Massive multi-player multi-armed bandits for IoT networks: An application on LoRa networks

Hiba Dakdouk, Raphaël Féraud, Nadège Varsier, Patrick Maillé, Romain Laroche

Massive Multi-Player Multi-Armed Bandits for IoT Networks: An Application on LoRa Networks

Hiba Dakdouk[a,*], Raphaël Féraud[a], Nadège Varsier[a], Patrick Maillé[b], Romain Laroche[c]

[a] *Orange Labs, France*
[b] *IMT Atlantique, France*
[c] *Microsoft Research Lab, Canada*

Abstract

More and more manufacturers, as part of the transition toward Industry 4.0, are using Internet of Things (IoT) networks for more efficient production. The wide and extensive expansion of IoT devices and the variety of applications generate different challenges, mainly in terms of reliability and energy efficiency. In this paper, we propose an approach to optimize the performance of IoT networks by making the IoT devices intelligent using machine learning techniques. We formulate the optimization problem as a massive multi-player multi-armed bandit and introduce two novel policies: Decreasing-Order-Reward-Greedy (`DORG`) focuses on the number of successful transmissions, while Decreasing-Order-Fair-Greedy (`DOFG`) also guarantees some measure of fairness between the devices. We then present an efficient way to manage the trade-off between transmission energy consumption and packet losses in Long-Range (LoRa) networks using our algorithms, by which LoRa nodes adjust their emission parameters (Spreading Factor and transmitting power). We implement our algorithms on a LoRa network simulator and show that such learning techniques largely outperform the Adaptive Data Rate (ADR) algorithm currently implemented in LoRa devices, in terms of both energy consumption and packet losses.

Keywords: IoT, Multi-player multi-armed bandits, LoRaWAN, resource allocation

1. Introduction

The Internet of Things (IoT) has gained a great attention in the last decade. The world has been witnessing such a massive growth in the node deployment that the IoT survey reported on the Forbes website [1] forecasts more than 75 billion connected IoT devices by 2025. Massive IoT applications require energy-efficient and low-complexity nodes. To support such requirements, Low Power Wide Area Networks (LPWANs), that provide large coverage areas, low transmission data rates with small packet data sizes, low device complexity and long battery life have evolved [2]. LPWANs include several technologies operating in the unlicensed industrial, scientific and medical (ISM) frequency band (868 MHz in Europe, 915 MHz in North America, and 433 MHz in Asia), and the Long Range Wide Area Network (LoRaWAN) developed by the LoRa alliance [3] is one of these massively deployed technologies for low power and long distance transmissions. However, with the great increase of IoT deployment, a major problem of systems' coexistence arises. Inside the unlicensed band, the different systems are not separated in the frequency domain but overlapping in the sense that they may use the same frequency resource at any time, causing interference and hence transmission failures. In this context, we propose in this paper an approach to optimize the communications in IoT networks by configuring IoT devices so that they are aware of the best operating parameters in order to avoid interference and packet loss while consuming as little energy as possible.

We consider a large number N of devices communicating through a unique gateway on a limited number K of orthogonal (independent) channels ($N \geq K$). The devices use an acknowledgement protocol slotted in time, where an acknowledgement is sent by the gateway to the transmitting device

*Corresponding author.
Email address: hiba.dakdouk@hotmail.com

after each successful transmission. A transmission fails if and only if a *collision* occurs. If this is the case, the packets of all colliding devices are lost and no acknowledgement is sent. There exist two types of collision. An *internal collision* occurs when two or more devices send packets to the gateway at the same time slot through the same channel. *External collisions* may also occur with unknown and uncontrolled devices. Therefore, even if only one device sends a packet on one channel at a given time slot, the packet may not be received by the gateway. Because of their nature, *external collisions* make the probabilities of successful transmission (and hence the channels' qualities) uncontrollably differ over channels. Another important feature of the studied problem is that, in the general case, the gateway cannot know that packets have been sent by some devices if a collision occurs. As a consequence, the estimation of the channel quality can only be done at the device side in a decentralized way. In such conditions and in order to maximize the number of successful transmissions of the IoT network while consuming as little energy as possible, we model our problem as a massive multi-player multi-armed Bandit.

Multi-Player Multi-Armed Bandits. Multi-Armed Bandit (MAB) refers to an online decision-making game where a *player* has to make decisions at specific time steps by selecting an arm from a set of K available arms. Each arm is associated with a sequence of rewards that are randomly and independently drawn according to an unknown distribution. At each turn, the player should select an arm and receive the reward corresponding to the selected arm. The player's goal is to maximize its cumulative reward over time by compromising between *exploring* the arms that have loosely estimates in order to build a better one, and *exploiting* the arm that seems to be the best in order to maximize the cumulative reward. The player should follow a certain *policy* that chooses the arm to play at each turn based on the previous outcomes.

In this work, we focus on *stochastic* MABs, where we assume that the rewards are generated independently from an unknown and constant distribution. **UCB** (Upper Confidence Bound) [4] is one of the most commonly used algorithms in stationary stochastic environments. It builds an upper confidence bound of the expected reward of each arm, and selects the arm with the highest bound at each iteration. **UCB** can be used in *selfish* MAB [5] for optimizing the packet data rate in IoT networks. Notice however that the basic assumption of *selfish* UCB does not hold in our setting: due to internal collisions and learning of other players, the reward evolves during time.

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

To set the aforementioned problem into the framework of multi-player MAB, each IoT device is considered as a player, a channel is considered as an arm, and the reward corresponds to the reception or not of the acknowledgement from the gateway.

Related Work. The *decentralized multi-player multi-armed bandits* have been studied for opportunistic spectrum access in [6, 7, 8, 9], where primary users have a strict priority over secondary users, which are allowed to *sense* a channel before sending a packet in order to check whether it is free. The objective of those works is to avoid collisions between concurrent secondary users, that share the same channels, while choosing the best channels, i.e., with the highest probabilities of being free of primary users. This line of work makes the assumption that there are less players than channels, that the collisions with other players are observed, and uses orthogonalization techniques to avoid collisions. In [10], the authors propose to use collisions to estimate in a first phase the number of players and the value of arms, and then applies a Musical Chair approach to allocate each player on a different N-best arm. In [11], the authors improve this approach by reducing the first phase to the estimation of the value of arms and then use a trekking approach to allocate each player on a different N-best arm without the knowledge of the number of players. In [12], the authors propose a communication protocol based on controlled collisions that achieves almost the same performance as

a centralized algorithm. In [13], the authors improve this result by electing a leader that explores the arms and allocates other players on different estimated N-best arms. The leader communicates to the other players the list of estimated N-best arms when it changes using the same communication protocol as in [12]. An interesting extension of the problem setting is proposed in [14] for handling the case where the mean rewards of arms are not the same for each player. Despite its merits, this thread of research makes the assumption that *sensing* information is available and the number of players is small ($N \leq K$), which are respectively impractical and unrealistic assumptions for IoT networks. In contrast, in this paper we do not consider any condition on the number of players, neither we consider primary/secondary user setting, and we do not allow sensing. Instead, players observe the success or failure of their transmissions.

