Journal of Machine Learning Research 25 (2024) 1-45 Submitted 6/22; Revised 10/23; Published 1/24

A Survey on Multi-player Bandits

Etienne Boursier ETIENNE.BOURSIER1@GMAIL.COM
INRIA, Université Paris Saclay, LMO, Orsay, France

Vianney Perchet VIANNEY.PERCHET@NORMALESUP.ORG
CREST, ENSAE Paris, Palaiseau, France
CRITEO AI Lab, Paris, France

Editor: Alexandre Proutiere

Abstract

Due mostly to its application to cognitive radio networks, multiplayer bandits gained a lot of interest in the last decade. A considerable progress has been made on its theoretical aspect. However, the current algorithms are far from applicable and many obstacles remain between these theoretical results and a possible implementation of multiplayer bandits algorithms in real communication networks. This survey contextualizes and organizes the rich multiplayer bandits literature. In light of the existing works, some clear directions for future research appear. We believe that a further study of these different directions might lead to theoretical algorithms adapted to real-world situations.

Keywords: Multiplayer bandits, Multi-armed bandits, Cognitive radio, Decentralized learning, Opportunistic Spectrum Access.

1. Introduction

The Multi-Armed Bandits (MAB) problem (Thompson, 1933) has been extensively studied in the last decades, probably because of its applications to online recommendation systems, and is one of the most used models of online learning. In its classical version, a single player sequentially chooses an action among a finite set $[K] := \{1, \dots, K\}$. After pulling the arm $k \in [K]$ at round t, the player receives the reward $X_k(t)$, which is drawn from some unknown distribution, of mean denoted by μ_k in the stochastic setting. Since only the reward of the pulled arm is observed, the player must balance between **exploration** (acquiring information on the different arms) and **exploitation** (pulling the seemingly best arm). This model has known many extensions such as contextual, combinatorial or Lipschitz bandits for example (Woodroofe, 1979; Cesa-Bianchi and Lugosi, 2012; Agrawal, 1995; Perchet and Rigollet, 2013).

The multiplayer variant of this problem has also known a recent interest, motivated by cognitive radio networks. It considers multiple decentralized players acting on a single Multi-Armed Bandits instance. If several of them pull the same arm at the same time, a *collision* occurs and causes a loss (of the message in the previous example). This leads to a decrease in the received reward and makes the problem much more intricate.

Section 2 first presents in more detail the motivations leading to the design of the multiplayer bandits model. The most classical version of multiplayer bandits is then described in Section 3, along with a first base study including the centralized case and a lower bound of the incurred regret. Section 4 presents the different results known for this model. In particular, collisions can be abusively used to reach regret bounds similar to the centralized case. Section 5 presents several practical considerations that can be added to the model, which might lead to more natural algorithms. Finally, Section 6 mentions the multi-agent bandits, competing bandits and dynamic queuing systems problems, which all bear similarities with multiplayer bandits, either in the model or in the used algorithms. Appendix A introduces single player stochastic MAB and the most common algorithms. Tables 3 and 4 in Appendix B summarize the main results presented in this survey.

2. Motivation for cognitive radio networks [1]

The problem of multiplayer bandits is mostly motivated by its applications to cognitive radio, whose paradigm has been first proposed by Mitola and Maguire (1999) and can be defined as a radio capable of learning its environment and choosing dynamically the best configuration for transmission. The definition of best configuration depends on the precise objective and is often considered as the maximal bandwidth usage rate (Haykin, 2005). Cognitive radios attract a lot of attention due to their numerous potential applications, such as Internet of Things (Shah et al., 2013), smart vehicles (Di Felice et al., 2012), public safety communication (Ferrus et al., 2012) and civil aviation (Zheng et al., 2023). Yet, cognitive radios are still at a development stage. Many technical issues remain to be solved before a possible use of cognitive radios in real world applications. In particular, improving the following features are crucial for a successful implementation of cognitive radios: spectrum awareness and exploitation (Sharma et al., 2015) and security (Attar et al., 2012). We refer to (Marinho and Monteiro, 2012; Garhwal and Bhattacharya, 2011) for surveys on different research directions on cognitive radios.

The most challenging approach to cognitive radio is **Opportunistic Spectrum Access** (OSA). Consider a spectrum partitioned into licensed bands. For each band, some designated Primary User (PUs), who pays a license, is given preferential access. In practice, many of these bands remain largely unused by PUs. OSA then aims at maximizing the spectrum usage by giving Secondary Users (SUs) the possibility to access channels left 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 transmit on this channel. Such devices yet have limited capabilities; especially, they proceed in a decentralized network and cannot sense different channels simultaneously. This last restriction justifies the bandit feedback assumed in formal models.

However, the previous sensing procedure is not perfect in practice: SUs might wrongly detect the absence of a PU on a channel, e.g., because of the hidden node problem (Sahai et al., 2004). Moreover, for other applications such as **Internet of Things** (IoT) networks, the devices have even lower power capabilities and there is no more licensed band. As a consequence, all devices behave as SUs (no PU) and do not require to, or cannot, sense the presence of another user before transmitting. These devices can still perform some form of learning: if the base station sends back downlink messages, the devices can determine afterward whether their transmission was successful, i.e., whether they received an

[1] We are grateful to Christophe Moy for his precious feedback on the application of multiplayer bandits to cognitive radio networks.

acknowledgement. As a consequence, similar learning models can be used for many different communication networks, and only differ from minor assumptions such as the received feedback (see Section 3.1).

Depending on the application, the user can also choose dynamically other transmission parameters, to maximise its throughput. For example in the case of Wi-Fi networks, the user can choose the transmission mode and rate (Combes and Proutière, 2014).

Using a Multi-Armed bandits model to improve spectrum exploitation for cognitive radios was first suggested by Jouini et al. (2009, 2010); Liu and Zhao (2008). In these first attempts in formalizing the problem, a single SU (player) repeatedly chooses among a choice of K arms (i.e., channels or any tunable transmission parameter) for transmission. The success of transmission is 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. In particular the Gilbert-Elliot model, with two states, is often considered as a good approximation of PUs activity (Tekin and Liu, 2011). A successful transmission corresponds then 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 messages, or in bandit terms, to minimize its regret. In real networks, devices observe whether their message is successfully transmitted using acknowledgement messages, but this aspect is disregarded here for the sake of simplicity.

Shortly after, Liu and Zhao (2010) extended this model to multiple users, taking into account the interaction between SUs in communication 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* in the literature.

Different Proof-of-Concepts, simulating networks in laboratory conditions, later justified the use of Reinforcement Learning, and more particularly Multi-Armed bandits model, for OSA (Robert et al., 2014; Toldov et al., 2016; Kumar et al., 2018), Wi-Fi networks[2] (Combes et al., 2018) and IoT networks (Moy and Besson, 2019).

While OSA currently remains a futuristic application, a real IoT deployment with bandits algorithms has already proven useful in a LoRaWAN network (Moy and Besson, 2019). In this work, Moy and Besson (2019) implemented a UCB algorithm on the device side in a real LoRa network, in the city of Rennes, France. The device dynamically chooses, with a duty cycle of 1% between three channels in the European ISM band at 868.1 MHz, 868.3 MHz and 868.5 MHz. These channels correspond to the authorized frequencies in France. To enable learning from the device side, the application server sends back an acknowledgement to the device in case of successful transmission. In only 129 transmissions, the learning device already outperforms by a factor 2 (in transmission rate) devices selecting their channel uniformly at random.

All these advanced experimentations demonstrate the potential utility of bandit policies for enhanced cognitive radio networks. We refer to (Jouini, 2012; Besson, 2019) for more details on the link between OSA, IoT and Multi-Armed bandits.

Besides the application to cognitive radios, which is at the origin of the multiplayer bandits problem, the algorithmic techniques and proof ideas developed for multiplayer bandits can also be adapted to other multi-agent sequential learning problems. Such other prob-

[2] Instead of the channel, the devices here adaptively choose the transmission rate and mode.

lems are mentioned in Section 6 and include applications to matching markets and packet routing in servers.

3. Baseline Results

This section describes the classical model of multiplayer bandits, with several variations of observation and arm means setting, and gives first results (derived from the centralized case), as well as notations used along this survey. Harder, more realistic variations are discussed in Section 5.

For completeness, Appendix A provides a short introduction to the classical (single player) stochastic MAB problem as well as the common algorithms and results known for this problem. Algorithms described in the remaining of the survey are based on these common algorithms and include UCB (Upper Confidence Bound), ε-Greedy, ETC (Explore-Then-Commit).

In this survey, we write $f(T) = \mathcal{O}\left(g(T)\right)$ if there exists a universal constant $c \in \mathbb{R}$ such that for any $T \geq 1$, $f(T) \leq cg(T)$. The constant c must not depend on any problem parameters. Conversely, we note $f(T) = \Omega(g(T))$ if $g(T) = \mathcal{O}\left(f(T)\right)$ and, furthermore, $f(T) = o\left(g(T)\right)$ if $\lim_{T\to\infty} \frac{f(T)}{g(T)} = 0$.

3.1 Model

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 $t \in [T] \coloneqq \{1, \ldots, T\}$, all players pull simultaneously an arm in $[K]$. We denote by $\pi^m(t)$ the arm pulled by player m at time t, who receives the individual reward

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

where $\eta_k(t) = \mathbb{1}\left(\#\left\{m \in [M] \mid \pi^m(t) = k\right\} > 1\right)$ is the collision indicator, and $\#A$ denotes the cardinality of a set A. The players are assumed to know the horizon T, even though this is not crucial (Degenne and Perchet, 2016), and use a common numbering of the arms.

A *matching* $\pi \in \mathcal{M}$ is an assignment of players to arms, i.e., mathematically, 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 (pseudo-)regret, which is the difference between the maximal expected reward and the algorithm cumulative reward:

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

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

$$\Delta \coloneqq U^* - \max\left\{U(\pi) \mid \pi \in \mathcal{M},\ U(\pi) < U^*\right\}.$$

In contrast to the classical bandits problem where only the received reward can be observed at each time step, multiplayer settings might differ in the received feedback at each step, which leads to at least four different settings[3], described in Table 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 1: Different observation settings considered. *Feedback* represents the observation of player m after 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, assuming there is no sensing failure (e.g., because of hidden nodes). On the other hand, no-sensing can model IoT networks as explained in Section 2. The no-sensing setting is obliviously the hardest one, since a 0 reward can either correspond to a low channel quality or a collision with another player.

