathbb{1}\left(\psi(P)(\omega) \neq \psi(P')(\omega)\right) \mathrm{d}\omega &= \sum_{n=1}^{K!} \mathbb{P}(A = P_n \text{ and } A' \neq P_n) \\ &= \frac{1}{2}\sum_{n=1}^{K!} \mathbb{P}(A = P_n \text{ and } A' \neq P_n) + \frac{1}{2}\sum_{n=1}^{K!} \mathbb{P}(A' = P_n \text{ and } A \neq P_n)\end{aligned}$$

$$= \frac{1}{2} \sum_{n=1}^{K!} \mathrm{vol}\left(\left[\sum_{j=1}^{n-1} z_j(P), \sum_{j=1}^{n} z_j(P) \right] \ominus \left[\sum_{j=1}^{n-1} z_j(P'), \sum_{j=1}^{n} z_j(P') \right] \right),$$

where vol denotes the volume of a set and $A \ominus B = (A \setminus B) \cup (B \setminus A)$ is the symmetric difference of A and B. The last equality comes from the expression of ψ with respect to the coefficients $z_j(P)$, thanks to Lemma 7.2.

It is easy to show that

$$\mathrm{vol}\left([a,b] \ominus [c,d]\right) \leq \left(|c-a| + |d-b|\right) \mathbb{1}\left(b > a \text{ or } c > d\right).$$

The previous equality then leads to

$$\begin{aligned}
\int_0^1 \mathbb{1}\left(\psi(P)(\omega) \neq \psi(P')(\omega)\right) \mathrm{d}\omega &\leq \frac{1}{2} \sum_{n=1}^{K!} \left(\left| \sum_{j=1}^{n-1} z_j(P) - z_j(P') \right| + \left| \sum_{j=1}^{n} z_j(P) - z_j(P') \right| \right) \cdot \\
&\qquad \mathbb{1}\left(z_n(P) + z_n(P') > 0\right) \\
&\leq \sum_{n=1}^{K!} \left| \sum_{j=1}^{n} z_j(P) - z_j(P') \right| \mathbb{1}\left(z_n(P) + z_n(P') > 0\right).
\end{aligned} \tag{7.18}$$

The last inequality holds because $\sum_{j=1}^{k} z_j(P) - z_j(P')$ is counted twice when $z_k(P) + z_k(P')$ is positive: when $n = k$ and for the next n such that the elements are counted in the sum.

Thanks to Lemma 7.2, only $2K^2$ elements $z_j(P)$ and $z_j(P')$ are non-zero. Let k_n be the index of the n-th non-zero element of $(z_s(P)+z_s(P'))_{1 \leq s \leq K!}$. Note that $z_s(P')$ can be non-zero while $z_s(P)$ is zero (or conversely). Let also

$$\begin{aligned}
(i_{k_n}, j_{k_n}) &\in \operatorname*{arg\,min}_{(i,j) \in \mathrm{supp}(P_{k_n})} P_{i,j} - \sum_{l < k_n} z_l(P) \mathbb{1}\left((i,j) \in \mathrm{supp}(P_{k_n})\right) \\
(i'_{k_n}, j'_{k_n}) &\in \operatorname*{arg\,min}_{(i,j) \in \mathrm{supp}(P_{k_n})} P'_{i,j} - \sum_{l < k_n} z_l(P') \mathbb{1}\left((i,j) \in \mathrm{supp}(P_{k_n})\right)
\end{aligned}$$

It then comes, thanks to Lemma 7.2

$$\begin{aligned}
z_{k_n}(P) - z_{k_n}(P') &\leq P_{i'_{k_n}, j'_{k_n}} - P'_{i'_{k_n}, j'^{k_n}} - \sum_{l < k_n} \left(z_l(P) - z_l(P')\right) \mathbb{1}\left((i'_{k_n}, j'_{k_n}) \in \mathrm{supp}(P_{k_n})\right) \\
&\leq P_{i'_{k_n}, j'_{k_n}} - P'_{i'_{k_n}, j'_{k_n}} - \sum_{l < n} \left(z_{k_l}(P) - z_{k_l}(P')\right) \mathbb{1}\left((i'_{k_n}, j'_{k_n}) \in \mathrm{supp}(P_{k_n})\right)
\end{aligned}$$

The second inequality holds, because for $l' \notin \{k_l \mid l < 2K^2\}$, the term in the sum is zero by definition of the sequence k_l.

A similar inequality holds for $z_{k_n}(P') - z_{k_n}(P)$, which leads to

$$|z_{k_n}(P) - z_{k_n}(P')| \leq \|P - P'\|_\infty + \sum_{l<n} |z_{k_l}(P) - z_{k_l}(P')|.$$

By induction, it thus holds

$$|z_{k_n}(P) - z_{k_n}(P')| \leq 2^{n-1}\|P - P'\|_\infty.$$

We finally conclude using Equation (7.18)

$$\begin{aligned}
\int_0^1 \mathbb{1}\left(\psi(P)(\omega) \neq \psi(P')(\omega)\right) \mathrm{d}\omega &\leq \sum_{n=1}^{K!} \left|\sum_{j=1}^{n} z_j(P) - z_j(P')\right| \mathbb{1}\left(z_n(P) + z_n(P') > 0\right) \\
&\leq \sum_{n=1} \left|\sum_{j=1}^{k_n} z_j(P) - z_j(P')\right| \\
&\leq \sum_{n=1} \left|\sum_{l=1}^{n} z_{k_l}(P) - z_{k_l}(P')\right| \\
&\leq \sum_{n=1}^{2K^2-1} \sum_{j=1}^{n} 2^{j-1}\|P - P'\|_\infty \\
&\leq 2^{2K^2}\|P - P'\|_\infty.
\end{aligned}$$

In the fourth inequality, the last term of the sum is ignored. It is indeed 0 as z and z' both sum to 1.

Proof of Theorem 7.5

First recall below a useful version of Chernoff bound.

Lemma 7.8. *For any independent variables $X_1, \ldots, X_n$ in $[0,1]$ and $\delta \in (0,1)$,*

$$\mathbb{P}\left(\sum_{i=1}^{n} X_i \leq (1-\delta)\sum_{i=1}^{n} \mathbb{E}[X_i]\right) \leq e^{-\frac{\delta^2 \sum_{i=1}^{n} \mathbb{E}[X_i]}{2}}.$$

We now prove the following concentration lemma.

Lemma 7.9. *For any time $t \geq (N+K)^5$ and $\varepsilon \in (0, \frac{1}{4})$,*

$$\mathbb{P}\left(|\hat{\mu}_k^i(t) - \mu_k| \geq \varepsilon\right) \leq 3\exp\left(-\lambda_i\left(t^{\frac{4}{5}} - 1\right)\varepsilon^2\right)$$

$$\mathbb{P}\left(|\hat{\lambda}_j^i(t) - \lambda_j| \geq \varepsilon\right) \leq 6\exp\left(-\lambda_i K\underline{\mu}\frac{t^{\frac{4}{5}}-1}{145}\varepsilon^2\right).$$

Proof.

Concentration for $\hat{\mu}$. Consider agent i in the following and denote by $N_k(t)$ the number of *exploratory pulls* of this agent on server k at time t. By definition, the probability to proceed to an exploratory pull on the server k at round t is at least $\lambda_i \min(t^{-\frac{1}{5}}, \frac{1}{N+K})$. The term λ_i here appears as a pull is guaranteed if a packet appeared at the current time step. Yet the number of exploratory pulls might be much larger in practice as queues should accumulate a large number of uncleared packets at the beginning.

For $t \geq (N+K)^5$, it holds:

$$\begin{aligned}
\sum_{n=1}^{t} \min(n^{-\frac{1}{5}}, \frac{1}{N+K}) &= \sum_{n=1}^{(N+K)^5} \frac{1}{N+K} + \sum_{n=(N+K)^5+1}^{t} n^{-\frac{1}{5}} \\
&\geq (N+K)^4 + \int_{(N+K)^5}^{t} x^{-\frac{1}{5}}\mathrm{d}x - 1 \\
&\geq \frac{1}{4}\left(5t^{\frac{4}{5}} - (N+K)^4 - 4\right) \\
&\geq t^{\frac{4}{5}} - 1.
\end{aligned}$$

Lemma 7.8 then gives for $N_k(t)$:

$$\begin{aligned}
\mathbb{P}\left(N_k(t) \leq (1-\delta)\mathbb{E}[N_k(t)]\right) &\leq \exp\left(-\frac{\delta^2\mathbb{E}[N_k(t)]}{2}\right) \\
\mathbb{P}\left(N_k(t) \leq (1-\delta)\lambda_i\left(t^{\frac{4}{5}}-1\right)\right) &\leq \exp\left(-\frac{\lambda_i\delta^2\left(t^{\frac{4}{5}}-1\right)}{2}\right).
\end{aligned}$$

Which leads for $\delta = \frac{1}{2}$ to

$$\mathbb{P}\left(N_k(t) \leq \frac{\lambda_i}{2}\left(t^{\frac{4}{5}}-1\right)\right) \leq \exp\left(-\lambda_i\frac{t^{\frac{4}{5}}-1}{8}\right). \tag{7.19}$$

The number of exploratory pulls and the observations on the server k are independent. Thanks to this, Hoeffding's inequality can be directly used as follows

$$\mathbb{P}\left(|\hat{\mu}_k^i(t) - \mu_k| \geq \varepsilon \mid N_k(t)\right) \leq 2\exp\left(-2N_k(t)\varepsilon^2\right).$$

Using Equation (7.19) now gives the first concentration inequality for $\varepsilon \leq \frac{1}{4} \leq \frac{1}{2\sqrt{2}}$:

$$\begin{aligned}\mathbb{P}\left(|\hat{\mu}_k^i(t) - \mu_k| \geq \varepsilon\right) &\leq 2\exp\left(-\lambda_i\left(t^{\frac{4}{5}}-1\right)\varepsilon^2\right) + \exp\left(-\lambda_i\frac{t^{\frac{4}{5}}-1}{8}\right) \\ &\leq 3\exp\left(-\lambda_i\left(t^{\frac{4}{5}}-1\right)\varepsilon^2\right).\end{aligned}$$

Concentration for $\hat{\lambda}$. Consider agent i in the following. First show a concentration inequality for $\widetilde{\mu}$. Denote by $N(t)$ the total number of exploratory pulls on servers proceeded by player i at round t, i.e., $N(t) = \sum_{k=1}^{K} N_k(t)$. Similarly to Equation (7.19), it can be shown that

$$\mathbb{P}\left(N(t) \leq \lambda_i K\frac{t^{\frac{4}{5}}-1}{2}\right) \leq \exp\left(-\lambda_i K\frac{t^{\frac{4}{5}}-1}{8}\right).$$

Lemma 7.8 then gives for $\delta \in (0,1)$:

$$\begin{aligned}\mathbb{P}\left(|\widetilde{\mu} - \overline{\mu}| \geq \delta\overline{\mu}\right) &\leq 2\exp\left(-\lambda_i K\delta^2\overline{\mu}\frac{t^{\frac{4}{5}}-1}{8}\right) + \exp\left(-\lambda_i K\frac{t^{\frac{4}{5}}-1}{8}\right) \\ &\leq 3\exp\left(-\lambda_i K\delta^2\overline{\mu}\frac{t^{\frac{4}{5}}-1}{8}\right).\end{aligned}$$

Note that $|\widetilde{\mu} - \overline{\mu}| \leq \delta\overline{\mu}$ implies $|\frac{1}{\overline{\mu}} - \frac{1}{\widetilde{\mu}}| \leq \frac{\delta}{(1-\delta)\overline{\mu}}$. So this gives the following inequality:

$$\mathbb{P}\left(\left|\frac{1}{\overline{\mu}} - \frac{1}{\widetilde{\mu}}\right| \geq \frac{\delta}{(1-\delta)\overline{\mu}}\right) \leq 3\exp\left(-\lambda_i K\delta^2\overline{\mu}\frac{t^{\frac{4}{5}}-1}{8}\right). \tag{7.20}$$

A concentration bound on $\hat{S}_j^i$ can be shown similarly for any $\delta \in (0,1)$

$$\mathbb{P}\left(\left|\hat{S}_j^i(t) - (1-\frac{\lambda_j}{2})\overline{\mu}\right| \geq \delta\overline{\mu}\right) \leq 3\exp\left(-\lambda_i K\delta^2\overline{\mu}\frac{t^{\frac{4}{5}}-1}{8}\right). \tag{7.21}$$

Now recall that the estimate of λ_j is defined by $\hat{\lambda}_j = 2 - \frac{2\hat{S}_j^i}{\widetilde{\mu}}$. We then have the following identity:

$$\hat{\lambda}_j - \lambda_j = 2(\frac{1}{\widetilde{\mu}} - \frac{1}{\overline{\mu}})\hat{S}_j^i + \frac{2}{\overline{\mu}}\left((1-\frac{\lambda_j}{2})\overline{\mu} - \hat{S}_j^i\right).$$

Since $\hat{S}_j \in [0,1]$, it yields for $\varepsilon \leq \frac{1}{4}$ and $x \in (0,1)$:

$$\mathbb{P}\left(\left|\hat{\lambda}_j^i(t) - \lambda_j\right| \geq \varepsilon\right) \leq \mathbb{P}\left(\left|2(\frac{1}{\widetilde{\mu}} - \frac{1}{\overline{\mu}})\hat{S}_j^i\right| \geq x\varepsilon \text{ or } \left|\frac{2}{\overline{\mu}}\left((1-\frac{\lambda_j}{2})\overline{\mu} - \hat{S}_j\right)\right| \geq (1-x)\varepsilon\right)$$

$$\leq \mathbb{P}\left(\left|\frac{1}{\widetilde{\mu}} - \frac{1}{\overline{\mu}}\right| \geq \frac{x\varepsilon}{2\hat{S}_j^i} \mid \hat{S}_j^i \leq (1 + \frac{1-x}{8})\overline{\mu}\right) + \mathbb{P}\left(\left|\hat{S}_j^i(t) - (1 - \frac{\lambda_j}{2})\overline{\mu}\right| \geq \frac{(1-x)\overline{\mu}\varepsilon}{2}\right)$$
$$\leq \mathbb{P}\left(\left|\frac{1}{\widetilde{\mu}} - \frac{1}{\overline{\mu}}\right| \geq \frac{\delta}{(1-\delta)\overline{\mu}} \text{ for } \delta = \frac{4x\varepsilon}{9}\right) + \mathbb{P}\left(\left|\hat{S}_j^i(t) - (1 - \frac{\lambda_j}{2})\overline{\mu}\right| \geq \frac{(1-x)\overline{\mu}\varepsilon}{2}\right)$$

Taking $x = \frac{9}{17}$ leads to $\frac{4x}{9} = \frac{1-x}{2}$ and thus, using Equations (7.20) and (7.21):

$$\mathbb{P}\left(\left|\hat{\lambda}_j^i(t) - \lambda_j\right| \geq \varepsilon\right) \leq 6\exp\left(-\lambda_i K\left(\frac{8}{17}\right)^2 \overline{\mu}\frac{t^{\frac{4}{5}} - 1}{32}\varepsilon^2\right)$$
$$\leq 6\exp\left(-\lambda_i K\overline{\mu}\frac{t^{\frac{4}{5}} - 1}{145}\varepsilon^2\right)$$

□

In the following, let $c_1 = 0.1$ and $c_2 = \frac{4}{(1-2c_1)/\sqrt{e} - 2c_1} \approx 14$. For a problem instance, let the good event $\mathcal{E}_t$ at time t be defined as

$$\mathcal{E}_t \coloneqq \left\{\|(\hat{\lambda}^i - \lambda, \hat{\mu}^i - \mu)\|_\infty \leq \frac{0.1\Delta^2}{2c_2 2^{2K^2} KN},\ \forall i \in [N]\right\}.$$

As Δ is smaller than 1, the right hand term in the definition of $\mathcal{E}_t$ is smaller than $c_1\Delta$. Thanks to Lemmas 7.1 and 7.3, $\mathcal{E}_t$ then guarantees that any player will collide with another player with probability at most 0.1Δ, i.e., , $\forall i \in [N]$,

$$\mathbb{P}_{\omega\sim\mathcal{U}(0,1)}\left(\exists j \in [N], \hat{A}_t^i(\omega) \neq \hat{A}_t^j(\omega) \mid \mathcal{E}_t\right) \leq 0.1\Delta.$$

Moreover, thanks to Lemma 7.1, under $\mathcal{E}_t$,

$$\lambda_i \leq \sum_{k=1}^{K} \hat{P}_{i,k}\mu_k - \left(\frac{1-2c_1}{\sqrt{e}} - 2c_1\right)\Delta.$$

These two last inequalities lead to the following lemma.

Lemma 7.10. *For $t \geq \frac{2^5K^5}{0.08^5\Delta^5}$, denote by $\mathcal{H}_t$ the history of observations up to round t. Then*

$$\mathbb{E}\left[S_t^i \mid \mathcal{E}_t, \mathcal{H}_t\right] \geq \lambda_i + 0.1\Delta.$$

Proof. This is a direct consequence of the following decomposition:

$$\mathbb{E}\left[S_t^i \mid \mathcal{E}_t, \mathcal{H}_t\right] \geq \overbrace{(1 - (N+K)t^{-\frac{1}{5}})}^{\text{proba to exploit}}\left(\overbrace{\hat{P}_{i,k}\mu_k}^{\text{proba to clear}} - \overbrace{\mathbb{P}\left(\exists j \in [N], \hat{A}_t^i(\omega) \neq \hat{A}_t^j(\omega)\exists \mid \mathcal{E}_t\right)}^{\text{collision proba}}\right)$$

$$\geq (1-(N+K)t^{-\frac{1}{5}})(\lambda_i + \left(\frac{1-2c_1}{\sqrt{e}} - 2c_1\right)\Delta - 0.1\Delta)$$
$$\geq (1-(N+K)t^{-\frac{1}{5}})(\lambda_i + 0.18\Delta).$$

The last inequality is given by $c_1 = 0.1$ and it leads to

$$\mathbb{E}\left[S_t^i \mid \mathcal{E}_t, \mathcal{H}_t\right] \geq \lambda_i + 0.18\Delta - (N+K)t^{-\frac{1}{5}}.$$

For $t \geq \frac{2^5K^5}{0.08^5\Delta^5}$, the last term is smaller than 0.08Δ, giving Lemma 7.10. □

Define the stopping time

$$\tau \coloneqq \min\left\{t \geq \frac{2^5K^5}{0.08^5\Delta^5} \mid \forall t \geq s, \mathcal{E}_s \text{ holds}\right\}. \tag{7.22}$$

Lemma 7.11. *With probability* 1, $\tau < +\infty$ *and for any integer* $r \geq 1$,

$$\mathbb{E}[\tau^r] = \mathcal{O}\left(KN\left(\frac{N^{\frac{5}{2}}K^{\frac{5}{2}}2^{5K^2}}{(\min(1,K\overline{\mu})\underline{\lambda})^{\frac{5}{4}}\Delta^5}\right)^r\right),$$

where the $\mathcal{O}$ *notation hides constant factors that only depend on* r.

Proof. Define for this proof $t_0 = \left\lceil \frac{2^5K^5}{0.08^5\Delta^5} \right\rceil$. By definition, if $\mathcal{E}_t$ does not hold for $t > t_0$, then $\tau \geq t$. As a consequence, for any $t > t_0$ and thanks to Lemma 7.9:

$$\mathbb{P}(\tau \geq t) \leq \mathbb{P}(\neg\mathcal{E}_t)$$
$$\leq (3eKN + 6eN^2)\exp\left(-ct^{\frac{4}{5}}\right),$$

where $c = c_0\frac{\min(1,K\overline{\mu})\underline{\lambda}\Delta^4}{N^2K^22^{4K^2}}$ for some universal constant $c_0 \leq 1$. Note that $\sum_{t=0}^{\infty}\mathbb{P}(\tau = t) < +\infty$ by comparison. Borel-Cantelli lemma then implies that τ is finite with probability 1.

We can now bound the moments of τ:

$$\mathbb{E}[\tau^r] = r\int_0^{\infty} t^{r-1}\mathbb{P}\left(\tau \geq t\right)\mathrm{d}t$$
$$\leq t_0^r + (3eKN + 6eN^2)r\int_0^{\infty} t^{r-1}e^{-ct^{\frac{4}{5}}}\mathrm{d}t.$$

Using the change of variable $u = ct^{\frac{4}{5}}$, it can be shown that

$$\int_0^{\infty} t^{r-1}e^{-ct^{\frac{4}{5}}}\mathrm{d}t = \frac{5}{4}c^{-\frac{5r}{4}}\Gamma\left(\frac{5r}{4}\right),$$

where Γ denotes the Gamma function. It finally allows to conclude:

$$\begin{aligned}\mathbb{E}[\tau^r] &= \mathcal{O}\left(\frac{K^{5r}}{\Delta^{5r}} + KNc^{-\frac{5r}{4}}\right) \\ &= \mathcal{O}\left(KN\left(\frac{N^{\frac{5}{2}}K^{\frac{5}{2}}2^{5K^2}}{(\min(1,K\overline{\mu})\underline{\lambda})^{\frac{5}{4}}\,\Delta^5}\right)^r\right).\end{aligned}$$

□

Let X_t be a random walk biased towards 0 with the following transition probabilities:

$$\begin{aligned}\mathbb{P}(X_{t+1} = X_t + 1) = p,\ \mathbb{P}(X_{t+1} = X_t - 1 | X_t > 0) = q, \\ \mathbb{P}(X_{t+1} = X_t | X_t > 0) = 1 - p - q,\ \mathbb{P}(X_{t+1} = X_t | X_t = 0) = 1 - p,\end{aligned} \tag{7.23}$$

and $X_0 = 0$.

Lemma 7.12. *The non-asymptotic moments of the random walk defined by Equation* (7.23) *are bounded. For any $t > 0, r > 0$:*

$$\mathbb{E}\left[(X_t)^r\right] \leq \frac{r!}{\left(\ln\left(q/p\right)\right)^r}.$$

Proof: Let π be the stationary distribution of the random walk. It verifies the following system of equations:

$$\begin{cases}\pi(z) = p\pi(z-1) + q\pi(z+1) + (1-p-q)\pi(z),\ \forall z > 0 \\ \pi(0) = (1-p)\pi(0) + q\pi(1) \\ \sum \pi(z) = 1\end{cases}$$

which gives:

$$\pi(z) = \frac{q-p}{q}\left(\frac{p}{q}\right)^z.$$

Equivalently, $\pi(z) = \mathbb{P}(\lfloor Y \rfloor = z)$ with Y an exponential random variable of parameter $\ln(q/p)$. This gives:

$$\mathbb{E}_{X\sim\pi}\left[(X)^r\right] \leq \frac{r!}{\left(\ln\left(q/p\right)\right)^r}.$$

Let $\widetilde{X}_t$ be the random walk with the same transition probabilities as X_t and $\widetilde{X}_0 \sim \pi$. For any $t > 0$, $\widetilde{X}_t \sim \pi$. Moreover, for any $t > 0$, $\widetilde{X}_t$ stochasticaly dominates X_t, which terminates the proof. □

Proof of Theorem 7.5. For τ the stopping time defined by Equation (7.22), Lemma 7.11 bounds its moments as follows

$$\mathbb{E}[\tau^r] = \mathcal{O}\left(KN\left(\frac{N^{\frac{5}{2}}K^{\frac{5}{2}}2^{5K^2}}{(\min(1, K\overline{\mu})\underline{\lambda})^{\frac{5}{4}}\,\Delta^5}\right)^r\right).$$

Let

$$p_i = \lambda_i(1 - \lambda_i - 0.1\Delta) \text{ and } q_i = (\lambda_i + 0.1\Delta)(1 - \lambda_i).$$

Let X_t^i be the random walk biased towards 0 with parameters p_i and q_i, with $X_t^i = 0$ for any $t \leq 0$. According to Lemma 7.10, past time τ, Q_t^i is stochastically dominated by the random process $\tau + X_{t-\tau}^i$. Thus, for any $t > 0$, for any $r > 0$

$$\begin{aligned}
\mathbb{E}[\left(Q_i^t\right)^r] &\leq \max(1, 2^{r-1})\left(\mathbb{E}[\tau^r] + \mathbb{E}[(X_{t-\tau}^i)^r]\right) \\
&= \mathcal{O}\left(KN\left(\frac{N^{\frac{5}{2}}K^{\frac{5}{2}}2^{5K^2}}{(\min(1, K\overline{\mu})\underline{\lambda})^{\frac{5}{4}}\,\Delta^5}\right)^r + \frac{1}{\ln(q_i/p_i)^r}\right) \\
&= \mathcal{O}\left(KN\left(\frac{N^{\frac{5}{2}}K^{\frac{5}{2}}2^{5K^2}}{(\min(1, K\overline{\mu})\underline{\lambda})^{\frac{5}{4}}\,\Delta^5}\right)^r + \Delta^{-r}\right) \\
&= \mathcal{O}\left(KN\left(\frac{N^{\frac{5}{2}}K^{\frac{5}{2}}2^{5K^2}}{(\min(1, K\overline{\mu})\underline{\lambda})^{\frac{5}{4}}\,\Delta^5}\right)^r\right).
\end{aligned}$$

□

Chapter 8

Utility/Privacy Trade-off as Regularized Optimal Transport

Strategic information is valuable either by remaining private (for instance if it is sensitive) or, on the other hand, by being used publicly to increase some utility. These two objectives are antagonistic and leaking this information by taking full advantage of it might be more rewarding than concealing it. Unlike classical solutions that focus on the first point, we consider instead agents that optimize a natural trade-off between both objectives. We formalize this as an optimization problem where the objective mapping is regularized by the amount of information revealed to the adversary (measured as a divergence between the prior and posterior on the private knowledge). Quite surprisingly, when combined with the entropic regularization, the Sinkhorn loss naturally emerges in the optimization objective, making it efficiently solvable via better adapted optimization schemes. We empirically compare these different techniques on a toy example and apply them to preserve some privacy in online repeated auctions.

8.1 Introduction

In many economic mechanisms and strategic games involving different agents, asymmetries of information (induced by a private type, some knowledge on the hidden state of Nature, etc.) can and should be leveraged to increase one's utility. When these interactions between agents are repeated over time, preserving some asymmetry (i.e., not revealing private information) can be crucial to guarantee a larger utility in the long run. Indeed, the small short term utility of publicly using information can be overwhelmed by the long term effect of revealing it (Aumann et al., 1995).

Informally speaking, an agent should use, and potentially reveal some private information only if she gets a subsequent utility increase in return. Keeping this information private is no longer a constraint (as in other classical privacy concepts such as differential privacy Dwork et al., 2006) but becomes part of the objective, which is then to decide how and when to use it. For instance, it might happen that revealing everything is optimal or, on the contrary, that a non-revealing policy is the best one. This is roughly similar to a poker player deciding whether to bluff or not. In some situations, it might be interesting to focus solely on the utility even if it implies losing the whole knowledge advantage, while in other situations, the immediate profit for using this advantage is so small that playing independently of it (or bluffing) is better.

After a rigorous mathematical formulation of this utility vs. privacy trade-off, it appears that this problem can be recast as a regularized optimal transport minimization. In the specific case of entropic regularization, this problem has received a lot of interest in the recent years as it induces a computationally tractable way to approximate an optimal transport distance between distributions and has thus been used in many applications (Cuturi, 2013). Our work showcases how the new Privacy Regularized Policy problem benefits in practice from this theory.

Private Mechanisms. Differential privacy is the most widely used private learning framework (Dwork, 2011; Dwork et al., 2006; Reed and Pierce, 2010) and ensures that any single element of the whole dataset cannot be retrieved from the output of the algorithm. This constraint is often too strong for economic applications (as illustrated before, it is sometimes optimal to disclose publicly some private information). f-divergence privacy costs have thus been proposed in recent literature as a promising alternative (Chaudhuri et al., 2019). These f-divergences, such as Kullback-Leibler, are also used by economists to measure the cost of information from a Bayesian perspective, as in the rational inattention literature (Sims, 2003; Matějka and McKay, 2015; Maćkowiak and Wiederholt, 2015). It was only recently that this approach has been considered to measure "privacy losses" in economic mechanisms (Eilat et al., 2019). This model assumes that the designer of the mechanism has some prior belief on the unobserved and private information. After observing the action of the player, this belief is updated and the cost of information corresponds to the KL between the prior and posterior distributions of this private information.

Optimal privacy preserving strategies with privacy constraints have been recently studied in this setting under specific conditions (Eilat et al., 2019). Loss of privacy can however be directly considered as a cost in the overall objective and an optimal strategy reveals information only if it actually leads to a significant increase in utility. Meanwhile, constrained strategies systematically reveal as much as allowed by the constraints, without incorporating the additional cost of this revelation.