Motivated by IoT networks, in [5] the authors propose a new problem setting where *sensing* is not allowed, the number of players is larger than the number of channels, and the players asynchronously play: each player has the same probability of sending a packet at each time slot. The authors show experimentally that *selfish UCB*, which consists in each player independently playing *UCB* [4], i.e., a classic commonly-used MAB algorithm, works surprisingly well. This experimental result has been confirmed in the case of LoRa networks using stochastic and non-stochastic multi-armed bandits in [15, 16] or in the case of the IEEE 802.15.4 time-slotted channel hopping protocol [17]. Despite its good experimental performance, this algorithm has no theoretical guarantees, and it has been shown that *selfish UCB* can fail badly on some cases [18]. With a similar problem setting but with different probabilities to send packets, the authors in [19] propose a cooperative algorithm that aims to find a set of optimal arms while minimizing the number of plays. However, that work does not optimize the number of optimal arms to find, and the exploitation policy followed by the players is uniform, which is clearly sub-optimal. These limitations are resolved with our proposed algorithms.

Finally, the optimization problem we propose to solve is related to the slotted-Aloha protocol [20], where each player n transmits a packet with a probability p_n at the beginning of a slot. For instance, in [21] the authors formulate the decentralized throughput maximization problem in an Aloha network with a single channel in a way that is close to our optimization problem. However, that work considers a single channel, and the decision variable is the sending probability p_n rather than the choice of the channel. If the probabilities of sending a message are optimized, then the application constraints of IoT (frequency of sending messages or real-time messages) cannot be respected. In [22], the authors propose a best-response algorithm which solves the throughput maximization problem for the multi-channel Aloha protocol. They notably show that the best-response algorithm converges to a Nash Equilibrium in a finite time. However the authors consider that the channel capacities and the strategies of other players are known, and that each player has the same probability of sending a message at each slot, which is unrealistic and restrictive for IoT networks.

Contributions and paper organization. In this paper, we study the extension of the problem proposed in [5], where at each time slot, each device n has a probability p_n to send a packet to the gateway [19]. We propose an *explore-then-exploit* approach, where a decentralized exploration algorithm outputs an estimation of the parameters. Players send these estimates to the gateway in order to centralize the decision making. Then, the gateway computes a policy to be used during the exploitation phase. We then test our approach on LoRa networks using a LoRa network simulator, and compare it with the already-implemented Adaptive Data Rate (ADR) algorithm.

In section 2, after discussing the assumptions and simplifications done in comparison to a real IoT network, we formalize the objective of optimizing the successful transmissions. We show in Theorem 1, that there exists a deterministic policy (an assignment of players over arms) that is optimal. Then, in section 3, we propose two deterministic policies: `DORG` (decreasing-order-reward-greedy) aims to optimize the number of successful transmissions, while `DOFG` (decreasing-order-fair-greedy) guarantees fairness between players in terms of successful transmission rate. In Theorem 2, we show that `DORG` is optimal at least in the setting proposed in [5] (when $\forall n,\ p_n = p$), while Theorem 3 establishes fairness guarantees for `DOFG`. We then compare the performance of the two policies in preliminary experiments in section 3.3. In section 4, we propose a collaborative exploration algorithm, which has to be decentralized since the packet loss can only be observed by players. The

players output unbiased estimates of the mean rewards of arms, *i.e.* the probability of not suffering an external collision, with classic concentration properties. Theorem 5 proves an upper bound on the number of time steps needed to output a controlled approximation of the arms that is near optimal in comparison to the lower bound of K biased coin estimations in $\Omega\left(K/\epsilon^2 \log 1/\delta\right)$ [23]. Furthermore, Theorem 4 guarantees its communication efficiency by stating an upper bound on its communication cost in $O\left(NK \log(NK+N)/\delta\right)$. Theorem 7 establishes a pseudo-regret lower bound in $\Omega\left(T^{2/3}\frac{\log T}{N}\right)$, which holds for any *explore-then-exploit* algorithm, and unveils the hardness of the studied problem in comparison to the multi-armed bandit and multi-player bandit problems. Then, in the specific setting when $\forall n$, $p_n = p$ (proposed in [5]), Theorem 6 demonstrates that `DORG` enjoys a pseudo-regret upper bound that is optimal in T. Finally, Theorem 8 states fairness guarantees of our *explore-then-exploit* algorithm with `DOFG`. In section 4.4, we compare our approach with the state-of-the-art methods on a large set of synthetic problems. Our experiments reveal that when using `DORG`, the proposed algorithm outperforms the baselines in terms of successful communication rate, and when using `DOFG` it outperforms them in terms of fairness between players. In section 5, we implement the proposed algorithms and some MAB baselines into LoRaWAN technology. The experiments done on a realistic simulator show that the Adaptive Data Rate (ADR) algorithm, which is currently implemented in LoRa protocol, is largely outperformed by our algorithms in terms of energy consumption and packet losses. We moreover show that if a team of nodes uses ADR while another team uses our approach, the first team consumes more energy and suffers of more packet losses. We finally conclude in section 6 by suggesting directions for future work. The reader will find the societal impact in appendix Appendix .1, additional experiments in appendix Appendix .2 and the proofs in appendix Appendix .3.

2. Massively Multi-Player Multi-Armed Bandits

In the following, we model the problem of optimizing the communications in IoT networks as a massive MP-MAB after presenting the main assumptions that we make.