The above description corresponds to the *heterogeneous* setting, where the arm means differ among the players. In practice, the quality of a channel might vary among players, notably because of propagation problems such as frequency selective fading and path loss. In the following, the easier *homogeneous* setting is also considered, where the arm means are common to all players: $\mu_k^m = \mu_k$ for all $(m, k) \in [M] \times [K]$. If $\mu_{(k)}$ denotes the k-th largest mean, i.e., $\mu_{(1)} \geq \mu_{(2)} \ldots \geq \mu_{(K)}$, 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 players in the literature. In the following, we consider by default common statistics between players and precise when otherwise. Note that this has no influence in both collision and no-sensing settings.

3.2 Centralized Case

To set baseline results, we consider first, in this section, the easier centralized model, where all players in the game described in Section 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 thus only is determining 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

[3] 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.

single player has to pull M arms among a set of K arms at each round. Anantharam et al. (1987) introduced this problem long before multiplayer bandits and provided an asymptotic lower bound for this problem, given by Theorem 2 in Section 3.3. They also provided an optimal algorithm, asymptotically reaching this exact regret bound.

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, $\mathcal{A}$ is 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 $\mathcal{A}$ is the set of matchings between players and arms.

Chen et al. (2013) proposed the CUCB algorithm, which yields a $\mathcal{O}\left(\frac{M^2K}{\Delta}\log(T)\right)$ regret in the heterogeneous setting (Kveton et al., 2015). While CUCB performs well for any correlation between the arms, Combes et al. (2015) leverage 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 computing upper confidence bounds for any (meta-)action. The number of such actions indeed scales combinatorially with the number of arms and players. 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). Cuvelier et al. (2021) proposed a computationally efficient, asymptotically optimal algorithm for many combinatorial bandits problems with independent arms, including the centralized heterogeneous one. The optimal regret bound is yet not explicit, but characterized as the minimum of some optimization problem. Degenne and Perchet (2016) and Perrault et al. (2020) respectively extended ESCB and combinatorial TS for the intermediate case of any fixed correlation between the arms.

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

Definition 1 *We call an algorithm uniformly good if for every parameter configuration $\boldsymbol{\mu}, K, M$, it yields for every $\alpha > 0$ that $R(T) = o\left(T^\alpha\right)$.*

Theorem 2 (Anantharam et al. 1987) *For any uniformly good algorithm and all instances of homogeneous multiplayer bandits with $\mu_{(1)} > \ldots > \mu_{(K)}$,*

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

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

Combes et al. (2015) proved a lower bound for general combinatorial bandits, determined as the solution of an optimization problem. Luckily for the specific case of matchings, its value is simplified. Especially, for some problem instances of the heterogeneous setting, any uniformly good algorithm regret is $\Omega\left(\frac{KM}{\Delta}\log(T)\right)$. In the centralized case, studying the heterogeneous setting is already more intricate than the homogeneous one. This gap also holds 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 that an additional M factor, the number of players, would appear in the regret bounds of any decentralized algorithm (Liu and Zhao, 2010; Besson and Kaufmann, 2018). This actually only holds if the players do not use any information from the collisions incurred with other players (Besson and Kaufmann, 2019); but as soon as the players use this information, only the centralized bound holds.

4. Reaching Centralized Optimal Regret

We shall consider from now on the decentralized multiplayer bandits problem. This section shows how the collision information has been used in the literature, from a coordination 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 aspect.

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 number of collisions. In this direction, Besson and Kaufmann (2018) 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. Boursier and Perchet (2019) later proved that Selfish yields a regret scaling linearly with T for machines with infinite precision (i.e., that can distinguish an irrational number from an arbitrarily close rational number). It yet remains to be proved for machines with finite bit representation of real numbers.

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 (i.e., unique IDs), player j then just focuses on pulling the j-th best arm. Anandkumar et al. (2010) proposed the first algorithm in this line, 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 similar 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 in the absence of pre-agreement. A significant objective of multiplayer bandits is then to *orthogonalize* players, i.e., reach a state where all players pull different arms and no collision happens.

A first routine for orthogonalization, called Rand Orthogonalisation is given by Algorithm 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 up 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 pulling it for a long time.

Algorithm 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 $k$ uniformly in $\mathcal{S}$
    Pull arm $k$
end
```

Algorithm 2: Musical Chairs

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

Rosenski et al. (2016) later introduced a faster routine for orthogonalization, Musical Chairs described in Algorithm 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 afterward. 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 afterward 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.

Besson and Kaufmann (2018) adapted both orthogonalization 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 orthogonalization. Lugosi and Mehrabian (2021) 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 (2021) remove this assumption by instead using a Successive Accept and Reject (SAR[4]) algorithm (Bubeck et al., 2013) with epochs of increasing sizes. At the end of each epoch, players eliminate the arms appearing suboptimal and accept arms appearing optimal. The remaining arms still have to be explored in the next phases. To avoid collisions on the remaining arms, players proceed to Musical Chairs at the beginning of each new epoch.

Kumar et al. (2018) 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 currently assigned arm when possible.

With a few exceptions (Avner and Mannor, 2014; Kumar et al., 2018), 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, 2021). 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, 2021).

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 only proposed solutions reaching Pareto optimal matchings. A matching is Pareto optimal if no two players can exchange their assigned arms (or choose a free arm) while both receiving the same or a larger reward.

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 a small probability p, a player becomes a leader. The leader then suggests switching 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.

Pareto optimal matchings yet do not guarantee a large social welfare. Figure 1 below gives an example of Pareto optimal matching (in green) which yields an arbitrarily small utility when $\varepsilon \to 0$, while the welfare maximising matching (in grey, dashed) yields a utility 1. In Game Theory lingo, this corresponds to a game with a *Price of Anarchy* $\frac{1}{2\varepsilon}$ (Koutsoupias and Papadimitriou, 1999).

[4] SAR is an extension of the ETC algorithm to the multiple plays bandits problem.

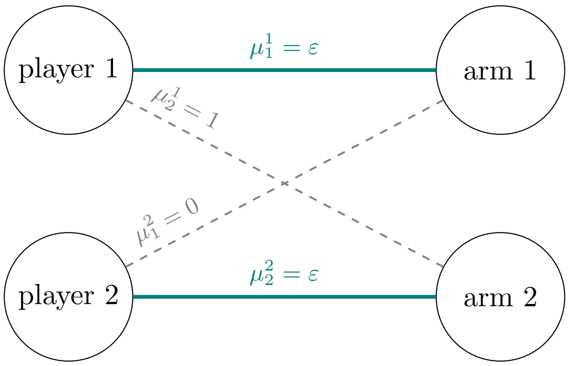

Figure 1: Example of *bad* Pareto optimal matching.

4.2 Enhancing Communication

Most of the literature described in Section 4.1 used collision information as a 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 information about their arms means.

For this purpose, Kalathil et al. (2014) assumed that players were able to send real numbers to each other 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. The algorithm proceeds in epochs of doubling size. Each epoch starts with a decision phase, where players bid according to UCB indexes of their arms. After this phase, players are attributed via ε-optimal matching for these indexes and pull this matching for the whole exploitation phase. This algorithm was later improved and adapted to ETC and TS strategies (Nayyar et al., 2016).

Although these works provide the 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, when smartly used. In the following of this section, we consider the collision sensing setting if not specified, so that a collision is systematically detected.

4.2.1 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 proceeds in epochs of increasing sizes. Each epoch is divided in an exploration phase where players estimate the arm means; a Game of Thrones (GoT) phase, which follows a Markov chain dynamic to attribute the best-estimated matching to players; and an exploitation phase where players pull the matching attributed by the GoT phase. This algorithm reaches a regret scaling with the horizon as $\log^{1+\delta}(T)$ for any choice of parameter $\delta > 0$ in the heterogeneous setting (the regret obviously also depends in the parameters δ, μ, K and M).

The main interest of the algorithm comes from the GoT phase, described in Algorithm 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 visit more often the best matching according to the estimated means $\hat{\mu}_k^j$ so far. Each player, while content, pulls her assigned arm in the optimal matching most often. This phase thus allows estimating the optimal matching between arms and players as proved by Bistritz and Leshem (2020).

Algorithm 3: Game of Thrones subroutine