Optimal Transport. Finding an appropriate way to compare probability distributions is a major challenge in learning theory. Optimal Transport manages to provide powerful tools to compare distributions in metric spaces (Villani, 2008). As a consequence, it has received an increasing interest these past years (Santambrogio, 2015), especially for generative models (Arjovsky et al., 2017; Genevay et al., 2018; Salimans et al., 2018). However, such powerful distances often come at the expense of heavy and intractable computations, which might not be suitable to learning algorithms. It was recently showcased that adding an entropic regularization term enables fast computations of approximated distances using Sinkhorn algorithm (Sinkhorn, 1967; Cuturi, 2013). Since then, the Sinkhorn loss has also shown promising results for applications such as generative models (Genevay et al., 2016; Genevay et al., 2018), domain adaptation (Courty et al., 2014) and supervised learning (Frogner et al., 2015), besides having interesting theoretical properties (Peyré and Cuturi, 2019; Feydy et al., 2019; Genevay et al., 2019).

Contributions and organization of the chapter. The new framework of Privacy Regularized Policy is motivated by several applications, presented in Section 8.2 and is formalized in Sec-

tion 8.3. This problem is mathematically formulated as some optimization problem (yet eventually in an infinite dimensional space), which is convex if the privacy cost is an f-divergence, see Section 8.4. Also, if the private information space is discrete, this problem admits an optimal discrete distribution. The minimization problem then becomes dimensionally finite, but non-convex.

If the Kullback-Leibler divergence between the prior and the posterior is considered for the cost of information, the equivalence with a Sinkhorn loss minimization problem is shown in Section 8.5. Although non-convex, this new problem formulation allows different optimization techniques developed in Section 8.6 to efficiently compute partially revealing policies. Finally, with a linear utility cost, the problem is equivalent to the minimization of the difference of two convex functions. Using the theories of these specific problems, different optimization methods can be compared, which illustrates the practical aspect of our new model. This is done in Section 8.7, where we also compute partially revealing strategies for repeated auctions.

8.2 Some Applications

Our model is motivated by different applications described in this section: online repeated auctions and learning models on external servers.

8.2.1 Online repeated auctions

When a website wants to sell an advertisement slot, firms such as Google or Criteo take part in an auction to buy this slot for one of their customer, a process illustrated in Figure 8.1. As this interaction happens each time a user lands on the website, this is no longer a one-time auction problem, but repeated auctions where the seller and/or the competitor might observe not just one bid, but a distribution of bids. As a consequence, if a firm were bidding truthfully, seller and other bidders would have access to its true value distribution μ. This has two possible downsides.

First, if the value distribution μ was known to the auctioneer, she could maximize her revenue at the expense of the bidder utility (Amin et al., 2013; Amin et al., 2014; Feldman et al., 2016; Golrezaei et al., 2019), for instance with personalized reserve prices. Second, the auctioneer can sometimes take part in the auction and becomes a direct concurrent of the bidder (this might be a unique characteristic of online repeated auctions for ads). For instance, Google is both running some auction platforms and bidding on some ad slots for their client. As a consequence, if the distribution μ was perfectly known to some concurrent bidder, he could use it in the future, by bidding more or less aggressively or by trying to conquer new markets.

It is also closely related to online pricing or repeated posted price auctions. When a user

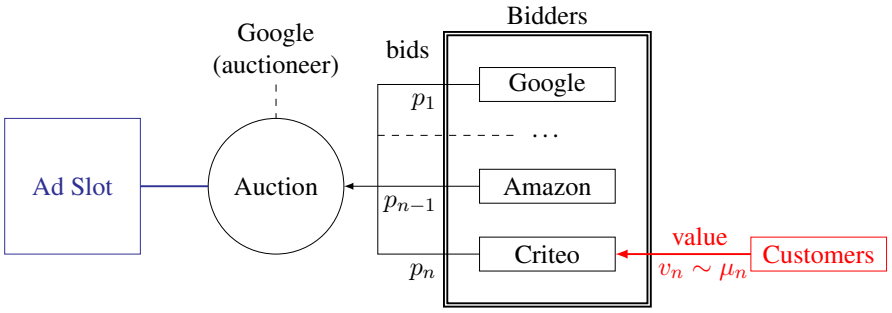


Figure 8.1: Online advertisement auction system.

wants to buy a flight ticket (or any other good), the selling company can learn the value distribution of the buyer and then dynamically adapts its prices in order to increase its revenue. The user can prevent this behavior in order to maximize her long term utility, even if it means refusing some apparently good offers in the short term (in poker lingo, she would be "bluffing").

As explained in Section 8.3.1 below, finding the best possible long term strategy is intractable, as the auctioneer could always adapt to the bidding strategy, leading to an arm race where the bidder and the auctioneer successively adapt to the other one's strategy. Such an arm race is instead avoided by trading-off between the best possible response to the auctioneer's fixed strategy as well as the leaked quantity of information. The privacy loss here aims at bounding the incurred loss in bidder's utility if the auctioneer adapts her strategy using the revealed information.

8.2.2 Learning through external servers

Nowadays, several servers or clusters allow their clients to perform heavy computations remotely, for instance to learn some model parameters (say a deep neural net) for a given training set. The privacy concern when querying a server can sometimes be handled using homomorphic encryption (Gilad-Bachrach et al., 2016; Bourse et al., 2018; Sanyal et al., 2018), if the cluster is designed in that way (typically a public model has been learned on the server). In this case, the client sends an encrypted testing set to the server, receives encrypted predictions and locally recovers the accurate ones. This technique, when available, is powerful, but requires heavy local computations.

Consider instead a client wanting to learn a new model (say, a linear/logistic regression or any neural net) on a dataset that has some confidential component. Directly sending the training set would reveal the whole data to the server owner, besides the risk of someone else observing it. The agent might instead prefer to send noised data, so that the computed model remains close to the accurate one, while keeping secret the true data. If the data contain sensitive information

on individuals, then differential privacy is an appropriate solution. However, it is often the case that the private part is just a single piece of information of the client itself (say, its margin, its current wealth or its total number of users for instance) that is crucial to the final learned model but should not be totally revealed to a competitor. Then differential privacy is no longer the solution, as there is only a single element to protect and/or to use. Indeed, some privacy leakage is allowed and can lead to much more accurate parameters returned by the server and a higher utility at the end; the Privacy Regularized Policy aims at computing the best dataset to send to the server, in order to maximize the utility-privacy trade-off.

8.3 Model

We first introduce a simple toy example in Section 8.3.1 giving insights into the more general problem, whose formal and general formulation is given in Section 8.3.2.

8.3.1 Toy Example

Suppose an agent is publicly playing an action $x \in \mathcal{X}$ to minimize a loss $x^\top c_k$, where c_k is some vector. The true type $k \in [K]$ is only known to the agent and drawn from a prior p_0. Without privacy concern, the agent would then solve for every k: $\min_{x\in\mathcal{X}} x^\top c_k$.
Let us denote by x_k^* the optimal solution of that problem. Besides maximizing her reward, the agent actually wants to protect the secret type k. After observing the action x taken by the agent, an adversary updates her posterior distribution of the hidden type p_x.

If the agent were to play deterministically x_k^* when her type is k, then the adversary could infer the true value of k based on the played action. The agent should instead choose her action randomly to hide her true type to the adversary. Given a type k, the strategy of the agent is then a probability distribution μ_k over $\mathcal{X}$ and her expected reward is $\mathbb{E}_{x\sim\mu_k}[x^\top c_k]$. In this case, the posterior distribution after playing the action x is computed using Bayes rule and if the different μ_k have overlapping supports, then the posterior distribution is no longer a Dirac mass, i.e., some asymmetry of information is maintained.

The agent aims at simultaneously minimizing both the utility loss and the amount of information given to the adversary. A common way to measure the latter is given by the Kullback-Leibler (KL) divergence between the prior and the posterior (Sims, 2003): $\mathrm{KL}(p_x, p_0) = \sum_{k=1}^K \log\left(\frac{p_x(k)}{p_0(k)}\right) p_x(k)$, where $p_x(k) = \frac{p_0(k)\mu_k(x)}{\sum_{l=1}^K p_0(l)\mu_l(x)}$. If the information cost scales in utility with $\lambda > 0$, the regularized loss of the agent is then $x^\top c_k + \lambda \mathrm{KL}(p_x, p_0)$ instead of $x^\top c_k$.

Overall, the global objective of the agent is the following minimization:

$$\min_{\mu_1,\dots,\mu_K} \sum_{k=1}^{K} p_0(k)\mathbb{E}_{x\sim\mu_k}\left[x^\top c_k + \lambda \mathrm{KL}(p_x, p_0)\right].$$

In the limit case $\lambda = 0$, the agent follows a totally revealing strategy and deterministically plays x_k^* given k. When $\lambda = \infty$, the agent focuses on perfect privacy and looks for the best action chosen independently of the type: $x \perp\!\!\!\perp k$. It corresponds to a so called non-revealing strategy in game theory and the best strategy is then to play $\arg\min_x x^\top c[p_0]$ where $c[p_0] = \sum_{k=1}^{K} p_0(k)c_k$. For a positive λ, the behavior of the player will then interpolate between these two extreme strategies.

This problem is related to repeated games with incomplete information (Aumann et al., 1995), where players have private information affecting their utility functions. Playing some action leaks information to the other players, who then change their strategies in consequence. The goal is then to control the amount of information leaked to the adversaries in order to maximize one's own utility. In practice, it can be impossible to compute the best adversarial strategy, e.g., the player is unaware of how the adversaries would adapt. The utility loss caused by adversarial actions is then modeled as a function of the amount of revealed information.

8.3.2 General model

We now introduce formally the general model sketched by the previous toy example. The agent (or player) has a private type $y \in \mathcal{Y}$ drawn according to a prior p_0 whose support can be infinite. She then chooses an action $x \in \mathcal{X}$ to maximize her utility, which depends on both her action and her type. Meanwhile, she wants to hide the true value of her type y. A strategy is thus a mapping $\mathcal{Y} \to \mathcal{P}(\mathcal{X})$, where $\mathcal{P}(\mathcal{X})$ denotes the set of distributions over $\mathcal{X}$; for the sake of conciseness, we denote by $X|Y \in \mathcal{P}(\mathcal{X})^{\mathcal{Y}}$ such a strategy. In the toy example, this mapping was given by $k \mapsto \mu_k$. The adversary observes her action x and tries to infer the type of the agent. We assume a perfect adversary, i.e., she can compute the exact posterior distribution p_x.

Let $c(x, y)$ be the utility loss for playing $x \in \mathcal{X}$ with the type $y \in \mathcal{Y}$. The cost of information is $c_{\text{priv}}(X, Y)$ where (X, Y) is the joint distribution of the action and the type. In the toy example given in Section 8.3.1, the utility cost was given by $c(x, k) = x^\top c_k$ and the privacy cost was the expected KL divergence between p_x and p_0. The previous frameworks aimed at minimizing the utility loss with a privacy cost below some threshold $\varepsilon > 0$, i.e., minimize $\mathbb{E}_{(x,y)\sim(X,Y)}\left[c(x, y)\right]$ such that $c_{\text{priv}}(X, Y) \leq \varepsilon$. Here, this privacy loss has some utility scaling with $\lambda > 0$, which can be seen as the value of information. The final objective of the agent is then to minimize the

following loss:

$$\inf_{X|Y\in\mathcal{P}(\mathcal{X})^{\mathcal{Y}}} \mathbb{E}_{(x,y)\sim(X,Y)}\big[c(x,y)\big] + \lambda\, c_{\text{priv}}(X,Y). \tag{8.1}$$

As mentioned above, the cost of information is here defined as a measure between the posterior p_x and the prior distribution p_0 of the type, i.e., $c_{\text{priv}}(X,Y) = \mathbb{E}_{x\sim X} D(p_x, p_0)$ for some function D[1]. In the toy example of Section 8.3.1, $D(p_x,p_0) = \mathrm{KL}(p_x,p_0)$, which is a classical cost of information in economics.

For a distribution $\gamma \in \mathcal{P}(\mathcal{X}\times\mathcal{Y})$, we denote by $\pi_{1\#}\gamma$ (resp. $\pi_{2\#}\gamma$) the marginal distribution of X (resp. Y): $\pi_{1\#}\gamma(A) = \gamma(A\times\mathcal{Y})$ and $\pi_{2\#}\gamma(B) = \gamma(\mathcal{X}\times B)$. In order to have a simpler formulation of the problem, we remark that instead of defining a strategy by the conditional distribution $X|Y$, it is equivalent to see it as a joint distribution γ of (X,Y) with a marginal over the type equal to the prior: $\pi_{2\#}\gamma = p_0$. The remaining of the chapter focuses on the problem below, which we call **Privacy Regularized Policy.** With the privacy cost defined as above, the minimization problem (8.1) is equivalent to

$$\inf_{\substack{\gamma\in\mathcal{P}(\mathcal{X}\times\mathcal{Y})\\ \pi_{2\#}\gamma=p_0}} \int_{\mathcal{X}\times\mathcal{Y}} [c(x,y) + \lambda\, D(p_x,p_0)]\,\mathrm{d}\gamma(x,y). \tag{PRP}$$

8.4 A convex minimization problem

In this section, we study some theoretical properties of the Problem (PRP). We first recall the definition of an f-divergence.

Definition 8.1. *D is an f-divergence if for all distributions P, Q such that P is absolutely continuous w.r.t. Q, $D(P,Q) = \int_{\mathcal{Y}} f\left(\frac{\mathrm{d}P(y)}{\mathrm{d}Q(y)}\right)\mathrm{d}Q(y)$ where f is a convex function defined on $\mathbb{R}_+^*$ with $f(1)=0$.*

The set of f-divergences includes common divergences such as the Kullback-Leibler divergence ($t\log(t)$), the reverse Kullback-Leibler ($-\log(t)$) or the Total Variation distance ($0.5|t-1|$).

Also, the min-entropy defined by $D(P,Q) = \log\left(\operatorname{ess\,sup} \mathrm{d}P/\mathrm{d}Q\right)$ is widely used for privacy (Tóth et al., 2004; Smith, 2009). It corresponds to the limit of the Renyi divergence $\ln\left(\sum_{i=1}^n p_i^\alpha q_i^{1-\alpha}\right)/(\alpha-1)$, when $\alpha\to+\infty$ (Rényi, 1961; Mironov, 2017). Although it is not an f-divergence, the Renyi divergence derives from the f-divergence associated to the convex function $t\mapsto (t^\alpha-1)/(\alpha-1)$. f-divergence costs have been recently considered in the computer science literature in a non-Bayesian case and then present the good properties of convexity, composition and post-processing invariance (Chaudhuri et al., 2019).

[1] We here favor ex-ante costs as they suggest that the value of information can be heterogeneous among types.

In the remaining of this chapter, D is an f-divergence. (PRP) then becomes a convex minimization problem.

Theorem 8.1. *If D is an f-divergence,* (PRP) *is a convex problem in $\gamma \in \mathcal{P}(\mathcal{X}\times\mathcal{Y})$*[2].

Proof. The constraint set is obviously convex. The first part of the integral is linear in γ. It thus remains to show that the privacy loss is also convex in γ. As D is an f-divergence, the privacy cost is

$$\begin{aligned} c_{\text{priv}}(\gamma) &\coloneqq \int_{\mathcal{X}\times\mathcal{Y}} D\left(p_x, p_0\right) \mathrm{d}\gamma(x,y) \\ &= \int_{\mathcal{X}}\int_{\mathcal{Y}} f\Big(\frac{\mathrm{d}\gamma(x,y)}{\mathrm{d}\gamma_1(x)\mathrm{d}p_0(y)}\Big)\mathrm{d}p_0(y)\mathrm{d}\gamma_1(x), \end{aligned}$$

where $\gamma_1 = \pi_{1\#}\gamma$. For $t \in (0,1)$ and two distributions γ and μ, we can define the convex combination $\nu = t\gamma + (1-t)\mu$. By linearity of the projection π_1, $\nu_1 = t\gamma_1 + (1-t)\mu_1$. The convexity of c_{priv} actually results from the convexity of the *perspective* of f defined by $g(x_1,x_2) = x_2 f(x_1/x_2)$ (Boyd and Vandenberghe, 2004). It indeed implies

$$f\Big(\frac{\mathrm{d}\nu}{\mathrm{d}\nu_1\mathrm{d}p_0}\Big)\mathrm{d}\nu_1 \leq tf\Big(\frac{\mathrm{d}\gamma}{\mathrm{d}\gamma_1\mathrm{d}p_0}\Big)\mathrm{d}\gamma_1 + (1-t)f\Big(\frac{\mathrm{d}\mu}{\mathrm{d}\mu_1\mathrm{d}p_0}\Big)\mathrm{d}\mu_1.$$

The result then directly follows when summing over $\mathcal{X}\times\mathcal{Y}$. □

Although $\mathcal{P}(\mathcal{X}\times\mathcal{Y})$ has generally an infinite dimension, it is dimensionally finite if both sets $\mathcal{X}$ and $\mathcal{Y}$ are discrete. A minimum can then be found using classical optimization methods. In the case of bounded low dimensional spaces $\mathcal{X}$ and $\mathcal{Y}$, they can be approximated by finite grids. However, the size of the grid grows exponentially with the dimension and another approach is needed for large dimensions of $\mathcal{X}$ and $\mathcal{Y}$.

8.4.1 Discrete type space

We assume here that $\mathcal{X}$ is an infinite action space and $\mathcal{Y}$ is of cardinality K (or equivalently, p_0 is a discrete prior of size K), so that $p_0 = \sum_{k=1}^K p_0^k \delta_{y_k}$. For a fixed joint distribution γ, let the measure μ_k be defined for any $A \subset \mathcal{X}$ by $\mu_k(A) = \gamma(A\times\{y_k\})$ and $\mu = \sum_{k=1}^K \mu_k = \pi_{1\#}\gamma$. The function $p^k(x) = \frac{\mathrm{d}\mu_k(x)}{\mathrm{d}\mu(x)}$, defined over the support of μ by absolute continuity, is the posterior probability of having the type k when playing x. The tuple $(\mu, (p^k)_k)$ exactly

[2] It is convex in a usual sense and not geodesically here.

determines γ. (PRP) is then equivalent to:

$$\inf_{\substack{\mu,(p^k(\cdot))\\ p^k\geq 0,\sum_{l=1}^K p^l(\cdot)=1}} \sum_k \int_{\mathcal{X}} [p^k(x)c(x,y_k) + \lambda p_0^k f\left(\frac{p^k(x)}{p_0^k}\right)]\mathrm{d}\mu(x) \quad \text{such that for all } k\leq K, \int_{\mathcal{X}} p^k(x)\mathrm{d}\mu(x) = p_0^k. \tag{8.2}$$

For fixed posterior distributions p^k, this is a generalized moment problem on the distribution μ (Lasserre, 2001). The same types of arguments can then be used for the existence and the form of optimal solutions.

Theorem 8.2. *If the prior is dicrete of size K, for all $\varepsilon > 0$,* (PRP) *has an ε-optimal solution such that $\pi_{1\#}\gamma = \mu$ has a finite support of at most $K+2$ points.*
Furthermore, if $\mathcal{X}$ is compact and $c(\cdot, y_k)$ is lower semi-continuous for every k, then it also holds for $\varepsilon = 0$.

Proof. For $\varepsilon > 0$, let $(p^k)_k$ and μ be an ε-optimal solution. We define

$$\begin{cases} g_0(x) \coloneqq \sum_k [p^k(x)c(x,y_k) + \lambda p_0^k(x) f\left(\frac{p^k(x)}{p_0^k}\right)], \\ g_k(x) \coloneqq p^k(x) \text{ for } k \in \{1,\dots,K\}. \end{cases}$$

Let $\alpha_j(\mu) = \int_{\mathcal{X}} g_j \mathrm{d}\mu$ for $j \in \{0,\dots,K\}$. The considered solution μ is included in a convex hull as follows:

$$(\alpha_j(\mu))_{0\leq j\leq K} \in \mathrm{Conv}\{(g_j(x))_{0\leq j\leq K} \,/\, x\in\mathcal{X}\}.$$

So by Caratheodory theorem, there are $K+2$ points $x_i \in \mathcal{X}$ and $(t_i) \in \Delta_{K+2}$ such that $\alpha_j(\mu) = \sum_{i=1}^{K+2} t_i g_j(x_i)$ for any j. Let $\mu' = \sum_{i=1}^{K+2} t_i \delta_{x_i}$. We then have $\alpha_j(\mu') = \alpha_j(\mu)$ for all j, which means that $(\mu', (p^k)_k)$ is also an ε-optimal solution of the problem (8.2) and the support of μ' is of size at most $K+2$.

Now assume that $\mathcal{X}$ is compact and the $c(\cdot, y_k)$ are lower semi-continuous. The first part of Theorem 8.2 that we just proved leads to Corollary 8.1, which is given below and claims that (PRP) is equivalent to its discrete version given by equation (8.3). We consider the formulation of equation (8.3) in the remaining of the proof.

Define $h_k(\gamma_i) \coloneqq \left(\sum_{m=1}^K \gamma_{i,m}\right) f\left(\frac{\gamma_{i,k}}{p_0^k \sum_{m=1}^K \gamma_{i,m}}\right)$, with the conventions $f(0) = \lim_{x\to 0} f(x) \in \mathbb{R}\cup\{+\infty\}$ and $h_k(\gamma_i) = 0$ if $\sum_{m=1}^K \gamma_{i,m} = 0$.

The privacy cost is then the sum of the $h_k(\gamma_i)$ for all k and i. The case $\varepsilon = 0$ comes from the lower semi-continuity of the objective function, as claimed by Lemma 8.1 proven below.

Lemma 8.1. *For any k in $\{1,\dots,K\}$, h_k is lower semi-continuous.*

Let $(\gamma^{(n)}, x^{(n)})_n$ be a feasible sequence whose value converges to this infimum. By compacity, we can assume after extraction that $(x^{(n)}, \gamma^{(n)}) \to (x, \gamma)$. As $c(\cdot, y_k)$ and h_k are all lower semi-continuous, the infimum is reached in (γ, x). □

Proof of Lemma 8.1. f is convex and thus continuous on $\mathbb{R}_+^*$. If $\lim_{x\to 0^+} f(x) \in \mathbb{R}$, then f can be extended as a continuous function on $\mathbb{R}_+$ and all the h_k are thus continuous.

Otherwise by convexity, $\lim_{x\to 0^+} f(x) = +\infty$. Thus, h_k is continuous at γ_i as soon as $\gamma_{i,j} > 0$ for every j. If $\gamma_{i,k} = 0$, but the sum $\sum_{l=1}^K \gamma_{i,l}$ is strictly positive, then $h_k(\gamma_i) = +\infty$; but as soon as $\rho \to \gamma$, we also have an infinite limit.

If $\sum_{l=1}^K \gamma_{i,l} = 0$, then $\liminf_{\rho\to\gamma} f\Big(\frac{\rho_{i,k}}{p_0^k \sum_l \rho_{i,l}}\Big) \in \mathbb{R} \cup \{+\infty\}$. This term is multiplied by a factor going to 0, so $\liminf_{\rho\to\gamma} h_k(\rho_i) \geq 0 = h_k(\gamma_i)$. Finally, h_k is lower semi-continuous in all the cases. □

If the support of γ is included in $\{(x_i, y_k) \mid 1 \leq i \leq K+2,\ 1 \leq k \leq K\}$, it can be denoted it as a matrix $\gamma_{i,k} \coloneqq \gamma(\{(x_i, y_k)\})$.

Corollary 8.1. *In the case of a discrete prior,* (PRP) *is equivalent to:*

$$\begin{gathered}\inf_{(\gamma,x)\in\mathbb{R}_+^{(K+2)\times K}\times\mathcal{X}^{K+2}} \sum_{i,k} \gamma_{i,k}\, c(x_i, y_k) + \lambda \sum_{i,k} \gamma_{i,k} D(p_{x_i}, p_0) \\ \text{such that } \forall k \leq K,\ \sum_i \gamma_{i,k} = p_0^k.\end{gathered} \tag{8.3}$$

Proof. Theorem 8.2 claims that (PRP) is equivalent to the problem of Corollary 8.1 if we also impose $x_i \neq x_j$ for $i \neq j$. The value of problem (8.3) is thus lower than the value of (PRP) as we consider a larger feasible set. Let us consider a redundant solution (γ, x) with $x_i = x_j$ for $i \neq j$. It remains to show that a non redundant version of this solution has a lower value.

The functions h_k defined in the proof of Theorem 8.2 are convex as the perspectives of convex functions (Boyd and Vandenberghe, 2004). Also, they are obviously homogeneous of degree 1. These two properties imply that the h_k are subadditive. Thus, let (γ', x') be defined by

$$\begin{cases} \gamma'_{l,k} \coloneqq \gamma_{l,k} \text{ for any } l \notin \{i, j\}, \\ \gamma'_{i,k} \coloneqq \gamma_{i,k} + \gamma_{j,k}, \\ \gamma'_{j,k} \coloneqq 0 \end{cases} \quad \text{and} \quad \begin{cases} x'_l \coloneqq x_l \text{ for any } l \neq j, \\ x'_j \in \mathcal{X} \setminus \{x_l \mid 1 \leq l \leq K+2\}. \end{cases}$$

The subadditivity of h_k implies $h_k(\gamma'_i) + h_k(\gamma'_j) \leq h_k(\gamma_i) + h_k(\gamma_j)$ for any k. The other terms

in the objective function will be the same for (γ, x) and (γ', x'). It thus holds

$$\sum_{i,k} \gamma_{i,k} c(x_i, y_k) + \lambda \sum_{i,k} p_0^k h_k(\gamma_i) \geq \sum_{i,k} \gamma'_{i,k} c(x'_i, y_k) + \lambda \sum_{i,k} p_0^k h_k(\gamma'_i).$$

(γ', x') is in the feasible set of the problem of Corollary 8.1 and we removed a redundant condition from x. We can thus iteratively construct a solution $(\widetilde{\gamma}, \widetilde{x})$ until reaching non redundancy. We then have $(\widetilde{\gamma}, \widetilde{x})$ a non redundant solution with a lower value than (γ, x), i.e., allowing redundancy does not change the infimum. □

Although it seems easier to consider the dimensionally finite problem given by Corollary 8.1, it is not jointly convex in (γ, x). No general algorithms exist to efficiently minimize non-convex problems. We refer the reader to (Horst et al., 2000) for an introduction to non-convex optimization.

The next sections reformulate the problem to better understand its structure, leading to optimization methods reaching better local minima.

8.5 Sinkhorn Loss minimization

Formally, (PRP) is expressed as Optimal Transport Minimization for the utility cost c with a regularization given by the privacy cost. This section considers the Kullback-Leibler divergence for privacy cost. In this case, the problem becomes a Sinkhorn loss minimization, which presents computationally tractable schemes (Peyré and Cuturi, 2019). If the privacy cost is the KL divergence between the posterior and the prior, i.e., $f(t) = t\log(t)$, then the regularization term corresponds to the mutual information $I(X;Y)$, which is the classical cost of information in economics.

The Sinkhorn loss for distributions $(\mu, \nu) \in \mathcal{P}(\mathcal{X}) \times \mathcal{P}(\mathcal{Y})$ is defined by

$$\begin{aligned} \mathrm{OT}_{c,\lambda}(\mu, \nu) \coloneqq \min_{\gamma \in \Pi(\mu,\nu)} & \int c(x,y) \mathrm{d}\gamma(x,y) \\ & + \lambda \int \log\left(\frac{\mathrm{d}\gamma(x,y)}{\mathrm{d}\mu(x)\mathrm{d}\nu(y)}\right) \mathrm{d}\gamma(x,y), \end{aligned} \tag{8.4}$$

where $\Pi(\mu,\nu) = \{\gamma \in \mathcal{P}(\mathcal{X}\times\mathcal{Y}) \mid \pi_{1\#}\gamma = \mu \text{ and } \pi_{2\#}\gamma = \nu\}$. Problem (PRP) with the privacy cost given by the Kullback-Leibler divergence is actually a Sinkhorn loss minimization problem.

Theorem 8.3. *Problem* (PRP) *with* $D = \mathrm{KL}$ *is equivalent to*

$$\inf_{\mu \in \mathcal{P}(\mathcal{X})} \mathrm{OT}_{c,\lambda}(\mu, p_0). \tag{8.5}$$

Proof. Observe that $\frac{\mathrm{d}\gamma(x,y)}{\mathrm{d}\mu(x)}$ is the posterior probability $\mathrm{d}p_x(y)$, thanks to Bayes rule. The regularization term in equation (8.4) then corresponds to $D(p_x, p_0)$ as $p_0 = \nu$ and $D = \mathrm{KL}$ here. The minimization problem given by equation (8.4) is thus equivalent to equation (PRP) with the additional constraint $\pi_{1\#}\gamma = \mu$. Minimizing without this constraint is thus equivalent to minimizing the Sinkhorn loss over all action distributions μ. □

While the regularization term is usually only added to speed up the computations of optimal transport, it here directly appears in the cost of the original problem since it corresponds to the privacy cost! An approximation of $\mathrm{OT}_{c,\lambda}(\mu, \nu)$ can then be quickly computed for discrete distributions using Sinkhorn algorithm (Cuturi, 2013), described in Section 8.5.1.