2.1. Underlying assumptions

To best formulate our optimization problem, we model an IoT network by considering the following:

1. **The number of devices could be greater than the number of channels:** Unlike most of previously mentioned works, we do not assume the number of devices is less than the number of channels. Indeed, in Internet of Things (IoT) networks, a large number of devices are connected to the Internet through wireless gateways, and hence the number of devices cannot be lesser than the number of channels.
2. **Each successful uplink transmission is followed by a downlink acknowledgement:** The communication protocols used in IoT allow to assign a binary outcome for each transmission (success or not) since each uplink transmission is followed by time windows during which the device listens to the gateway to receive the acknowledgement of the uplink transmission.
3. **Sensing information is not possible:** The players cannot distinguish internal and external collisions, rather they can only observe the success or failure of their transmissions. This is known to be a difficult case for multi- player multi-armed bandits, however it is realistic for IoT networks, where sensing information is too costly in terms of energy consumption.
4. **Downlink transmissions do not fail:** We do not consider that collisions could occur when the gateway sends acknowledgements. Indeed, these downlink collisions require that at least two acknowledgements are sent from the gateway at the same time to different devices located at the same place, which cannot happen with a unique gateway using a protocol slotted in time, and which would be unlikely in a real Internet of Things (IoT) network, where a finite number of gateways is positioned to cover the maximum area.
5. **Each player has a probability of sending a packet at each time step:** The frequency of sending packets through the gateway depends on the application (healthcare, security, smart cities, marketing, home automation...). Moreover, for several real-time applications, the device has to send a packet when an unknown and uncontrolled event occurs. For instance, a user's device can interact with its

environment in real-time, to get a green light when the user faces a crossroad, an ad when the user is in front of a shop, a ticket when getting on the bus, and more critical applications such as healthcare ones. Such packets has to be sent and processed as soon as possible, and therefore the authors in [16] suggest a modification in the LoRA@FIIT [24] link-layer protocol, so such emergency packets are given the priority to be retransmitted in case of failure over other types of normal packets in order to guarantee QoS in LoRa networks. In this work, in order to model the packets' delivery rate, we assume that each player has a probability of sending a packet at each time step.

6. **Players are Socratic**[1]**:** Considering that the probability of sending a packet depends mainly on the type of devices, we assume that each player knows its own probability of sending a packet.
7. **Known number of players:** We assume that the number of players is known by the gateway, which is realistic in IoT protocols (the gateway can keep track of all the devices it has received packets from), and that the gateway sends this information to each player at the beginning of the game.
8. **Players can share information by including their messages in the payload of the packet they need to send:** We allow the devices to share information by sending messages to other devices through the gateway using the IoT protocol. As in IoT networks the payload of each packet can contain up to 255 bytes [25, 26], we assume that in the same packet 8 bytes of the payload can be used to send a message to other players. We hereby distinguish between the two terms: a *packet* that corresponds to the regular transmissions of a device, and a *message* that corresponds to the information shared between the players.

2.2. Problem Formulation

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

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

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

$$\mu_{n,\boldsymbol{\theta}}^k(\pi) = \theta^k \prod_{n'=1, n'\neq n}^{N} (1 - p_{n'}\pi_{n'}^k). \qquad (1)$$

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

[1] from the ancient Greek aphorism "know thyself" attributed to Socrates.

$$\mu_{\boldsymbol{\theta}}(\pi) = \sum_{k=1}^{K} \theta^k \sum_{n=1}^{N} p_n . \pi_n^k \prod_{n' \in [N] \setminus \{n\}} (1 - p_{n'} \pi_{n'}^k). \tag{2}$$

This performance metric corresponds to the expected number of successful transmissions per time slot. The optimization problem in Equation (2) with respect to π has a solution, since the objective function is continuous and the set of decision variables is compact. But the problem itself is not convex with respect to π_n^k, hence classical convex optimization methods cannot be applied.

Another approach is to consider a particular subset of policies: the subset of deterministic policies is obtained when $\forall (k,n) \in [K] \times [N]$, $\pi_n^k \in \{0,1\}$. Let k_n be the arm assigned to player n. The expected reward per time slot in model $\boldsymbol{\theta} = \{\theta^1, ..., \theta^K\}$ of a deterministic policy π can then be written as:

$$\mu_{\boldsymbol{\theta}}(\pi) = \sum_{n=1}^{N} p_n \theta^{k_n} \prod_{n' \neq n, \text{ s.t. } k_{n'} = k_n} (1 - p_{n'}) \tag{3}$$

$$= \sum_{k=1}^{K} \theta^k \underbrace{\prod_{n \in [N], \text{ s.t. } k_n = k} (1 - p_n)}_{z^k} \underbrace{\sum_{n \in [N], \text{ s.t. } k_n = k} \frac{p_n}{1 - p_n}}_{\ell^k},$$

where z^k is the probability that no player assigned to arm k sends a packet, and ℓ^k is the sum of the activation odds for all players assigned to arm k.

Theorem 1. *There exists a policy maximizing the overall network utility (equation (2)) that is deterministic.*

Theorem 1 states that at least one solution is a deterministic policy, which justifies to consider only the subset of deterministic policies. However, as a deterministic policy is an assignment of players over arms, there are N^K deterministic policies. This means that when N and K are not small, finding the optimal policy is hopeless, and this even if the model $\boldsymbol{\theta}$ is known in advance, which is not the case.

2.3. Discussion

In face of these impossibility results for both stochastic and deterministic policies, for handling massively multi-player multi-armed bandits, we aim to find reasonably good deterministic target policies in the next section. Then, in section 4, we will propose an exploration algorithm that finds an unbiased and controlled approximation of the model $\boldsymbol{\theta}$ for computing the target policy. This *explore-then-exploit* approach allows to compute a controlled approximation of a target policy, even in the case where N and K are not small. Moreover if N and K are small, a controlled approximation of the optimal policy can be obtained. The alternative approach consisting of using an optimal multi-armed bandit algorithm that consider each deterministic policy as an arm will lead to a regret lower bound in $\Omega\left(\sqrt{N^K T}\right)$ [27]. Notice that even when N and K are small (for instance in the order of 10) the dominant term of the regret lower bound is not T, but N^K.

3. Collaborative Exploitation in Massively Multi-Player Bandits

3.1. Reward greedy algorithm

In this section, we propose a greedy algorithm that aims to maximize the network utility (equation (3)).

Lemma 1. *For a deterministic policy π, let $\mu_{\boldsymbol{\theta}}(\pi[n])$ denote the expected reward when only players $1, ..., n$ are playing (all players $n' > n$ are deactivated). Then we have the recursive expression:*

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

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

Lemma 1 reveals a recurrence relation over n of the expected total reward. Under the assumption that the problem parameters are known, Lemma 1 paves the way to the definition of `DORG`, decreasing-order-reward-greedy (Algorithm 1), a recursive algorithm that assigns player n to arm k_n such that the right-hand term of the recursive equation in Lemma 1 is maximized. The result is highly dependent on the order in which the players are added to the pool, but the following theorem shows Algorithm 1 can lead to an actual optimum.