```
input: time $T_0$, starting arm $\bar{a}_t$, player $j$, parameters $\varepsilon$ 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 $\begin{cases} 1 - \varepsilon^c \text{ if } k = \bar{a}_t \\ \varepsilon^c/(K-1) \text{ otherwise} \end{cases}$
    else pull $k$ with probability $1/K$
    if $k \neq \bar{a}_t$ or $\eta_k(t) = 0$ or $S_t = D$ then
        $\bar{a}_t, S_t \leftarrow \begin{cases} k, C \text{ with probability } \frac{\hat{\mu}_k^j \eta_k(t)}{u_{\max}} \varepsilon^{u_{\max} - \hat{\mu}_k^j \eta_k(t)} \\ k, D \text{ otherwise} \end{cases}$
end
```

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

This routine elegantly assigns the optimal matching to 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. Also, the algorithm requires a good tuning of the GoT parameters ε and c, which depend on the suboptimality gap Δ.

4.2.2 Collision Information as Bits.

In a different work, Boursier and Perchet (2019) suggested 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 based on Musical chairs (Boursier and Perchet, 2019), which also quickly estimates the number of players M. This initialization procedure, later improved by Wang et al. (2020), allows to reach the pre-agreement[5] mentioned in Section 4.1 after a time of order $\mathcal{O}\left(K^2 M\right)$.

Homogeneous setting. After this initialization, the SAR-based algorithm of Boursier and Perchet (2019) runs epochs of doubling size. Each epoch is divided in an exploration

[5] In the more difficult settings considered in Section 5, whether this pre-agreement can be reached at a negligible cost remains an open problem.

phase, where players pull all 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, without colliding.

Note that the communication regret of SIC-MMAB can directly be improved using a leader who gathers all the information and communicates the arms to pull to other players (Boursier et al., 2019).

As the players share their statistics altogether, Boursier and Perchet (2019) showed that the centralized lower bound was achievable with decentralization up to constant factors, contradicting first intuitions. Besides the initialization procedure of SIC-MMAB, Wang et al. (2020) also improved the regret due to exploration. In particular, they proposed to elect a leader, who is the only one to explore, while other players greedily exploit the best empirical arms. When the set of best empirical arms changes (which only happens $\mathcal{O}\left(\frac{K^{3/2}M^2}{\Delta^2}\right)$ times), the leader communicates to the other players the new set of best empirical arms. Their algorithm exactly matches the theoretical lower bound for the homogeneous setting.

Theorem 3 (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*

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

A similar leader-followers idea was also proposed by Verma et al. (2019). 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 without 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. (2022) 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 afterward, with high probability, they have elected an arm $\bar{k}$, which is the same for all players. Moreover, players have a common lower bound of $\mu_{\bar{k}}$, which is of the same order as $\mu_{(1)}$. Thanks to this, the players can then send information on this arm in $\frac{\log(T)}{\mu_{(1)}}$ 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, players then follow an algorithm similar to Shi et al. (2020), with a few modifications to ensure that players only communicate on the good arm $\bar{k}$. 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.

Pacchiano et al. (2023) also proposed a no-sensing algorithm without prior knowledge of μ. Although it proceeds to much fewer communication rounds than Huang et al. (2022), it leads to a larger regret bound and requires a pre-agreement on the players' ranks.

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 compute the optimal matching. If the suboptimality gap Δ is known, then a natural algorithm (Magesh and Veeravalli, 2019) 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.

Boursier et al. (2019) extended 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. However, exploring all matchings leads to a combinatorial regret and computational complexity of the algorithm. Players instead explore arm-player pairs and the SAR procedure thus accept/reject pairs that are sure to be involved/missing 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. Besides holding only with a unique optimal matching, this bound incurs an additional M factor with respect to centralized algorithms such as CUCB. In the case of several optimal matchings (in opposition to works mentioned above), they provide an algorithm with a regret scaling with horizon as $\log^{1+\delta}(T)$ for any $\delta > 0$, using longer exploration phases.

Shi et al. (2021) recently adapted the well-known CUCB algorithm to heterogeneous multiplayer bandits. Their BEACON algorithm reaches the centralized optimal performance for arbitrary correlations between the arms, even with several optimal matchings. Adapting CUCB to the multiplayer setting is possible thanks to two main ideas. First, CUCB is run in a batched version where the players pull the maximal matching (in terms of confidence bound) for n_p rounds during epoch p. The length of the epoch p here depends on the previous epochs and is thus communicated by the leader for each new epoch during the communication phase. The second main idea is to use *adaptive differential communication*. Namely, instead of communicating their empirical means $\hat{\mu}_k^j(p)$ to the leader at the end of epoch p, the players only communicate the difference $\hat{\mu}_k^j(p) - \hat{\mu}_k^j(p-1)$ between their current empirical means and the ones at the end of the previous epoch. While sending $\hat{\mu}_k^j(p)$ can be done in p rounds, the difference can instead be sent in expectation in $\mathcal{O}(1)$ rounds, drastically reducing the communication cost without loss of information. Using these two techniques, BEACON thus reaches the $\mathcal{O}\left(\frac{M^2 K \log(T)}{\Delta}\right)$ regret guarantee of CUCB , up to logarithmic terms in K.

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 5. Especially, 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 4.2 is then significant.

Some works instead focus on which level of regret is possible without collision information at all in the homogeneous setting. In that case, no communication can happen between the players. Naturally, these works 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 4.1, provides a first algorithm using no collision information. Boursier and Perchet (2020) later reached the asymptotic regret bound $M \sum_{k>M} \frac{\mu_{(k)}-\mu_{(M)}}{\mathrm{kl}\left(\mu_{(M)},\mu_{(k)}\right)}$, adapting the main phase of DPE1 in this setting. This 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. (2021) instead focus on reaching a minimax regret scaling with the horizon as $\sqrt{T \log(T)}$ 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 avoids any collision at all 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 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 regret of the algorithm by Bubeck et al. (2021) still suffers a dependency $MK^{11/2}$, which increases considerably with the number of channels K.

5. Towards Realistic Considerations

Section 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 3.1 is not well designed. In particular, it might be too simple with respect to the real problem of cognitive radio networks.

Section 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). It yet remains to study such a condition in the multiplayer setting.

This section proposes other reasons for this discrepancy and removes some simplifications from the multiplayer model, in the hope of obtaining algorithms adapted to real-world situations. First, the stochasticity of the reward X_k is questioned in Section 5.1 and replaced by either Markovian, abruptly changing, or adversarial rewards. The current collision model is then relaxed in Section 5.2. It instead considers a more difficult model where players only observe a decrease in reward when colliding. Section 5.3 considers non-collaborative players, who can be either adversarial or strategic. Section 5.4 finally questions the time synchronization that is assumed in most of the literature.

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.

5.1.1 Markovian Rewards.

A first more complex model is given by Markovian rewards. In this model introduced by Anantharam et al. (1987), 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. (1987) 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, following the same lines as their algorithm described in Section 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 proper tuning of its different parameters.

In more recent work, Gafni and Cohen (2022) 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. The Gilbert-Elliot model, which is often considered as a good model for the activity of primary users (Tekin and Liu, 2011), is a particular case of restless Markov chain. Using an ETC approach, they were able to reach a stable matching in a logarithmic time. Their result yet assumes knowledge of the suboptimality gap Δ and the uniqueness of the stable matching. Moreover, it only reaches a stable matching instead of an optimal one, similarly to the algorithms described in Section 4.1 for the heterogeneous setting. 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.

5.1.2 Abruptly Changing Rewards.

Although Markovian rewards are closer to reality, the resulting algorithms are very similar to the stochastic case. Indeed, the goal is still to pull the arm with the maximal mean reward, with respect to the stationary distribution. A stronger model assumes instead that the expected rewards abruptly change over time, e.g., the mean vector μ is piecewise constant in time, and each change is a *breakpoint*. 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 a regret scaling with the horizon as $T^{\frac{1+\nu}{2}}\log(T)$, when 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.

5.1.3 Adversarial Rewards.

The hardest model for rewards is the adversarial case, where the rewards are fixed by an adversary. In this case, the goal is to provide a minimax regret bound that holds under any problem instance. Bubeck et al. (2020) 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 regret lower bound is linear with T. The literature thus focuses on an *oblivious* adversary, who chooses beforehand the sequences of 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. Afterward, the player keeps pulling this arm until the end of the block. This algorithm yields a regret scaling with time as $T^{3/4}$. 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 time as $T^{2/3}$.

Shi and Shen (2021) 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 scaling with time as $T^{\frac{2+\alpha}{3}}$.

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 (2021) here use error detection code and randomized communication rounds to avoid such situations.

Bubeck et al. (2020) were the first to propose an algorithm with a regret scaling with the horizon as $\sqrt{T}$ 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 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. (2020) also proposed an algorithm with a regret scaling with time as $T^{1-\frac{1}{2M}}$ for the no-sensing setting. For two players, the first, low-switching player runs an algorithm on the arms $\{2, \dots, K\}$ and divides the time into fixed blocks whose length is 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 outside S_t and adds them to 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. (2020) generalize this algorithm using blocks of different sizes for different players.

5.2 Different Collision Models

As shown in Section 4.2, the collision information allows communication between the different players. The discrepancy between the theoretical and practical algorithms might be due to the collision model, which assumes that a collision systematically corresponds to a 0 reward. Such a model is motivated by the carrier-sense multiple access with collision avoidance protocol, which is for instance used in WiFi.

Non-zero collision reward.

In some modern wireless systems, central scheduling may allow to share the resource between users, e.g., using multiplexing (Sesia et al., 2011). Reduced-rate communications can then be successful in the presence of multiple users. In that case, collisions only lead to a decrease in reward and not necessarily a 0 reward. Also, 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 give 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.

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 problem by assuming that the decrease in mean reward when an additional player pulls an arm is large enough with respect to the noise. As a consequence, the observed rewards on a single arm can then be perfectly clustered and 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 (2019) 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 any k and m. From there, the players can reach the optimal allocation using a Game of Thrones routine similar to Algorithm 3. This work thus extended the results of Bistritz and Leshem (2020) 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.

When the arm mean is exactly inversely proportional to the number of players pulling it, i.e., $\mu_k^j(m) = \frac{\mu_k^j(1)}{m}$, Boyarski et al. (2021) exploit this assumption to design a simple algorithm with a regret scaling with time as $\log^{3+\delta}(T)$. 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 the preferences of the arm. This setting, introduced by Liu et al. (2020), is discussed in Section 6.2.

5.3 Non-collaborative Players

Assuming perfectly collaborative players might be another oversimplification in multiplayer bandits. 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*. These security threats thus 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 to 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 incur tremendous losses. Because of this, they restrict the jammers' strategy 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. Afterward, they exploit by sequentially pulling the top J-arms where J is chosen to maximize the earned reward.

Fairness. A first attempt at preventing 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, 2018). 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,[6] which can still be easily reached by making the players sequentially pull all the top-M arms in a round robin fashion when exploiting.

The notion of fairness becomes intricate 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 any player at each round. They propose an ETC algorithm that 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 incur a huge loss to other players at any cost, selfish players have a different objective: they maximize their own individual rewards. In the algorithms

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

mentioned so far, a selfish player could largely improve her earned regret at the expense of the other players. Boursier and Perchet (2020) propose algorithms robust to selfish players, being ε-Nash equilibria, where ε scales logarithmically with time. Without collision information, they adapt DPE1 without communication between the players. The main difficulty comes from designing a robust initialization protocol to assign ranks and estimate M. With collision information, they even show that robust communication-based algorithms are possible, thanks to a Grim Trigger strategy that 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, they 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).

Xu et al. (2023) later proposed an algorithm that is robust to selfish behaviors with non-zero reward collisions. More precisely, they consider a homogeneous setting where the reward of an arm is split (unevenly) between the players pulling it. In this model, reaching maximal social welfare might be impossible with selfish players (price of anarchy larger than 1). Instead, they propose an algorithm inspired by DPE1 , which aims at pulling the unique (up to players' permutation) Nash equilibrium of the game. It then yields an asymptotically optimal regret with respect to the utility of this Nash equilibrium.

Brânzei and Peres (2021) 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 Bayesian bandit game. Players observe both their obtained reward and the choice of the other player. In this setting, they compare the different Nash equilibria when players are either collaborative (maximizing the 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 commit on the best arms to keep them for a long time.

5.4 Beyond Time Synchronization

Most of the multiplayer algorithms depend on a high level of synchronization between the players. In particular, they assume two major simplifications: the players are assumed to be synchronous and also start and end the game at the same time.

Asynchronous players. First, the synchronous assumption considers that all players pull arms simultaneously (we say they are active) at each time step. Although assuming a common time discretization might be reasonable, e.g., through the use of *orthogonal frequency-division multiplexing* (Schmidl and Cox, 1997), the devices might not be active at each timestep in practice. For example in Europe, the 868 MHz bandwidth, which is used for IoT transmissions, limits the duty cycle at 1% to avoid saturation of the network. This means that each device can emit at most 1% of the time. As a consequence, each device is only active a small fraction of the time, which is specific to each device.

Bonnefoi et al. (2017) first considered the *asynchronous* setting, where each player is active with a probability p at each time step. They first focus on offline policies, as even determining the optimal policy is not straightforward here, and compare empirically single-player online strategies (UCB and TS) to these baselines.