Notice that the definition of Sinkhorn loss sometimes differs in the literature and instead uses $\int \log(\mathrm{d}\gamma(x,y))\,\mathrm{d}\gamma(x,y)$ for the regularization term. When μ and ν are both fixed, the optimal transport plan γ remains the same. As μ is varying here, these notions yet become different. For this alternative definition, a minimizing distribution μ would actually be easy to compute. It is much more complex in our problem because of the presence of μ in the denominator of the logarithmic term.

With a discrete prior, we can then look for a distribution $\mu = \sum_{j=1}^{K+2} \alpha_j \delta_{x_j}$. In case of a continuous prior, it could still be approximated using sampled discrete distributions as previously done for generative models (Genevay et al., 2018; Genevay et al., 2019).

Besides being a new interpretation of Sinkhorn loss, this reformulation allows a better understanding of the problem structure and reduces the dimension of the considered distributions.

8.5.1 Computing Sinkhorn loss

It was recently suggested to use the Sinkhorn algorithm, which has a linear convergence rate, to compute $\mathrm{OT}_{c,\lambda}(\mu, \nu)$ for distributions $\mu = \sum_{i=1}^{n} \alpha_i \delta_{x_i}$ and $\nu = \sum_{j=1}^{m} \beta_j \delta_{y_j}$ (Knight, 2008; Cuturi, 2013). With K the exponential cost matrix defined by $K_{i,j} = e^{-\frac{c(x_i, y_j)}{\lambda}}$, the unique matrix γ solution of the problem (8.4) has the form $\mathrm{diag}(u) K \mathrm{diag}(v)$. The Sinkhorn algorithm then updates alternatively $u \leftarrow \alpha / Kv$ and $v \leftarrow \beta / K^\top u$ (with component-wise division) for n iterations or until convergence.

8.6 Minimization schemes

Despite the equivalence between (PRP) and the minimization of Sinkhorn loss given by equation (8.5), minimizing this quantity remains an open problem. This section suggests different possible optimization methods in this direction.

8.6.1 Optimization methods

Convex minimization over a distribution set. Problems (PRP) and (8.5) are both of the form

$$\min_{\mu \in \mathcal{P}(\mathcal{X})} J(\mu), \tag{8.6}$$

with J convex. Although solving such a problem is unknown in general, some methods are possible in specific cases (see e.g., Chizat and Bach, 2018, for a short overview).

For polynomial costs, this problem can be solved using generalized moment approaches (Lasserre, 2001), but the complexity explodes with the degree of the polynomial.

$\mathcal{P}(\mathcal{X})$ is the convex hull of Dirac distributions on $\mathcal{X}$, so Frank-Wolfe algorithm might be a good choice (Jaggi, 2013), especially to guarantee sparsity of the returned distribution using *away-steps* technique (Guélat and Marcotte, 1986; Clarkson, 2010). Unfortunately, the Franke-Wolfe algorithm requires at each step to solve a subproblem, which is here equivalent to

$$\arg\max_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p_0(y) \exp\left(\frac{g(y) - c(x,y)}{\varepsilon}\right),$$

where g depends on the previous optimization step. This problem is computationally intractable for most cost functions, making Frank-Wolfe methods unadapted to our problem.

Non-convex minimization. Minimizing over the set of distributions remains solved only for specific cases. The most common approach instead approximates problem (8.6) by discretizing it as

$$\min_{\substack{x \in \mathcal{X}^m \\ \alpha \in \Delta_m}} J\left(\sum_{i=1}^{m} \alpha_i \delta_{x_i}\right). \tag{8.7}$$

Although this dimensionally finite problem is not convex, recent literature has shown the absence of spurious local minima for a large number of particles m (over-parameterization). These results yet hold only under restrictive conditions on the loss function and problem structure (Li and Yuan, 2017; Soudry and Hoffer, 2017; Soltanolkotabi et al., 2018; Venturi et al., 2018; Chizat and Bach, 2018), which are adapted to optimization with neural networks. None of these conditions are satisfied here, making the benefit from over-parameterization uncertain. The empirical results in Section 8.7.2 yet suggest that such a phenomenon might also hold in our setting.

In general, reaching global optimality in non-convex minimization is intractable (Hendrix and Boglárka, 2010; Sergeyev et al., 2013), so we only aim at computing local minima. In practice, RMSProp and ADAM are often considered as the best algorithms in such cases, as they tend to avoid bad local minima thanks to the use of specific momentums (Hinton et al.,

2012; Kingma and Ba, 2014). They yet remain little understood in theory (Reddi et al., 2019; Zou et al., 2019).

Minimax formulation. Note that the dual formulation (Peyré and Cuturi, 2019, Proposition 4.4) of Equation (8.4) allows the following formulation of the optimization problem (8.5):

$$\min_{\mu\in\mathcal{P}(\mathcal{X})} \max_{\substack{f\in\mathcal{C}(\mathcal{X})\\ g\in\mathcal{C}(\mathcal{Y})}} \langle \mu, f\rangle + \langle p_0, g\rangle - \lambda\big\langle\, \mu\otimes p_0,\, \exp\left((f\oplus g - c)/\lambda\right)\big\rangle, \tag{8.8}$$

where $\langle \mu, f\rangle \coloneqq \int_{\mathcal{X}} f(x)\mathrm{d}\mu(x)$ for a distribution μ and a continuous function f on $\mathcal{X}$, $\mu\otimes p_0$ is the product distribution and $f\oplus g(x,y) = f(x)+g(y)$. This corresponds to a minimax problem of the form $\min_x \max_y \psi(x,y)$ where $\psi(\cdot, y)$ is convex for any y and $\psi(x,\cdot)$ is concave for any x. Such problems appear in many applications and have been extensively studied. We refer to (Nedić and Ozdaglar, 2009; Chambolle and Pock, 2016; Thekumparampil et al., 2019; Lin et al., 2020) for detailed surveys on the topic.

As we are considering the discretized problem (8.7), we are actually in the nonconvex-concave setting where ψ is nonconvex on its first variable and concave on its second. Algorithms with theoretical convergence rates to local minima have been studied in this specific setting (Rafique et al., 2018; Lin et al., 2019; Nouiehed et al., 2019; Thekumparampil et al., 2019; Lu et al., 2020; Ostrovskii et al., 2020; Lin et al., 2020). Most of them alternate (accelerated) gradient descent on x and gradient ascent on y, while considering a regularized version ψ_ε of ψ.

Their interests are mostly theoretical as ADAM and RMSProp on the first coordinate instead of gradient descent should converge to better local minima in practice, similarly to nonconvex minimization. In practice, they still provide good heuristics as shown in Section 8.7.2.

On minimizing Sinkhorn divergence. Ballu et al. (2020) recently proposed a method to solve the minimization problem (8.5). Unfortunately, they consider discrete distributions and focus on reducing the dependency in the size of their supports. More importantly, this method adds a regularization term $\eta KL(\mu,\beta)$ for some reference measure β and requires this regularizer to be more significant than the one originally in the Sinkhorn loss, i.e., $\eta \geq \lambda$. While this does not add any trouble when considering regimes where both are close to 0, we here consider fixed λ, potentially far from 0 as explained in Section 8.5. The scaling factor η thus cannot be negligible, making this method unadapted to our case.

8.6.2 Different algorithms

Using these previous formulations, we propose several algorithms to solve the optimization problem (8.5), which are compared experimentally in Section 8.7.2. As explained above, we consider the discrete but non-convex formulation:

$$\min_{\substack{x\in\mathcal{X}^m\\ \alpha\in\Delta_m}} \mathrm{OT}_{c,\lambda}\left(\sum_{i=1}^m \alpha_i\delta_{x_i}, p_0\right). \tag{8.9}$$

We first consider ADAM and RMSProp algorithms for this problem. Note that the gradient of the Sinkhorn loss (Feydy et al., 2019) is given by $\nabla\mathrm{OT}_{c,\lambda}(\mu,\nu) = (f,g)$, where f and g are the solutions of the dual problem given by equation (8.8), i.e., $(f,g) = \lambda(\ln(u),\ln(v))$ where u and v are the vectors computed by the Sinkhorn algorithm presented in Section 8.5.1. The gradient of $\mathrm{OT}_{c,\lambda}$ can then only be approximated, as it is the solution of an optimization problem. Luckily, first order optimization methods can still be used with inexact gradients (Devolder et al., 2014). Two approximations of the gradient are possible.

***Analytic Differentiation*:** $\nabla\mathrm{OT}_{c,\lambda}(\mu,\nu)$ is approximated by $(f^{(n)},g^{(n)})$, which are the dual variables obtained after n iterations of the Sinkhorn algorithm.

***Automatic Differentiation*:** the gradient is computed via the chain rule over the successive operations processed during the Sinkhorn algorithm.

These two methods have been recently compared by Ablin et al. (2020) and showed to roughly perform similarly for the same computation time.

For each optimization step, the gradient $\nabla\mathrm{OT}_{c,\lambda}$ is approximated by computing $(u_t^{(k+1)}, v_t^{(k+1)}) \leftarrow (\alpha/Kv_t^{(k)}, \beta/K^\top u_t^{(k+1)})$ for n iterates. However, if the distribution μ_t did not significantly change since the last step, the gradient does not change too much as well. Instead of starting the Sinkhorn algorithm from scratch ($u_t^{(0)} = \mathbf{1}$), we instead want to use the last optimization step ($u_t^{(0)} = u_{t-1}^{(n)}$) to converge faster. Note that this technique, which we call *warm restart*, cannot be coupled with *automatic differentiation* as it would require nt backpropagation operations for the optimization step t.

The iteration step $(u,v) \leftarrow (\alpha/Kv, \beta/K^\top u)$ actually corresponds to a gradient ascent step on (f,g) in the minimax formulation given by equation (8.8). The *warm restart* technique then just corresponds to alternating optimization steps between the primal and dual variables, which is classical in minimax optimization.

To summarize, here are the different features of the optimization scheme to compare in Section 8.7.2.

Optimizer: the general used algorithm, i.e., ADAM, RMSProp or accelerated gradient descent (AGD).

Differentiation: whether we use automatic or analytic differentiation.

Warm restart: whether we use the warm restart technique, which is only compatible with analytic differentiation.

8.7 Experiments and particular cases

In this section, the case of linear utility cost is first considered and shown to have relations with DC programming. The performances of different optimization schemes are then compared on a simple example. Simulations based on the Sinkhorn scheme are then run for the real problem of online repeated auctions. The code is publicly available at `github.com/eboursier/regularized_private_learning`.

8.7.1 Linear utility cost

Section 8.4 described a general optimization scheme for (PRP) with a discrete type prior. Its objective is to find local minima, for a dimensionally finite, non-convex problem, using classical algorithms (Wright, 2015). However in some particular cases, better schemes are possible as claimed in Sections 8.5 and 8.6 for the particular case of entropic regularization. In the case of a linear utility with any privacy cost, it is related to DC programming (Horst et al., 2000). A standard DC program is of the form $\min_{x\in\mathcal{X}} f(x) - g(x)$, where both f and g are convex functions. Specific optimization schemes are then possible (Tao and An, 1997; Horst and Thoai, 1999; Horst et al., 2000). In the case of linear utility costs over a hyperrectangle, (PRP) can be reformulated as a DC program stated in Theorem 8.4.

Theorem 8.4. *If* $\mathcal{X} = \prod_{l=1}^{d} [a_l,\, b_l]$ *and* $c(x,y) = x^\top y$, *define* $\phi(y)^l := (b_l - a_l)y^l/2$ *and* $h_k(\gamma_i) := \big(\sum_{m=1}^{K} \gamma_{i,m}\big) f\big(\frac{\gamma_{i,k}}{p_0^k \sum_{m=1}^{K} \gamma_{i,m}}\big)$. *Then* (PRP) *is equivalent to the following DC program:*

$$\min_{\gamma\in\mathbb{R}_+^{(K+2)\times K}} \lambda \sum_{i,k} p_0^k h_k(\gamma_i) - \sum_{i=1}^{K+2} \left\| \sum_{k=1}^{K} \gamma_{i,k}\phi(y_k) \right\|_1,$$

$$\text{such that for all } k \le K,\ \sum_{i=1}^{K+2} \gamma_{i,k} = p_0^k.$$

Proof. Let ψ be the rescaling of $\mathcal{X}$ to $[-1,1]^d$, i.e., $\psi(x)^l := \frac{2x^l - b_l - a_l}{b_l - a_l}$. Then, $c(x,y) = \psi(x)^T\phi(y) + \eta(y)$ where $\phi(y)^l := (b_l - a_l)\frac{y^l}{2}$ and $\eta(y) = \sum_{l=1}^{d} \frac{a_l + b_l}{b_l - a_l} y^l$. The problem given by Corollary 8.1 is then equivalent to minimizing

$$\sum_{i,k} \gamma_{i,k}(x_i^\top \phi(y_k) + \eta(y_k)) + \lambda \sum_{i,k} p_0^k h_k(\gamma_i),$$

for $x \in [-1,1]^{d\times(K+2)}$. Because of the marginal constraints, $\sum_{i,k} \gamma_{i,k}\eta(y_k) = \sum_k p_0^k \eta(y_k)$. This sum does not depend neither on x nor γ, so that the terms $\eta(y_k)$ can be omitted, i.e., we minimize

$$\sum_i x_i^\top \left(\sum_k \gamma_{i,k}\phi(y_k) \right) + \lambda \sum_{i,k} p_0^k h_k(\gamma_i).$$

It is clear that for a fixed γ, the best x_i corresponds to $x_i^l = -\text{sign}(\sum_k \gamma_{i,k}\phi(y_k)^l)$ and the term $x_i^\top \left(\sum_k \gamma_{i,k}\phi(y_k) \right)$ then corresponds to the opposite of the 1-norm of $\sum_k \gamma_{i,k}\phi(y_k)$, i.e., the problem then minimizes

$$-\sum_i \| \sum_k \gamma_{i,k}\phi(y_k)\|_1 + \lambda \sum_{i,k} p_0^k h_k(\gamma_i).$$

□

More generally, if the cost c is concave and the action space $\mathcal{X}$ is a polytope, optimal actions are located on the vertices of $\mathcal{X}$. In that case, $\mathcal{X}$ can be replaced by the set of its vertices and the problem becomes dimensionally finite. Unfortunately, for some polytopes such as hyper-rectangles, the number of vertices grows exponentially with the dimension and the optimization scheme is no longer tractable in large dimensions.

8.7.2 Minimize Sinkhorn loss on the toy example

This section compares empirically different ways of minimizing the Sinkhorn loss as described in Section 8.6.2. We consider the linear utility loss $c(x,y) = x^\top y$ over the space $\mathcal{X} = [-1,1]^d$ and the Kullback-Leibler divergence for privacy cost, so that both DC and Sinkhorn schemes are possible. The comparison with DC scheme is available in Section 8.7.3.

We optimized using well tuned learning rates. The prior p_0^k is chosen proportional to e^{Z_k} for any $k \in [K]$, where Z_k is drawn uniformly at random in $[0,1]$ and $K = 100$. Each y_i^k is taken uniformly at random in $[-1,1]$ and is rescaled so that $\|y_i\|_1 = 1$. The values are averaged over 200 runs.

Figure 8.2 compares the different features described at the end of Section 8.6.2 for different problem parameters. As suggested by Ablin et al. (2020), the algorithms perform similarly with automatic and analytic differentiation. However, the analytic differentiation allows to use the *warm restart* technique which, coupled with RMSProp, yields better performances as shown in Figure 8.2.

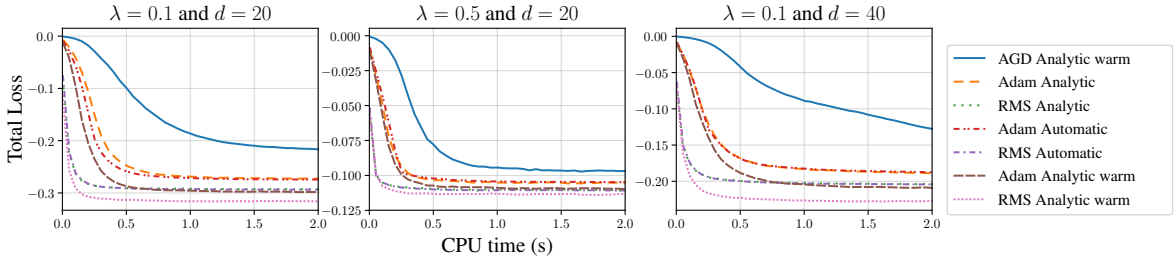


Figure 8.2: Comparison of different features for Sinkhorn minimization.

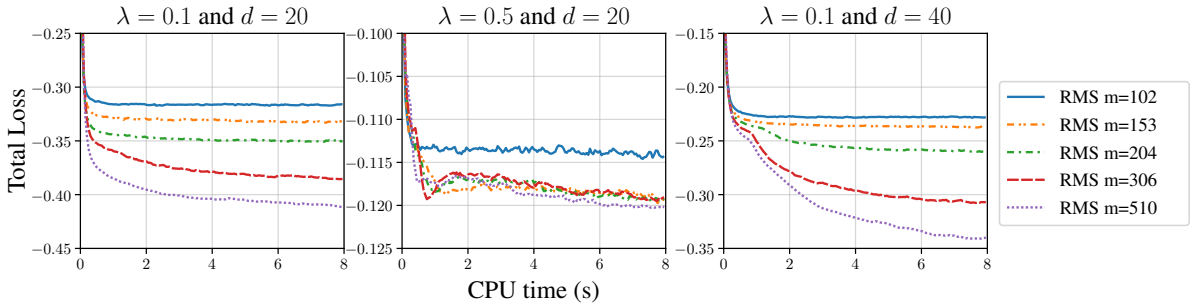


Figure 8.3: Influence of number of actions m.

Figure 8.3 on the other hand studies the influence of the chosen number of actions[3], which is the parameter m in equation (8.9). As expected, the larger the number of actions, the better. Note that for $\lambda = 0.5$, increasing the number of actions has no real influence after $m \geq 153$. The global minimum might always be reached in this case; and this minimum does not depend on m as soon as it is greater than $K + 2$, thanks to Theorem 8.2. It yet remains unkown whether the reached minima are global minima when the number of actions tends to infinity (over-parameterization).

8.7.3 Comparing methods on the toy example

We now compare the performance of Sinkhorn minimization with different algorithms on the toy example described in Section 8.7.2 for $m = K + 2$ actions.

[3]The comparison is done with RMSProp and warm restart, since it yields the best results for a fixed number of actions.

Different methods exist for DC programming and they compute either a local or a global minimum. We here choose the DCA algorithm (Tao and An, 1997) as it computes a local minimum and is thus comparable to the other considered schemes. Figure 8.4 compares the best Sinkhorn scheme in Section 8.7.2 with DCA and PRP method, which uses ADAM or RMSProp optimizers for the minimization problem (8.3).

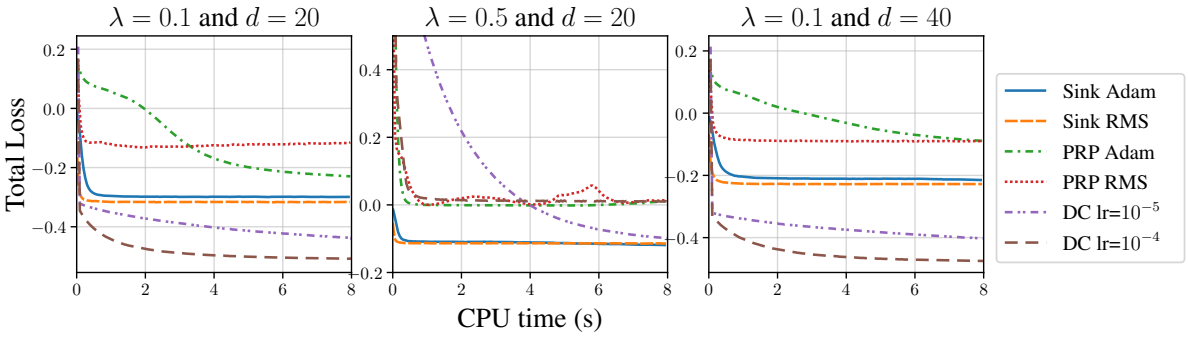


Figure 8.4: Comparison of optimization schemes. `lr` is the learning rate used for DC.

The DC method finds better local minima than the other ones. This was already observed in practice (Tao and An, 1997) and confirms that it is more adapted to the structure of the problem, despite being only applicable in very specific cases such as linear cost on hyperrectangles. Also, the PRP method converges to worse spurious local minima as it optimizes in higher dimensional spaces than the Sinkhorn method. We also observed in our experiments that PRP method is more sensitive to problem parameters than Sinkhorn method.

The Sinkhorn method seems to perform better for larger values of λ. Indeed, given the actions, the Sinkhorn method computes the best joint distribution for each iteration and thus performs well when the privacy cost is predominant, while DCA computes the best actions given a joint distribution and thus performs well when the utility cost is predominant. It is thus crucial to choose the method which is most adapted to the problem structure as it can lead to significant improvement in the solution.

8.7.4 Utility-privacy in repeated auctions

For repeated second price auctions following a precise scheme (Leme et al., 2016), there exist numerical methods to implement an optimal strategy for the bidder (Nedelec et al., 2019). However, if the auctioneer knows that the bidder plays such a strategy, he can still infer the bidder's type and adapt to it. We thus require to add a privacy cost to avoid this kind of behavior from the auctioneer as described in Section 8.2.1.

For simplicity, bidder's valuations are assumed to be exponential distributions, so that the

private type y is the parameter of this distribution, i.e., its expectation: $y = \mathbb{E}_{v\sim\mu_y}[v]$. Moreover, we assume that the prior p_0 over y is the discretized uniform distribution on $[0, 1]$ with a support of size $K = 10$; let $\{y_j\}_{j=1,\dots,K}$ be the support of p_0.

In repeated auctions, values v are repeatedly sampled from the distribution μ_{y_j} and a bidder policy is a mapping $\beta(\cdot)$ from values to bids, i.e., she bids $\beta(v)$ if her value is v. So a type y_j and a policy $\beta(\cdot)$ generate the bid distribution $\beta_{\#}\mu_{y_j}$, which corresponds to an action in $\mathcal{X}$ in our setting. As a consequence, the set of actions of the agent are the probability distributions over $\mathbb{R}_+$ and an action ρ_i is naturally generated from the valuation distribution via the optimal monotone transport map denoted by β^i_j, i.e., $\rho_i = \beta^i_{j\#}\mu_{y_j}$ (Santambrogio, 2015). In the particular case of exponential distributions, this implies that $\beta^j_i(v) = \beta_i(v/y_j)$ where β_i is the unique monotone transport map from $\mathrm{Exp}(1)$ to ρ_i. The revenue of the bidder is then deduced for exponential distributions (Nedelec et al., 2019) as

$$\begin{aligned} r(\beta_i, y_j) &= 1 - c(\beta_i, y_j) \\ &= \mathbb{E}_{v\sim\mathrm{Exp}(1)}\big[(y_j v - \beta_i(v) + \beta_i'(v))G\big(\beta_i(v)\big)\mathbb{1}_{\beta_i(v)-\beta_i'(v)\geq 0}\big], \end{aligned}$$

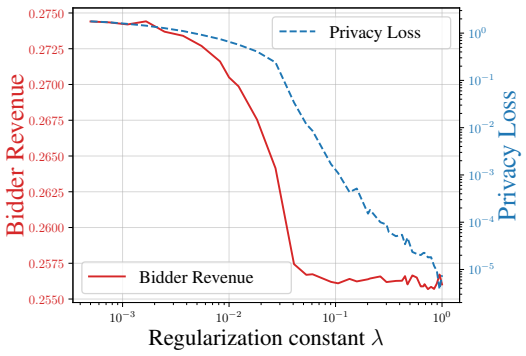
where G is the *c.d.f.* of the maximum bid of the other bidders. We here consider a single truthful opponent with a uniform value distribution on $[0, 1]$, so that $G(x) = \min(x, 1)$. This utility is averaged over 10^3 values drawn from the corresponding distribution at each training step and 10^6 values for the final evaluation.

Considering the KL for privacy cost, we compute a strategy (γ, β) using the Sinkhorn scheme yielding the best results in Section 8.7.2. Every action β_i is parametrized as a single layer neural network of 100 ReLUs. Figure 8.5a represents both utility and privacy as a function of the regularization factor λ.

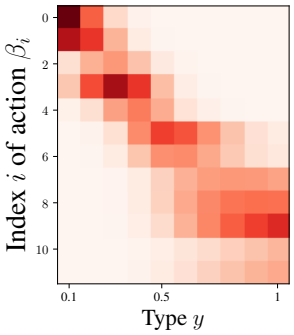
Naturally, both the bidder revenue and the privacy loss decrease with λ, going from revealing strategies for $\lambda \simeq 10^{-3}$ to non-revealing strategies for larger λ. They significantly drop at a critical point near 0.05, which can be seen as the cost of information here. There is a 7% revenue difference[4] between the non revealing strategy and the partially revealing strategy shown in Figure 8.5b. The latter randomizes the type over its neighbors and reveals more information when the revenue is sensible to the action, i.e., for low types y_j here. This strategy thus takes advantage from the fact that the value of information is here heterogeneous among types, as desired in the design of our model.

Figure 8.6 shows the most used action for different types and λ. In the revealing strategy ($\lambda = 0$), the action significantly scales with the type. But as λ grows, this rescaling shrinks so

[4]Which is significant for large firms such as those presented in Figure 8.1 besides the revenue difference brought by considering non truthful strategies (Nedelec et al., 2019).



(a) Evolution of privacy-utility with λ.



(b) Joint distribution map for $\lambda = 0.01$. The intensity of a point (i, j) corresponds to the value of $\gamma(\beta_i, y_j)$.

Figure 8.5: Privacy-utility trade-off in online repeated auctions.

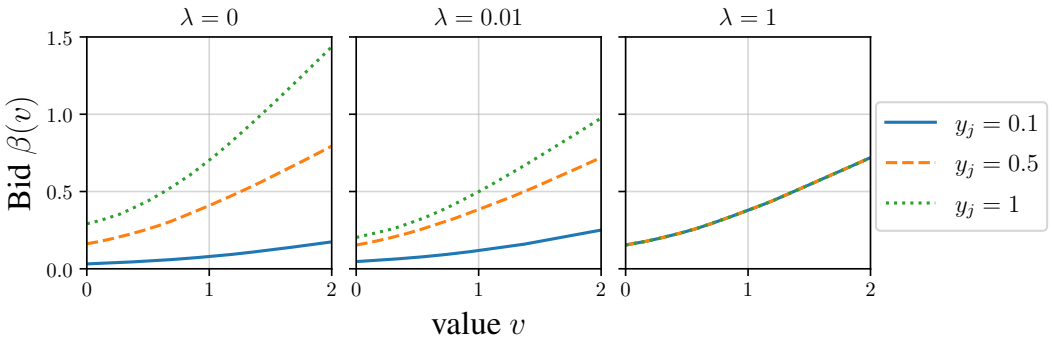


Figure 8.6: Evolution of the bidding strategy with the type and the regularization constant.

that the actions perform for several types, until having a single action in the non-revealing strategy. This shrinkage is also more important for large values of y_j. This confirms the observation made above: the player loses less by hiding her type for large values than for low values and she is thus more willing to hide her type when it is large.

Besides confirming expected results, this illustrates how the Privacy Regularized Policy is adapted to complex utility costs and action spaces, such as distributions or function spaces.

Chapter 9

Social Learning in Non-Stationary Environments

Potential buyers of a product or service, before making their decisions, tend to read reviews written by previous consumers. We consider Bayesian consumers with heterogeneous preferences, who sequentially decide whether to buy an item of unknown quality, based on previous buyers' reviews. The quality is multi-dimensional and may occasionally vary over time; the reviews are also multi-dimensional. In the simple uni-dimensional and static setting, beliefs about the quality are known to converge to its true value. This chapter extends this result in several ways. First, a multi-dimensional quality is considered, second, rates of convergence are provided, third, a dynamical Markovian model with varying quality is studied. In this dynamical setting the cost of learning is shown to be small.

9.1 Introduction

In our society many forms of learning do not stem from direct experience, but rather from observing the behavior of other people who themselves are trying to learn. In other words, people engage in social learning. For instance, before deciding whether to buy a product or service, consumers observe the past behavior of previous consumers and use this observation to make their own decision. Once their decision is made, this becomes a piece of information for future consumers. In the old days, it was common to consider a crowd in a restaurant as a sign that the food was likely good. Nowadays, there are more sophisticated ways to learn from previous consumers. After buying a product and experiencing its features, people often leave reviews on sites such as Amazon, Tripadvisor, Yelp, etc. When consumers observe only the purchasing behavior of previous consumers, there is a risk of a cascade of bad decisions: if the first agents make the wrong decision, the following agents may follow them thinking that what they did was optimal and herding happens. Interestingly enough, this is not necessarily the effect of bounded rationality. It can actually be the outcome of a Bayesian equilibrium in a game with fully rational players. It seems reasonable to conjecture that, if consumers write reviews about the product that they bought, then social learning will be achieved. This is not always the case when consumers are heterogeneous and the reviews that they write depend on the quality of the object but also on their idiosyncratic attitude towards the product they bought.