Theorem 2. *If $\sum_{n \in [N]} \frac{p_n}{1-p_n} \leq K + 1$, then there exists an ordering over players $\sigma^* : [N] \to [N]$ such that Algorithm 1 returns an optimal policy.*

Algorithm 1 Reward Greedy
(DORG if players are sorted in p_n decreasing order)

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

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

Algorithm 2 Fairness Greedy
(DOFG if players are sorted in p_n decreasing order)

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

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

Remark 1. *When $\forall n, p_n = p$ [5] Theorem 2 states that DORG returns the optimal policy. The precondition of Theorem 2 clearly holds in IoT networks, where the duty cycle p is commonly set to less than* 0.01 *[26].*

In DORG, we sort the players in the decreasing order of their probabilities to send packets so that the most frequently-playing players, which are responsible of a major part of sent packets, are assigned to the best arms in order to maximize the number of successful transmissions. In the next section, we experimentally show that scheduling players by decreasing activity values is a good heuristic, significantly outperforming the random scheduling.

3.2. Fairness greedy algorithm

Theorem 1 states that the resource assignment of an optimal deterministic policy is a Pareto optimum: as the network utility is maximum, if a user increases its own utility (equation (1)) another user has necessarily to decrease its utility (due to equation (2)). Notice that a Pareto optimum does not provide any guarantee about the *fairness* of the resource allocation among players. In this section, we design a policy to ensure fairness among players, for which we will use the definition below.

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

Building a fair policy can be done by balancing the load with respect to the mean rewards of arms. The fair greedy algorithm (see Algorithm 2) assigns sequentially each player to the arm that maximizes the reward of the arm times the probability of no internal collision. The player scheduling also plays an important role and we prove a lower bound on the fairness of Algorithm 2, when players are sorted in decreasing order of p_n. In that case we coin this algorithm DOFG, which stands for decreasing-order-fair-greedy.

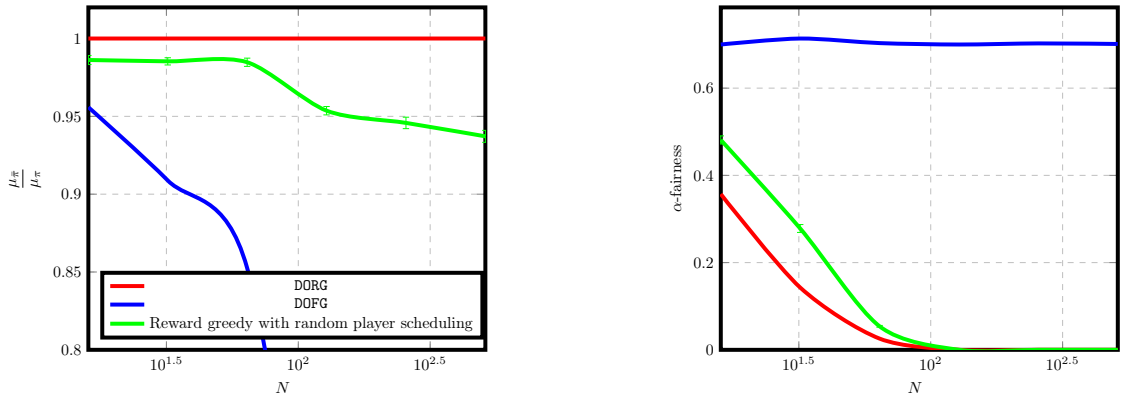
Theorem 3. *DOFG generates α-fair policies, with $\alpha \geq 1 - \max_{n\in[N]} p_n$.*

Theorem 3 implies that when the probability of sending packets of the most frequent player is not high, which is the case in IoT networks, DOFG is a fair policy. In the following section, experimental evidence about performance and fairness of DORG and DOFG is provided.

3.3. Preliminary Experiments

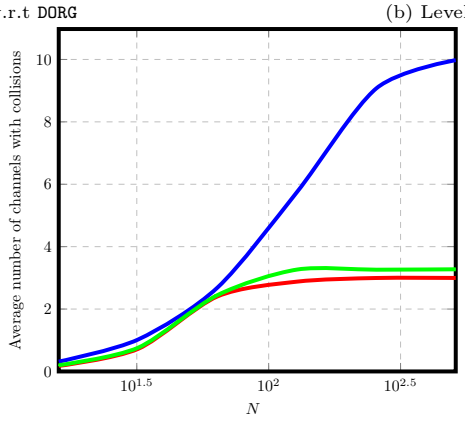
In this section, we perform the following experiment: the problem parameters are sampled as follows: $\forall n \in [N], p_n \sim \mathcal{U}(0, 0.3)^2$ and $\forall k \in [K], \theta^k \sim \mathcal{U}(0, 1)$. Figure 1 compares the performance of DORG, DOFG, and Reward Greedy (Algorithm 1) with random ordering, where each point is the average of 10,000 runs. Figure 1a that compares the expected reward ratio of the algorithms with respect to DORG, where $\bar{\pi}$ denotes the policy to be compared with DORG, reveals that sorting the players in decreasing order is a good policy. However, it has to be noted that the difference between DORG and a random ordering is much thinner when p_n are smaller, as expected in a real setting. We also notice that DOFG expected reward loss, as compared to DORG, is below 20% until $N \approx 75$. Figure 1b illustrates the result of Theorem 3, and indicates that the fairness lower bound is tight. It also shows that, while DOFG only loses 20% rewards when $N \approx 75$ as

[2]Such high values for p_n are used to graphically observe the expected properties.



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

(b) Level of fairness between players



(c) Number of channels with internal collisions

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

compared to DORG, its fairness is approximately 30 times larger.

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

Similar experiments with different settings are presented in Appendix Appendix .2.

4. Collaborative Exploration in Massively Multi-Player Bandits

The policies *Reward Greedy* and *Fairness Greedy* necessitate the knowledge of the model θ and the probabilities to send packets of the players p_n which are unknown to the players. Therefore, we propose in the following a collaborative exploration algorithm that returns unbiased estimates of the mean rewards of the arms.

4.1. Principle

Decentralized explore-then-exploit approach: The choice of the policy depends on the metric to be maximized: for maximizing network utility, DORG policy (Algorithm 1) should be used, while to guarantee some fairness among players, DOFG policy (Algorithm 2) is to be used. However

both policies require an estimate of the model $\boldsymbol{\theta}$, which can only be obtained after sufficient exploration. Since the gateway cannot observe the collisions (packet losses), the learning (exploration) is done at the device (player) side. Therefore, we propose a decentralized exploration algorithm performed with the packets that the players have to send, i.e. they do not send extra packets dedicated for exploration but just the packets they need to send with probability p_n, hence they do not lose any of their packets neither consume higher energy. Then, the exploration is followed by an exploitation phase, i.e. *explore-then-exploit* approach: an exploration algorithm shares the probabilities of sending packets of players at the beginning, which is necessary to compute the exploration policy of each player and outputs an ϵ-approximation of the model $\boldsymbol{\theta}$ with high probability for a sufficiently small ϵ, and then a target policy is used during the exploitation phase.