While considered homogeneous in this work, the activation rates might be heterogeneous in practice, for example, due to the heterogeneity of the devices and their applications in IoT networks. Dakdouk et al. (2021) consider activation probabilities p_m varying among the players. Computing the optimal policy becomes even more intricate in this case, and they propose an offline greedy policy, which reduces to the optimal policy with homogeneous activation rates. They also propose a greedy policy with fairness guarantees. Two Explore-then-Commit algorithms based on these baseline policies are then proposed and yield regret scaling with time as $T^{2/3}$.

Richard et al. (2023) recently proposed two centralized algorithms in the presence of homogeneous activation rates. First, a UCB-based algorithm yields a minimax regret scaling with the horizon as $\sqrt{T \log(T)}$. Second, they propose Cautious Greedy algorithm, which greedily pulls the best possible assignment, but only assigns no player to an arm if it is confident enough that the optimal assignment does not assign any player to this arm. This algorithm has two different regimes of regret: in the case where the optimal policy assign at least one player to each arm, it yields a regret constant in time; while it is scales logarithmically with time when some arm is assigned no player by the optimal policy. Extending this algorithm to a decentralized setting yet remains open and an interesting direction for future work.

Dynamic model.

Second, the players are assumed to respectively start and end the game at times $t = 1$ and T. In practice, secondary users enter and leave the network at different time steps, making this assumption oversimplifying. The dynamic model removes this assumption: players enter and leave the bandits instance at different times.

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 4.1 still reaches a regret of order $NT^{\frac{2}{3}}$ in this case when omitting the dependency in K and the means μ. N is here 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 quickly 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 $\sqrt{NT}$. Precisely, the total incurred regret by this algorithm is $\mathcal{O}\left(\sqrt{NT}\frac{K^2 \log(T)}{\Delta^2}\right)$.

This technique can be used to adapt any static algorithm but 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 a regret scaling with time in $T^{\frac{3}{4}}$ in the dynamic setting.

On the other hand, Bande and Veeravalli (2019); 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 synchronization, 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, Boursier and Perchet (2019) consider a simpler dynamic setting, where players can enter at any time but all leave the game at time T. They 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 to the best-known available arm. This algorithm leads to a regret $\mathcal{O}\left(\frac{NK\log(T)}{\bar{\Delta}^2}\right)$, with $\bar{\Delta}_{(M)} = \min_{i\in[M]} \mu_{(i)} - \mu_{(i+1)}$.

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.

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 and some level of interaction between these agents. Because of the similarities with the multiplayer bandits problem considered in this survey, the methods and techniques mentioned above can be directly used or adapted to these related problems.

The widely studied problem of multi-agent bandits is first mentioned. Section 6.2 then introduces the problem of competing bandits, motivated by matching markets. Section 6.3 finally discusses the problem of queuing systems, motivated by packet routing in servers.

Although multiplayer bandits is a particular instance of Multi-Agent Reinforcement Learning (MARL), MARL is not discussed in this survey. MARL is indeed a much more complicated problem, since determining the optimal policy (with perfect knowledge of problem parameters) is a hard problem on its own. On the other hand, determining the optimal policy in multiplayer bandits is straightforward once the arm means are known. This difference in difficulties comes from the simple structure of the rewards in multiplayer bandits, and especially because of its simple dependency in other players' actions. In particular, designing centralized MARL algorithms is still of major interest in the current MARL literature. Moreover because of the complexity of the problem, MARL constitutes a research field on its own and we refer the reader to Busoniu et al. (2008); Zhang et al. (2021) for dedicated surveys.

6.1 Multi-agent Bandits

The multi-agent bandits problem (also called cooperative bandits or 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 neighbors 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) with 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 (bistochastic) matrix. To respect the communication graph structure, $P_{i,j} > 0$ if and only if the edge (i, j) is in the graph $\mathcal{G}$. 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 a gossip UCB algorithm. Their regret decomposes in two terms: a centralized term approaching the regret incurred by a centralized algorithm and a second term, which is constant in T but depends on the spectral gap of the communication matrix P and can be seen as the *delay* to pass a message through the graph with the gossip procedure. Improving this graph-dependent term is thus the main focus of many works. Martínez-Rubio et al. (2019) 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 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. Despite having an optimal regret bound in T, the second term of the regret due to communication scales with the diameter of the graph $\mathcal{G}$. This algorithm only requires for the players to send 1-bit messages at each time step, while most multi-agent bandits works 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 a play distribution, based on EXP.3, to their followers. Instead of focusing on the collective regret as usually done, they provide good individual regret guarantees, leading to a fair algorithm.

Another line of work assumes that a player observes the rewards of all her neighbors at each time step. Cesa-Bianchi et al. (2019) 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 reaching 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 of 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 to 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. (2021) recently proposed an algorithm that is robust to malicious players. While malicious players cannot create collisions on purpose here, they can send corrupted information to their neighbors, leading to bad behaviors.

6.2 Competing Bandits

The problem of competing bandits was first introduced by Liu et al. (2020), 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 here:

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

As often in bipartite matching problems, stable matchings constitute the oracle baselines. A matching is *stable* if any unmatched pair (j, k) would prefer to be matched. Mathematically, this corresponds to the following definition.

Definition 4 *A matching $\pi : [M] \to [K]$ is stable if for any $j \neq j'$, either $\mu_{\pi(j)}^j > \mu_{\pi(j')}^j$ or $j' \succ_{\pi(j')} j$ and for any unmatched arm k, $\mu_{\pi(j)}^j > \mu_k^j$.*

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 $\bar{k}_j$:

$$\bar{R}_j(T) = \mu_{\bar{k}_j}^j T - \sum_{t=1}^{T} \mu_{\pi^j(t)}^j \cdot (1 - \eta_{\pi^j(t)}^j(t)).$$

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

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

Liu et al. (2020) 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 (2022) showed that a logarithmic optimal regret is reachable for this algorithm if the platform can also choose transfers between the players and arms, which here play a symmetric role. The idea is to smartly choose the transfers so that the optimal matching in social welfare is the only stable matching when taking into account these transfers. Their notion of equilibrium is yet weak, as the players and arms do not negotiate the transfers fixed by the platform. Jagadeesan et al. (2021) instead consider the stronger notion of equilibrium where the agents also negotiate these transfers. They define the *subset instability*, which represents the distance of a market outcome from equilibrium and is considered as the incurred loss at each round. Using classical UCB-based algorithms, they are then able to minimize the regret with respect to this measure of utility loss.

Liu et al. (2020) propose an ETC algorithm reaching a logarithmic optimal regret without any transfer. 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_k^j. This decentralized algorithm yet requires prior knowledge of Δ. Basu et al. (2021) extend this algorithm without knowing Δ, but the optimal regret incurred then scales with time as $\log^{1+\varepsilon}(T)$ for any parameter $\varepsilon > 0$.

Liu et al. (2021) 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 incurs a pessimal regret of order $\log^2(T)$ with exponential dependence 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, the algorithm of Liu et al. (2021) yields a logarithmic regret without exponential dependency in this case.

Using this simplified structure, Sankararaman et al. (2021) also propose a decentralized UCB algorithm with a 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.[7] 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 unique stable matching even in more general settings than globally ranked players. In particular, this is the case when the preferences profile satisfies uniqueness consistency. Basu et al. (2021) then adapt the aforementioned algorithm to this setting, by using a more subtle collision avoidance mechanism.

Recently, Zhang et al. (2022) proposed an ETC algorithm with a communication protocol based on SIC-MMAB. The communication protocol is yet more technical as some

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

players might not be able to communicate with each other. The arms preferences indeed fix the possible communications between player pairs. However, if two players cannot communicate, this means that one of them is always preferred over the other one; her optimal assigned arm thus does not depend on the other player's preferences. Zhang et al. (2022) thus propose a *multi-layered communication* that still allows to exchange enough information between the players to determine the optimal matching. Their proposed algorithm allows to reach a collective optimal regret scaling as $\sum_{j=1}^{M} \bar{R}_j(T) = \mathcal{O}\left(\frac{MK}{\Delta^2}\log(T)\right)$, which is thus the first $\log(T)$ optimal regret bound in the general decentralized setting. The $\frac{1}{\Delta^2}$ dependency yet remains sub-optimal. Improving upon this dependency and getting a sublinear minimax regret are possible directions for future work.

6.3 Queuing Systems

Gaitonde and Tardos (2020) extended the queuing systems introduced by Krishnasamy et al. (2016) to the multi-agent setting. This problem remains largely open as it is quite recent, but similarly to the competing bandits problem, it 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, only the oldest packet is treated and is then cleared 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 $o\left(t^{\alpha}\right)$ for any $\alpha > 0$.

A crucial quantity of interest is the largest $\eta \in \mathbb{R}$ such that

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

In the centralized case, stability of all queues is possible if and only if $\eta > 1$. Gaitonde and Tardos (2020) study whether a similar result is possible, even 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, 2021), 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 effect over the rounds. The issue of a round indeed depends on the past since a server treats the oldest packet sent by a player. A player thus can have an 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 (2021) consider the following game: all players have perfect knowledge of λ and μ and play repeatedly 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. In this game, strategic players are even stable in situations where $\eta \leq 2$ as claimed by Theorem 5 below.

Theorem 5 (Gaitonde and Tardos 2021) *If $\eta > \frac{e}{e-1}$ and all players follow a Nash equilibrium of the game described above, the system is stable.*

The limit ratio $\frac{e}{e-1}$ thus corresponds to the *price of anarchy* of this game. Yet this result holds only with prior knowledge of the game parameters and the price of "learning" remains unknown. The policy regret (Arora et al., 2012) can be seen as a patient version of regret here, as it takes into account the long-term effects of queues' actions on future rewards. Sentenac et al. (2021) show that no policy-regret strategies are no better than no regret strategies in general. In particular, they show that no policy-regret strategies can still be unstable as soon as $\eta < 2$.

They instead propose a particular decentralized strategy, such that the system is stable for any $\eta > 1$ when all the queues follow this strategy, thus being comparable to centralized performances. Similar to many multiplayer bandits algorithms, this strategy takes advantage of synchronization between the players, and extending this kind of result to the dynamic/asynchronous setting remains challenging.

Freund et al. (2022) considered a different preferences' model by the servers, which allows the players to give a bidding amount when pulling an arm. The player with the highest bid then gets to pull the arm. This allows to design a simpler decentralized algorithm, based on ascending-price auction. Besides yielding better finite time bounds on the number of left packets with respect to Sentenac et al. (2021), their algorithm is valid in the more general no-sensing, heterogeneous setting.

7. Conclusion

The multiplayer bandits problem has been widely studied in the past years. In particular, centralized optimal regret bounds are reachable in common models, by leveraging collision information as a way of communicating between players. However, these optimal algorithms might seem *ad hoc* and are undesirable in real cognitive radio networks. This suggests that the current formulation of the multiplayer bandits problem is oversimplified and does not reflect real situations and leads to undesirable optimal algorithms. Several more realistic models were suggested in the literature to overcome this issue. In particular, the dynamic and asynchronous models seem of crucial interest since time synchronization is not verified in practice and seems crucial to communicate through collision information between players. We personally believe that developing efficient strategies for both asynchronous and dynamic settings is a major direction for future research, which should lead to implementable algorithms in real applications. More generally, studying further non-stationary environments (e.g., in arm means or number of players) in multiplayer bandits is of crucial interest. Some preliminary works focus on these aspects, but yet remain far from solving the general settings.

Besides theoretical investigation of multiplayer bandits, successfully implementing multiplayer bandits algorithms in real world applications such as IoT networks remains an obvious direction of interest, that could lead to a much better quality of service of real world networks and would boost the interest in multiplayer bandits.

Besides its main application for communication networks, the multiplayer bandits problem is also related to several sequential multi-agent problems, notably motivated by distributed networks, matching markets and packet routing. Exploring further the potential relations that might exist between these different problems could be of great interest to any of them. Although the problem of multiplayer bandits seems “solved” for its classical formulation, proposing efficient algorithms adapted to real applications remains largely open.

Appendix A. Classical Stochastic Multi-Armed Bandits

This section shortly describes the stochastic MAB problem, as well as the main results and algorithms for this classical instance, which provide insights for the different algorithms and results in the multiplayer bandits literature. We refer the reader to (Bubeck and Cesa-Bianchi, 2012; Lattimore and Szepesvári, 2020; Slivkins et al., 2019) for extensive surveys on MAB and for the adversarial setting.

A.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 $X_k(t)$ of mean $\mu_k = \mathbb{E}[X_k(t)]$. 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.

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 and $\mu_{(k)}$ denotes the k-th largest mean.

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.

Definition 6 *We call an algorithm uniformly good for if for every parameter configuration $\boldsymbol{\mu}, K$ and $\alpha > 0$, it yields $R(T) = o\left(T^{\alpha}\right)$.*

The cumulated reward is of order $T\mu_{(1)}$ 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. First, Theorem 7 lower bounds the achievable regret in the classical stochastic MAB. We call an instance of stochastic MAB a setting with fixed K and arm means $(\mu_k)_{k\in[K]}$.

Theorem 7 (Lai and Robbins 1985) *Consider a bandit instance with Bernoulli distributions, i.e., $X_k(t) \sim \mathrm{Bernoulli}(\mu_k)$, then any uniformly good algorithm has an asymptotic regret bounded as*

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

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

However, the above equation does not directly imply that the maximal regret incurred at some given time T over all the possible instances is not linear in T (if μ_k is arbitrarily close to $\mu_{(1)}$, the right-hand side term can be arbitrarily large). 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 8 (Auer et al. 1995) *For any MAB algorithm and horizon $T \in \mathbb{N}$, there exists a problem instance such that*

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

A.2 Common Algorithms

This section describes the following bandit algorithms: ε-greedy, Upper Confidence Bound (UCB), Thompson Sampling and Explore-then-commit (ETC). 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 learning the problem.[8]

A.2.1 ε-greedy Algorithm

Algorithm 4: ε-greedy algorithm