Consumers also tend to give higher value to recent reviews. As highlighted in a survey (Murphy, 2019) run on a panel of a thousand consumers, "48% of consumers only pay attention to reviews written within the past two weeks," and this trend is growing over time. A justification for this behavior may be that customers perceive the quality of the product that they consider buying as variable over time. The more recent the review, the more informative it is about the current state of the product. This chapter considers a dynamical environment and shows that, under some conditions, the outcome of the learning process in stationary and non-stationary environments are overall comparable.

9.1.1 Main contribution

We consider a model where heterogeneous consumers arrive sequentially at a monopolistic market and—before deciding whether to buy a product of unknown quality—observe the reviews (e.g., like/dislike) provided by previous buyers. Consumers are Bayesian and buy the product if and only if their expected utility of buying is larger than 0 (the utility of the outside option). Each buyer posts a sincere review that summarizes the experienced quality of the product and an idiosyncratic attitude to it. Ifrach et al. (2019) studied this model in the case where the intrinsic quality of the product is one-dimensional, fixed over time, and can assume just two values; they studied conditions for social learning to be achieved. We extend their results in two main directions. First, we allow the quality to be multidimensional, i.e., to have different features that consumers experience and evaluate. Second, we consider a model where the quality can occasionally change over time.

We start examining a benchmark model where the quality is actually static and we provide rates of convergence for the posterior distribution of the quality. We then move to the more challenging dynamical model where quality may change over time. The criterion that we use in this dynamical setting is the utility loss that a non-informed consumer incurs with respect to a fully informed consumer, who at every time knows the true quality of the product. We show that the learning cost is a logarithmic factor of the changing rate of the quality.

Table 9.1 below summarizes the proved bounds for the different settings. In the analysis we also consider the case of imperfect learners, who are not aware of the dynamical nature of the quality, and we quantify the loss they incur.

Type of model	**Utility Loss**	**Tight Bound**
stationary	$\mathcal{O}\left(Md\right)$	✓
dynamical	$\mathcal{O}\left(Md\ln(2/\eta)\eta T\right)$	✓

Table 9.1: Bounds summary, where the reward function is M-Lipschitz and d is the dimension of the quality space. In a non-stationary environment, the quality changes with probability η at each round, while the utility loss is summed over T rounds.

9.1.2 Related literature

The problem of social learning goes back to Banerjee (1992) and Bikhchandani et al. (1992) who considered models where Bayesian rational agents arrive at a market sequentially, observe the actions of the previous agents, and decide based on their private signals and the public observations. These authors showed that in equilibrium, consumers may herd into a sequence of

bad decisions; in other words, social learning fails with positive probability. Smith and Sørensen (2000) showed that this learning failure is due to the fact that signals are bounded. In the presence of unbounded signals that can overcome any observed behavior, herding cannot happen.

Different variations of the above model have been considered, where either agents observe only a subset of the previous agents (see e.g., Çelen and Kariv, 2004; Acemoglu et al., 2011; Lobel and Sadler, 2015), or the order in which actions are taken is not determined by a line, but rather by a lattice (Arieli and Mueller-Frank, 2019). A general analysis of social learning models can be found in (Arieli and Mueller-Frank, 2021).

A more recent stream of literature deals with models where agents observe not just the actions of the previous agents, but also their ex-post reaction to the actions they took. For instance, before buying a product of unknown quality, consumers read the reviews written by the previous consumers. In particular, Besbes and Scarsini (2018) dealt with some variation of a model of social learning in the presence of reviews with heterogeneous consumers. In one case, agents observe the whole history of reviews and can use Bayes rule to compute the conditional expectation of the unknown quality and learning is achieved. In the other case they only observe the mean of past reviews. Interestingly, even in this case, learning is achieved and the speed of convergence is of the same order. Ifrach et al. (2019) studied a model where the unknown quality is binary and the reviews are also binary (like or dislike). They considered the optimal pricing policy and looked at conditions that guarantee social learning. Correa et al. (2020) also considered the optimal dynamic pricing policy when consumers have homogeneous preferences. A non-Bayesian version of the model was considered by Crapis et al. (2017), where mean-field techniques were adopted to study the learning trajectory.

Papanastasiou and Savva (2017) studied a market where strategic consumers can delay their purchase anticipating the fact that other consumers will write reviews in the meanwhile. They examined the implication on pricing of this strategic interaction between consumers and a monopolist. Feldman et al. (2019) examined the role of social learning from reviews in the monopolist's design of a product in a market with strategic consumers. Kakhbod and Lanzani (2021) studied heterogeneity of consumers' reviews and its impact on social learning and price competition. Maglaras et al. (2020) considered a model of social learning with reviews where consumers have different buying options and a platform can affect consumers' choice by deciding the order in which different brands are displayed. Park et al. (2021) dealt with the effect of the first review on the long-lasting success of a product. Chen et al. (2021) considered the issue of bias in reviews from a theoretical viewpoint. They quantified the acquisition bias and the impact on the rating of an arriving customer, characterized the asymptotic outcome of social learning, and we show the effect of biases and social learning on pricing decisions.

The speed of convergence in social learning was considered by Rosenberg and Vieille (2019)

in models where only the actions of the previous agents are observed and by Acemoglu et al. (2017) when reviews are present. This last paper is the closest to the spirit of this chapter.

Learning problems in non-stationary environment have been considered, for instance, by Besbes et al. (2015) and Besbes et al. (2019) in a context where the function that is being learned changes smoothly, rather than abruptly as in our model in Section 9.4.

9.1.3 Organization of the chapter

Section 9.2 introduces the model of social learning from consumer reviews. Section 9.3 studies the stationary setting where the quality is fixed. Section 9.4 introduces the dynamical setting, where the quality changes over time. Section 9.5 consider a model with naive consumers and shows that knowledge of the dynamical structure is crucial for the consumer utility.

Section 9.A contains additional proofs and Section 9.B studies the continuous model where the quality space $\mathcal{Q}$ is convex.

9.2 Model

We consider a model of social learning where consumers read reviews before making their purchase decisions. A monopolist sells a product of unknown quality to consumers who arrive sequentially at the market. The quality may vary over time, although variations are typically rare. The quality of the product at time t is denoted by Q_t and the set of possible qualities is $\mathcal{Q} = \{0,1\}^d$. For a vector x, we denote by $x^{(i)}$ its i-th component, i.e., $Q_t^{(i)}$ represents the i-th feature of the product at time t and has a binary value (low or high).

The prior distribution of the quality at time 1 is π_1. Consumers are indexed by the time of their arrivals $t \in \mathbb{N}\backslash\{0\}$. They are heterogeneous and consumer t has an idiosyncratic preference $\theta_t \in \Theta$ for the product. This preference θ_t is private information. These preferences are assumed to be i.i.d. according to some known distribution. In game-theoretic terms, θ_t could be seen as the *type* of consumer t. The sequences of preferences θ_t and of qualities Q_t are independent.

A consumer who buys the product posts a review in the form of a multi-dimensional numerical grade. The symbol Z_t denotes the review posted by consumer t. The notation $Z_t = *$ indicates that consumer t did not buy the product. We call $\mathcal{H}_t \coloneqq \{Z_1, \dots, Z_{t-1}\}$ the history before the decision of consumer t. We set $\mathcal{H}_1 \coloneqq \varnothing$.

Since the preferences are independent of the quality, a no-purchase decision does not carry any information on the quality. As a consequence, the history $\mathcal{H}_t$ is informationally equivalent to the reduced history $\widetilde{\mathcal{H}}_t$ that includes only the reviews of the buyers up to $t-1$. This differentiates this model from the classical social learning models, where consumers have private signals that

are correlated with the quality.

Based on the history $\mathcal{H}_t$ of past observations and her own preference θ_t, consumer t decides whether to buy the product. In case of purchase, she receives the utility $u_t \coloneqq r(Q_t, \theta_t)$ where r is the reward function. A consumer who does not buy the product gets $u_t = 0$.

Bayesian rationality is assumed, so consumer t buys the product if and only if her conditional expected utility of purchasing is positive, that is, if and only if $\mathbb{E}[r(Q_t, \theta_t) \mid \mathcal{H}_t, \theta_t] > 0$. Consumer t then reviews the product by giving the feedback $Z_t = f(Q_t, \theta_t, \varepsilon_t) \in \mathcal{Z} \subset \mathbb{R}^d$ where ε_t are i.i.d. variables independent from θ_t. Also, the feedback function is assumed to take a finite number of values in $\mathbb{R}^d$ and to be of the form

$$f(Q, \theta, \varepsilon) = (f^{(i)}(Q^{(i)}, \varepsilon, \theta))_{i=1,\dots,d}\,.$$

In words, for each different feature $Q^{(i)}$ of the quality Q, consumers provide a separate feedback. Previous works (Acemoglu et al., 2017; Ifrach et al., 2019) considered $\mathcal{Z} = \{0, 1\}$ as the reviews were only the likes or dislikes of consumers. This model allows a more general and richer feedback, such as ratings on a five-star scale for each feature, or even sparse feedback where consumers do not necessarily review each feature.

In a model without noise ε_t, the learning process is much simpler, as already noted by Ifrach et al. (2019). Indeed, in this case, a single negative review rules out many possibilities as it means that the quality was overestimated. To depict a more interesting learning process, we consider noise, which corresponds to variations caused by different factors, e.g., fluctuations in the product quality or imperfect perception of the quality by the consumer.

In the following, π_t denotes the posterior distribution of Q_t given $\mathcal{H}_t$ and, for any $i \in [d]$, $\pi_t^{(i)}(q^{(i)}) = \mathbb{P}[Q_t^{(i)} = q^{(i)} \mid \mathcal{H}_t]$ is the i-th marginal of the posterior.

We also introduce the function G and its componentwise equivalent $G^{(i)}$, defined as

$$G(z, \pi, q) = \mathbb{P}[Z_t = z \mid \pi_t = \pi, Q_t = q], \tag{9.1}$$

$$G^{(i)}(z^{(i)}, \pi, q^{(i)}) = \mathbb{P}[Z_t^{(i)} = z^{(i)} \mid \pi_t = \pi, Q_t^{(i)} = q^{(i)}]. \tag{9.2}$$

In the following, we also use the notations

$$G(z, \pi) = \mathbb{E}_{q\sim\pi}[G(z, \pi, q)], \tag{9.3}$$

$$G^{(i)}(z^{(i)}, \pi) = \mathbb{E}_{q\sim\pi}[G^{(i)}(z^{(i)}, \pi, q^{(i)})]. \tag{9.4}$$

The following two assumptions will be used in the sequel.

Assumption 9.1 (Purchase guarantee)**.** *The reward function r is monotonic in each feature $q^{(i)}$ and for any $q \in \mathcal{Q}$, $\mathbb{P}_{\theta_t}(r(q, \theta_t) > 0) > 0$, i.e., there is always a fraction of consumers who buy*

the product.

Assumption 9.1 excludes situations where consumers stop buying if the expected quality becomes low. Without this condition, social learning fails with positive probability (Acemoglu et al., 2017; Ifrach et al., 2019).

Assumption 9.2 (Identifiability)**.** *For any $i \in [d]$, any quality posterior $\pi \in \mathcal{P}(\mathcal{Q})$ and quality $q^{(i)}$, we have $G^{(i)}(\,\cdot\,, \pi, q^{(i)}) > 0$. Moreover, for $q^{(i)} \neq q'^{(i)}$, there exists some $z \in \mathcal{Z}$ such that $G^{(i)}(z^{(i)}, \pi, q^{(i)}) \neq G^{(i)}(z^{(i)}, \pi, q'^{(i)})$.*

Assumption 9.2 is needed to distinguish different qualities based on past reviews. The positivity of G is required to avoid trivial situations. The case of $G = 0$ for some variables is similar to the absence of noise ε_t, as a single observation can definitely rule out several possibilities.

An interesting choice of reward function is, for instance, $r(Q, \theta) = \langle Q, \theta_t \rangle$ where $\langle\,\cdot\,,\,\cdot\,\rangle$ is the scalar product. In this case, $\theta_t^{(i)}$ is the weight that customer t gives to feature i of the service.

In practice, customers might also only focus on the best or worst aspects of the service, meaning their reward might only depend on the maximal or minimal value of the $Q^{(i)}$'s. The ordered weighted averaging operators (Yager, 1988) model these behaviors. In an additive model similar to the classical case in the literature, this leads to a reward function $r(Q, \theta) = \sum_{i=1}^{n} w^{(i)}(Q + \theta)^{(\sigma(i))}$ where σ is a permutation such that $(Q + \theta)^{(\sigma(i))}$ is the i-th largest component of the vector $(Q^{(i)} + \theta^{(i)})_{i=1,\dots,d}$. If $w^{(i)} = 1/d$ for all i, this is just an average of all features' utilities. When $w^{(1)} = 1$ and all other terms are 0, consumers are only interested in the maximal utility among all features.

Much of the existing literature has focused on the following unidimensional setting

$$\begin{aligned} r(Q, \theta) &= Q + \theta - p, \\ f(Q, \theta, \varepsilon) &= \operatorname{sign}\left(Q + \theta + \varepsilon - p\right), \end{aligned}$$

where p is an exogenously fixed price. Since consumers review separately each feature of the service, the feedback function is a direct extension of the above unidimensional setting. It is then of the form

$$f^{(i)}(Q^{(i)}, \varepsilon, \theta) = \operatorname{sign}\left(Q^{(i)} + \theta^{(i)} + \varepsilon^{(i)} - p^{(i)}\right), \tag{9.5}$$

for some constant price $p^{(i)}$.

Having a sparse feedback is very common on platform reviews, where consumers only review a few features. This case can be modeled by

$$f^{(i)}(Q^{(i)}, \varepsilon, \eta, \theta) = \operatorname{sign}\left(Q^{(i)} + \theta^{(i)} + \varepsilon^{(i)} - p^{(i)}\right) \xi^{(i)}, \tag{9.6}$$

with $\varepsilon \in \mathbb{R}^d$ and $\xi \in \{0,1\}^d$. Although the noise vector is here given by the tuple (ε, ξ) instead of ε alone, this remains a specific case of our model.

A multiplicative model can also be considered where the relevant quantity is $Q^{(i)}\theta^{(i)}$, rather than $Q^{(i)} + \theta^{(i)}$. This model is very similar to the additive one when using a logarithmic transformation.

9.3 Stationary Environment

As mentioned before, our aim is to consider a model where the quality of the product may occasionally change over time. As a benchmark, we start considering the case where the quality is constant: $Q_t = Q_1$ for all $t \in \mathbb{N}$. We will leverage this case, when dealing with the dynamic model of variable quality. In the unidimensional case $\mathcal{Q} = \{0,1\}$, Ifrach et al. (2019) showed that the posterior almost surely converges to the true quality, and Acemoglu et al. (2017) showed an asymptotic exponential convergence rate. Besides extending these results to the multidimensional model, this section shows anytime convergence rates of the posterior. The study of convergence rates in social learning is just a recent concern (Acemoglu et al., 2017; Rosenberg and Vieille, 2019), despite being central to online learning (Bottou, 1999) and Bayesian estimation (Ghosal et al., 2000). Moreover, convergence rates are of crucial interest when facing a dynamical quality. The main goal of this section is thus to lay the foundation for the analysis of Section 9.4.

The posterior update is obtained using Bayes' rule for any $q \in \mathcal{Q}$,

$$\pi_{t+1}(q) = \frac{G\left(Z_t, \pi_t, q\right)}{G\left(Z_t, \pi_t\right)}\, \pi_t(q). \tag{9.7}$$

Theorem 9.1 below gives a convergence rate of the posterior to the true quality. Similarly to Acemoglu et al. (2017, Theorem 2), it shows an exponential convergence rate. While their result yields an asymptotic convergence rate, we provide an anytime, but slower, rate with similar assumptions. We focus on anytime rates as they are highly relevant in the model with a dynamical, evolving quality considered in Section 9.4.

Theorem 9.1. *For $q \neq q'$, we have*

$$\mathbb{E}[\pi_{t+1}(q') \mid Q = q] \leq \exp\left(-\frac{t\delta^4}{2\gamma^2 + 4\delta^2}\right) \frac{1}{\max_{i\in[d]} \pi_1^{(i)}(q^{(i)})},$$

$$\textit{where} \quad \delta \coloneqq \min_{i\in[d], \pi\in\mathcal{P}(\mathcal{Q})} \sum_{z\in\mathcal{Z}} |G^{(i)}(z^{(i)}, \pi, 1) - G^{(i)}(z^{(i)}, \pi, 0)| \tag{9.8}$$

$$\textit{and} \quad \gamma \coloneqq 2 \max_{i \in [d], \pi \in \mathcal{P}(\mathcal{Q}), z \in \mathcal{Z}} \left| \ln \left(\frac{G^{(i)}(z^{(i)}, \pi, 1)}{G^{(i)}(z^{(i)}, \pi, 0)} \right) \right|. \tag{9.9}$$

Notice that δ is the minimal total variation between $Z_t^{(i)}$ conditioned either on $(\pi, Q_t^{(i)} = 1)$ or $(\pi, Q_t^{(i)} = 0)$. Thanks to Assumption 9.2, both δ and γ are positive and finite. This guarantees an exponential convergence rate of the posterior as $\pi_t(q) = 1 - \sum_{q' \neq q} \pi_t(q')$.

Proof of Theorem 9.1. Assume without loss of generality $Q_1^{(i)} = 1$. The proof of Theorem 9.1 follows directly from the following inequality, which we prove in the following:

$$\mathbb{E}[\pi_{t+1}^{(i)}(0) \mid Q_1^{(i)} = 1] \leq \exp\left(-\frac{t\delta^4}{2\gamma^2 + 4\delta^2} \right) \frac{1}{\pi_1^{(i)}(1)}. \tag{9.10}$$

Similarly to (9.7), we have the Bayesian update

$$\pi_{t+1}^{(i)}(q^{(i)}) = \frac{G^{(i)}\left(Z_t^{(i)}, \pi_t, q^{(i)}\right)}{G^{(i)}\left(Z_t^{(i)}, \pi_t\right)} \pi_t^{(i)}(q^{(i)}). \tag{9.11}$$

This leads by induction to

$$\ln\left(\frac{\pi_{t+1}^{(i)}(1)}{\pi_{t+1}^{(i)}(0)} \right) = \ln\left(\frac{\pi_1^{(i)}(1)}{\pi_1^{(i)}(0)} \right) + \sum_{s=1}^{t} \ln\left(\frac{G^{(i)}\left(Z_s^{(i)}, \pi_s, 1\right)}{G^{(i)}\left(Z_s^{(i)}, \pi_s, 0\right)} \right).$$

In the following, we use the notation $\mathrm{KL}(\mu, \nu)$ for the Kullback-Leibler divergence between the distributions μ and ν, which is defined as

$$\mathrm{KL}(\mu, \nu) = \mathbb{E}_{x \sim \mu}\left[\ln\left(\frac{\mu(x)}{\nu(x)} \right) \right]. \tag{9.12}$$

Define now

$$X_t \coloneqq \ln\left(\frac{G^{(i)}(Z_t^{(i)}, \pi_t, 1)}{G^{(i)}(Z_t^{(i)}, \pi_t, 0)} \right) - \mathrm{KL}\left(G^{(i)}(\cdot, \pi_t, 1), G^{(i)}(\cdot, \pi_t, 0) \right). \tag{9.13}$$

Notice that $\mathbb{E}[X_t \mid \mathcal{H}_t, Q_1^{(i)} = 1] = 0$. Also, by definition of γ, $X_t \in [Y_t, Y_t + \gamma]$ almost surely for some $\mathcal{H}_t$-measurable variable Y_t. Azuma-Hoeffding's inequality (see, e.g., Cesa-Bianchi and Lugosi, 2006, Lemma A.7) then yields for any $\lambda \geq 0$:

$$\mathbb{P}\left[\sum_{s=1}^{t} X_s \leq -\lambda \,\middle|\, Q_1^{(i)} = 1 \right] \leq \exp\left(-\frac{2\lambda^2}{t\gamma^2} \right),$$

which is equivalent to

$$\mathbb{P}\left[\frac{\pi_{t+1}^{(i)}(0)}{\pi_{t+1}^{(i)}(1)} \geq \exp\left(\lambda - \sum_{s=1}^{t} \mathrm{KL}\left(G^{(i)}(\cdot,\pi_s,1), G^{(i)}(\cdot,\pi_s,0)\right)\right)\frac{\pi_1^{(i)}(0)}{\pi_1^{(i)}(1)} \,\Big|\, Q_1^{(i)} = 1\right] \leq \exp\left(-\frac{2\lambda^2}{t\gamma^2}\right). \tag{9.14}$$

By Pinsker's inequality (see, e.g., Tsybakov, 2009, Lemma 2.5), we have

$$\mathrm{KL}\left(G^{(i)}(\cdot,\pi_s,1), G^{(i)}(\cdot,\pi_s,0)\right) \geq \delta^2/2,$$

so Equation (9.14) becomes

$$\mathbb{P}\left[\pi_{t+1}^{(i)}(0) \geq \exp\left(\lambda - \frac{t\delta^2(1,0)}{2}\right)\frac{\pi_1^{(i)}(0)}{\pi_1^{(i)}(1)} \,\Big|\, Q_1^{(i)} = 1\right] \leq \exp\left(-\frac{2\lambda^2}{t\gamma^2}\right),$$

where we used the fact that $\pi_{t+1}^{(i)}(1) \leq 1$. This then yields

$$\begin{aligned}\mathbb{E}[\pi_{t+1}^{(i)}(0) \mid Q_1^{(i)} = 1] &\leq \exp\left(\lambda - \frac{t\delta^2}{2}\right)\frac{\pi_1^{(i)}(0)}{\pi_1^{(i)}(1)} + \mathbb{P}\left[\pi_{t+1}^{(i)}(0) \geq \exp\left(\lambda - t\delta^2/2\right) \mid Q_1^{(i)} = 1\right] \\ &\leq \exp\left(\lambda - \frac{t\delta^2}{2}\right)\frac{\pi_1^{(i)}(0)}{\pi_1^{(i)}(1)} + \exp\left(-\frac{2\lambda^2}{t\gamma^2}\right).\end{aligned}$$

Let $x = t\gamma^2/4$ and $y = t\delta^2/2$. Setting $\lambda = -x + \sqrt{2xy + x^2}$ equalizes the exponential terms:

$$\begin{aligned}\mathbb{E}[\pi_{t+1}^{(i)}(0) \mid Q_1^{(i)} = 1] &\leq (1 + \frac{\pi_1^{(i)}(0)}{\pi_1^{(i)}(1)}) \exp\left(-x - y + \sqrt{x^2 + 2xy}\right) \\ &\leq \frac{1}{\pi_1^{(i)}(1)} \exp\big(-\frac{y^2}{2(x+y)}\big).\end{aligned}$$

The second inequality is given by the convex inequality

$$\sqrt{a} - \sqrt{a+b} \leq -\frac{b}{2\sqrt{a+b}}, \quad \text{for } a = x^2 + 2xy \text{ and } b = y^2.$$

From the definitions of x and y, this yields

$$\mathbb{E}[\pi_{t+1}^{(i)}(0) \mid Q_1^{(i)} = 1] \leq \frac{1}{\pi_1^{(i)}(1)} \exp\left(-\frac{t\delta^4}{2\gamma^2 + 4\delta^2}\right).$$

We conclude by noting that $\pi_t^{(i)}(q'^{(i)}) \geq \pi_t(q')$. □

9.4 Dynamical Environment

We now model a situation where the quality Q may change over time. We consider a general Markovian model given by the transition matrix P. Moreover, at each time step, the quality might change with probability at most $\eta \in (0, 1)$:

$$\begin{aligned} &\mathbb{P}\left(Q_{t+1} = q' \mid Q_t = q\right) = P_{q,q'}, \\ &\text{with } P(q,q) \geq 1 - \eta \text{ for all } q \in \mathcal{Q}. \end{aligned} \tag{9.15}$$

The use of a Markovian model is rather usual in such dynamical models. Assuming that the diagonal terms of the transition matrix P are large ensures that changes of quality are rare. Consumers thus have some time to learn the current quality of the product.

Studying the convergence of the posterior is irrelevant, as the quality regularly changes. Instead, we measure the quality of the posterior variations in term of the total utility loss

$$R_T \coloneqq \sum_{t=1}^{T} \mathbb{E}[r(Q_t, \theta_t)_+ - u_t], \tag{9.16}$$

also known as "regret". The first term $r(Q_t, \theta_t)_+$ corresponds to the utility a consumer would get if she knew the quality Q_t, whereas u_t is the utility she actually gets.

Lemma 9.1. *If r is M-Lipschitz in its first argument for any $\theta \in \Theta$, i.e., $|r(q,\theta) - r(q',\theta)| \leq M\|q - q'\|_1$ for any $q, q' \in \mathcal{Q}$, we have*

$$R_T \leq M \sum_{i=1}^{d} \sum_{t=1}^{T} \mathbb{E}[1 - \pi_t^{(i)}(Q_t^{(i)})].$$

Lemma 9.1, proved in Section 9.A.1, shows that bounding the cumulated estimation error $\sum_{t=1}^{T} \mathbb{E}[1 - \pi_t^{(i)}(Q_t^{(i)})]$ for each coordinate is sufficient to bound the total regret.

We consider in this section consumers who have perfect knowledge of the model, i.e., they know that the quality might change following (9.15). Recall that the prior is assumed uniform on $\mathcal{Q}$. If G is defined as in (9.1), the posterior update is given by

$$\pi_{t+1}(q) = \sum_{q' \in \mathcal{Q}} P(q,q') \frac{G\left(Z_t, \pi_t, q'\right)}{G\left(Z_t, \pi_t\right)} \pi_t(q'). \tag{9.17}$$

The effect of the old reviews is mitigated by the multiplications with the transition matrix P. Consumers thus value more recent reviews in this model, as wished in its design. By induction,

the previous inequality leads to the following expression.

$$\pi_{t+1}(q) = \sum_{\substack{(q_s)\in\mathcal{Q}^t \\ q_{t+1}=q}} \pi_1(q_1) \prod_{s=1}^{t} P(q_s, q_{s+1}) \frac{G(Z_s, \pi_s, q_s)}{G(Z_s, q_s)} \tag{9.18}$$

This expression is more complex than the one in the stationary case, leading to a more intricate proof of error bounds. We actually bound the estimation error for a simpler, imperfect bayesian estimator, which directly bounds the true utility loss, by optimality of the bayesian estimator.

Theorem 9.2 below shows that the cumulated loss is of order $\ln(2/\eta)\eta T$. Perfect learners, who could directly observe Q_{t-1} before making the decision at time t, would still suffer a loss of order ηT as there is a constant uncertainty η about the next step quality. Theorem 9.2 thus shows that the cost of learning is just a logarithmic factor in the dynamical setting.

Theorem 9.2. *If r is M-Lipschitz, then $R_T = \mathcal{O}\left(Md\ln\left(2/\eta\right)\eta T\right)$.*
Moreover, if $\eta T = \Omega(1)$, there is some M-Lipschitz reward r and some transition matrix P verifying the conditions of Equation (9.15) *such that $R_T = \Omega(Md\ln(2/\eta)\eta T)$.*

The hidden constants in the $\mathcal{O}\left(\cdot\right)$ and $\Omega(\cdot)$ above only depend on the values of δ and γ defined in Theorem 9.1.

The proof of Theorem 9.2 is divided into two parts: first, the upper bound $R_T = \mathcal{O}\left(Md\ln(2/\eta)\eta T\right)$ and, second, the lower bound $R_T = \Omega\left(Md\ln(2/\eta)\eta T\right)$. The proof of the lower bound is postponed to Section 9.A.2.

The assumption $\eta T = \Omega(1)$ guarantees that changes of quality actually have a non-negligible chance to happen in the considered time window. Without it, we would be back to the stationary case. In the extreme case $\eta T \approx 1$, the error is thus of order $\ln(T)$ against 1 in the stationary setting. This larger loss is actually the time needed to achieve the same precision in posterior belief anew after a change of quality. Indeed, let the posterior be very close to the true quality q, i.e., $\pi_t(q') \approx 0$ for $q' \neq q$; if the quality suddenly changes to q', it will take a while to have a correct estimation again, i.e., to get $\pi_t(q') \approx 1$.

Proof of the Upper Bound.