Definition 2 (ϵ-approximation)**.** *$\hat{\theta}^k$ is said to be an ϵ-approximation of arm k, if the difference between it and θ^k is less than ϵ : $|\theta^k - \hat{\theta}^k| \leq \epsilon$.*

Collaboration: In order to reduce the exploration time needed to find an ϵ-approximation of each arm, we propose to distribute the exploration task on the players: each player is responsible of a predefined number of samples t_n^* for each arm according to its probability of sending a packet, so that all players would finish their estimations almost at the same time. At the end of the exploration phase, each player sends its ϵ-approximation of each arm to other players through the gateway. Then, the target policy can be computed in a centralized way (by the gateway) or separately within each player. Our exploration algorithm as the algorithm in [28], belongs to the federated multi-armed bandits as defined in [29], as the players learn independently on different data and share their knowledge afterwards. In Algorithm 3, we assume that a message to other player can be sent with the packet at the same time slot (see section 2).

4.2. Description of the algorithm

The function send(s), used in Algorithm 3, means that message s is sent by player n and broadcast to other players through the gateway on a channel chosen uniformly over K. The function send(s) returns 1 if an acknowledgement is received by player n from the gateway or 0 else. When player n receives the probabilities of sending packets of all other players (Algorithm 3 line 11), it computes the required number of samples of each arm t_n^* according to Lemma 2. When player n samples at least t_n^* times an arm k, it sends its estimation $\hat{\theta}_n^k$ and t_n^k to other players (Algorithm 3 lines 16,18) each in a distinct message (in distinct time slots). $\hat{\theta}_n^k$ is computed according to equation (4). The exploration phase ends when the arms have been sampled enough by a subset of players and the estimations of this subset have been successfully sent (Algorithm 3 line 20). Finally, the players compute the global estimations of arms by combining the received local ones (Algorithm 3 line 21).

Algorithm 3 Collaborative Exploration in Massively Multi-Player Multi-Armed Bandits

Inputs: $[K]$, $[N]$, $\epsilon \in [0,1]$, $\delta \in (0,1)$
Output: $\hat{\theta} = \{\hat{\theta}^k, \forall k \in [K]\}$
Init: $t := 0$; $\forall n \in [N] : t_n^* := \infty$, $ack1_n := 0$; $\forall (n,k) \in [N] \times [K]$: $ack2_n^k := 0, ack3_n^k := 0$

1 **repeat**
2 $\quad \mathcal{N}_t := \{n \in [N], a_n \sim \mathcal{B}(p_n), a_n = 1\}$
3 $\quad$ **for** $n \in \mathcal{N}_t$ **do**
4 $\quad\quad k_n \sim \mathcal{U}(1,K)$
5 $\quad\quad Y_n^{k_n}(t_n^{k_n}) := I_n^{k_n} E^{k_n}$
6 $\quad\quad \hat{\mu}_n^{k_n}(\pi_u) := \sum_{t=1}^{t=t_n^{k_n}} Y_n^{k_n}(t)/t_n^{k_n}$
7 $\quad\quad t_n^{k_n} := t_n^{k_n} + 1$
8 $\quad\quad$ **if** $ack1_n = 0$ **then**
9 $\quad\quad\quad ack1_n := send(p_n)$
10 $\quad\quad$ **else**
11 $\quad\quad\quad$ **if** $\forall i \in [N], ack1_i = 1$ **then**
$\quad\quad\quad\quad \forall i, t_i^* := \dfrac{p_i \log(2K/\delta)}{2(\epsilon \rho_i^k(\pi_u))^2 \sum_{j=1}^N p_j}$
$\quad\quad\quad$ **end if**
12 $\quad\quad\quad$ **if** $\exists k, t_n^k \geq t_n^*$ **then**
13 $\quad\quad\quad\quad$ **if** $ack2_n^k = 0$ **then**
14 $\quad\quad\quad\quad\quad ack2_n^k := \text{send}(\hat{\theta}_n^k)$
15 $\quad\quad\quad\quad$ **else if** $ack3_n^k = 0$ **then**
16 $\quad\quad\quad\quad\quad ack3_n^k := \text{send}(t_n^k)$
17 $\quad\quad\quad\quad$ **end if**
18 $\quad\quad\quad$ **end if**
19 $\quad\quad$ **end if**
20 $\quad$ **end for**
21 $\quad t = t + 1$
until $\exists \mathcal{N}' \subset \mathcal{N}, \begin{cases} \forall k \sum\limits_{n \in \mathcal{N}'} t_n^k \geq \sum\limits_{n \in \mathcal{N}} t_n^* \\ \forall k \sum\limits_{n \in \mathcal{N}'} ack2_n^k = |\mathcal{N}'| \end{cases}$
all players calculate $\hat{\theta}^k := \dfrac{\sum_{n \in \mathcal{N}'} \hat{\theta}_n^k t_n^k}{\sum_{n \in \mathcal{N}'} t_n^k}$

The sampling strategy used in *collaborative exploration* is the *Uniform Policy* π_u: $\forall n, \forall k, \pi_n^k = \frac{1}{K}$. Then, player n can estimate the mean reward of arms using:

$$\hat{\theta}_n^k = \frac{\hat{\mu}_n^k(\pi_u)}{\rho_n^k(\pi_u)}, \text{ where} \tag{4}$$

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

Lemma 2. *With Algorithm 3, to obtain with a probability* $1-\delta$ *an* ϵ*-approximation of the mean rewards of arms, player* n *needs to sample each arm at least*

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

In the following we study the communication cost and the exploration duration of the proposed exploration algorithm.

4.3. Analysis of the algorithm

The communication cost presents the number of transmissions needed to successfully send the messages of Algorithm 3.

Theorem 4. ***Communication cost.*** *When Algorithm 3 stops, the number of messages sent is, with probability* $1-\delta$*, less than* $C(N(1+2K))$*, where*

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

Theorem 4 states an upper bound on the number of messages issued by the N players for sharing the probabilities of sending packets, and for sharing their estimations that is in $O\left(NK \log \frac{NK+N}{\delta}\right)$.