```
input: $(\varepsilon_t)_t \in [0,1]^{\mathbb{N}}$
for $t = 1, \ldots, K$ do pull arm $t$
for $t = K+1, \ldots, T$ do $\begin{cases} \text{pull } k \sim \mathcal{U}([K]) \text{ with probability } \varepsilon_t; \\ \text{pull } k \in \operatorname{argmax}_{i \in [K]} \hat{\mu}_i(t) \text{ otherwise.} \end{cases}$
```

The ε-greedy algorithm described in Algorithm 4 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. 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 linear regret in 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 9.

[8] Note that it coincides with the definition given in Section 3.1.

Theorem 9 (Slivkins et al. 2019) *ε-greedy algorithm with exploration probabilities $\varepsilon_t = \left(\frac{K\log(t)}{t}\right)^{1/3}$ has a regret bounded as*

$$R(T) = \mathcal{O}\left((K\log(T))^{1/3} T^{2/3} \right).$$

A.2.2 Upper Confidence Bound Algorithm

Algorithm 5: UCB algorithm

```
for $t = 1, \ldots, K$ do pull arm $t$
for $t = K+1, \ldots, T$ do  pull $k \in \mathrm{argmax}_{i\in[K]}\, \hat{\mu}_i(t) + B_i(t)$
```

As explained above, greedily choosing the best empirical arm leads to a considerable regret, because of an under-estimation of the optimal arm mean. The UCB algorithm, given by Algorithm 5 below, 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 ensuring that the mean estimates are (slightly) positively biased with high probability. Theorem 10 bounds the regret of the UCB algorithm with its most common choice of confidence bound.

Theorem 10 (Auer et al. 2002) *The UCB algorithm with $B_i(t) = \sqrt{\frac{2\log(t)}{N_i(t)}}$ verifies the following instance dependent and minimax bounds*

$$R(T) = \mathcal{O}\left(\sum_{k:\mu_k<\mu_{(1)}} \frac{\log(T)}{\mu_{(1)} - \mu_k} \right), \tag{1}$$

$$R(T) = \mathcal{O}\left(\sqrt{KT\log(T)} \right).$$

The UCB algorithm 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).

A.2.3 Thompson Sampling Algorithm

Algorithm 6: Thompson sampling algorithm

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

The Thompson sampling algorithm described in Algorithm 6 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 maximising θ_k. It then updates its posterior distribution using the observed reward, according to the Bayes rule.

Theorem 11 (Kaufmann et al. 2012) *There exists a function f depending only on the means vector $\boldsymbol{\mu}$ such that for every problem instance with Bernoulli reward distributions 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. 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 two specific aforementioned cases.

A.2.4 Explore-then-Commit Algorithm

Algorithm 7: 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 $t = T$
```

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 can be adapted to each arm as described in Algorithm 7. 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 12.

Theorem 12 (Perchet and Rigollet 2013) *Algorithm 7 with $B_i(t) = \sqrt{\frac{2\log(T)}{N_i(t)}}$ has a regret bounded as*

$$R(T) = \mathcal{O}\left(\sum_{k:\mu_k<\mu_{(1)}} \frac{\log(T)}{\mu_{(1)} - \mu_k}\right),$$
$$R(T) = \mathcal{O}\left(\sqrt{KT\log(T)}\right).$$

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, since we can use the doubling trick (Degenne and Perchet, 2016). On the other hand, Successivation Eliminations has the advantage of being simpler to analyse (especially for more complex problems) as it clearly separates both exploration and exploitation.

Appendix B. Summary Tables

Tables 3 and 4 below summarize the theoretical guarantees of the algorithms presented in this survey. Unfortunately, some significant algorithms such as GoT (26) 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.

The upper bounds given in Tables 3 and 4 hide universal constant factors. Theoretical analyses of bandits algorithms often yield very large constant factors. These constants are often much larger than the ones observed in practice, because of looseness in the theoretical analysis. Because of these constant factors, some algorithms can outperform others in practice, despite having worse theoretical bounds.

Here is a list of the different notations used in Tables 3 and 4.

Δ	$= \min\{U^* - U(\pi) > 0 \mid \pi \in \mathcal{M}\}$
$\Delta_{(k,m)}$	$= \min\{U^* - U(\pi) > 0 \mid \pi \in \mathcal{M}$ and $\pi(m) = k\}$
$\bar{\Delta}_{(M)}$	$= \min_{k\leq M} \mu_{(k)} - \mu_{(k+1)}$
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
$U(\pi)$	$= \sum_{m=1}^{M} \mu_{\pi(m)}^{m}$
U^*	$= \max_{\pi\in\mathcal{M}} U(\pi)$

Model	Reference	Prior knowledge	Extra consideration	Upper bound
Centralized	CUCB (44)	M	-	$\sum_{\substack{\pi\in\mathcal{M}\\ U(\pi)<U^*}} \frac{\log(T)}{U^*-U(\pi)}$
Centralized	CTS (123)	M	Independent arms	$\sum_{m=1}^{M}\sum_{k=1}^{K} \frac{\log^2(M)\log(T)}{\Delta_{(k,m)}}$
Coll. sensing	dE^3 (94)	T, Δ, M	communicating players	$M^3K^2\frac{\log(T)}{\Delta^2}$
Coll. sensing	D-MUMAB(87)	T, Δ	unique optimal matching	$\frac{M\log(T)}{\Delta^2}+\frac{KM^3\log(\frac{1}{\Delta})\log(T)}{\log(M)}$
Coll. sensing	ELIM-ETC (31)	T	$\delta = 0$ if unique optimal matching	$\sum_{k=1}^{K}\sum_{m=1}^{M}\left(\frac{M^2\log(T)}{\Delta_{(k,m)}}\right)^{1+\delta}$
Coll. sensing	BEACON (111)	-	-	$\sum_{k=1}^{K}\sum_{m=1}^{M}\frac{M\log(T)}{\Delta_{(k,m)}}+M^2K\log(K)\log(T)$

Table 3: Summary of presented algorithms in the heterogeneous setting. The last column provides the asymptotic upper bound, up to universal multiplicative constant.