In order to prove that $R_T = \mathcal{O}\left(Md\ln(2/\eta)\eta T\right)$, we actually show the result marginally on each dimension, i.e., for any $i \in [d]$

$$\sum_{t=1}^{T} 1 - \pi_t^{(i)}(q^{(i)}) = \mathcal{O}\left(\ln(2/\eta)\eta T\right). \tag{9.19}$$

Lemma 9.1 then directly leads to the upper bound. To prove Equation (9.19), we first consider another $\mathcal{H}_t$-measurable estimator defined for any i by

$$\widetilde{\pi}_1^{(i)} = \pi_1^{(i)} \quad \text{and} \quad \widetilde{\pi}_{t+1}^{(i)}(q^{(i)}) = (1-2\eta)\frac{G^{(i)}(Z_t^{(i)}, \pi_t, (q^{(i)}))}{G^{(i)}(Z_t^{(i)}, \pi_t)}\widetilde{\pi}_t^{(i)}(q^{(i)}) + \eta. \tag{9.20}$$

The estimator $\widetilde{\pi}_t$ can be seen as the bayesian estimator, for the worst case of transition matrix, where each feature i changes with probability η at each step. As perfect bayesian consumers' decisions minimize the utility loss among the classes of $\mathcal{H}_t$ measurable decisions, having an $\mathcal{O}\left(\ln(2/\eta)\eta T\right)$ error for $\widetilde{\pi}_t^{(i)}$ directly yields Equation (9.19).

We consider small η in the following, as the bound trivially holds for η larger than some constant.

To prove Equation (9.19), we partition $\mathbb{N}^*$ into blocks $[t_k^{(i)}+1, t_{k+1}^{(i)}]$ of fixed quality (for the i-th coordinate) and show that the error of $\widetilde{\pi}_t^{(i)}$ on each block individually is $\mathcal{O}\left(\ln(2/\eta)\right)$:

$$t_1^{(i)} \coloneqq 0 \quad \text{and} \quad t_{k+1}^{(i)} \coloneqq \min\left\{t > t_k^{(i)} \mid Q_{t+1}^{(i)} \neq Q_{t_k^{(i)}+1}^{(i)}\right\}. \tag{9.21}$$

We only aim at bounding the estimation error on a single block k. In the rest of the proof, we assume w.l.o.g. that $Q_t^{(i)} = 1$ on this block.

Define the stopping time

$$\tau_k^{(i)} \coloneqq \min\left(\left\{t \in [t_k^{(i)}+1, t_{k+1}^{(i)}] \;\middle|\; \frac{\widetilde{\pi}_t^{(i)}(1)}{\widetilde{\pi}_t^{(i)}(0)} \geq 1\right\} \cup \{t_{k+1}^{(i)}\}\right). \tag{9.22}$$

This is the first time[1] in block k where the posterior belief of the true quality (for $\widetilde{\pi}_t^{(i)}$) exceeds the one of the wrong quality. The error on the block is then decomposed as the terms before $\tau_k^{(i)}$, which contribute to at most 1 per timestep, and the terms after $\tau_k^{(i)}$. Lemma 9.2 bounds the first part.

Lemma 9.2. *For any k,*

$$\mathbb{P}\left[\tau_k^{(i)} - t_k^{(i)} \geq 2 + \frac{2\gamma^2 + 4\delta^2}{\delta^4}\ln\left(\frac{1}{\eta}\right)\right] \leq \eta,$$

where δ and γ are defined as in Theorem 9.1.

Proof of Lemma 9.2. As a consequence of the posterior update of $\widetilde{\pi}_t$ given by Equation (9.20),

[1] It is set as the largest element of the block if such a criterion is never satisfied.

for $t+1 \leq \tau_k^{(i)}$,

$$\widetilde{\pi}_{t+1}^{(i)}(0) \leq \frac{G^{(i)}(Z_t^{(i)}, \pi_t, 0)}{G(Z_t, \pi_t)} \widetilde{\pi}_t(0) \quad \text{and} \quad \widetilde{\pi}_{t+1}^{(i)}(1) \geq \frac{G^{(i)}(Z_t^{(i)}, \pi_t, 1)}{G(Z_t, \pi_t)} \widetilde{\pi}_t(1).$$

We then get by induction

$$\frac{\widetilde{\pi}_{t+1}^{(i)}(0)}{\widetilde{\pi}_{t+1}^{(i)}(1)} \leq \frac{1}{\eta} \prod_{s=t_k^{(i)}+1}^{t} \frac{G^{(i)}(Z_s^{(i)}, \pi_s, 0)}{G^{(i)}(Z_s^{(i)}, \pi_s, 1)}, \tag{9.23}$$

as $\widetilde{\pi}_{t_k^{(i)}+1}^{(i)}(1) \geq \eta$. For $n = \left\lceil \frac{2\gamma^2+4\delta^2}{\delta^4} \ln\left(\frac{1}{\eta}\right) \right\rceil$, it has been shown in the proof of Theorem 9.1 that:

$$\mathbb{P}\left[\prod_{s=t_k^{(i)}+1}^{t_k^{(i)}+n} \frac{G(Z_s^{(i)}, \pi_s, 0)}{G(Z_s^{(i)}, \pi_s, 1)} > \eta \;\middle|\; \pi_{t_k^{(i)}+1}, \forall s \in [t_k^{(i)}+1, t_k^{(i)}+n], Q_s^{(i)} = 1 \right] \leq \eta.$$

Note that by definition of $\tau_k^{(i)}$, $\frac{\widetilde{\pi}_{\tau_k^{(i)}}^{(i)}(0)}{\widetilde{\pi}_{\tau_k^{(i)}}^{(i)}(1)} \leq 1$. The above concentration inequality and (9.23) imply that $\mathbb{P}[\tau_k^{(i)} - t_k^{(i)} \geq n+1] \leq \eta$. □

In Lemma 9.3 below we show that, past this stopping time $\tau_k^{(i)}$, the quantity $1/\widetilde{\pi}_t^{(i)}(1)$ cannot exceed some constant term in expectation.

Lemma 9.3. *For any $k \in \mathbb{N}^*$ and $t \in [\tau_k^{(i)}, t_{k+1}^{(i)}]$,*

$$\mathbb{E}\left[\frac{1}{\widetilde{\pi}_t^{(i)}(Q_t^{(i)})} \;\middle|\; \tau_k^{(i)}, (t_n^{(i)})_n \right] \leq 2.$$

Proof of Lemma 9.3. By definition of $G^{(i)}$ and the posterior update, given by Equations (9.2) and (9.20) respectively, we have

$$\mathbb{E}\left[\frac{1}{\widetilde{\pi}_{t+1}^{(i)}(1)} \;\middle|\; Q_t^{(i)} = 1, \mathcal{H}_t \right] = \sum_{z^{(i)}: z \in \mathcal{Z}} G^{(i)}(z^{(i)}, \pi_t^{(i)}, 1)\, h\left(\frac{G^{(i)}(z^{(i)}, \pi_t)}{G^{(i)}(z^{(i)}, \pi_t, 1)\widetilde{\pi}_t^{(i)}(1)} \right),$$
$$\text{with } h(x) = \frac{1}{\eta + \frac{1-2\eta}{x}}. \tag{9.24}$$

Note that h is concave on $\mathbb{R}_+^*$, so by Jensen's inequality:

$$\mathbb{E}\left[\frac{1}{\widetilde{\pi}_{t+1}^{(i)}(1)} \;\middle|\; Q_t^{(i)} = 1, \mathcal{H}_t \right] \leq h\left(\frac{1}{\widetilde{\pi}_t^{(i)}(1)} \right). \tag{9.25}$$

Lemma 9.3 then follows by induction

$$\begin{aligned}
&\mathbb{E}\left[\frac{1}{\widetilde{\pi}^{(i)}_{t+\tau_k^{(i)}+1}(1)} \;\middle|\; \tau_k^{(i)}, \forall s \in [\tau_k^{(i)}, t+\tau_k^{(i)}], Q_s^{(i)} = 1\right] \\
&\quad \leq \mathbb{E}\left[h\left(\frac{1}{\widetilde{\pi}^{(i)}_{t+\tau_k^{(i)}}(1)}\right) \;\middle|\; \tau_k^{(i)}, \forall s \in [\tau_k^{(i)}, t+\tau_k^{(i)}], Q_s^{(i)} = 1\right] \\
&\quad \leq h\left(\mathbb{E}\left[\frac{1}{\widetilde{\pi}^{(i)}_{t+\tau_k^{(i)}}(1)} \;\middle|\; \tau_k^{(i)}, \forall s \in [\tau_k^{(i)}, t+\tau_k^{(i)}], Q_s^{(i)} = 1\right]\right) \\
&\quad \leq h\left(2\right) = 2.
\end{aligned}$$

The first inequality is a direct consequence of Equation (9.25), the second is Jensen's inequality again, while the third one is obtained by induction using the fact that h is increasing and $\widetilde{\pi}^{(i)}_{\tau_k^{(i)}}(1) \geq \frac{1}{2}$. □

Similarly to the proof of Theorem 9.1, Azuma-Hoeffding's inequality on a single block leads to

$$\mathbb{E}\left[\prod_{s=n}^{t-1} \frac{G^{(i)}(Z_s^{(i)}, \pi_s, 0)}{G^{(i)}(Z_s^{(i)}, \pi_s, 1)} \;\middle|\; \pi_n, \forall s \in [n, t-1], Q_s^{(i)} = 1\right] \leq \exp\left(-\frac{(t-n)\delta^4}{2\gamma^2 + 4\delta^2}\right). \tag{9.26}$$

Also, note that Equation (9.20) leads to

$$\frac{G^{(i)}(Z_t^{(i)}, \pi_t, 1)}{G^{(i)}(Z_t^{(i)}, \pi_t)} \leq \frac{\widetilde{\pi}^{(i)}_{t+1}(1)}{(1-2\eta)\widetilde{\pi}^{(i)}_t(1)}.$$

By induction, we get

$$\prod_{s=n}^{t-1} \frac{G^{(i)}(Z_s^{(i)}, \pi_s, 1)}{G^{(i)}(Z_s^{(i)}, \pi_s)} \leq \frac{1}{\widetilde{\pi}^{(i)}_n(1)(1-2\eta)^{t-n}}. \tag{9.27}$$

Multiplying the left hand side of (9.26) by the left hand side of (9.27), we obtain

$$\mathbb{E}\left[\prod_{s=n}^{t-1} \frac{G^{(i)}(Z_s^{(i)}, \pi_s, 0)}{G^{(i)}(Z_s^{(i)}, \pi_s)} \;\middle|\; \pi_n, \forall s \in [n, t-1], Q_s^{(i)} = 1\right] \leq \frac{(1-2\eta)^{-(t-n)}}{\widetilde{\pi}^{(i)}_n(1)} \exp\left(-\frac{(t-n)\delta^4}{2\gamma^2 + 4\delta^2}\right). \tag{9.28}$$

Similarly to Equation (9.18), starting from $n_0 \geq 1$, for the i-th coordinate it can be shown that

$$\widetilde{\pi}^{(i)}_{t+1}(q^{(i)}) = (1-2\eta)^{t-n_0+1}\widetilde{\pi}^{(i)}_{n_0}(q^{(i)}) \prod_{s=n_0}^{t} \frac{G^{(i)}\left(Z^{(i)}_s, \pi_s, q^{(i)}\right)}{G^{(i)}\left(Z^{(i)}_s, \pi_s\right)} + \eta \sum_{s=0}^{t-n_0} (1-2\eta)^s \prod_{l=t-s+1}^{t} \frac{G^{(i)}(Z^{(i)}_l, \pi_l, q^{(i)})}{G^{(i)}(Z^{(i)}_l, \pi_l)}.$$

Define $A^t_{\tau^{(i)}_k} \coloneqq \left\{\forall s \in [\tau^{(i)}_k, \tau^{(i)}_k + t], Q^{(i)}_s = 1\right\}$. Combining this formula with Equation (9.28), we obtain

$$\mathbb{E}\left[\widetilde{\pi}^{(i)}_{\tau^{(i)}_k + t}(0) \mid \mathcal{H}_{\tau^{(i)}_k}, A^t_{\tau^{(i)}_k}\right] \leq \frac{\widetilde{\pi}^{(i)}_{\tau^{(i)}_k}(0)}{\widetilde{\pi}^{(i)}_{\tau^{(i)}_k}(1)} \exp\left(-\frac{t\delta^4}{2\gamma^2 + 4\delta^2}\right) + 2\eta \sum_{s=0}^{t-1} \mathbb{E}\left[\frac{1}{\widetilde{\pi}^{(i)}_{\tau^{(i)}_k + t - s}(1)} \,\middle|\, \mathcal{H}_{\tau^{(i)}_k}, A^t_{\tau^{(i)}_k}\right] \exp\left(-\frac{s\delta^4}{2\gamma^2 + 4\delta^2}\right).$$

Thanks to Lemma 9.3,

$$\mathbb{E}\left[\frac{1}{\widetilde{\pi}^{(i)}_{\tau^{(i)}_k + t - s}(1)} \,\middle|\, \mathcal{H}_{\tau^{(i)}_k}, A^t_{\tau^{(i)}_k}\right] \leq 2 \quad \text{and} \quad \frac{\widetilde{\pi}^{(i)}_{\tau^{(i)}_k}(0)}{\widetilde{\pi}^{(i)}_{\tau^{(i)}_k}(1)} \leq 1,$$

so that

$$\begin{aligned} \mathbb{E}\left[\widetilde{\pi}^{(i)}_{\tau^{(i)}_k + t}(0) \mid \mathcal{H}_{\tau^{(i)}_k}, A^t_{\tau^{(i)}_k}\right] &\leq \exp\left(-\frac{t\delta^4}{2\gamma^2 + 4\delta^2}\right) + 4\eta \sum_{s=0}^{t-1} \exp\left(-\frac{s\delta^4}{2\gamma^2 + 4\delta^2}\right) \\ &\leq \exp\left(-\frac{t\delta^4}{2\gamma^2 + 4\delta^2}\right) + \frac{4\eta}{1 - \exp\left(-\frac{\delta^4}{2\gamma^2 + 4\delta^2}\right)}. \end{aligned} \tag{9.29}$$

Finally, the estimation error for $\widetilde{\pi}^{(i)}_t$ incurred during the block k is at most

$$\tau^{(i)}_k - t^{(i)}_k + \sum_{t=0}^{t^{(i)}_{k+1} - t^{(i)}_k - 1} \left(\exp\left(-\frac{t\delta^4}{2\gamma^2 + 4\delta^2}\right) + \frac{4\eta}{1 - \exp\left(-\frac{\delta^4}{2\gamma^2 + 4\delta^2}\right)}\right),$$

i.e., it is of order $\tau^{(i)}_k - t^{(i)}_k + \eta(t^{(i)}_{k+1} - t^{(i)}_k)$. Lemma 9.2 then yields

$$\mathbb{E}[\tau^{(i)}_k - t^{(i)}_k \mid (t_n)_n] \leq 2 + \frac{2\gamma^2 + 4\delta^2}{\delta^4} \ln\left(\frac{1}{\eta}\right) + \eta(t^{(i)}_{k+1} - t^{(i)}_k).$$

Thus in expectation, given $(t_n)_n$, the estimation error of $Q_t^{(i)}$ over the block k for $\widetilde{\pi}_t$ is of order $\ln(2/\eta) + \eta(t_{k+1}^{(i)} - t_k^{(i)})$. Note that $t_{k+1}^{(i)} - t_k^{(i)}$ is stochastically dominated by a geometric distribution of parameter η. In expectation the number of blocks counted before T is thus $\mathcal{O}(\eta T)$ and summing over all these blocks yields

$$\sum_{t=1}^{T} \mathbb{E}[1 - \widetilde{\pi}_t^{(i)}(Q_t^{(i)})] = \mathcal{O}(\ln(2/\eta)\eta T).$$

When summing over all coordinates, this implies that the regret incurred by the estimator $\widetilde{\pi}_t$ is of order $\mathcal{O}(Md\ln(2/\eta)\eta T)$. Since the exact estimator π_t minimizes the expected utility loss among the class of all $\mathcal{H}_t$-measurable estimators, the upper bound follows.

Proof of the Lower Bound.

The proof of the lower bound is postponed to Section 9.A.2. The idea is that the posterior cannot converge faster than exponentially on a single block. Thus, if the posterior converged in the last block, e.g., $\pi_t(q') \approx \eta$ in a block of quality q, then it would require a time $\ln(2/\eta)$ before $\pi_t(q') \geq 1/2$ in the new block of quality q', leading to a loss at least $\ln(2/\eta)$ on this block.

9.5 Naive Learners

In Section 9.4 we showed that learning occurs for Bayesian consumers who are perfectly aware of the environment, and especially of its dynamical aspect. In some learning problems, Bayesian learners can still have small regret, despite having an imperfect knowledge of the problem parameters or even ignoring some aspects of the problem.

This section shows that awareness of the problem's dynamical structure is essential here. In particular, naive learners incur a considerable utility loss.

In the following, we consider the setting described in Section 9.4 with naive learners, i.e., consumers who are unaware of possible quality changes over time. As a consequence, their posterior distribution π_t^{naive} follows the exact same update rule as in the stationary case:

$$\pi_{t+1}^{\text{naive}}(q) = \frac{G(Z_t, \pi_t^{\text{naive}}, q)}{G(Z_t, \pi_t^{\text{naive}})} \pi_t^{\text{naive}}(q).$$

The regret for naive learners is then

$$\sum_{t=1}^{T} \mathbb{E}\left[r(Q_t, \theta_t)_+ - u_t^{\text{naive}}\right],$$

$$\text{where } u_t^{\text{naive}} = r(Q_t, \theta_t)\mathbb{1}\left(\sum_{q\in\mathcal{Q}} \pi_t^{\text{naive}}(q) r(q, \theta_t) \geq 0\right).$$

u_t^{naive} is the utility achieved by naive learners who make their decisions based on π_t^{naive}.

Theorem 9.3 below states that the utility loss for naive learners is non-negligible, i.e., of order T, which displays the significance of taking into account the dynamical structure of the problem in the learning process.

Theorem 9.3. *If $\eta T = \Omega(1)$, then there is some M-Lipschitz reward r and some transition matrix P verifying the conditions given by Equation* (9.15) *such that*

$$R_T^{\text{naive}} = \Omega(MdT).$$

The proof of Theorem 9.3 can be found in Section 9.A.3 and bears similarities with the proof of the lower bound in Theorem 9.2. The posterior of naive learners converges quickly to the true quality on a single block. Because of this, after a change of quality, it takes a long time before the posterior belief of naive learners becomes accurate again with respect to the new quality.

Appendix

9.A Omitted proofs

This section contains detailed proofs of lemmas and theorems postponed to the Appendix.

9.A.1 Proof of Lemma 9.1

The inequality actually holds individually for each term of the sum when conditioned on π_t, i.e., $\mathbb{E}[r(Q_t,\theta_t)_+ - u_t \mid \pi_t] \leq M\sum_{i=1}^{d}(1-\pi_t^{(i)}(Q_t^{(i)}))$, which directly implies Lemma 9.1. By definition, $u_t = r(Q_t,\theta_t)\mathbb{1}\left(\sum_{q\in\mathcal{Q}}\pi_t(q)r(q,\theta_t)\geq 0\right)$ and so it comes

$$
\begin{aligned}
r(Q_t,\theta_t)_+ - u_t &= r(Q_t,\theta_t)\left(\mathbb{1}\left(r(Q_t,\theta_t)\geq 0\right) - \mathbb{1}\left(\sum_{q\in\mathcal{Q}}\pi_t(q)r(q,\theta_t)\geq 0\right)\right)\\
&= r(Q_t,\theta_t)\Bigg(\mathbb{1}\left(r(Q_t,\theta_t)\geq 0\geq \sum_{q\in\mathcal{Q}}\pi_t(q)r(q,\theta_t)\right)\\
&\qquad\qquad - \mathbb{1}\left(\sum_{q\in\mathcal{Q}}\pi_t(q)r(q,\theta_t)\geq 0\geq r(Q_t,\theta_t)\right)\Bigg)\\
&\leq \left|r(Q_t,\theta_t) - \sum_{q\in\mathcal{Q}}\pi_t(q)r(q,\theta_t)\right|\\
&= \left|\sum_{q\in\mathcal{Q}}\pi_t(q)(r(Q_t,\theta_t)-r(q,\theta_t))\right|\\
&\leq \sum_{q\in\mathcal{Q}}\pi_t(q)\left|r(Q_t,\theta_t)-r(q,\theta_t)\right|\\
&\leq M\sum_{q\in\mathcal{Q}}\pi_t(q)\|Q_t-q\|_1\\
&= M\sum_{i=1}^{d}(1-\pi_t^{(i)}(Q_t^{(i)})).
\end{aligned}
$$

9.A.2 Proof of the lower bound of Theorem 9.2

In this proof we consider the following transition matrix:

$$P(q,q) = 1-\eta \quad \text{and } P(q,\mathbf{1}-q) = \eta,$$

i.e., all the features change simultaneously with probability η at each round. We also assume that the prior is only split between the vectors $\mathbf{0}$ and $\mathbf{1}$, i.e., the features are either all 0 or all 1. If we take the reward function $r(q,\theta) = M\sum_{i=1}^{d} q_i + \theta_i$, then the regret scales as

$$\begin{aligned} R_T &= \Omega\left(M\sum_{i=1}^{d}\sum_{t=1}^{T}\mathbb{E}[1-\pi_t^{(i)}(Q_t^{(i)})]\right) \\ &= \Omega\left(Md\sum_{t=1}^{T}\mathbb{E}[1-\pi_t(Q_t)]\right). \end{aligned} \tag{9.30}$$

In this model, we thus have the following posterior update

$$\pi_{t+1}(\mathbf{1}) = (1-2\eta)\frac{G(Z_t,\pi_t,\mathbf{1})}{G(Z_t,\pi_t)}\pi_t(\mathbf{1}) + \eta. \tag{9.31}$$

This proof uses a partitioning in blocks as follows

$$t_1 \coloneqq 0 \quad \text{and} \quad t_{k+1} \coloneqq \min\left\{t > t_k \mid Q_{t+1} \neq Q_{t_k+1}\right\}. \tag{9.32}$$

Consider the block k and assume w.l.o.g. that $Q_t = \mathbf{1}$ for this block. Define the stopping time

$$\tau_k \coloneqq \min\left(\left\{t \in [t_k+1, t_{k+1}] \;\middle|\; \pi_t(\mathbf{1}) \geq \frac{1}{2}\right\} \cup \{t_{k+1}\}\right), \tag{9.33}$$

and similarly for τ_{k+1} (with $\mathbf{0}$).

The estimation error incurred during blocks k and $k+1$ is at least $(\tau_k - t_k + \tau_{k+1} - t_{k+1})/2$.

Given the posterior update, $\pi_{t+1}(\mathbf{1}) \leq c\pi_t(\mathbf{1})$ where $c = 1 + \max_{\pi,z}\frac{G(z,\pi,\mathbf{1})}{G(z,\pi,\mathbf{0})}$. As a consequence, $\tau_{k+1} - t_{k+1} \geq \min\left(-\frac{\ln(2\pi_{t_{k+1}}(\mathbf{0}))}{\ln(c)}, t_{k+2} - t_{k+1}\right)$. Assume in the following that $t_{k+2} - t_{k+1} \geq -\frac{\ln(2\eta)}{\ln(c)}$, so that we actually have $\tau_{k+1} - t_{k+1} \geq -\frac{\ln(2\pi_{t_{k+1}}(0))}{\ln(c)}$.

We now bound $\ln(\pi_{t_{k+1}}(\mathbf{0}))$ in expectation. By concavity of the logarithm,

$$\mathbb{E}[\ln(\pi_{t_{k+1}}(\mathbf{0})) \mid (t_n)_n, \tau_k] \leq \ln\left(\mathbb{E}[\pi_{t_{k+1}}(\mathbf{0}) \mid (t_n)_n, \tau_k]\right).$$

Note that the estimator $\widetilde{\pi}_t$ in the proof of the upper bound is similar to π_t for the transition

matrix considered here. Equation (9.29) then yields

$$\mathbb{E}[\pi_{t_{k+1}}(\mathbf{0}) \mid (t_n)_n, \tau_k] \le \exp\left(-\frac{(t_{k+1}-\tau_k)\overline{\delta}^4}{2\overline{\gamma}^2+4\overline{\delta}^2}\right) + \frac{\eta}{1-\exp\left(-\frac{\overline{\delta}^4}{2\overline{\gamma}^2+4\overline{\delta}^2}\right)},$$

where

$$\overline{\delta} \coloneqq \min_{\pi\in\mathcal{P}(\mathcal{Q})} \sum_{z\in\mathcal{Z}} |G(z,\pi,\mathbf{1}) - G(z,\pi,\mathbf{0})| \quad \text{and} \quad \overline{\gamma} \coloneqq 2 \max_{\pi\in\mathcal{P}(\mathcal{Q}),z\in\mathcal{Z}} \left|\ln\left(\frac{G(z,\pi,\mathbf{1})}{G(z,\pi,\mathbf{0})}\right)\right|.$$

And so, with $t_{k+2}-t_{k+1} \ge -\frac{\ln(\eta)}{\ln(c)}$,

$$\begin{aligned}\mathbb{E}[\tau_k - t_k + \tau_{k+1} - t_{k+1} \mid (t_n)_n, \tau_k] &\ge \tau_k - t_k + \Omega\left(-\ln\left(\exp\left(-\frac{(t_{k+1}-\tau_k)\overline{\delta}^4}{2\overline{\gamma}^2+4\overline{\delta}^2}\right) + \frac{\eta}{1-\exp\left(-\frac{\overline{\delta}^4}{2\overline{\gamma}^2+4\overline{\delta}^2}\right)}\right)_+\right)\\ &\ge \tau_k - t_k + \Omega\left(\left(-\ln(\eta) - \frac{1}{\eta}\exp\left(-\frac{(t_{k+1}-\tau_k)\overline{\delta}^4}{2\overline{\gamma}^2+4\overline{\delta}^2}\right)\right)_+\right).\end{aligned}$$

Where we used the convex inequality $-\ln(x+y) \ge -\ln(x) - y/x$.

When looking at the variations of the right hand side with τ_k, it is minimized either when $\tau_k = t_k$ or when the second term is equal to 0, i.e., $t_{k+1}-\tau_k = \Omega(\ln(1/\eta))$. Finally this yields when $t_{k+2}-t_{k+1} \ge -\frac{\ln(2\eta)}{\ln(c)}$:

$$\begin{aligned}\mathbb{E}[\tau_k - t_k + \tau_{k+1} - t_{k+1} \mid (t_n)_n] \ge \Omega\Big(\min\Big(\quad &\ln(1/\eta) - \frac{1}{\eta}\exp\left(-\frac{(t_{k+1}-t_k)\overline{\delta}^4}{2\overline{\gamma}^2+4\overline{\delta}^2}\right),\\ &\ln(1/\eta) + t_{k+1} - t_k\Big)\Big).\end{aligned} \tag{9.34}$$

Case $\eta T \ge 32$. Recall that $t_{k+1}-t_k$ are i.i.d. geometric variables of parameter η. Lemma 9.4 below provides some concentration bound for the sum of such variables. Its proof is given at the end of the section.