Theorem 5. ***Exploration duration.*** *With a probability at least* $1-\delta$*, when* $N \geq 2$ *Algorithm 3 stops while finding the* ϵ*-approximations of model* $\boldsymbol{\theta} = \{\theta^1, ..., \theta^K\}$ *at:*

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

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

Theorem 5 states an upper bound on the number of time slots needed by all players to finish their estimations of the mean rewards of the arms and to share them. The left term in $O\left(K^{3/2}/\epsilon^2 \log K/\delta\right)$ is the dominating term of the upper bound of the sample complexity. It is near optimal in comparison to the lower bound of K biased coin estimations in $\Omega\left(K/\epsilon^2 \log 1/\delta\right)$ [23].

For the regret analysis of the proposed algorithm, we define the pseudo-regret as follows:

Definition 3 (Pseudo-regret)**.** *Let* π_t *be a policy generated at time* t *by an algorithm, and* $\mu_{\boldsymbol{\theta}}(\pi_t)$ *be its value in model* $\boldsymbol{\theta} = \{\theta^1, ..., \theta^K\}$*, we define the pseudo-regret with respect to the optimal policy* $\pi_{\boldsymbol{\theta}}^*$ *as* $R(T) = \sum_{t=1}^{T} (\mu_{\boldsymbol{\theta}}(\pi_{\boldsymbol{\theta}}^*) - \mu_{\boldsymbol{\theta}}(\pi_t))$.

Theorem 6. ***Pseudo-regret upper bound.*** *when* $\forall n \in [N], p_n = p$*, and* $N \geq 3$*, the pseudo-regret with respect to the optimal policy* $\pi_{\boldsymbol{\theta}}^*$ *of Algorithm 3 followed by the policy* $\pi_{\hat{\boldsymbol{\theta}}}^*$ *is upper bounded by:*

$$R(T) \leq O\left(\frac{T^{2/3} \log NKT}{p^{3/2}(1-p/K)^{2N-2}N} + K^{9/4} T^{2/3}\right).$$

To show how tight this bound is we provide below a lower bound on the pseudo-regret of any explore-then-exploit approach.

Theorem 7. ***Pseudo-regret lower bound.*** *There exists a model* $\boldsymbol{\theta} = \{\theta^1, ..., \theta^k\}$ *and a distribution of players* $p_1, ..., p_N$ *such that the pseudo-regret with respect to the deterministic optimal policy* $\pi_{\boldsymbol{\theta}}^*$ *of any exploration algorithm that outputs an* ϵ*-approximation of each arm* θ^k *with probability at least* $1 - 1/T$ *and which is followed by the optimal policy using the estimated model is at least:*

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

Theorem 7 reveals the difficulty of the studied problem in comparison to the multi-armed bandit and multi-player bandit problems. Indeed, in the case of bandit, the pseudo-regret lower bound of *explore-then-exploit* algorithms is in $\Omega(\sqrt{KT \log T})$ [30], and in the case of multi-player bandit, there exists an *explore-then-exploit* algorithm with a regret upper bound in $O(K\sqrt{T \log T})$ [12]. The difference in power of T of the pseudo-regret lower

bounds of bandits and massively multi-player bandits is due to the fact that, in the studied problem, the whole model $\boldsymbol{\theta}$ is needed to compute the optimal policy, and not only the N best arms: when the exploration stops, there is no guarantee that the arms are sufficiently sampled to compute the optimal policy without mistakes of assignment of players over arms. The independence of K of the pseudo-regret lower bound of massively multi-player bandits is due to the fact that, at each time step, K players can sample the arms. Finally, the pseudo-regret lower and upper bounds are tight in T, since the pseudo-regret upper bound of 3 followed by the policy $\pi^*_{\hat{\boldsymbol{\theta}}}$ reaches the pseudo-regret lower bound (Theorem 6).

Theorem 8. ***Fairness.*** *Applying Algorithm 3 followed by* `DOFG` *(Algorithm 2) on $\hat{\boldsymbol{\theta}}$ returns with a probability $1-\delta$ an α-fair policy in the true model $\boldsymbol{\theta}$, with*

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

Theorem 8 implies that, using ϵ-approximations of arms, with high probability `DOFG` still has the same fairness guarantee minus a term that decreases with ϵ.

4.4. Experiments on simulated environment

In order to illustrate and complete the analysis of the aforementioned algorithms, we first compare the performance of *collaborative exploration* (Algorithm 3) with *selfish exploration*, where each player explores selfishly, and with *follow-the-leader exploration (FtL)*, where only the most frequent player explores. Then we compare *collaborative exploration* followed by `DORG`$(\hat{\boldsymbol{\theta}})$ and `DOFG`$(\hat{\boldsymbol{\theta}})$, with *selfish UCB* [5] and *selfish EXP3* [31], which respectively consist in independently playing *UCB* and *EXP3* on each player, and with CBAIMPB [19], where the players find $(\epsilon', m)-$ optimal arms and exploit them uniformly with $m = 5, \epsilon' = 0.2$. We run simulations with various values of N, and $K = 10$, such that $\forall k, \theta^k \sim \mathcal{U}(0,1)$. The distribution of players is uniform and the upper bound of the distribution is chosen such that the internal collision rate does not exceed 0.15 when the number of players reaches 1300 and play the arms uniformly, so $\forall n, p_n \sim \mathcal{U}(3.10^{-4}, 2.2.10^{-3})$. $\delta = 0.05$, $\epsilon = 0.1$. The curves are averaged over 10 trials and run on 10^6 time steps.

In figure 2a, we observe that the exploration time of *collaborative exploration* is two orders of magnitude less than *follow-the-leader exploration* and three orders of magnitude less than *selfish exploration* but one order of magnitude more than *CBAIMPB*, which stops exploration when it finds the best arms. Concerning the communication cost, we observe that the communication cost of the *collaborative exploration* is only one order of magnitude greater than other exploration algorithms, however it is more than two times less than the upper bound stated in Theorem 4, which is in the order of $O\left(NK \log \frac{NK+N}{\delta}\right)$. This is due to the fact that the stopping condition of Algorithm 3 does not imply that all players have been sampled enough, but that the arms have been sampled enough. As a consequence, all the estimations of all players do not need to be shared, but only those of players that have finished their estimations.