Model	Reference	Prior knowledge	Extra consideration	Upper bound
Centralized	MP-TS (73)	M	-	$\sum_{k>M} \frac{\log(T)}{\mu_{(M)}-\mu_{(k)}}$
Full sensing	SIC-GT (30)	T	robust to selfish players	$\sum_{k>M} \frac{\log(T)}{\mu_{(M)}-\mu_{(k)}} + MK^2 \log(T)$
Stat. sensing	MCTopM (23)	M	-	$M^3 \sum_{1\leq i<k\leq K} \frac{\log(T)}{\left(\mu_{(i)}-\mu_{(k)}\right)^2}$
Stat. sensing	RR-SW-UCB# (124)	T, M, rank	$\mathcal{O}\left(T^\nu\right)$ changes of $\boldsymbol{\mu}$	$\frac{K^2 M}{\Delta^2} T^{\frac{1+\nu}{2}} \log(T)$
Stat. sensing	Selfish-Robust MMAB (30)	T	robust to selfish players	$M \sum_{k>M} \frac{\log(T)}{\mu_{(M)}-\mu_{(k)}} + \frac{MK^3}{\mu_{(K)}} \log(T)$
Coll. sensing	MEGA (13)	-	-	$M^2 K T^{\frac{2}{3}}$
Coll. sensing	MC (100)	$T, \mu_{(M)}-\mu_{(M+1)}$	-	$\frac{MK \log(T)}{\left(\mu_{(M)}-\mu_{(M+1)}\right)^2}$
Coll. sensing	SIC-MMAB (29)	T	-	$\sum_{k>M} \frac{\log(T)}{\mu_{(M)}-\mu_{(k)}} + MK \log(T)$
Coll. sensing	DPE1 (121)	T	-	$\sum_{k>M} \frac{\log(T)}{\mu_{(M)}-\mu_{(k)}}$
Coll. sensing	C&P (3)	T	Adversarial rewards	$K^{\frac{4}{3}} M^{\frac{2}{3}} \log(M)^{\frac{1}{3}} T^{\frac{2}{3}}$
Coll. sensing	(36)	T, rank, two players	Adversarial rewards	$K^2 \sqrt{T \log(K) \log(T)}$
No-sensing	(86)	T, M	-	$\frac{MK \log(T)}{\left(\mu_{(M)}-\mu_{(M+1)}\right)^2}$
No-sensing	(86)	$T, M, \mu_{(M)}$	-	$\frac{MK^2}{\mu_{(M)}} \log^2(T) + MK \frac{\log(T)}{\Delta}$
No-sensing	(110)	$T, \mu_{(K)}, \Delta$	-	$\sum_{k>M} \frac{\log(T)}{\mu_{(M)}-\mu_{(k)}} + M^2 K \frac{\log(\frac{1}{\Delta}) \log(T)}{\mu_{(K)}}$
No-sensing	(64)	T	-	$\sum_{k>M} \frac{\log(T)}{\mu_{(M)}-\mu_{(k)}} + MK^2 \log(\frac{1}{\Delta})^2 \log(T)$
No-sensing	A2C2 (109)	$T, M,\ \alpha$	Adversarial rewards attackability $\mathcal{O}\left(T^\alpha\right)$	$M^{\frac{4}{3}} K^{\frac{1}{3}} \log(K)^{\frac{2}{3}} T^{\frac{2+\alpha+\delta}{3}}$
No-sensing	(37)	M, rank shared randomness	No collision with high proba	$MK^{\frac{11}{2}} \sqrt{T \log(T)}$
No-sensing	(36)	M, rank	Adversarial rewards	$MK^{\frac{3}{2}} T^{1-\frac{1}{2M}} \sqrt{\log(K)}$
No-sensing Non zero collision ($M \geq K$)	(16)	T, M, Δ	Small variance of noise	$\frac{KM}{\Delta} e^{\frac{M-1}{K-1}} \log(T)$
No-sensing Non zero collision ($M \geq K$)	(17)	M, rank	-	$\frac{M^3 K}{\Delta^2} \log(T)$
Dynamic, coll. sensing	DMC (100)	$T, \bar{\Delta}_{(M)}$	-	$\frac{M\sqrt{K \log(T) T}}{\bar{\Delta}^2_{(M)}}$
Dynamic, coll. sensing	(16)	T	Adversarial rewards $N \leq \mathcal{O}\left(\sqrt{T}\right)$	$\frac{K^{K+2}}{\sqrt{K \log(K)}} T^{\frac{3}{4}} + NK\sqrt{T}$
Dynamic, no-sensing	DYN-MMAB (29)	T	All players end at T	$\frac{MK \log(T)}{\bar{\Delta}^2_{(M)}} + \frac{M^2 K \log(T)}{\mu_{(M)}}$

Table 4: Summary of presented algorithms in the homogeneous setting. The last column provides the asymptotic upper bound, up to universal multiplicative constant.

References

[1] Atila Abdulkadiroğlu and Tayfun Sönmez. Random serial dictatorship and the core from random endowments in house allocation problems. *Econometrica*, 66(3):689–701, 1998.

[2] Rajeev Agrawal. The continuum-armed bandit problem. *SIAM journal on control and optimization*, 33(6):1926–1951, 1995.

[3] Pragnya Alatur, Kfir Y Levy, and Andreas Krause. Multi-player bandits: The adversarial case. *Journal of Machine Learning Research*, 21, 2020.

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

[5] Animashree Anandkumar, Nithin Michael, Ao Kevin Tang, and Ananthram Swami. Distributed algorithms for learning and cognitive medium access with logarithmic regret. *IEEE Journal on Selected Areas in Communications*, 29(4):731–745, 2011.

[6] Venkatachalam Anantharam, Pravin Varaiya, and Jean Walrand. Asymptotically efficient allocation rules for the multiarmed bandit problem with multiple plays-part i: I.i.d. rewards. *IEEE Transactions on Automatic Control*, 32(11):968–976, 1987.

[7] Venkatachalam Anantharam, Pravin Varaiya, and Jean Walrand. Asymptotically efficient allocation rules for the multiarmed bandit problem with multiple plays-part ii: Markovian rewards. *IEEE Transactions on Automatic Control*, 32(11):977–982, 1987.

[8] Raman Arora, Ofer Dekel, and Ambuj Tewari. Online bandit learning against an adaptive adversary: from regret to policy regret. In *Proceedings of the 29th International Conference on Machine Learning*, pages 1747–1754, 2012.

[9] Alireza Attar, Helen Tang, Athanasios V Vasilakos, F Richard Yu, and Victor CM Leung. A survey of security challenges in cognitive radio networks: Solutions and future research directions. *Proceedings of the IEEE*, 100(12):3172–3186, 2012.

[10] P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2-3):235–256, 2002.

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

[12] Peter Auer, Yifang Chen, Pratik Gajane, Chung-Wei Lee, Haipeng Luo, Ronald Ortner, and Chen-Yu Wei. Achieving optimal dynamic regret for non-stationary bandits without prior information. In *Conference on Learning Theory*, pages 159–163. PMLR, 2019.

[13] Orly Avner and Shie Mannor. Concurrent bandits and cognitive radio networks. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 66–81. Springer, 2014.

[14] Orly Avner and Shie Mannor. Multi-user communication networks: A coordinated multi-armed bandit approach. *IEEE/ACM Transactions on Networking*, 27(6):2192–2207, 2019.

[15] Baruch Awerbuch and Robert Kleinberg. Competitive collaborative learning. *Journal of Computer and System Sciences*, 74(8):1271–1288, 2008.

[16] Meghana Bande and Venugopal V Veeravalli. Multi-user multi-armed bandits for uncoordinated spectrum access. In *2019 International Conference on Computing, Networking and Communications (ICNC)*, pages 653–657. IEEE, 2019.

[17] Meghana Bande, Akshayaa Magesh, and Venugopal V Veeravalli. Dynamic spectrum access using stochastic multi-user bandits. *IEEE Wireless Communications Letters*, 10(5):953–956, 2021.

[18] Yogev Bar-On and Yishay Mansour. Individual regret in cooperative nonstochastic multi-armed bandits. *Advances in Neural Information Processing Systems*, 32:3116–3126, 2019.

[19] Soumya Basu, Karthik Abinav Sankararaman, and Abishek Sankararaman. Beyond $\log^2(t)$ regret for decentralized bandits in matching markets. In *International Conference on Machine Learning*, pages 705–715. PMLR, 2021.

[20] Mohsen Bayati, Nima Hamidi, Ramesh Johari, and Khashayar Khosravi. Unreasonable effectiveness of greedy algorithms in multi-armed bandit with many arms. *Advances in Neural Information Processing Systems*, 33, 2020.

[21] Dimitri P Bertsekas. Auction algorithms for network flow problems: A tutorial introduction. *Computational optimization and applications*, 1(1):7–66, 1992.

[22] Lilian Besson. *Multi-Players Bandit Algorithms for Internet of Things Networks.* PhD thesis, CentraleSupélec, 2019.

[23] Lilian Besson and Emilie Kaufmann. Multi-player bandits revisited. In *Algorithmic Learning Theory*, pages 56–92. PMLR, 2018.

[24] Lilian Besson and Emilie Kaufmann. Lower bound for multi-player bandits: Erratum for the paper multi-player bandits revisited, 2019.

[25] Lilian Besson, Emilie Kaufmann, Odalric-Ambrym Maillard, and Julien Seznec. Efficient change-point detection for tackling piecewise-stationary bandits. 2020.

[26] Ilai Bistritz and Amir Leshem. Game of thrones: Fully distributed learning for multiplayer bandits. *Mathematics of Operations Research*, 2020.