Lemma 9.4. *Denote by $Y(n,p)$ the sum of n i.i.d. geometric variables of parameter p. We have the following concentration bounds on $Y(n,p)$:*

1. *For $k \le 1$ and $kn/p \in \mathbb{N}$, $\mathbb{P}[Y(n,p) < kn/p] \le \exp\left(-\frac{(1-1/k)^2 kn}{1+1/k}\right)$.*
2. *For $k \ge 1$ and $kn/p \in \mathbb{N}$, $\mathbb{P}[Y(n,p) > kn/p] \le \exp\left(-\frac{(1-1/k)^2 kn}{2}\right)$.*

Let $\beta \in [\frac{1}{4}, \frac{1}{2}]$ such that $\beta\eta T \in 2\mathbb{N}$ and note that $\mathbb{1}\left(t_{k+1} - t_k \geq x\right)$ follows a Bernoulli distribution of parameter smaller than $(1-\eta)^{\lceil x \rceil}$. We then have the following concentration bounds:

$$\mathbb{P}\left[\sum_{k=1}^{\beta\eta T} t_{k+1} - t_k > T\right] \leq \exp\left(-\frac{(1-\beta)^2\eta T}{2}\right) \leq \exp\left(-\frac{\eta T}{8}\right) \leq e^{-4}. \tag{9.35}$$

and

$$\mathbb{P}\left[\sum_{k=1}^{\beta\eta T/2} \mathbb{1}\left(t_{2k+1} - t_{2k} \geq \frac{1}{\eta}\right)\mathbb{1}\left(t_{2k+2} - t_{2k+1} \geq -\frac{\ln(2\eta)}{\ln(c)}\right) \leq \frac{\beta\eta T}{4}(1-\eta)^{\frac{1}{\eta} - \frac{\ln(2\eta)}{\ln(c)}}\right] \leq \exp\left(-\frac{\beta\eta T(1-\eta)^{\frac{1}{\eta} - \frac{\ln(2\eta)}{\ln(c)}}}{16}\right). \tag{9.36}$$

The first bound is a direct consequence of Lemma 9.4 while the second one is an application of Chernoff bound to Bernoulli variables of parameter $(1-\eta)^{\lceil \frac{1}{\eta} \rceil - \lfloor \frac{\ln(2\eta)}{\ln(c)} \rfloor}$. Recall that we only consider small η. We can thus assume that η is small enough so that $\frac{1}{\eta} \geq -\frac{\ln(2\eta)}{\ln(c)}$. The second bound then becomes:

$$\mathbb{P}\left[\sum_{k=1}^{\beta\eta T/2} \mathbb{1}\left(t_{2k+1} - t_{2k} \geq \frac{1}{\eta}\right)\mathbb{1}\left(t_{2k+2} - t_{2k+1} \geq -\frac{\ln(2\eta)}{\ln(c)}\right) \leq \frac{\beta\eta T}{4}(1-\eta)^{\frac{2}{\eta}}\right] \leq \exp\left(-\frac{\beta\eta T(1-\eta)^{\frac{2}{\eta}}}{16}\right).$$

Note that for any $x \in (0, \frac{1}{2})$, $e^{-3} \leq (1-x)^{2/x}$, so that the last inequality implies for $\eta \leq \frac{1}{2}$

$$\mathbb{P}\left[\sum_{k=1}^{\beta\eta T/2} \mathbb{1}\left(t_{2k+1} - t_{2k} \geq \frac{1}{\eta}\right)\mathbb{1}\left(t_{2k+2} - t_{2k+1} \geq -\frac{\ln(\eta)}{\ln(c)}\right) \leq \frac{\beta\eta T}{4}e^{-3}\right] \leq \exp\left(-\frac{\beta\eta T e^{-3}}{16}\right) \leq \exp\left(-\frac{7e^{-3}}{8}\right).$$

Now note that $e^{-4} + e^{-\frac{7e^{-3}}{8}} < 1$ so that neither the event in Equation (9.35) nor in Equation (9.36) hold with some constant probability. In that case, Equation (9.35) means that the $\beta\eta T$ first blocks fully count in the regret. Equation (9.36) implies that Equation (9.34) holds for at least $\Omega(\eta T)$ pairs of blocks and for each of them, the incurred error is at least $\Omega(\ln(1/\eta))$. This finally implies that $\mathcal{L}_T = \Omega(\ln(1/\eta)\eta T)$ and similarly for the regret. We conclude by summing over all the coordinates.

Case $\eta T \leq 32$. Since $\eta T = \Omega(1)$, we can consider a constant $c_0 > 0$ such that $\eta T > c_0$. In that case, the desired bound can actually be obtained on the two first blocks only. Assume

w.l.o.g. for simplicity that T is a multiple of 4.

$$\begin{aligned}\mathbb{P}\left(t_1 - t_0 \in [T/4, T/2] \text{ and } t_2 - t_1 \in [T/4, T/2]\right) &= \left((1-\eta)^{T/4} - (1-\eta)^{T/2}\right)^2 \\ &= e^{\frac{T}{2}\ln(1-\eta)}\left(1 - e^{\frac{T}{4}\ln(1-\eta)}\right)^2 \\ &= e^{-\frac{\eta T}{2}}\left(1 - e^{-\eta T/2}\right)^2.\end{aligned}$$

With a positive probability depending only on c_0, the two first blocks are completed before T, $t_1 - t_0 \geq T/4$ and $t_2 - t_1 \geq T/4$. Assuming w.l.o.g. that η is small enough so that $T/4 \geq -\frac{\ln(2\eta)}{\ln(c)}$, Equation (9.34) then gives that the loss incurred during the two first blocks is $\Omega\left(\ln(1/\eta)\right)$. As $\eta T = \mathcal{O}\left(1\right)$ in this specific case, this still leads to

$$\sum_{t=1}^{T} \mathbb{E}[1 - \pi_t(Q_t)] = \Omega\left(\ln(1/\eta)\eta T\right).$$

This allows to conclude using Equation (9.30).

Proof of Lemma 9.4. Note that the probability that the sum of n i.i.d. geometric variables of parameter p are smaller than kn/p is exactly the probability that the sum of kn/p i.i.d. Bernoulli variables are larger than n. We can then use the Chernoff bound on these kn/p Bernoulli variables. The same reasoning also leads to the second inequality. □

9.A.3 Proof of Theorem 9.3

This proof uses the block partitioning given by Equation (9.32) and the same transition matrix and reward as in Section 9.A.2. It relies on intermediate results given by Lemma 9.5. Its proof can be found below.

Lemma 9.5. *For any $t \in [t_k + 1, t_{k+1}]$,*

1. $\pi_t^{\text{naive}}(Q_t) \leq c^{t-t_k}\pi_{t_k}^{\text{naive}}(Q_t)$;
2. $\mathbb{E}\left[\ln(\pi_t^{\text{naive}}(q)) \mid (t_n)_n, \pi_{t_k}^{\text{naive}}, Q_t \neq q\right] \leq -(t-t_k)\frac{\delta^4}{2\gamma^2+4\delta^2} - \ln(\pi_{t_k}^{\text{naive}}(Q_t))$;
3. $\mathbb{P}\Big[\ln(\pi_t^{\text{naive}}(q)) - \mathbb{E}\left[\ln(\pi_t^{\text{naive}}(q)) \mid (t_n)_n, \pi_{t_k}^{\text{naive}}\right] \geq \lambda\gamma d\sqrt{t-t_k} \mid (t_n)_n, \pi_{t_k}^{\text{naive}}\Big] \leq \exp\left(-2\lambda^2\right)$;

where $c = \max_{\pi,z,q,q'} \frac{G(z,\pi,q)}{G(z,\pi,q)}$.

Consider two successive blocks k and $k+1$, where the quality is q on the block k and q' on the block $k+1$. Similarly to the proof of Theorem 9.2, define $\tau_{k+1} = \min\Big(\{t \in [t_{k+1}+1, t_{k+2}] \mid \pi_t^{\text{naive}}(q') \geq 1/2\} \cup \{t_{k+2}\}\Big)$. We define τ_k similarly.

The first point of Lemma 9.5 implies that $\tau_{k+1}-t_{k+1} \geq \min\left(t_{k+2}-t_{k+1}, \frac{-\ln(2)-\ln(\pi^{\text{naive}}_{t_{k+1}}(q'))}{\ln(c)}\right)$.

Moreover, thanks to the second and third points of Lemma 9.5, with probability at least $1-e^{-2\lambda^2}$ for some $\lambda > 0$,

$$-\ln(\pi^{\text{naive}}_{t_{k+1}}(q')) \geq (t_{k+1}-t_k)\frac{\delta^4}{2\gamma^2+4\delta^2} + \ln(\pi^{\text{naive}}_{t_k}(q)) - \lambda\gamma d\sqrt{t_{k+1}-t_k}. \tag{9.37}$$

Either $\pi^{\text{naive}}_{t_k}(q) \geq \exp\left(-(t_{k+1}-t_k)\frac{\delta^4}{4\gamma^2+8\delta^2}\right)$, in which case the two first terms in Equation (9.37) are larger than $(t_{k+1}-t_k)\frac{\delta^4}{4\gamma^2+8\delta^2}$.

Otherwise, $\pi^{\text{naive}}_{t_k}(q) \leq \exp\left(-(t_{k+1}-t_k)\frac{\delta^4}{4\gamma^2+8\delta^2}\right)$. Using the first point of Lemma 9.5, this yields that for the $\frac{-\ln(2)}{\ln(c)} + (t_{k+1}-t_k)\frac{\delta^4}{(4\gamma^2+8\delta^2)\ln(c)}$ first steps of the block k, $\pi^{\text{naive}}_t(q) \leq \frac{1}{2}$, i.e., $\tau_k - t_k \geq \frac{-\ln(2)}{\ln(c)} + (t_{k+1}-t_k)\frac{\delta^4}{(4\gamma^2+8\delta^2)\ln(c)}$.

So we can actually bound the error in expectation:

$$\begin{aligned}\mathbb{E}[\tau_{k+1}-t_{k+1}+\tau_k-t_k|(t_n)_n] \geq &(1-e^{-2\lambda^2})\min\left(t_{k+2}-t_{k+1}, \frac{\delta^4\,(t_{k+1}-t_k)}{(4\gamma^2+8\delta^2)\max(1,\ln(c))}\right)\\ &- \frac{\ln(2)+\lambda\gamma d\sqrt{t_{k+1}-t_k}}{\ln(c)}.\end{aligned} \tag{9.38}$$

Case $\eta \geq 32$. Consider $\beta \in [\frac{1}{4}, \frac{1}{2}]$ such that $\beta\eta T \in 2\mathbb{N}^*$. As $t_{k+1}-t_k$ are dominated by geometric variables of parameter η, we can show similarly to Equations (9.35) and (9.36) in the proof of Theorem 9.2 that

1. $\mathbb{P}\left[\sum_{k=1}^{\beta\eta T} t_{k+1}-t_k > T\right] \leq e^{-4}$;
2. $\mathbb{P}\left[\sum_{k=1}^{\beta\eta T/2} \mathbb{1}\left(t_{2k+1}-t_{2k} \geq \frac{2}{\eta}\right)\mathbb{1}\left(t_{2k+2}-t_{2k+1} \geq \frac{2}{\eta}\right) \leq \frac{\beta\eta T}{4}(1-\eta/2)^{\frac{4}{\eta}}\right] \leq \exp\left(-\frac{7e^{-3}}{8}\right)$.

Similarly to the proof of Theorem 9.2, the sum of these two probabilities is below 1, so that none of these two events can happen with probability $\Omega(1)$. When it is the case, the first point yields that the $\beta\eta T$ first blocks totally count in the estimation error before T. The second point implies, thanks to Equation (9.38), that the estimation loss is $\Omega(T)$ in this case.

Case $\eta T \leq 32$. Since $\eta T = \Omega(1)$, we can consider a constant $c_0 > 0$ such that $\eta T > c_0$. Similarly to the case $\eta T \leq 32$ in the proof of Theorem 9.2, we can show that with a positive probability depending only on c_0, the two first blocks are completed before T and

$\min(t_1 - t_0, t_2 - t_1) \geq T/4$. In that case, Equation (9.38) yields that the estimation loss incurred during the two first blocks is $\Omega(T)$, which leads to a regret $\Omega(MdT)$.

Proof of Lemma 9.5.
1) This is a direct consequence of the posterior update given by Equation (9.7).

2) Jensen's inequality gives that

$$\mathbb{E}\left[\ln(\pi_t^{\text{naive}}(q)) \mid (t_n)_n, \pi_{t_k}^{\text{naive}}\right] \leq \ln\left(\mathbb{E}\left[\pi_t^{\text{naive}}(q) \mid (t_n)_n, \pi_{t_k}^{\text{naive}}\right]\right).$$

Theorem 9.1 claims that

$$\mathbb{E}\left[\pi_t^{\text{naive}}(q) \mid (t_n)_n, \pi_{t_k}^{\text{naive}}\right] \leq \exp\left(-(t-t_k)\frac{\delta^4}{2\gamma^2+4\delta^2}\right)\frac{1}{\pi_{t_k}^{\text{naive}}(Q_t)},$$

leading to the second point.

3) Recall that $\ln(\pi_t^{\text{naive}}(q)) = \ln(\pi_{t_k}^{\text{naive}}(q)) + \sum_{s=t_k}^{t-1} \ln\left(\frac{G(Z_s,\pi_s,q)}{G(Z_s,\pi_s)}\right)$ and that $\ln\left(\frac{G(Z_s,\pi_s,q)}{G(Z_s,\pi_s)}\right) \in [Y_s, Y_s + \gamma d]$ for some variable Y_s. The third point is then a direct application of Azuma-Hoeffding's inequality as used in the proof of Theorem 9.1. □

9.B Continuous quality

We consider in this section the continuous case where $\mathcal{Q}$ is some continuous set and show that, in the dynamic model described by Equation (9.15), the regret is upper bounded by $\mathcal{O}\left(M\eta^{1/4}T\right)$ and lower bounded by $\Omega(M\eta^{1/2}T)$ when the reward function is M-Lipschitz. Closing the gap between these two bounds is left open for future work.

9.B.1 Continuous model

In the whole section, the quality space $\mathcal{Q}$ is a convex and compact subset of $\mathbb{R}^d$. Assumption 9.1 is specific to the discrete model and we use an equivalent assumption in the continuous case.

Assumption 9.3 (Purchase guarantee, continuous case)**.** *The function r is non-decreasing in each feature $q^{(i)}$ and there is some $\underline{q} \in \mathbb{R}^d$ such that $\forall i \in [d], q \in \mathcal{Q}, \underline{q}^{(i)} \leq q^{(i)}$ and $\mathbb{P}_{\theta_t}(r(\underline{q}, \theta_t) > 0) > 0$.*

In the continuous case, an additional assumption is required to get fast convergence of the posterior.

Assumption 9.4 (Monotone feedback)**.** *For any $i \in \{1, \dots, d\}$ and $\pi_t \in \mathcal{P}(\mathcal{Q})$, $G^{(i)}(z^{(i)}, \pi_t, \cdot)$ defined by Equation* (9.2) *is continuously differentiable and strictly monotone in $q^{(i)}$ for some $z \in \mathcal{Z}$.*

This assumption guarantees that for two different qualities, the distributions of observed feedbacks are different enough. Note that G_i does not have to be strictly monotone in $q^{(i)}$ for all $z \in \mathcal{Z}$, but only for one of them. For instance in the sparse feedback model, the probability of observing $z^{(i)} = *$ indeed does not depend on the quality as it corresponds to the absence of review. Requiring the monotonicity only for some z_i is thus much weaker than for all of them.

9.B.2 Stationary environment

Consider as a warmup in this section the static case $Q_t = Q_1$ for all $t \in \mathbb{N}$. The arguments from Section 9.3 cannot be adapted to this case for two reasons. First, the pointwise convergence was shown using the fact that the posterior was upper bounded by 1, but a similar bound does not hold for density functions. Second, even the pointwise convergence of the posterior does not give a good enough rate of convergence for the estimated quality. Instead, we first show the existence of a "good" non-Bayesian estimator. The Bayes estimator will also have similar, if not better, performances as it minimizes the Bayesian risk.

We first show the existence of a good non-Bayesian estimator. Define $L_t(z)$ as the empirical probability of observing the feedback z, i.e., $L_t(z) = \frac{1}{t}\sum_{s=1}^{t-1} \mathbb{1}(Z_s = z)$. Also define for any posterior π and quality q:

$$\psi(\pi, q) \coloneqq (z \mapsto G(z, \pi, q)), \tag{9.39}$$

where G is defined by Equation (9.1). The function $\psi(\pi, q)$ is simply the probability distribution of the feedback, given the posterior π and the quality q.

Lemma 9.6. *Under Assumptions 9.3 and 9.4,*

$$\mathbb{E}\left[\left\|\overline{\psi}^{\dagger}_{t+1}(L_{t+1}) - Q\right\|_2^2\right] = \mathcal{O}(1/t),$$

where

$$\overline{\psi}_{t+1}(\cdot) \coloneqq \frac{1}{t}\sum_{s=1}^{t} \psi(\pi_s, \cdot)$$

and

$$\begin{aligned}\overline{\psi}^{\dagger}_{t+1}(L_{t+1}) &\coloneqq \operatorname*{arg\,min}_{Q \in \mathcal{Q}} \|L_{t+1} - \overline{\psi}_{t+1}(Q)\|_2^2 \\ &= \operatorname*{arg\,min}_{Q \in \mathcal{Q}} \sum_{z \in \mathcal{Z}} (L_{t+1}(z) - \overline{\psi}_{t+1}(Q)(z))^2.\end{aligned}$$

The † operator is a generalized inverse operator, i.e., $f^\dagger$ is the composition of f^{-1} with the projection on the image of f. For a bijective function, it is then exactly its inverse.

The $\arg\min$ above is well defined by continuity of $\overline{\psi}_{t+1}$ and compactness of $\mathcal{Q}$. Assumption 9.4 implies that $\overline{\psi}_{t+1}$ is injective. Thanks to this, the function $\overline{\psi}^\dagger_{t+1}$ is well defined.

Here L_{t+1} is the empirical distribution of the feedback. The function $\overline{\psi}^\dagger_{t+1}$ then returns the quality that best fits this empirical distribution.

Proof. Note that $L_{t+1}(z) = \frac{1}{t}\sum_{s=1}^t \mathbb{1}\left(Z_s = z\right)$, where $\mathbb{E}\left[\mathbb{1}\left(Z_s = z\right) \mid \mathcal{H}_s, Q\right] = G(z, \pi_s, Q)$. As we consider the variance of a sum of martingales, we have

$$\mathbb{E}\left[\Big(L_{t+1}(z) - \overline{\psi}_{t+1}(Q)(z)\Big)^2 \;\Big|\; Q\right] = \frac{1}{t^2}\sum_{s=1}^t \mathrm{Var}(\mathbb{1}\left(Z_s = z\right) \mid Q, \pi_s).$$

From this, we deduce a convergence rate $1/t$:

$$\begin{aligned}\mathbb{E}\left[\|L_{t+1} - \overline{\psi}_{t+1}(Q)\|_2^2 \,\Big|\, Q\right] &\leq \frac{1}{t^2}\sum_{s=1}^t \sum_{z\in\mathcal{Z}} \mathrm{Var}(\mathbb{1}\left(Z_s = z\right) \mid Q, \pi_s) \\ &\leq \frac{1}{t^2}\sum_{s=1}^t \sum_{z\in\mathcal{Z}} \mathbb{P}(Z_s = z \mid Q, \pi_s) = \frac{1}{t}. \end{aligned} \tag{9.40}$$

As $G^{(i)}$ is strictly monotone in $q^{(i)}$ and continuously differentiable on $\mathcal{Q}$ for some $z^{(i)}$, the absolute value of its derivative in $q^{(i)}$ is lower bounded by some positive constant. As a consequence, for some $\lambda > 0$,

$$\forall q, q' \in \mathcal{Q}, \|q - q'\| \leq \lambda \|\overline{\psi}_{t+1}(q) - \overline{\psi}_{t+1}(q')\|. \tag{9.41}$$

For $\hat{Q} = \overline{\psi}^\dagger_{t+1}(L_{t+1}) = \arg\min_{Q\in\mathcal{Q}} \|L_{t+1} - \overline{\psi}_{t+1}(Q)\|^2$, it follows

$$\begin{aligned}\mathbb{E}\left[\|\hat{Q} - Q\|_2^2 \,\Big|\, Q\right] &\leq \lambda \mathbb{E}\left[\|\overline{\psi}_{t+1}(\hat{Q}) - \overline{\psi}_{t+1}(Q)\|_2^2 \,\Big|\, Q\right] \\ &\leq 2\lambda \mathbb{E}\left[\|L_{t+1} - \overline{\psi}_{t+1}(\hat{Q})\|_2^2 \,\Big|\, Q\right] + 2\lambda \mathbb{E}\left[\|L_{t+1} - \overline{\psi}_{t+1}(Q)\|_2^2 \,\Big|\, Q\right] \\ &\leq 4\lambda \mathbb{E}\left[\|L_{t+1} - \overline{\psi}_{t+1}(Q)\|_2^2 \,\Big|\, Q\right] \leq \frac{\lambda}{t}.\end{aligned}$$

The third inequality is given by the definition of $\hat{Q}$ as a minimizer of the distance to L_{t+1}, and Lemma 9.6 follows thanks to Equation (9.40). □

In the sequel we use the notation $M_t = \mathbb{E}[Q \mid \mathcal{H}_t]$. Lemma 9.6 gives a non-Bayesian estimator that converges to Q at rate $1/t$ in quadratic loss. Using arguments similar to Besbes and Scarsini, 2018, this implies that $M_t \xrightarrow{a.s.} Q$, thanks to a result from Le Cam and Yang, 2000. Theorem 9.4 yields a different result: M_t converges to Q at a rate $1/\sqrt{t}$ in average.

Theorem 9.4. *Under Assumptions 9.3 and 9.4, then* $\mathbb{E}\Big[\,\|M_{t+1}-Q\|_1\,\Big]=\mathcal{O}\left(\sqrt{d/t}\right)$.

The hidden constant in the $\mathcal{O}(\cdot)$ above depends only the parameter λ appearing in Equation (9.41), which depends on the functions $G^{(i)}$. The above bound directly leads to a $\mathcal{O}\left(\sqrt{d/t}\right)$ regret when the reward is M-Lipschitz (for the 1-norm).

Note that the rate $\mathcal{O}\left(\sqrt{d/t}\right)$ is the best rate possible even if the reviews report exactly $Q+\varepsilon_t$. Indeed, the best estimator in this case is the average quality $\frac{1}{t}\sum_{s<t}(Q+\varepsilon_s)$, which behaves as a Gaussian variable with a variance of order $1/t$ by the central limit theorem on each coordinate. The error $\mathbb{E}\Big[\,\|M_{t+1}-Q\|\,\Big]$ is then of the same order as the square root of the trace of the covariance matrix of M_{t+1}, i.e., $\sqrt{d/t}$.

Proof. A characterization of the Bayes estimator is that it minimizes the Bayesian mean square error among all $\mathcal{H}_t$-measurable functions. In particular,

$$\mathbb{E}\Big[\,\|M_{t+1}-Q\|_2^2\,\Big]\le\mathbb{E}\left[\left\|\overline{\psi}^{\dagger}_{t+1}(L_{t+1})-Q\right\|_2^2\right].$$

Thanks to Lemma 9.6, this term is $\mathcal{O}(1/t)$ and Theorem 9.4 then follows by comparison of the 1 and 2-norms. □

9.B.3 Dynamical environment

We now consider the dynamical setting given by Equation (9.15). The Markov chain is here continuous, but the quality still has a probability to stay the same $1-\eta$ at each round. As in the stationary case, we first expose a satisfying non-Bayesian estimator, implying similar bounds on the posterior distribution.

In the stationary case, our non-Bayesian estimator comes from the empirical distribution of the feedback. As highlighted by Equation (9.18), with a dynamical quality, recent reviews have a larger weight in the posterior. This leads to the following adapted discounted estimator for $\eta_1\in(0,1)$:

$$L_t^{\eta_1}(z)\coloneqq\eta_1\textstyle\sum_{s=1}^{t-1}(1-\eta_1)^{t-s-1}\mathbb{1}\left(Z_s=z\right). \tag{9.42}$$

Lemma 9.7 below bounds the mean error for the estimator $L_t^{\eta_1}$.

Lemma 9.7. *Under Assumptions 9.3 and 9.4, for* $\eta_1=\sqrt{\eta}$,

$$\sum_{t=1}^{T}\sqrt{\mathbb{E}\left[\left\|L_t^{\eta_1}(z)-\overline{\psi}_{t,\eta_1}(Q_t)\right\|_2^2\right]}=\mathcal{O}\left(\eta^{1/4}T\right),$$

where

$$\overline{\psi}_{t,\eta_1}(Q)(z) \coloneqq \eta_1 \sum_{s=1}^{t-1}(1-\eta_1)^{t-s-1}G(z,\pi_s,Q).$$

Proof. First fix the qualities $(Q_s)_s$ and blocks $(t_n)_n$ defined as in Equation (9.32). Note that $G(z,\pi_t,Q_t)$ is exactly the expectation of $\mathbb{1}(Z_t = z)$ given $\mathcal{H}_t$ and Q_t. Similarly to the stationary case, we have

$$\mathbb{E}\left[\left(L_t^{\eta_1}(z) - \eta_1\sum_{s=1}^{t-1}(1-\eta_1)^{t-s-1}G(z,\pi_s,Q_s)\right)^2 \;\middle|\; (Q_s)_s\right] = \eta_1^2\sum_{s=1}^{t-1}(1-\eta_1)^{2(t-s-1)}\mathrm{Var}(\mathbb{1}(Z_s=z)\mid Q_s).$$

When summing over all $z \in \mathcal{Z}$, we get the following inequality

$$\mathbb{E}\left[\left\|L_t^{\eta_1} - \eta_1\sum_{s=1}^{t-1}(1-\eta_1)^{t-s-1}G(\cdot,\pi_s,Q_s)\right\|^2 \;\middle|\; (Q_s)_s\right] \le \frac{\eta_1^2}{1-(1-\eta_1)^2} \le \eta_1. \tag{9.43}$$

For $t \in [t_i+1, t_{i+1}]$, we can relate the expected value of $L_t^{\eta_1}(z)$ to $\overline{\psi}_{t,\eta_1}(Q_t)(z)$:

$$\begin{aligned}\Big(\eta_1\sum_{s=1}^{t-1}(1-\eta_1)^{t-s-1}G(z,\pi_s,Q_s) - \overline{\psi}_{t,\eta_1}(Q_t)(z)\Big)^2 &= \eta_1^2\Big(\sum_{s=1}^{t-1}(1-\eta_1)^{t-s-1}\left(G(z,\pi_s,Q_s) - G(z,\pi_s,Q_t)\right)\Big)^2\\ &= \eta_1^2\Big(\sum_{s=1}^{t_i}(1-\eta_1)^{t-s-1}\left(G(z,\pi_s,Q_s) - G(z,\pi_s,Q_t)\right)\Big)^2\\ &\le (1-\eta_1)^{2t-2(t_i+1)}.\end{aligned} \tag{9.44}$$

The second equality holds because $Q_s = Q_t$ for $s > t_i$ by definition of the blocks. In the last inequality, we used the fact that G has values in $[0,1]$, besides comparing the partial sum with $(1-\eta_1)^{t-(t_i+1)}/\eta_1$. This finally gives, for $h(t) \coloneqq \max\left(\{t' < t \mid Q_{t'} \neq Q_t\} \cup \{0\}\right)$,

$$\mathbb{E}\left[\Big(\eta_1\sum_{s=1}^{t-1}(1-\eta_1)^{t-s-1}G(z,\pi_s,Q_s) - \overline{\psi}_{t,\eta_1}(Q_t)(z)\Big)^2 \;\middle|\; (Q_s)_s\right] \le (1-\eta_1)^{2(t-h(t)-1)}. \tag{9.45}$$

When reversing the time, note that $t-h(t)-1$ is the minimum between a geometric variable of parameter η and $t-1$. It follows

$$\begin{aligned}\mathbb{E}\left[\Big(\eta_1\sum_{s=1}^{t-1}(1-\eta_1)^{t-s-1}G(z,\pi_s,Q_s) - \overline{\psi}_{t,\eta_1}(Q_t)(z)\Big)^2\right] &\le \eta\sum_{s=0}^{\infty}(1-\eta)^s(1-\eta_1)^{2s}\\ &\le \frac{\eta}{1-(1-\eta)(1-\eta_1)^2}\end{aligned}$$

$$\leq \frac{\eta}{\eta + \eta_1 - \eta_1 \eta}. \tag{9.46}$$

Noting that $2x^2 + 2y^2 \geq (x+y)^2$, we can now use Equations (9.43) and (9.46) to bound the total error on a round:

$$\mathbb{E}\left[\left(L_t^{\eta_1}(z) - \overline{\psi}_{t,\eta_1}(Q_t)(z)\right)^2\right] \leq 2\eta_1 + \frac{2\eta}{\eta + \eta_1 - \eta_1 \eta}.$$

The error on a single round is of order $\mathcal{O}\left(\eta_1 + \frac{\eta}{\eta+\eta_1}\right)$ in average; and for $\eta_1 = \sqrt{\eta}$, it is then $\mathcal{O}\left(\sqrt{\eta}\right)$ in average. Summing the square root of this term over all rounds finally yields Lemma 9.7. □

Theorem 9.5. *If the reward is M-Lipschitz (for the 1-norm), the regret of Bayesian consumers in the dynamical continuous case is bounded as $R_T = \mathcal{O}\left(M\sqrt{d}\eta^{1/4}T\right)$ under Assumptions 9.3 and 9.4.*

As in the stationary case, the hidden constant in the $\mathcal{O}(\cdot)$ above depends only the parameter λ appearing in Equation (9.41).