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

Concerning fairness, `DOFG`$(\hat{\boldsymbol{\theta}})$ clearly outperforms *selfish UCB1*, *selfish Exp3* and `DORG`$(\hat{\boldsymbol{\theta}})$, while `DORG`$(\hat{\boldsymbol{\theta}})$ is outperformed by them when N is high (Figure 2e). *CBAIMPB* offers a high fairness between players due to the uniform selection of the arms by all players during both exploration and exploitation phases. The use of *selfish exploration* leads to high fairness level due to its very long uni-

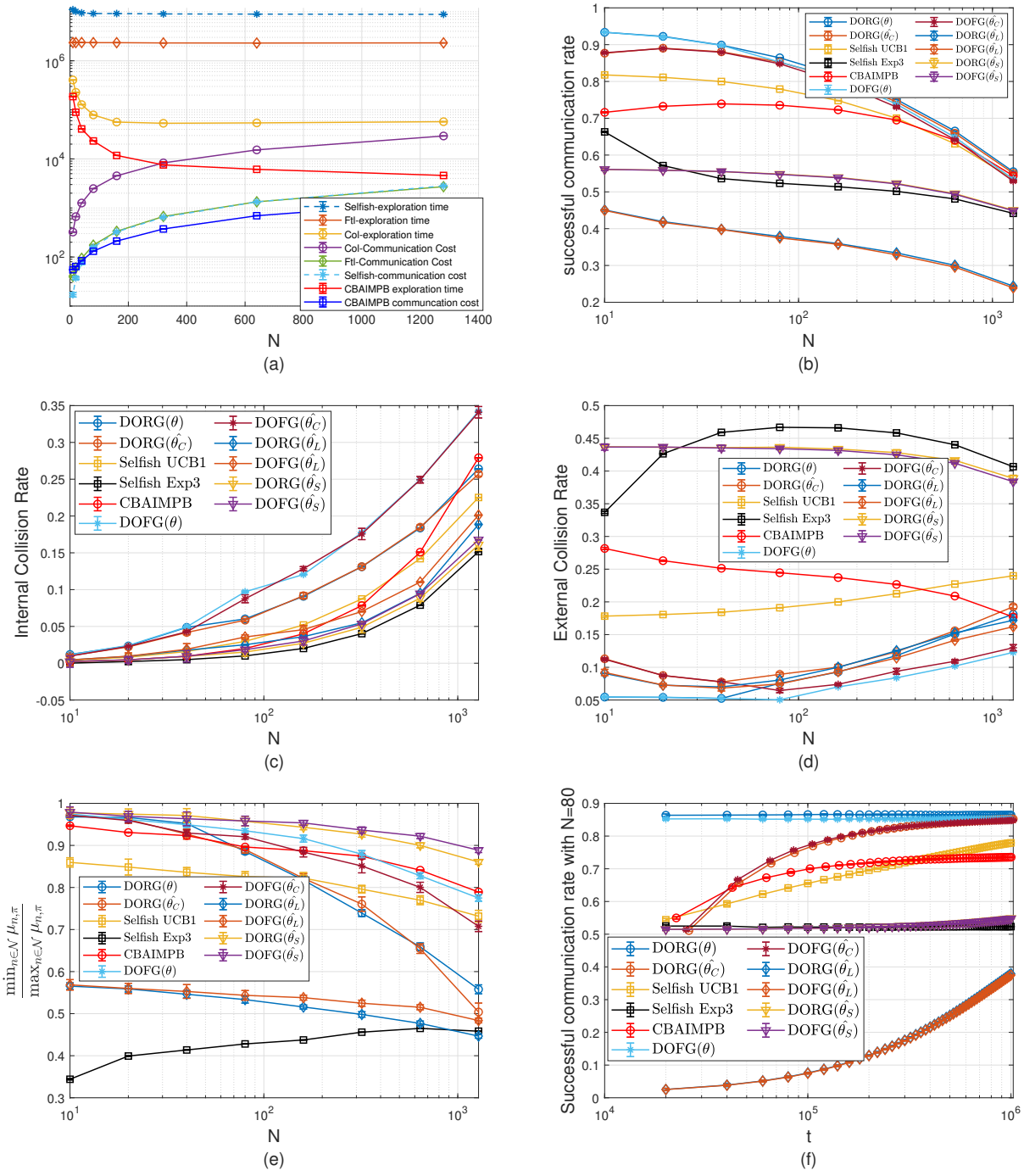


Figure 2: (a) exploration phase, (b) successful communication rate, (c) internal collision rate, (d) external collision rate, (e) fairness, (f) successful communication rate versus time. The successful communication and collision rates are cumulative over time. $\hat{\theta}_{\mathbf{C}}$ when *collaborative exploration* is used, $\hat{\theta}_{\mathbf{S}}$ when *selfish exploration* is used, and $\hat{\theta}_{\mathbf{L}}$ when *follow-the-leader exploration* is used. θ is the ground truth.

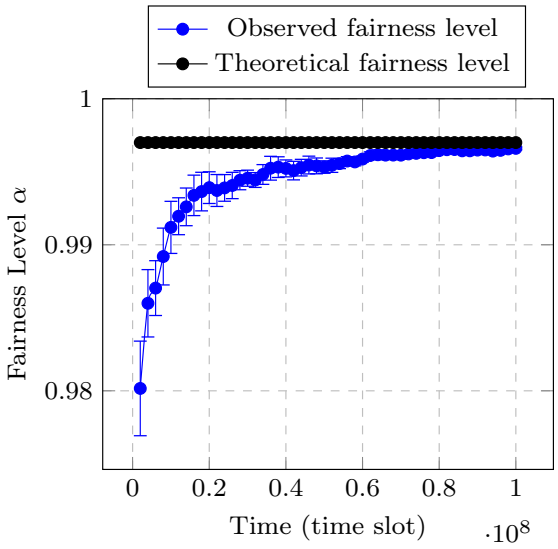


Figure 3: Fairness level achieved by DOFG(θ) as a function of time with 10 players.

form exploration phase, in contrast to *follow-the-leader exploration* that suffers of very low fairness level due to the fact that, during the exploration time, only the leader can send messages.

The observed fairness of DOFG($\boldsymbol{\theta}$) in figure 2e differs from the theoretical one (Theorem 3). This is due to the fact that the mean rewards of players are observed on a finite number of time slots (10^6). Figure 3 shows the progress of the fairness level achieved by DOFG(θ) policy as time passes. The black plot corresponds to the theoretical fairness level proved in Theorem 3. In order to reach the theoretical fairness level, the observed mean rewards of all players have to reach their expected values. Due to the low probabilities of sending packets of the players, this would take a long time. As shown by figure 3, the observed fairness tends to the theoretical fairness in 10^8 times steps for 10 players.

In the next section, we propose an adaptive modelling of the LoRa communications such that we can apply DORG and DOFG to optimize not only packet delivery but also to minimize the energy consumption.