[27] Ilai Bistritz, Tavor Z Baharav, Amir Leshem, and Nicholas Bambos. One for all and all for one: Distributed learning of fair allocations with multi-player bandits. *IEEE Journal on Selected Areas in Information Theory*, 2021.

[28] Rémi Bonnefoi, Lilian Besson, Christophe Moy, Emilie Kaufmann, and Jacques Palicot. Multi-armed bandit learning in iot networks: Learning helps even in non-stationary settings. In *International Conference on Cognitive Radio Oriented Wireless Networks*, pages 173–185. Springer, 2017.

[29] Etienne Boursier and Vianney Perchet. SIC-MMAB: synchronisation involves communication in multiplayer multi-armed bandits. *NeurIPS*, 2019.

[30] Etienne Boursier and Vianney Perchet. Selfish robustness and equilibria in multi-player bandits. In *Conference on Learning Theory*, 2020.

[31] Etienne Boursier, Emilie Kaufmann, Abbas Mehrabian, and Vianney Perchet. A practical algorithm for multiplayer bandits when arm means vary among players. *AISTATS*, 2019.

[32] Tomer Boyarski, Amir Leshem, and Vikram Krishnamurthy. Distributed learning in congested environments with partial information. *arXiv preprint arXiv:2103.15901*, 2021.

[33] Simina Brânzei and Yuval Peres. Multiplayer bandit learning, from competition to cooperation. In *Conference on Learning Theory*, pages 679–723. PMLR, 2021.

[34] Sébastien Bubeck and Thomas Budzinski. Coordination without communication: optimal regret in two players multi-armed bandits. In *Conference on Learning Theory*, pages 916–939. PMLR, 2020.

[35] Sébastien Bubeck and Nicolo Cesa-Bianchi. Regret analysis of stochastic and non-stochastic multi-armed bandit problems. *Machine Learning*, 5(1):1–122, 2012.

[36] Sébastien Bubeck, Yuanzhi Li, Yuval Peres, and Mark Sellke. Non-stochastic multi-player multi-armed bandits: Optimal rate with collision information, sublinear without. In *Conference on Learning Theory*, pages 961–987, 2020.

[37] Sébastien Bubeck, Thomas Budzinski, and Mark Sellke. Cooperative and stochastic multi-player multi-armed bandit: Optimal regret with neither communication nor collisions. In *Conference on Learning Theory*, pages 821–822. PMLR, 2021.

[38] Séebastian Bubeck, Tengyao Wang, and Nitin Viswanathan. Multiple identifications in multi-armed bandits. In *International Conference on Machine Learning*, pages 258–265. PMLR, 2013.

[39] Lucian Busoniu, Robert Babuska, and Bart De Schutter. A comprehensive survey of multiagent reinforcement learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 38(2):156–172, 2008.

[40] Sarah H Cen and Devavrat Shah. Regret, stability & fairness in matching markets with bandit learners. In *International Conference on Artificial Intelligence and Statistics*, pages 8938–8968. PMLR, 2022.

[41] Nicolo Cesa-Bianchi and Gábor Lugosi. Combinatorial bandits. *Journal of Computer and System Sciences*, 78(5):1404–1422, 2012.

[42] Nicolo Cesa-Bianchi, Claudio Gentile, and Yishay Mansour. Delay and cooperation in nonstochastic bandits. *The Journal of Machine Learning Research*, 20(1):613–650, 2019.

[43] Nicolò Cesa-Bianchi, Tommaso Cesari, and Claire Monteleoni. Cooperative online learning: Keeping your neighbors updated. In *Algorithmic Learning Theory*, pages 234–250. PMLR, 2020.

[44] Wei Chen, Yajun Wang, and Yang Yuan. Combinatorial multi-armed bandit: General framework and applications. In *International Conference on Machine Learning*, pages 151–159, 2013.

[45] Richard Combes and Alexandre Proutière. Dynamic rate and channel selection in cognitive radio systems. *IEEE Journal on Selected Areas in Communications*, 33(5): 910–921, 2014.

[46] Richard Combes, Sadegh Talebi, Alexandre Proutière, and Marc Lelarge. Combinatorial bandits revisited. In *Advances in Neural Information Processing Systems*, pages 2116–2124, 2015.

[47] Richard Combes, Jungseul Ok, Alexandre Proutiere, Donggyu Yun, and Yung Yi. Optimal rate sampling in 802.11 systems: Theory, design, and implementation. *IEEE Transactions on Mobile Computing*, 18(5):1145–1158, 2018.

[48] Thibaut Cuvelier, Richard Combes, and Eric Gourdin. Asymptotically optimal strategies for combinatorial semi-bandits in polynomial time. In *Algorithmic Learning Theory*, pages 505–528. PMLR, 2021.

[49] Hiba Dakdouk, Raphaël Féraud, Romain Laroche, Nadège Varsier, and Patrick Maillé. Collaborative exploration and exploitation in massively multi-player bandits. 2021.

[50] Sumit J Darak and Manjesh K Hanawal. Multi-player multi-armed bandits for stable allocation in heterogeneous ad-hoc networks. *IEEE Journal on Selected Areas in Communications*, 37(10):2350–2363, 2019.

[51] Rémy Degenne and Vianney Perchet. Anytime optimal algorithms in stochastic multi-armed bandits. In *International Conference on Machine Learning*, pages 1587–1595. PMLR, 2016.

[52] Rémy Degenne and Vianney Perchet. Combinatorial semi-bandit with known co-variance. In *Advances in Neural Information Processing Systems*, pages 2972–2980, 2016.

[53] Riccardo Della Vecchia and Tommaso Cesari. An efficient algorithm for cooperative semi-bandits. In *Algorithmic Learning Theory*, pages 529–552. PMLR, 2021.

[54] Marco Di Felice, Rahman Doost-Mohammady, Kaushik R Chowdhury, and Luciano Bononi. Smart radios for smart vehicles: Cognitive vehicular networks. *IEEE Vehicular Technology Magazine*, 7(2):26–33, 2012.

[55] Ramon Ferrus, Oriol Sallent, Gianmarco Baldini, and Leonardo Goratti. Public safety communications: Enhancement through cognitive radio and spectrum sharing principles. *IEEE Vehicular Technology Magazine*, 7(2):54–61, 2012.

[56] Daniel Freund, Thodoris Lykouris, and Wentao Weng. Efficient decentralized multi-agent learning in asymmetric queuing systems. In *Conference on Learning Theory*, pages 4080–4084. PMLR, 2022.

[57] Tomer Gafni and Kobi Cohen. Distributed learning over markovian fading channels for stable spectrum access. *IEEE Access*, 10:46652–46669, 2022.

[58] Yi Gai, Bhaskar Krishnamachari, and Rahul Jain. Combinatorial network optimization with unknown variables: Multi-armed bandits with linear rewards and individual observations. *IEEE/ACM Transactions on Networking*, 20(5):1466–1478, 2012.

[59] Jason Gaitonde and Éva Tardos. Stability and learning in strategic queuing systems. In *Proceedings of the 21st ACM Conference on Economics and Computation*, pages 319–347, 2020.

[60] Jason Gaitonde and Éva Tardos. Virtues of patience in strategic queuing systems. In *Proceedings of the 22nd ACM Conference on Economics and Computation*, pages 520–540, 2021.

[61] Anita Garhwal and Partha Pratim Bhattacharya. A survey on dynamic spectrum access techniques for cognitive radio. *International Journal of Next-Generation Networks*, 3(4):15, 2011.

[62] A. Garivier and O. Cappé. The KL-UCB algorithm for bounded stochastic bandits and beyond. In *Conference On Learning Theory*, pages 359–376, 2011.

[63] Simon Haykin. Cognitive radio: brain-empowered wireless communications. *IEEE journal on selected areas in communications*, 23(2):201–220, 2005.

[64] Wei Huang, Richard Combes, and Cindy Trinh. Towards optimal algorithms for multi-player bandits without collision sensing information. In *Conference on Learning Theory*, pages 1990–2012. PMLR, 2022.

[65] Meena Jagadeesan, Alexander Wei, Yixin Wang, Michael Jordan, and Jacob Steinhardt. Learning equilibria in matching markets from bandit feedback. *Advances in Neural Information Processing Systems*, 34:3323–3335, 2021.

[66] Matthieu Jedor, Jonathan Louëdec, and Vianney Perchet. Be greedy in multi-armed bandits. *arXiv preprint arXiv:2101.01086*, 2021.

[67] Himani Joshi, Rohit Kumar, Ankit Yadav, and Sumit Jagdish Darak. Distributed algorithm for dynamic spectrum access in infrastructure-less cognitive radio network. In *2018 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1–6. IEEE, 2018.

[68] Wassim Jouini. *Contribution to learning and decision making under uncertainty for Cognitive Radio.* PhD thesis, 2012.

[69] Wassim Jouini, Damien Ernst, Christophe Moy, and Jacques Palicot. Multi-armed bandit based policies for cognitive radio's decision making issues. In *2009 3rd International Conference on Signals, Circuits and Systems (SCS)*, pages 1–6. IEEE, 2009.

[70] Wassim Jouini, Damien Ernst, Christophe Moy, and Jacques Palicot. Upper confidence bound based decision making strategies and dynamic spectrum access. In *2010 IEEE International Conference on Communications*, pages 1–5. IEEE, 2010.

[71] Dileep Kalathil, Naumaan Nayyar, and Rahul Jain. Decentralized learning for multiplayer multiarmed bandits. *IEEE Transactions on Information Theory*, 60(4):2331–2345, 2014.

[72] Emilie Kaufmann, Nathaniel Korda, and Rémi Munos. Thompson Sampling: An Asymptotically Optimal Finite Time Analysis. *Algorithmic Learning Theory*, 2012.

[73] Junpei Komiyama, Junya Honda, and Hiroshi Nakagawa. Optimal regret analysis of thompson sampling in stochastic multi-armed bandit problem with multiple plays. In *International Conference on Machine Learning*, pages 1152–1161. PMLR, 2015.

[74] Elias Koutsoupias and Christos Papadimitriou. Worst-case equilibria. In *Annual symposium on theoretical aspects of computer science*, pages 404–413. Springer, 1999.

[75] Subhashini Krishnasamy, Rajat Sen, Ramesh Johari, and Sanjay Shakkottai. Regret of queueing bandits. *Advances in Neural Information Processing Systems*, 29:1669–1677, 2016.