Proof of Theorem 9.5 . Similarly to the stationary setting, the error of the Bayesian estimator can be bounded by the error of the non-Bayesian one since the former is the minimizer of the quadratic loss among all $\mathcal{H}_t$-measurable functions:

$$\mathbb{E}\Big[\left\|\mathbb{E}\left[Q_t \mid \mathcal{H}_t\right] - Q_t\right\|^2\Big] \leq \mathbb{E}\left[\left\|\overline{\psi}^{\dagger}_{t,\eta_1}(L_t^{\eta_1}) - Q_t\right\|^2\right].$$

Thanks to Assumption 9.4, $\overline{\psi}_{t,\eta_1}$ verifies Equation (9.41) for some constant $\lambda > 0$ independent from η_1 and T for any $t \geq \frac{1}{\eta_1}$. As we consider $\eta T = \Omega(1)$, the $\frac{1}{\eta_1}$ first terms in the loss are negligible compared $\eta^{1/4}T$. The convergence rate is thus preserved when composing with $\overline{\psi}^{\dagger}_{t,\eta_1}$. Theorem 9.5 then follows using Lemma 9.7 and Jensen's inequality as in the proof of Theorem 9.4. □

In contrast to the discrete case, determining a tight bound in the continuous case remains open for the dynamical setting. Note that the total error is of order at least $M\sqrt{d\eta}T$. Indeed, in the stationary case, no estimator converges faster than a rate $\sqrt{d/t}$. As the length of a block is around $1/\eta$, the loss per block is thus $\Omega\left(\sqrt{d/\eta}\right)$. Thanks to this, a tight bound should be between $M\sqrt{d\eta}T$ and $M\sqrt{d}\eta^{1/4}T$.

A reason for such a discrepancy between the discrete and continuous case might be that the analysis is not tight enough. Especially, the considered non-Bayesian estimators might have a much larger regret than the Bayesian estimator.

On the other hand, reversing the posterior belief after a change of quality already takes a considerable amount of time in the discrete case and causes a loss $\ln(1/\eta)$ against 1 in the stationary case. Here as well, reversing this belief might take a larger time, causing a loss $\eta^{-\frac{3}{4}}$ per block, against $\eta^{-\frac{1}{2}}$ in the stationary case. Showing a tighter lower bound is yet much harder than for the discrete case, as working directly on the Bayesian estimator is more intricate.

Lemma 9.7 uses the non-Bayesian estimator $L_t^{\eta_1}$ with the parameter η_1. Quite surprisingly, $\sqrt{\eta}$ seems to be the best choice for the parameter η_1, despite η being the natural choice. Figure 9.1 below confirms this point empirically on a toy example. The code used for this experiment can be found in the supplementary material. The experiment considers the classical unidimensional setting with:

$$r(Q,\theta) = Q + \theta,$$
$$f(Q,\theta,\varepsilon) = \operatorname{sign}(Q + \theta + \varepsilon).$$

Here, $\mathcal{Q} = [0,1]$, $\eta = 10^{-4}$ and θ and ε both have Gaussian distributions. The Markov Chain is here given as follows

$$\begin{cases} Q_{t+1} = Q_t \text{ with probability } 1-\eta \\ Q_{t+1} = X_{t+1} \text{ otherwise,} \end{cases}$$

where (X_t) is an i.i.d. sequence of random variables drawn from the uniform distribution on $[0,1]$.

Computing the exact posterior $M_t = \mathbb{E}\left[Q_t \mid \mathcal{H}_t\right]$ is intractable, so we remedy this point by assuming $M_t = 1$ all the time. This simplification does not affect the experiments run here as $\overline{\psi}_{t,\eta_1}$ uses M_t only to determine the population of potential buyers.

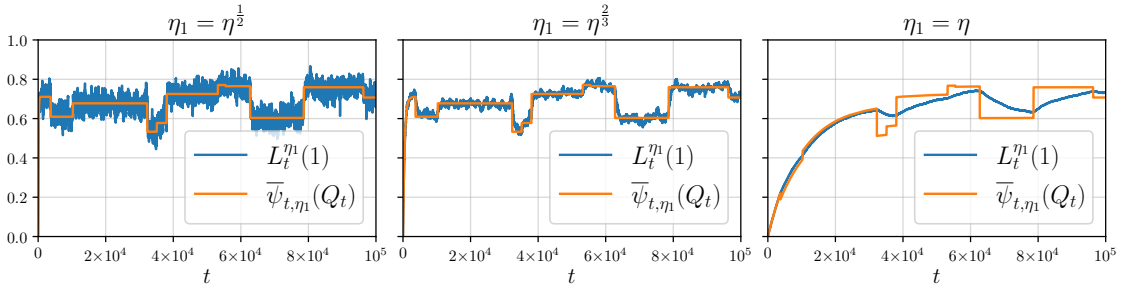
A larger η_1 allows to forget faster past reviews and thus gives a better adaptation after a quality change. However, a larger η_1 also yields a less accurate estimator in stationary phases.

The choice $\eta^{2/3}$ seems to be the best trade-off in Figure 9.1. The optimal choice of η_1 does not only depend on η but also on the distributions of θ and ε. In the considered experiments, η is thus not small enough to ignore these other dependencies. Figure 9.1 yet illustrates the trade-off between small variance and fast adaptivity when tuning η_1.

Figure 9.1: Behavior of L^{η_1} for different η_1.

Value of η_1	$\eta^{1/3}$	$\eta^{1/2}$	$\eta^{2/3}$	η
Error	10166	4780	3060	6462

(a) Estimation error of L^{η_1}. The error is $\sum_{t=1}^{T} \sqrt{\mathbb{E}\left[\left\|L_t^{\eta_1} - \overline{\psi}_{t,\eta_1}(Q_t)\right\|_2^2\right]}$ for $T = 10^5$, where the expectation is estimated by averaging over 2000 instances.



(b) Tracking of $\overline{\psi}_{t,\eta_1}(Q_t)(1)$ by $L_t^{\eta_1}(1)$ over a single instance.

Conclusion

This thesis considered several instances of interacting learning agents and studied how they might behave to maximize either their earned individual reward in the case of selfish agents or the total social welfare in the case of cooperative agents.

In the latter case, we showed that decentralized agents could still reach performances similar to centralized algorithms, implying a negligible cost of decentralization. In particular, we proposed algorithms comparable to centralized ones in multiplayer bandits using collision information as a medium of communication between the players. In the homogeneous case, Chapter 4 proved that the optimal centralized performance is also reachable for decentralized agents. We also proposed in Chapter 5 the first $\log(T)$-regret algorithm in the heterogeneous setting. Similarly, Chapter 7 proposed the first stable decentralized algorithm when the ratio between service and arrival rates is larger than 1, which is also the criterion of stability for centralized algorithms.

The existence of decentralized strategies comparable to centralized algorithms might be due to an oversimplification in the considered models. For example in multiplayer bandits, the proposed algorithms use undesirable communication schemes in practice. Future work on these topics should work on lifting unrealistic model assumptions to avoid such behaviors. Several of these directions for multiplayer bandits were mentioned in Chapter 3 and include the following: more limited feedback (no sensing), different collision models, non-stochasticity of the reward, non-collaborative players or asynchronicity of the players.

Except for this last consideration, algorithms relying on similar communication schemes also yield good performances in these harder models. For instance, the gap between no sensing and collision sensing was recently almost closed (Huang et al., 2021). Also, Chapter 6 proposed an algorithm robust to selfish players that still reach a regret comparable to the centralized case, through the use of a Grim Trigger strategy. Assuming selfish players actually extended the gap between homogeneous and heterogeneous setting, as the optimal allocation is no more reachable and the existence of Grim Trigger strategies is only known with a limited level of heterogeneity. We thus believe that the most crucial and oversimplifying assumption is the synchronicity of the players. We propose in Chapter 4 a first $\log(T)$ regret algorithm in a particular dynamic

setting, yet a lot remains to be found in either more general dynamic or asynchronous settings, where players do not share common time slots. Our algorithms indeed heavily rely on this synchronisation assumption to coordinate the communication schemes between the different players, allowing to transmit messages between them perfectly.

In queuing systems, even selfish players have interest in cooperating, as the decentralized strategy proposed in Chapter 7 is a correlated equilibrium of the patient game introduced by Gaitonde and Tardos (2020b). Here again, the synchronisation assumption is widely used to allow the queues to accurately estimate both the clearing probabilities of the servers and the arrival rates of the other queues. While the model already presents some level of asynchronicity, studying a more asynchronous and/or dynamic model also forms a main focus of future work for this problem, for example through the use of adapted Glauber dynamics (see e.g., Shah and Shin, 2012).

The other problems considered in this thesis raised different issues that also help to grasp the different levels of interplay that might occur between multiple learning agents.

In Chapter 8, we proposed heuristics for balancing short term (greedily bidding) and long term costs (concealing private information) in repeated online auctions. We formalized a new utility-privacy trade-off problem to compute strategies revealing private information only if it induces a significant increase in utility. For classical Bayesian costs, it benefits from recent advances in Optimal Transport. It yet leads to a non-convex minimization problem for which the computation of global minima remains open. We believe that this work is a step towards the design of optimal utility vs. privacy trade-offs in economic mechanisms as well as for other applications. Its numerous connexions with recent topics of interest (optimization, optimal transport) motivate a better understanding of them as future work.

Motivated by the behavior of consumers on review platforms, Chapter 9 was a first attempt to use a change-point framework for social learning in online markets with reviews when the qualities of a product vary over time. We provided a tight bound of the utility loss when the quality space is $\{0,1\}^d$. For more general quality spaces (e.g., continuous), determining the incurred regret is a much harder problem. Many other directions also remain open for review based markets. For instance, it would be interesting to study a model with a slowly drifting quality, rather than abrupt changes. This work only focused on the consumer side, but the seller can also adaptively set the price of the item. What is a good seller strategy in this case? The selling platform can also design the format of the feedback given by the consumers. Determining the format which allows the best possible convergence rate might also be of great interest in practice. Considering perfect Bayesian consumers might be unrealistic. In reality, consumers have limited computation capacity or can be risk averse, leading to different behaviors. Studying

the effects of these limitations is also of great interest.

Finally, this thesis grasped only in a small part the kind of interactions that might happen between learning agents. The observed behaviors highly depend on the considered model as illustrated all along this thesis. Many questions then remain open and we believe that the questions mentioned above are among the most interesting and major questions to better understand sequential learning in strategical environments.

Bibliography

[1] A. Abdulkadiroğlu and T. Sönmez. "Random serial dictatorship and the core from random endowments in house allocation problems". In: *Econometrica* 66.3 (1998), pp. 689–701.

[2] P. Ablin, G. Peyré, and T. Moreau. "Super-efficiency of automatic differentiation for functions defined as a minimum". In: *arXiv preprint arXiv:2002.03722* (2020).

[3] D. Acemoglu, M. A. Dahleh, I. Lobel, and A. Ozdaglar. "Bayesian learning in social networks". In: *Rev. Econ. Stud.* 78.4 (2011), pp. 1201–1236.

[4] D. Acemoglu, A. Makhdoumi, A. Malekian, and A. Ozdaglar. "Fast and slow learning from reviews". Tech. rep. National Bureau of Economic Research, 2017.

[5] M. Adamczyk, P. Sankowski, and Q. Zhang. "Efficiency of truthful and symmetric mechanisms in one-sided matching". In: *International Symposium on Algorithmic Game Theory*. Springer. 2014, pp. 13–24.

[6] R. Agrawal. "The continuum-armed bandit problem". In: *SIAM journal on control and optimization* 33.6 (1995), pp. 1926–1951.

[7] P. Alatur, K. Y. Levy, and A. Krause. "Multi-player bandits: The adversarial case". In: *Journal of Machine Learning Research* 21 (2020).

[8] K. Amin, A. Rostamizadeh, and U. Syed. "Learning prices for repeated auctions with strategic buyers". In: *Advances in Neural Information Processing Systems*. 2013, pp. 1169–1177.

[9] K. Amin, A. Rostamizadeh, and U. Syed. "Repeated contextual auctions with strategic buyers". In: *Advances in Neural Information Processing Systems*. 2014, pp. 622–630.

[10] A. Anandkumar, N. Michael, A. K. Tang, and A. 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.

[11] A. Anandkumar, N. Michael, and A. Tang. "Opportunistic spectrum access with multiple users: Learning under competition". In: *2010 Proceedings IEEE INFOCOM*. IEEE. 2010, pp. 1–9.

[12] V. Anantharam, P. Varaiya, and J. Walrand. "Asymptotically efficient allocation rules for the multiarmed bandit problem with multiple plays-Part I: I.I.D. rewards". In: *IEEE Transactions on Automatic Control* 32.11 (1987), pp. 968–976.

[13] V. Anantharam, P. Varaiya, and J. Walrand. "Asymptotically efficient allocation rules for the multiarmed bandit problem with multiple plays-Part II: Markovian rewards". In: *IEEE Transactions on Automatic Control* 32.11 (1987), pp. 977–982.

[14] I. Arieli and M. Mueller-Frank. "A general analysis of sequential social learning". In: *Math. Oper. Res.* forthcoming (2021).

[15] I. Arieli and M. Mueller-Frank. "Multidimensional social learning". In: *Rev. Econ. Stud.* 86.3 (2019), pp. 913–940.

[16] M. Arjovsky, S. Chintala, and L. Bottou. "Wasserstein generative adversarial networks". In: *International Conference on Machine Learning*. 2017, pp. 214–223.

[17] R. Arora, O. Dekel, and A. Tewari. "Online bandit learning against an adaptive adversary: from regret to policy regret". In: *Proceedings of the 29th International Coference on International Conference on Machine Learning*. 2012, pp. 1747–1754.

[18] A. Attar, H. Tang, A. V. Vasilakos, F. R. Yu, and V. C. Leung. "A survey of security challenges in cognitive radio networks: Solutions and future research directions". In: *Proceedings of the IEEE* 100.12 (2012), pp. 3172–3186.

[19] J. Audibert, S. Bubeck, and G. Lugosi. "Regret in Online Combinatorial Optimization". In: *Math. Oper. Res.* 39.1 (2014), pp. 31–45.

[20] P. Auer, N. Cesa-Bianchi, and P. Fischer. "Finite-time analysis of the multiarmed bandit problem". In: *Machine learning* 47.2-3 (2002), pp. 235–256.

[21] P. Auer, Y. Chen, P. Gajane, C.-W. Lee, H. Luo, R. Ortner, and C.-Y. Wei. "Achieving optimal dynamic regret for non-stationary bandits without prior information". In: *Conference on Learning Theory*. PMLR. 2019, pp. 159–163.

[22] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. 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.

[23] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire. "The nonstochastic multiarmed bandit problem". In: *SIAM journal on computing* 32.1 (2002), pp. 48–77.

[24] R. Aumann, M. Maschler, and R. Stearns. “Repeated games with incomplete information”. MIT press, 1995.

[25] O. Avner and S. Mannor. “Concurrent bandits and cognitive radio networks”. In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer. 2014, pp. 66–81.

[26] O. Avner and S. Mannor. “Learning to coordinate without communication in multi-user multi-armed bandit problems”. In: *arXiv preprint arXiv:1504.08167* (2015).

[27] O. Avner and S. Mannor. “Multi-user communication networks: A coordinated multi-armed bandit approach”. In: *IEEE/ACM Transactions on Networking* 27.6 (2019), pp. 2192–2207.

[28] B. Awerbuch and R. Kleinberg. “Competitive collaborative learning”. In: *Journal of Computer and System Sciences* 74.8 (2008), pp. 1271–1288.

[29] R. Axelrod and W. D. Hamilton. “The evolution of cooperation”. In: *science* 211.4489 (1981), pp. 1390–1396.

[30] M. Ballu, Q. Berthet, and F. Bach. “Stochastic Optimization for Regularized Wasserstein Estimators”. In: *arXiv preprint arXiv:2002.08695* (2020).

[31] M. Bande and V. V. Veeravalli. “Multi-user multi-armed bandits for uncoordinated spectrum access”. In: *2019 International Conference on Computing, Networking and Communications (ICNC)*. IEEE. 2019, pp. 653–657.

[32] M. Bande, A. Magesh, and V. V. Veeravalli. “Dynamic Spectrum Access using Stochastic Multi-User Bandits”. In: *arXiv preprint arXiv:2101.04388* (2021).

[33] A. V. Banerjee. “A simple model of herd behavior”. In: *Quart. J. Econ.* 107.3 (1992), pp. 797–817.

[34] Y. Bar-On and Y. Mansour. “Individual regret in cooperative nonstochastic multi-armed bandits”. In: *arXiv preprint arXiv:1907.03346* (2019).

[35] S. Basu, K. A. Sankararaman, and A. Sankararaman. “Beyond $\log^2(T)$ Regret for Decentralized Bandits in Matching Markets”. In: *arXiv preprint arXiv:2103.07501* (2021).

[36] M. Bayati, N. Hamidi, R. Johari, and K. Khosravi. “Unreasonable Effectiveness of Greedy Algorithms in Multi-Armed Bandit with Many Arms”. In: *Advances in Neural Information Processing Systems* 33 (2020).

[37] J. Berger. “Schach-Jahrbuch fur 1899/1900 : fortsetzung des schach-jahrbuches fur 1892/93”. In: *Verlag von Veit* (1899).

[38] Q. Berthet and V. Perchet. “Fast rates for bandit optimization with upper-confidence Frank-Wolfe”. In: *Advances in Neural Information Processing Systems*. 2017, pp. 2225–2234.

[39] D. P. Bertsekas. “Auction algorithms for network flow problems: A tutorial introduction”. In: *Computational optimization and applications* 1.1 (1992), pp. 7–66.

[40] O. Besbes and M. Scarsini. “On information distortions in online ratings”. In: *Oper. Res.* 66.3 (2018), pp. 597–610.

[41] O. Besbes, Y. Gur, and A. Zeevi. “Non-stationary stochastic optimization”. In: *Oper. Res.* 63.5 (2015), pp. 1227–1244.

[42] O. Besbes, Y. Gur, and A. Zeevi. “Optimal exploration-exploitation in a multi-armed bandit problem with non-stationary rewards”. In: *Stoch. Syst.* 9.4 (2019), pp. 319–337.

[43] L. Besson and E. Kaufmann. “Lower Bound for Multi-Player Bandits: Erratum for the paper Multi-player bandits revisited”. 2019.

[44] L. Besson and E. Kaufmann. “Multi-Player Bandits Revisited”. In: *Algorithmic Learning Theory*. Lanzarote, Spain, 2018.

[45] L. Besson. “Multi-Players Bandit Algorithms for Internet of Things Networks”. PhD thesis. CentraleSupélec, 2019.

[46] L. Besson and E. Kaufmann. “What Doubling Tricks Can and Can’t Do for Multi-Armed Bandits”. In: *arXiv.org:1803.06971* (2018).

[47] L. Besson, E. Kaufmann, O.-A. Maillard, and J. Seznec. “Efficient Change-Point Detection for Tackling Piecewise-Stationary Bandits”. In: (2020).

[48] S. Bikhchandani, D. Hirshleifer, and I. Welch. “A theory of fads, fashion, custom, and cultural change as informational cascades”. In: *J. Polit. Econ.* 100.5 (1992), pp. 992–1026.

[49] G. Birkhoff. “Tres observaciones sobre el algebra lineal”. In: *Univ. Nac. Tucuman, Ser. A* 5 (1946), pp. 147–154.

[50] I. Bistritz and A. Leshem. “Distributed multi-player bandits-a game of thrones approach”. In: *Advances in Neural Information Processing Systems*. 2018, pp. 7222–7232.

[51] I. Bistritz and A. Leshem. “Game of Thrones: Fully Distributed Learning for Multiplayer Bandits”. In: *Mathematics of Operations Research* (2020).

[52] I. Bistritz, T. Z. Baharav, A. Leshem, and N. Bambos. “One for All and All for One: Distributed Learning of Fair Allocations with Multi-player Bandits”. In: *IEEE Journal on Selected Areas in Information Theory* (2021).

[53] A. Blum and Y. Monsour. “Learning, regret minimization, and equilibria”. In: *Algorithmic Game Theory* (2007).

[54] R. Bonnefoi, L. Besson, C. Moy, E. Kaufmann, and J. Palicot. “Multi-Armed Bandit Learning in IoT Networks: Learning helps even in non-stationary settings”. In: *International Conference on Cognitive Radio Oriented Wireless Networks*. Springer. 2017, pp. 173–185.

[55] A. Borodin, J. Kleinberg, P. Raghavan, M. Sudan, and D. P. Williamson. “Adversarial queueing theory”. In: *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*. 1996, pp. 376–385.

[56] L. Bottou. “On-line learning and stochastic approximations”. In: *On-Line Learning in Neural Networks*. Ed. by D. Saad. Publications of the Newton Institute. Cambridge University Press, 1999, pp. 9–42.

[57] F. Bourse, M. Minelli, M. Minihold, and P. Paillier. “Fast homomorphic evaluation of deep discretized neural networks”. In: *Annual International Cryptology Conference*. 2018, pp. 483–512.

[58] E. Boursier and V. Perchet. “Selfish robustness and equilibria in multi-player bandits”. In: *Conference on Learning Theory*. PMLR. 2020, pp. 530–581.

[59] E. Boursier and V. Perchet. “SIC-MMAB: Synchronisation Involves Communication in Multiplayer Multi-Armed Bandits”. In: *Advances in Neural Information Processing Systems* 32 (2019), pp. 12071–12080.

[60] E. Boursier and V. Perchet. “Utility/Privacy Trade-off through the lens of Optimal Transport”. In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2020, pp. 591–601.

[61] E. Boursier, E. Kaufmann, A. Mehrabian, and V. Perchet. “A Practical Algorithm for Multiplayer Bandits when Arm Means Vary Among Players”. In: *AISTATS 2020-23rd International Conference on Artificial Intelligence and Statistics*. 2020.

[62] E. Boursier, T. Garrec, V. Perchet, and M. Scarsini. “Making the most of your day: online learning for optimal allocation of time”. In: *arXiv preprint arXiv:2102.08087* (2021).

[63] E. Boursier, V. Perchet, and M. Scarsini. “Social Learning from Reviews in Non-Stationary Environments”. In: *arXiv preprint arXiv:2007.09996* (2020).

[64] T. Boyarski, A. Leshem, and V. Krishnamurthy. “Distributed learning in congested environments with partial information”. In: *arXiv preprint arXiv:2103.15901* (2021).

[65] S. Boyd and L. Vandenberghe. "Convex optimization". Cambridge university press, 2004.

[66] S. Brânzei and Y. Peres. "Multiplayer bandit learning, from competition to cooperation". In: *arXiv preprint arXiv:1908.01135* (2019).

[67] S. Bubeck and N. Cesa-Bianchi. "Regret analysis of stochastic and nonstochastic multi-armed bandit problems". In: *Foundations and Trends® in Machine Learning* 5.1 (2012), pp. 1–122.

[68] S. Bubeck, T. Wang, and N. Viswanathan. "Multiple Identifications in Multi-Armed Bandits". In: *International Conference on Machine Learning*. 2013, pp. 258–265.

[69] S. Bubeck. "Convex optimization: Algorithms and complexity". In: *arXiv preprint arXiv:1405.4980* (2014).

[70] S. Bubeck and T. Budzinski. "Coordination without communication: optimal regret in two players multi-armed bandits". In: *arXiv preprint arXiv:2002.07596* (2020).

[71] S. Bubeck, T. Budzinski, and M. Sellke. "Cooperative and Stochastic Multi-Player Multi-Armed Bandit: Optimal Regret With Neither Communication Nor Collisions". In: *arXiv preprint arXiv:2011.03896* (2020).

[72] S. Bubeck, Y. Li, Y. Peres, and M. Sellke. "Non-stochastic multi-player multi-armed bandits: Optimal rate with collision information, sublinear without". In: *Conference on Learning Theory*. 2020, pp. 961–987.

[73] B. Çelen and S. Kariv. "Observational learning under imperfect information". In: *Games Econom. Behav.* 47.1 (2004), pp. 72–86.

[74] S. H. Cen and D. Shah. "Regret, stability, and fairness in matching markets with bandit learners". In: *arXiv preprint arXiv:2102.06246* (2021).

[75] N. Cesa-Bianchi, T. Cesari, and V. Perchet. "Dynamic Pricing with Finitely Many Unknown Valuations". In: *Algorithmic Learning Theory*. 2019, pp. 247–273.

[76] N. Cesa-Bianchi and G. Lugosi. "Combinatorial bandits". In: *Journal of Computer and System Sciences* 78.5 (2012), pp. 1404–1422.

[77] N. Cesa-Bianchi and G. Lugosi. "Prediction, learning, and games". Cambridge university press, 2006.

[78] N. Cesa-Bianchi, T. Cesari, and C. Monteleoni. "Cooperative online learning: Keeping your neighbors updated". In: *Algorithmic Learning Theory*. PMLR. 2020, pp. 234–250.

[79] N. Cesa-Bianchi, C. Gentile, and Y. Mansour. "Delay and cooperation in nonstochastic bandits". In: *The Journal of Machine Learning Research* 20.1 (2019), pp. 613–650.

[80] A. Chambolle and T. Pock. “An introduction to continuous optimization for imaging”. In: *Acta Numerica* 25 (2016), pp. 161–319.

[81] K. Chaudhuri, J. Imola, and A. Machanavajjhala. “Capacity bounded differential privacy”. In: *Advances in Neural Information Processing Systems*. 2019, pp. 3469–3478.

[82] N. Chen, A. Li, and K. Talluri. “Reviews and self-selection bias with operational implications”. In: *Management Sci.* forthcoming (2021).

[83] W. Chen, Y. Wang, and Y. Yuan. “Combinatorial multi-armed bandit: General framework and applications”. In: *International Conference on Machine Learning*. 2013, pp. 151–159.

[84] L. Chizat and F. Bach. “On the global convergence of gradient descent for over-parameterized models using optimal transport”. In: *Advances in neural information processing systems*. 2018, pp. 3036–3046.

[85] K. L. Clarkson. “Coresets, sparse greedy approximation, and the Frank-Wolfe algorithm”. In: *ACM Transactions on Algorithms (TALG)* 6.4 (2010), pp. 1–30.

[86] R. Combes, M. S. Talebi, A. Proutiere, and M. Lelarge. “Combinatorial bandits revisited”. In: *Proceedings of the 28th International Conference on Neural Information Processing Systems-Volume 2*. 2015, pp. 2116–2124.

[87] J. Correa, M. Mari, and A. Xia. “Dynamic pricing with Bayesian updates from online reviews”. Tech. rep. 2020.

[88] N Courty, R. Flamary, and D. Tuia. “Domain adaptation with regularized optimal transport”. In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. 2014, pp. 274–289.

[89] D. Crapis, B. Ifrach, C. Maglaras, and M. Scarsini. “Monopoly pricing in the presence of social learning”. In: *Management Sci.* 63.11 (2017), pp. 3586–3608.

[90] M. Cuturi. “Sinkhorn distances: Lightspeed computation of optimal transport”. In: *Advances in Neural Information Processing Systems*. 2013, pp. 2292–2300.

[91] S. J. Darak and M. K. Hanawal. “Multi-player multi-armed bandits for stable allocation in heterogeneous ad-hoc networks”. In: *IEEE Journal on Selected Areas in Communications* 37.10 (2019), pp. 2350–2363.

[92] C. Daskalakis, P. W. Goldberg, and C. H. Papadimitriou. “The complexity of computing a Nash equilibrium”. In: *SIAM Journal on Computing* 39.1 (2009), pp. 195–259.

[93] R. Degenne and V. Perchet. “Anytime optimal algorithms in stochastic multi-armed bandits”. In: *International Conference on Machine Learning*. 2016, pp. 1587–1595.

[94] R. Degenne and V. Perchet. "Combinatorial semi-bandit with known covariance". In: *Advances in Neural Information Processing Systems*. 2016, pp. 2972–2980.

[95] R. Della Vecchia and T. Cesari. "An Efficient Algorithm for Cooperative Semi-Bandits". In: *Algorithmic Learning Theory*. PMLR. 2021, pp. 529–552.

[96] O. Devolder, F. Glineur, and Y. Nesterov. "First-order methods of smooth convex optimization with inexact oracle". In: *Mathematical Programming* 146.1-2 (2014), pp. 37–75.

[97] C. Dwork. "Differential privacy". In: *Encyclopedia of Cryptography and Security* (2011), pp. 338–340.

[98] C. Dwork, F. McSherry, K. Nissim, and A. Smith. "Calibrating noise to sensitivity in private data analysis". In: *Theory of cryptography conference*. Springer. 2006, pp. 265–284.

[99] R. Eilat, K. Eliaz, and X. Mu. "Optimal Privacy-Constrained Mechanisms". Tech. rep. C.E.P.R. Discussion Papers, 2019.

[100] M. Feldman, T. Koren, R. Livni, Y. Mansour, and A. Zohar. "Online pricing with strategic and patient buyers". In: *Advances in Neural Information Processing Systems*. 2016, pp. 3864–3872.

[101] P. Feldman, Y. Papanastasiou, and E. Segev. "Social learning and the design of new experience goods". In: *Management Sci.* 65.4 (2019), pp. 1502–1519.

[102] J. Feydy, T. Séjourné, F.-X. Vialard, S. i. Amari, A. Trouve, and G. Peyré. "Interpolating between Optimal Transport and MMD using Sinkhorn Divergences". In: *The 22nd International Conference on Artificial Intelligence and Statistics*. 2019, pp. 2681–2690.

[103] J. W. Friedman. "A non-cooperative equilibrium for supergames". In: *The Review of Economic Studies* 38.1 (1971), pp. 1–12.