[76] Rohit Kumar, Sumit J Darak, Ajay K Sharma, and Rajiv Tripathi. Two-stage decision making policy for opportunistic spectrum access and validation on usrp testbed. *Wireless Networks*, 24(5):1509–1523, 2018.

[77] Rohit Kumar, Ankit Yadav, Sumit Jagdish Darak, and Manjesh 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)*, pages 1–8. IEEE, 2018.

[78] Branislav Kveton, Zheng Wen, Azin Ashkan, and Csaba Szepesvari. Tight regret bounds for stochastic combinatorial semi-bandits. In *Artificial Intelligence and Statistics*, pages 535–543, 2015.

[79] Tze Leung Lai and Herbert Robbins. Asymptotically efficient adaptive allocation rules. *Advances in applied mathematics*, 6(1):4–22, 1985.

[80] Peter Landgren, Vaibhav Srivastava, and Naomi Ehrich Leonard. On distributed cooperative decision-making in multiarmed bandits. In *2016 European Control Conference (ECC)*, pages 243–248. IEEE, 2016.

[81] Tor Lattimore and Csaba Szepesvári. *Bandit algorithms*. Cambridge University Press, 2020.

[82] Keqin Liu and Qing 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*, pages 1–5. IEEE, 2008.

[83] Keqin Liu and Qing Zhao. Distributed learning in multi-armed bandit with multiple players. *IEEE transactions on signal processing*, 58(11):5667–5681, 2010.

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

[85] Lydia T Liu, Feng Ruan, Horia Mania, and Michael I Jordan. Bandit learning in decentralized matching markets. *Journal of Machine Learning Research*, 22(211): 1–34, 2021.

[86] Gábor Lugosi and Abbas Mehrabian. Multiplayer bandits without observing collision information. *Mathematics of Operations Research*, 2021.

[87] Akshayaa Magesh and Venugopal V Veeravalli. Multi-user mabs with user dependent rewards for uncoordinated spectrum access. In *2019 53rd Asilomar Conference on Signals, Systems, and Computers*, pages 969–972. IEEE, 2019.

[88] Akshayaa Magesh and Venugopal V Veeravalli. Multi-player multi-armed bandits with non-zero rewards on collisions for uncoordinated spectrum access. *Journal of Environmental Sciences (China) English Ed*, 2019.

[89] Jason R Marden, H Peyton Young, and Lucy Y Pao. Achieving pareto optimality through distributed learning. *SIAM Journal on Control and Optimization*, 52(5): 2753–2770, 2014.

[90] José Marinho and Edmundo Monteiro. Cognitive radio: survey on communication protocols, spectrum decision issues, and future research directions. *Wireless networks*, 18(2):147–164, 2012.

[91] David Martínez-Rubio, Varun Kanade, and Patrick Rebeschini. Decentralized cooperative stochastic bandits. *Advances in Neural Information Processing Systems*, 32: 4529–4540, 2019.

[92] Joseph Mitola and Gerald Q Maguire. Cognitive radio: making software radios more personal. *IEEE personal communications*, 6(4):13–18, 1999.

[93] Christophe Moy and Lilian Besson. Decentralized spectrum learning for iot wireless networks collision mitigation. In *2019 15th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, pages 644–651. IEEE, 2019.

[94] Naumaan Nayyar, Dileep Kalathil, and Rahul Jain. On regret-optimal learning in decentralized multiplayer multiarmed bandits. *IEEE Transactions on Control of Network Systems*, 5(1):597–606, 2016.

[95] Aldo Pacchiano, Peter Bartlett, and Michael Jordan. An instance-dependent analysis for the cooperative multi-player multi-armed bandit. In *International Conference on Algorithmic Learning Theory*, pages 1166–1215. PMLR, 2023.

[96] V. Perchet and P. Rigollet. The multi-armed bandit problem with covariates. *The Annals of Statistics*, 41(2):693–721, 2013.

[97] Pierre Perrault, Etienne Boursier, Michal Valko, and Vianney Perchet. Statistical efficiency of thompson sampling for combinatorial semi-bandits. *Advances in Neural Information Processing Systems*, 33, 2020.

[98] Hugo Richard, Etienne Boursier, and Vianney Perchet. Constant or logarithmic regret in asynchronous multiplayer bandits. *arXiv preprint arXiv:2305.19691*, 2023.

[99] Clément Robert, Christophe Moy, and Honggang Zhang. Opportunistic spectrum access learning proof of concept. *SDR-WinnComm'14*, page 8, 2014.

[100] Jonathan Rosenski, Ohad Shamir, and Liran Szlak. Multi-player bandits–a musical chairs approach. In *International Conference on Machine Learning*, pages 155–163. PMLR, 2016.

[101] Anant Sahai, Niels Hoven, and Rahul Tandra. Some fundamental limits on cognitive radio. In *Allerton Conference on Communication, Control, and Computing*, volume 16621671. Monticello, Illinois, 2004.

[102] Abishek Sankararaman, Soumya Basu, and Karthik Abinav Sankararaman. Dominate or delete: Decentralized competing bandits in serial dictatorship. In *International Conference on Artificial Intelligence and Statistics*, pages 1252–1260. PMLR, 2021.

[103] Suneet Sawant, Rohit Kumar, Manjesh K Hanawal, and Sumit J Darak. Learning to coordinate in a decentralized cognitive radio network in presence of jammers. *IEEE Transactions on Mobile Computing*, 19(11):2640–2655, 2019.

[104] Timothy M Schmidl and Donald C Cox. Robust frequency and timing synchronization for ofdm. *IEEE transactions on communications*, 45(12):1613–1621, 1997.

[105] Flore Sentenac, Etienne Boursier, and Vianney Perchet. Decentralized learning in online queuing systems. *Advances in Neural Information Processing Systems*, 34, 2021.

[106] Stefania Sesia, Issam Toufik, and Matthew Baker. *LTE-the UMTS long term evolution: from theory to practice.* John Wiley & Sons, 2011.

[107] Munam Ali Shah, Sijing Zhang, and Carsten Maple. Cognitive radio networks for internet of things: Applications, challenges and future. In *2013 19th International Conference on Automation and Computing*, pages 1–6. IEEE, 2013.

[108] Shree Krishna Sharma, Tadilo Endeshaw Bogale, Symeon Chatzinotas, Björn Ottersten, Long Bao Le, and Xianbin Wang. Cognitive radio techniques under practical imperfections: A survey. *IEEE Communications Surveys & Tutorials*, 17(4):1858–1884, 2015.

[109] Chengshuai Shi and Cong Shen. On no-sensing adversarial multi-player multi-armed bandits with collision communications. *IEEE Journal on Selected Areas in Information Theory*, 2021.

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

[111] Chengshuai Shi, Wei Xiong, Cong Shen, and Jing Yang. Heterogeneous multi-player multi-armed bandits: Closing the gap and generalization. *Advances in Neural Information Processing Systems*, 34, 2021.

[112] Aleksandrs Slivkins et al. Introduction to multi-armed bandits. *Foundations and Trends® in Machine Learning*, 12(1-2):1–286, 2019.

[113] Balazs Szorenyi, Róbert Busa-Fekete, István Hegedus, Róbert Ormándi, Márk Jelasity, and Balázs Kégl. Gossip-based distributed stochastic bandit algorithms. In *International Conference on Machine Learning*, pages 19–27, 2013.

[114] Cem Tekin and Mingyan Liu. Online learning in opportunistic spectrum access: A restless bandit approach. In *2011 Proceedings IEEE INFOCOM*, pages 2462–2470. IEEE, 2011.

[115] Cem Tekin and Mingyan Liu. Online learning in decentralized multi-user spectrum access with synchronized explorations. In *MILCOM 2012-2012 IEEE Military Communications Conference*, pages 1–6. IEEE, 2012.

[116] William R Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3-4):285–294, 1933.

[117] Harshvardhan Tibrewal, Sravan Patchala, Manjesh K Hanawal, and Sumit J Darak. Distributed learning and optimal assignment in multiplayer heterogeneous networks. In *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*, pages 1693–1701. IEEE, 2019.

[118] Viktor Toldov, Laurent Clavier, Valeria Loscrí, and Nathalie Mitton. A thompson sampling approach to channel exploration-exploitation problem in multihop cognitive radio networks. In *2016 IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, pages 1–6. IEEE, 2016.

[119] Arun Verma, Manjesh K Hanawal, and Rahul 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)*, pages 1–8. IEEE, 2019.

[120] Daniel Vial, Sanjay Shakkottai, and R Srikant. Robust multi-agent multi-armed bandits. In *Proceedings of the Twenty-second International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing*, pages 161–170, 2021.

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

[122] Qian Wang, Kui Ren, Peng Ning, and Shengshan Hu. Jamming-resistant multiradio multichannel opportunistic spectrum access in cognitive radio networks. *IEEE Transactions on Vehicular Technology*, 65(10):8331–8344, 2015.

[123] Siwei Wang and Wei Chen. Thompson sampling for combinatorial semi-bandits. In *International Conference on Machine Learning*, pages 5114–5122, 2018.

[124] Lai Wei and Vaibhav Srivastava. On distributed multi-player multiarmed bandit problems in abruptly changing environment. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 5783–5788. IEEE, 2018.

[125] Michael Woodroofe. A one-armed bandit problem with a concomitant variable. *Journal of the American Statistical Association*, 74(368):799–806, 1979.

[126] Renzhe Xu, Haotian Wang, Xingxuan Zhang, Bo Li, and Peng Cui. Competing for shareable arms in multi-player multi-armed bandits. In *International Conference on Machine Learning*. PMLR, 2023.

[127] Marie-Josépha Youssef, Venugopal Veeravalli, Joumana Farah, and Charbel Abdel 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.

[128] Kaiqing Zhang, Zhuoran Yang, and Tamer Başar. Multi-agent reinforcement learning: A selective overview of theories and algorithms. *Handbook of reinforcement learning and control*, pages 321–384, 2021.

[129] Yirui Zhang, Siwei Wang, and Zhixuan Fang. Matching in multi-arm bandit with collision. *Advances in Neural Information Processing Systems*, 35:9552–9563, 2022.

[130] Ruikang Zheng, Xuan Li, and Yudong Chen. An overview of cognitive radio technology and its applications in civil aviation. *Sensors*, 23(13):6125, 2023.