[104] C. Frogner, C. Zhang, H. Mobahi, M. Araya-Polo, and T. Poggio. "Learning with a Wasserstein Loss". In: *Advances in Neural Information Processing Systems*. 2015.

[105] D. Fudenberg and E. Maskin. "The folk theorem in repeated games with discounting or with incomplete information". In: *A Long-Run Collaboration On Long-Run Games*. World Scientific, 2009, pp. 209–230.

[106] D. Fudenberg, F. Drew, D. K. Levine, and D. K. Levine. "The theory of learning in games". Vol. 2. MIT press, 1998.

[107] T. Gafni and K. Cohen. "Distributed Learning over Markovian Fading Channels for Stable Spectrum Access". In: *arXiv preprint arXiv:2101.11292* (2021).

[108] Y. Gai, B. Krishnamachari, and R. Jain. "Combinatorial network optimization with unknown variables: Multi-armed bandits with linear rewards and individual observations". In: *IEEE/ACM Transactions on Networking* 20.5 (2012), pp. 1466–1478.

[109] J. Gaitonde and E. Tardos. "Stability and Learning in Strategic Queuing Systems". In: *arXiv preprint arXiv:2003.07009* (2020).

[110] J. Gaitonde and E. Tardos. "Virtues of Patience in Strategic Queuing Systems". In: *arXiv preprint arXiv:2011.10205* (2020).

[111] A. Garhwal and P. P. Bhattacharya. "A survey on dynamic spectrum access techniques for cognitive radio". In: *International Journal of Next-Generation Networks* 3.4 (2011), p. 15.

[112] A. Garivier and O. Cappé. "The KL-UCB algorithm for bounded stochastic bandits and beyond". In: *Conference On Learning Theory*. 2011, pp. 359–376.

[113] A. Genevay, G. Peyre, and M. Cuturi. "Learning Generative Models with Sinkhorn Divergences". In: *International Conference on Artificial Intelligence and Statistics*. 2018, pp. 1608–1617.

[114] A. Genevay, L. Chizat, F. Bach, M. Cuturi, and G. Peyré. "Sample Complexity of Sinkhorn divergences". In: *The 22nd International Conference on Artificial Intelligence and Statistics*. 2019, pp. 1574–1583.

[115] A. Genevay, M. Cuturi, G. Peyré, and F. Bach. "Stochastic optimization for large-scale optimal transport". In: *Advances in Neural Information Processing Systems*. 2016, pp. 3440–3448.

[116] L. Georgiadis, M. J. Neely, and L. Tassiulas. "Resource allocation and cross-layer control in wireless networks". Now Publishers Inc, 2006.

[117] S. Ghosal, J. K. Ghosh, and A. W. van der Vaart. "Convergence rates of posterior distributions". In: *Ann. Statist.* 28.2 (2000), pp. 500–531.

[118] R. Gilad-Bachrach, N. Dowlin, K. Laine, K. Lauter, M. Naehrig, and J. Wernsing. "Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy". In: *International Conference on Machine Learning*. 2016, pp. 201–210.

[119] N. Golrezaei, A. Javanmard, and V. Mirrokni. "Dynamic incentive-aware learning: Robust pricing in contextual auctions". In: *Advances in Neural Information Processing Systems*. 2019, pp. 9756–9766.

[120] J. Guélat and P. Marcotte. "Some comments on Wolfe's 'away step'". In: *Mathematical Programming* 35.1 (1986), pp. 110–119.

[121] S. Hart and A. Mas-Colell. "A simple adaptive procedure leading to correlated equilibrium". In: *Econometrica* 68.5 (2000), pp. 1127–1150.

[122] E. Hendrix, G. Boglárka, et al. "Introduction to nonlinear and global optimization". Vol. 37. Springer, 2010.

[123] G. Hinton, N. Srivastava, and K. Swersky. "Neural networks for machine learning lecture 6a overview of mini-batch gradient descent". In: *Cited on* 14.8 (2012).

[124] W. Hoeffding. "Probability inequalities for sums of bounded random variables". In: *Journal of the American Statistical Association* 58 (1963), pp. 13–30.

[125] R. Horst and N. Thoai. "DC programming: overview". In: *Journal of Optimization Theory and Applications* 103.1 (1999), pp. 1–43.

[126] R. Horst, P. Pardalos, and N. V. Thoai. "Introduction to global optimization". Springer Science & Business Media, 2000.

[127] W. Huang, R. Combes, and C. Trinh. "Towards Optimal Algorithms for Multi-Player Bandits without Collision Sensing Information". In: *arXiv preprint arXiv:2103.13059* (2021).

[128] B. Ifrach, C. Maglaras, M. Scarsini, and A. Zseleva. "Bayesian social learning from consumer reviews". In: *Oper. Res.* 67.5 (2019), pp. 1209–1221.

[129] M. Jaggi. "Revisiting Frank-Wolfe: Projection-free sparse convex optimization". In: *Proceedings of the 30th international conference on machine learning*. 2013, pp. 427–435.

[130] M. Jedor, J. Louëdec, and V. Perchet. "Be Greedy in Multi-Armed Bandits". In: *arXiv preprint arXiv:2101.01086* (2021).

[131] L. Jiang and J. Walrand. "A distributed CSMA algorithm for throughput and utility maximization in wireless networks". In: *IEEE/ACM Transactions on Networking* 18.3 (2009), pp. 960–972.

[132] H. Joshi, R. Kumar, A. Yadav, and S. J. Darak. "Distributed algorithm for dynamic spectrum access in infrastructure-less cognitive radio network". In: *2018 IEEE Wireless Communications and Networking Conference (WCNC)*. 2018, pp. 1–6.

[133] W. Jouini, D. Ernst, C. Moy, and J. Palicot. "Multi-armed bandit based policies for cognitive radio's decision making issues". In: *2009 3rd International Conference on Signals, Circuits and Systems (SCS)*. 2009.

[134] W. Jouini. "Contribution to learning and decision making under uncertainty for Cognitive Radio." PhD thesis. 2012.

[135] W. Jouini, D. Ernst, C. Moy, and J. Palicot. "Upper confidence bound based decision making strategies and dynamic spectrum access". In: *2010 IEEE International Conference on Communications*. IEEE. 2010, pp. 1–5.

[136] A. Kakhbod and G. Lanzani. "Dynamic trading in product markets with heterogeneous learning technologies". Tech. rep. MIT, 2021.

[137] D. Kalathil, N. Nayyar, and R. Jain. "Decentralized learning for multiplayer multiarmed bandits". In: *IEEE Transactions on Information Theory* 60.4 (2014), pp. 2331–2345.

[138] E. Kaufmann, N. Korda, and R. Munos. "Thompson Sampling: An Asymptotically Optimal Finite Time Analysis". In: *Algorithmic Learning Theory* (2012).

[139] D. Kingma and J. Ba. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).

[140] P. Knight. "The Sinkhorn–Knopp algorithm: convergence and applications". In: *SIAM Journal on Matrix Analysis and Applications* 30.1 (2008), pp. 261–275.

[141] J. Komiyama, J. Honda, and H. Nakagawa. "Optimal Regret Analysis of Thompson Sampling in Stochastic Multi-armed Bandit Problem with Multiple Plays". In: *International Conference on Machine Learning*. 2015, pp. 1152–1161.

[142] E. Koutsoupias and C. Papadimitriou. "Worst-case equilibria". In: *Annual Symposium on Theoretical Aspects of Computer Science*. Springer. 1999, pp. 404–413.

[143] S. Krishnasamy, R. Sen, R. Johari, and S. Shakkottai. "Regret of queueing bandits". In: *CoRR, abs/1604.06377* (2016).

[144] R. Kumar, A. Yadav, S. J. Darak, and M. K. Hanawal. "Trekking based distributed algorithm for opportunistic spectrum access in infrastructure-less network". In: *2018 16th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*. 2018.

[145] R. Kumar, S. J. Darak, A. K. Sharma, and R. Tripathi. "Two-stage decision making policy for opportunistic spectrum access and validation on USRP testbed". In: *Wireless Networks* 24.5 (2018), pp. 1509–1523.

[146] B. Kveton, Z. Wen, A. Ashkan, and C. Szepesvari. "Tight regret bounds for stochastic combinatorial semi-bandits". In: *Artificial Intelligence and Statistics*. 2015, pp. 535–543.

[147] T. L. Lai and H. Robbins. "Asymptotically efficient adaptive allocation rules". In: *Advances in applied mathematics* 6.1 (1985), pp. 4–22.

[148] P. Landgren, V. Srivastava, and N. E. Leonard. "On distributed cooperative decision-making in multiarmed bandits". In: *2016 European Control Conference (ECC)*. IEEE. 2016, pp. 243–248.

[149] J. Lasserre. "Global Optimization with Polynomials and the Problem of Moments". In: *SIAM Journal on Optimization* 11.3 (2001), pp. 796–817.

[150] T. Lattimore and C. Szepesvári. "Bandit algorithms". In: *preprint* (2018), p. 28.

[151] L. Le Cam and G. L. Yang. "Asymptotics in Statistics". Second. Springer-Verlag, New York, 2000, pp. xiv+285.

[152] R. P. Leme, M. Pal, and S. Vassilvitskii. "A field guide to personalized reserve prices". In: *Proceedings of the 25th international conference on world wide web*. 2016, pp. 1093–1102.

[153] Y. Li and Y. Yuan. "Convergence analysis of two-layer neural networks with relu activation". In: *Advances in neural information processing systems*. 2017, pp. 597–607.

[154] T. Lin, C. Jin, and M. Jordan. "Near-optimal algorithms for minimax optimization". In: *arXiv preprint arXiv:2002.02417* (2020).

[155] T. Lin, C. Jin, and M. Jordan. "On gradient descent ascent for nonconvex-concave minimax problems". In: *arXiv preprint arXiv:1906.00331* (2019).

[156] K. Liu and Q. Zhao. "Distributed Learning in Multi-Armed Bandit With Multiple Players". In: *IEEE Transactions on Signal Processing* 58.11 (2010), pp. 5667–5681.

[157] K. Liu and Q. Zhao. "A restless bandit formulation of opportunistic access: Indexablity and index policy". In: *2008 5th IEEE Annual Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks Workshops*. IEEE. 2008, pp. 1–5.

[158] L. T. Liu, F. Ruan, H. Mania, and M. I. Jordan. "Bandit Learning in Decentralized Matching Markets". In: *arXiv preprint arXiv:2012.07348* (2020).

[159] L. T. Liu, H. Mania, and M. Jordan. "Competing bandits in matching markets". In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2020, pp. 1618–1628.

[160] I. Lobel and E. Sadler. "Information diffusion in networks through social learning". In: *Theor. Econ.* 10.3 (2015), pp. 807–851.

[161] S. Lu, I. Tsaknakis, M. Hong, and Y. Chen. "Hybrid block successive approximation for one-sided non-convex min-max problems: algorithms and applications". In: *IEEE Transactions on Signal Processing* (2020).

[162] G. Lugosi and A. Mehrabian. “Multiplayer bandits without observing collision information”. In: *arXiv preprint arXiv:1808.08416* (2018).

[163] B. Maćkowiak and M. Wiederholt. “Business cycle dynamics under rational inattention”. In: *The Review of Economic Studies* 82.4 (2015), pp. 1502–1532.

[164] A. Magesh and V. Veeravalli. “Multi-player Multi-Armed Bandits with non-zero rewards on collisions for uncoordinated spectrum access”. In: *arXiv preprint arXiv:1910.09089* (2019).

[165] A. Magesh and V. V. Veeravalli. “Multi-User MABs with User Dependent Rewards for Uncoordinated Spectrum Access”. In: *2019 53rd Asilomar Conference on Signals, Systems, and Computers*. IEEE. 2019, pp. 969–972.

[166] C. Maglaras, M. Scarsini, and S. Vaccari. “Social learning from online reviews with product choice”. Tech. rep. Columbia Business School Research Paper No. 18-17, 2020.

[167] J. R. Marden, H. P. Young, and L. Y. Pao. “Achieving pareto optimality through distributed learning”. In: *SIAM Journal on Control and Optimization* 52.5 (2014), pp. 2753–2770.

[168] J. Marinho and E. Monteiro. “Cognitive radio: survey on communication protocols, spectrum decision issues, and future research directions”. In: *Wireless networks* 18.2 (2012), pp. 147–164.

[169] A. W. Marshall, I. Olkin, and B. C. Arnold. “Inequalities: theory of majorization and its applications”. Vol. 143. Springer, 1979.

[170] D. Martínez-Rubio, V. Kanade, and P. Rebeschini. “Decentralized Cooperative Stochastic Bandits”. In: *arXiv preprint arXiv:1810.04468* (2018).

[171] F. Matějka and A. McKay. “Rational inattention to discrete choices: A new foundation for the multinomial logit model”. In: *American Economic Review* 105.1 (2015), pp. 272–98.

[172] I. Mironov. “Rényi Differential Privacy”. In: *Proceedings of 30th IEEE Computer Security Foundations Symposium (CSF)*. 2017, pp. 263–275.

[173] J. Mitola and G. Q. Maguire. “Cognitive radio: making software radios more personal”. In: *IEEE Personal Communications* 6.4 (1999), pp. 13–18.

[174] J. Munkres. “Algorithms for the assignment and transportation problems”. In: *J. Soc. Indust. Appl. Math.* 5 (1957), pp. 32–38.

[175] R. Murphy. “Local Consumer Review Survey 2019”. In: *BrightLocal* (2019). URL: `brightlocal.com/research/local-consumer-review-survey`.

[176] N. Nayyar, D. Kalathil, and R. Jain. "On regret-optimal learning in decentralized multi-player multiarmed bandits". In: *IEEE Transactions on Control of Network Systems* 5.1 (2016), pp. 597–606.

[177] T. Nedelec, N. E. Karoui, and V. Perchet. "Learning to bid in revenue-maximizing auctions". In: *International Conference on Machine Learning*. 2019, pp. 4781–4789.

[178] A. Nedić and A. Ozdaglar. "Subgradient methods for saddle-point problems". In: *Journal of optimization theory and applications* 142.1 (2009), pp. 205–228.

[179] M. J. Neely, E. Modiano, and C.-P. Li. "Fairness and optimal stochastic control for heterogeneous networks". In: *IEEE/ACM Transactions On Networking* 16.2 (2008), pp. 396–409.

[180] M. Nouiehed, M. Sanjabi, T. Huang, J. Lee, and M. Razaviyayn. "Solving a class of non-convex min-max games using iterative first order methods". In: *Advances in Neural Information Processing Systems*. 2019, pp. 14934–14942.

[181] D. Ostrovskii, A. Lowy, and M. Razaviyayn. "Efficient search of first-order nash equilibria in nonconvex-concave smooth min-max problems". In: *arXiv preprint arXiv:2002.07919* (2020).

[182] Y. Papanastasiou and N. Savva. "Dynamic pricing in the presence of social learning and strategic consumers". In: *Management Sci.* 63.4 (2017), pp. 919–939.

[183] S. Park, W. Shin, and J. Xie. "The fateful first consumer review". In: *Marketing Sci.* 40.3 (2021), pp. 481–507.

[184] R. Pemantle and J. S. Rosenthal. "Moment conditions for a sequence with negative drift to be uniformly bounded in Lr". In: *Stochastic Processes and their Applications* 82.1 (1999), pp. 143–155.

[185] V. Perchet and P. Rigollet. "The multi-armed bandit problem with covariates". In: *The Annals of Statistics* 41.2 (2013), pp. 693–721.

[186] V. Perchet, P. Rigollet, S. Chassang, and E. Snowberg. "Batched Bandit Problems". In: *Proceedings of The 28th Conference on Learning Theory*. 2015, pp. 1456–1456.

[187] V. Perchet. "Approachability, regret and calibration: Implications and equivalences". In: *Journal of Dynamics & Games* 1.2 (2014), p. 181.

[188] P. Perrault, E. Boursier, M. Valko, and V. Perchet. "Statistical efficiency of thompson sampling for combinatorial semi-bandits". In: *Advances in Neural Information Processing Systems* 33 (2020).

[189] G. Peyré and M. Cuturi. "Computational optimal transport". In: *Foundations and Trends® in Machine Learning* 11.5-6 (2019), pp. 355–607.

[190] A. Proutiere and P. Wang. "An Optimal Algorithm in Multiplayer Multi-Armed Bandits". 2019.

[191] H. Rafique, M. Liu, Q. Lin, and T. Yang. "Non-convex min-max optimization: Provable algorithms and applications in machine learning". In: *arXiv preprint arXiv:1810.02060* (2018).

[192] S. Reddi, S. Kale, and S. Kumar. "On the convergence of adam and beyond". In: *arXiv preprint arXiv:1904.09237* (2019).

[193] J. Reed and B. Pierce. "Distance makes the types grow stronger: a calculus for differential privacy". In: *ACM Sigplan Notices*. Vol. 45. 9. 2010, pp. 157–168.

[194] A. Rényi. "On measures of entropy and information". In: *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*. 1961.

[195] H. Robbins. "Some aspects of the sequential design of experiments". In: *Bulletin of the American Mathematical Society* 58.5 (1952), pp. 527–535.

[196] C. Robert, C. Moy, and H. Zhang. "Opportunistic Spectrum Access Learning Proof of Concept". In: *SDR-WinnComm'14* (2014), p. 8.

[197] D. Rosenberg and N. Vieille. "On the efficiency of social learning". In: *Econometrica* 87.6 (2019), pp. 2141–2168.

[198] J. Rosenski, O. Shamir, and L. Szlak. "Multi-player bandits–a musical chairs approach". In: *International Conference on Machine Learning*. 2016, pp. 155–163.

[199] T. Roughgarden. "Algorithmic game theory". In: *Communications of the ACM* 53.7 (2010), pp. 78–86.

[200] T. Salimans, D. Metaxas, H. Zhang, and A. Radford. "Improving GANs using optimal transport". In: *6th International Conference on Learning Representations, ICLR 2018*. 2018.

[201] A. Sankararaman, S. Basu, and K. A. Sankararaman. "Dominate or Delete: Decentralized Competing Bandits with Uniform Valuation". In: *arXiv preprint arXiv:2006.15166* (2020).

[202] F. Santambrogio. "Optimal transport for applied mathematicians". In: *Birkäuser, NY* 55 (2015), pp. 58–63.

[203] A. Sanyal, M. Kusner, A. Gascon, and V. Kanade. "TAPAS: Tricks to Accelerate (encrypted) Prediction As a Service". In: *International Conference on Machine Learning*. 2018, pp. 4497–4506.

[204] S. Sawant, M. K. Hanawal, S. Darak, and R. Kumar. "Distributed learning algorithms for coordination in a cognitive network in presence of jammers". In: *2018 16th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*. IEEE. 2018, pp. 1–8.

[205] S. Sawant, R. Kumar, M. K. Hanawal, and S. J. Darak. "Learning to Coordinate in a Decentralized Cognitive Radio Network in Presence of Jammers". In: *IEEE Transactions on Mobile Computing* (2019).

[206] A. S. Schulz and N. E. S. Moses. "On the performance of user equilibria in traffic networks." In: *SODA*. Vol. 3. 2003, pp. 86–87.

[207] F. Sentenac, E. Boursier, and V. Perchet. "Decentralized Learning in Online Queuing Systems". In: *arXiv preprint arXiv:2106.04228* (2021).

[208] Y. Sergeyev, R. Strongin, and D. Lera. "Introduction to global optimization exploiting space-filling curves". Springer Science & Business Media, 2013.

[209] D. Shah and J. Shin. "Randomized scheduling algorithm for queueing networks". In: *The Annals of Applied Probability* 22.1 (2012), pp. 128–171.

[210] C. Shi and C. Shen. "On No-Sensing Adversarial Multi-player Multi-armed Bandits with Collision Communications". In: *arXiv preprint arXiv:2011.01090* (2020).

[211] C. Shi, W. Xiong, C. Shen, and J. Yang. "Decentralized Multi-player Multi-armed Bandits with No Collision Information". In: *arXiv preprint arXiv:2003.00162* (2020).

[212] J. F. Shortle, J. M. Thompson, D. Gross, and C. M. Harris. "Fundamentals of queueing theory". Vol. 399. John Wiley & Sons, 2018.

[213] C. Sims. "Implications of rational inattention". In: *Journal of monetary Economics* 50.3 (2003), pp. 665–690.

[214] R. Sinkhorn. "Diagonal equivalence to matrices with prescribed row and column sums". In: *The American Mathematical Monthly* 74.4 (1967), pp. 402–405.

[215] A. Slivkins. "Introduction to multi-armed bandits". In: *arXiv preprint arXiv:1904.07272* (2019).

[216] G. Smith. "On the foundations of quantitative information flow". In: *International Conference on Foundations of Software Science and Computational Structures*. 2009, pp. 288–302.

[217] L. Smith and P. Sørensen. “Pathological outcomes of observational learning”. In: *Econometrica* 68.2 (2000), pp. 371–398.

[218] M. Soltanolkotabi, A. Javanmard, and J. Lee. “Theoretical insights into the optimization landscape of over-parameterized shallow neural networks”. In: *IEEE Transactions on Information Theory* 65.2 (2018), pp. 742–769.

[219] D. Soudry and E. Hoffer. “Exponentially vanishing sub-optimal local minima in multilayer neural networks”. In: *arXiv preprint arXiv:1702.05777* (2017).

[220] L.-G. Svensson. “Strategy-proof allocation of indivisible goods”. In: *Social Choice and Welfare* 16.4 (1999), pp. 557–567.

[221] B. Szorenyi, R. Busa-Fekete, I. Hegedus, R. Ormándi, M. Jelasity, and B. Kégl. “Gossip-based distributed stochastic bandit algorithms”. In: *International Conference on Machine Learning*. 2013, pp. 19–27.

[222] P. Tao and L. An. “Convex analysis approach to DC programming: Theory, algorithms and applications”. In: *Acta mathematica vietnamica* 22.1 (1997), pp. 289–355.

[223] L. Tassiulas and A. Ephremides. “Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks”. In: *29th IEEE Conference on Decision and Control*. IEEE. 1990, pp. 2130–2132.

[224] C. Tekin and M. Liu. “Online learning in decentralized multi-user spectrum access with synchronized explorations”. In: *MILCOM 2012-2012 IEEE Military Communications Conference*. IEEE. 2012, pp. 1–6.

[225] K. Thekumparampil, P. Jain, P. Netrapalli, and S. Oh. “Efficient algorithms for smooth minimax optimization”. In: *Advances in Neural Information Processing Systems*. 2019, pp. 12680–12691.

[226] W. 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.

[227] H. Tibrewal, S. Patchala, M. K. Hanawal, and S. J. Darak. “Distributed Learning and Optimal Assignment in Multiplayer Heterogeneous Networks”. In: *IEEE INFOCOM 2019*. 2019, pp. 1693–1701.

[228] G. Tóth, Z. Hornák, and F. Vajda. “Measuring anonymity revisited”. In: *Proceedings of the Ninth Nordic Workshop on Secure IT Systems*. 2004, pp. 85–90.

[229] A. B. Tsybakov. “Introduction to Nonparametric Estimation”. Springer, New York, 2009, pp. xii+214.

[230] L. Venturi, A. Bandeira, and J. Bruna. "Spurious valleys in two-layer neural network optimization landscapes". In: *arXiv preprint arXiv:1802.06384* (2018).

[231] A. Verma, M. K. Hanawal, and R. Vaze. "Distributed algorithms for efficient learning and coordination in ad hoc networks". In: *2019 International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOPT)*. IEEE. 2019, pp. 1–8.

[232] D. Vial, S. Shakkottai, and R Srikant. "Robust Multi-Agent Multi-Armed Bandits". In: *arXiv preprint arXiv:2007.03812* (2020).

[233] C. Villani. "Optimal transport: old and new". Vol. 338. Springer Science & Business Media, 2008.

[234] M. J. Wainwright. "High-dimensional statistics: A non-asymptotic viewpoint". Vol. 48. Cambridge University Press, 2019.

[235] P.-A. Wang, A. Proutiere, K. Ariu, Y. Jedra, and A. Russo. "Optimal algorithms for multiplayer multi-armed bandits". In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2020, pp. 4120–4129.

[236] Q. Wang, K. Ren, P. Ning, and S. Hu. "Jamming-resistant multiradio multichannel opportunistic spectrum access in cognitive radio networks". In: *IEEE Transactions on Vehicular Technology* 65.10 (2015), pp. 8331–8344.

[237] S. Wang and W. Chen. "Thompson Sampling for Combinatorial Semi-Bandits". In: *International Conference on Machine Learning*. 2018, pp. 5114–5122.

[238] L. Wei and V. Srivastava. "On Distributed Multi-Player Multiarmed Bandit Problems in Abruptly Changing Environment". In: *2018 IEEE Conference on Decision and Control (CDC)*. IEEE. 2018, pp. 5783–5788.

[239] M. Woodroofe. "A one-armed bandit problem with a concomitant variable". In: *Journal of the American Statistical Association* 74.368 (1979), pp. 799–806.

[240] S. Wright. "Coordinate descent algorithms". In: *Mathematical Programming* 151.1 (2015), pp. 3–34.

[241] R. R. Yager. "On ordered weighted averaging aggregation operators in multicriteria decisionmaking". In: *IEEE Transactions on systems, Man, and Cybernetics* 18.1 (1988), pp. 183–190.

[242] M.-J. Youssef, V. Veeravalli, J. Farah, and C. A. Nour. “Stochastic Multi-Player Multi-Armed Bandits with Multiple Plays for Uncoordinated Spectrum Access”. In: *PIMRC 2020: IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*. 2020.

[243] Q. Zhao and B. M. Sadler. “A survey of dynamic spectrum access”. In: *IEEE signal processing magazine* 24.3 (2007), pp. 79–89.

[244] L. Zhou. “On a conjecture by Gale about one-sided matching problems”. In: *Journal of Economic Theory* 52.1 (1990), pp. 123–135.

[245] F. Zou, L. Shen, Z. Jie, W. Zhang, and W. Liu. “A sufficient condition for convergences of adam and rmsprop”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2019, pp. 11127–11135.

université PARIS-SACLAY

ÉCOLE DOCTORALE
de mathématiques
Hadamard (EDMH)

Titre: Apprentissage séquentiel dans un environnement stratégique

Mots clés: Apprentissage séquentiel, Bandits à plusieurs joueurs, Théorie des jeux, Jeux répétés

Résumé: En apprentissage séquentiel (ou jeux répétés), les données sont acquises et traitées à la volée et un algorithme (ou stratégie) apprend à se comporter aussi bien que s'il avait pu observer l'état de nature, par exemple les distributions des gains. Dans de nombreuses situations réelles, de tels agents intelligents ne sont pas seuls et interagissent ou interfèrent avec d'autres. Ainsi, leurs décisions ont un impact direct sur les autres agents et indirectement sur leurs propres gains à venir. Nous étudions de quelle manière les algorithmes d'apprentissage séquentiel peuvent se comporter dans des environnements stratégiques quand ils sont confrontés à d'autres agents.

Cette thèse considère différents problèmes où certaines interactions entre des agents intelligents apparaissent, pour lesquels nous proposes des algorithmes efficaces en termes de calcul avec de bonnes garanties de performance (faible regret).

Lorsque les agents sont coopératifs, la difficulté du problème vient de son aspect décentralisé, étant donné que les agents prennent leurs décisions en se basant seulement sur leurs propres observations. Dans ce cas, les algorithmes proposés non seulement coordonnent les agents afin d'éviter des interférences entre eux, mais ils utilisent également ces interférences pour transférer de l'information entre les agents. Cela permet d'obtenir des performances comparables aux meilleurs algorithmes centralisés. Avec des agents en concurrence, nous proposons des algorithmes avec des garanties satisfaisantes, à la fois en terme de performance et de stratégie (ε-équilibre de Nash par exemple).

Title: Sequential Learning in a strategical environment

Keywords: Online Learning, Multiplayer bandits, Game Theory, Repeated Games

Abstract: In sequential learning (or repeated games), data is acquired and treated on the fly and an algorithm (or strategy) learns to behave as well as if it got in hindsight the state of nature, e.g., distributions of rewards. In many real life scenarios, learning agents are not alone and interact, or interfere, with many others. As a consequence, their decisions have an impact on the other and, by extension, on the generating process of rewards. We aim at studying how sequential learning algorithms behave in strategic environments, when facing and interfering with each others. This thesis thus considers different problems, where some interactions between learning agents arise and provides computationally efficient algorithms with good performance (small regret) guarantees.

When agents are cooperative, the difficulty of the problem comes from its decentralized aspect, as the different agents take decisions solely based on their observations. In this case, we propose algorithms that not only coordinate the agents to avoid negative interference with each other, but also leverage the interferences to transfer information between the agents, thus reaching performances similar to centralized algorithms. With competing agents, we propose algorithms with both satisfying performance and strategic (e.g., ε-Nash equilibria) guarantees.

Université Paris-Saclay
Espace Technologique / Immeuble Discovery
Route de l'Orme aux Merisiers RD 128 / 91190 Saint-Aubin, France