5. Application to LoRa Network

5.1. LoRaWAN Technology

LoRaWAN is a LPWAN protocol designed to optimize LPWANs for battery lifetime, capacity, range, and cost. LoRaWAN follows a pure-ALOHA

Table 1: Spreading factors and corresponding SNR required for ADR [32], antenna sensitivities [33], and Inter-SF collision threshold [34]

SF	Required SNR for ADR (dB)	LoRa gateway antenna sensitivity(dBm)	Inter-SF collision threshold (dB)
SF7	-7.5	-123	-7.5
SF8	-10	-126	-9
SF9	-12.5	-139	-13.5
SF10	-15	-132	-15
SF11	-17.5	-134.5	-18
SF12	-20	-137	-22.5

principle and is basically a single-hop technology that relays messages from the nodes to the central server via gateways. It is based on the chirp spread spectrum modulation technique, that supports 6 orthogonal spreading factors corresponding to 6 different data rates: SF7 (50 kbps) to SF12 (300 bps). The different SFs are orthogonal, allowing simultaneous transmissions of multiple frames with different SFs. At each transmission, the node selects the communication parameters including the spreading factor, radio channel, and the transmitting power that varies between 2 dBm and 14 dBm. The higher the SF (i.e., the lower the data rate and the slower the transmission), the longer the communication range. Consequently, the choice of an SF can be seen as a trade-off between coverage and message duration (and thus, energy consumption) [35].

As any IoT technology, LoRa is bonded with many constraints, including the maximum duty cycle, which defines a maximum percentage of time during which an end-device can occupy a channel. Therefore, LoRaWAN nodes follow a pseudo-random channel hopping at each transmission while meeting the duty-cycle constraint which is 1% in EU 868 for example. The resulting frequency diversity makes the system more robust to interference. The choice of the spreading factor as well as the transmitting power is done at the gateway using the so-called Adaptive Data Rate (ADR) algorithm. ADR compromises between energy consumption and packet loss depending on the past performance of each end node. It was established for stationary end nodes and stable radio channel environments [36].

5.2. Adaptive Data Rate Algorithm

An Adaptive Data Rate (ADR) mechanism is built into LoRaWAN for dynamically managing each end node's link parameters in order to increase the packet delivery ratio and to decrease energy consumption. It is only suitable for static devices and should not be applied on mobile devices since the radio channel changes dramatically with every frame. We hereby present a simple baseline way to implement this decision mechanism recommended by **Semtech** [37].Its performance has been evaluated in [38]. This algorithm in its present form is limited to EU868 Industrial Scientific and Medical bands, and to 6 data rates (SF12/125kHz to SF7/125kHz). The ADR mechanism adjusts the data rate (SF) and the transmitting power of an end device based on the values of the Signal to Noise Ratio (SNR) of the last 20 transmissions (i.e., for each transmission, it considers the maximum of the various SNRs reported by the different gateways who received this given frame) following the steps below:

- an SNR margin is calculated such that:

  $\mathrm{SNR}_{\mathrm{margin}} = \mathrm{SNR}_{\mathrm{max}} - \mathrm{SNR(SF)} - \mathrm{margin\_db}$,

  where:

  - $\mathbf{SNR}_{\mathrm{max}}$ is the maximum SNR value among the last 20 received packets,
  - **margin_db** is the installation margin of the network which is a device specific static parameter, It is typically 10 dB in most networks [37],
  - **SNR(SF)** is the required SNR to successfully demodulate a frame, and is a function of the SF of the end-device's last received frame and presented in table 1.

- $\mathrm{N}_{\mathrm{step}} := \mathit{round}(\ \mathrm{SNR}_{\mathrm{margin}}/3)$ is calculated to determine the number of steps to perform:

  - if $\mathrm{N}_{\mathrm{step}}$ is negative (i.e. SNRs are low), the transmitting power is incremented by $3 \times \mathrm{N}_{\mathrm{step}}$ dBm,
  - if $\mathrm{N}_{\mathrm{step}}$ is positive (i.e. SNRs are high), SF is decreased by $\mathrm{N}_{\mathrm{step}}$, in order to decrease the time-on-air and save energy, if SF7 is reached and there are still steps remaining, then the transmitting power is decreased by 3 dBm for each remaining step until the minimum power (2 dBm) is reached.

The end-device has also the possibility to manage its transmit parameters itself by making use of ADR mechanism that resides at the end-device side. If the end-device does not receive any downlink frame from the gateway for a certain number of sent packets, it must try to regain connectivity by first stepping up the transmit power to default power (i.e., the max power 14 dBm). It must further lower its data rate (increase the SF) step by step every predefined number of sent packets until it reaches the lowest data rate (i.e., SF12) [3].

Notice that ADR is a heuristic and is not based on any optimization objective: it increases and decreases SF and transmitting power depending on the SNR values. It also treats each device individually regardless of other devices in the network. In this work, we contrarily aim to optimize the global network capacity by adapting massively multi-player multi-armed bandits for handling the trade-off between energy consumption and packet losses. We compare the performance of the ADR algorithm with different multi-armed bandit algorithms using a LoRa network simulator presented below.

5.3. LoRa Network Simulator

For our simulations we extended a realistic LoRa network simulator [39] and adapt it to our settings. It is described below.

Network Operation: By default, LoRa devices use pure ALOHA for transmissions. However, due to the need of synchronized nodes and referring to [40] that shows that slotted-ALOHA (where a device can only transmit data in the start of a time slot) outperforms pure-ALOHA in terms of packet error rate, throughput, collision, and energy consumption, we propose that the devices transmit according to the slotted-ALOHA protocol. Each node n transmits at the beginning of a time slot with a fixed probability p_n. The time slot is of a configurable duration that together with p_n respect a duty cycle of 1%. We consider devices of class A, which after each uplink transmission open two short reception windows in order to receive a downlink transmission from the gateway as an acknowledgement of their uplink transmission reception at the gateway. The devices always receive an acknowledgement if their uplink transmission is successful. In case of a packet loss, an end-device n retransmits its packet in

the next time slots with a probability $p_n^\dagger > p_n$ whose value depends on the application. The maximum possible number of retransmissions is configurable and depends on the device (we consider 8 maximum retransmissions in the simulator).

Transmission Success and Collision Rules: The success or failure of a transmission mainly depends on two important metrics: the Received Signal Strength Indicator (RSSI) which characterizes the power level of a received radio signal, and the Signal to Noise Ratio (SNR). A packet is successfully received by a gateway if it does not collide with any other packets, and if its RSSI is strictly greater than the antenna sensitivity. The antenna sensitivity depends on the SF of the sent transmission as reported in table 1. A collision may occur when two or more packets sent on the same radio channel are received simultaneously. There are two types of collisions:

- Intra-SF collisions: occurs when the colliding packets (packet a and packet b) are of the same SF. The packet with the highest power will be decoded if it is at least 6 dB higher than the other LoRa packets: $\text{RSSI}_a - \text{RSSI}_b \geq 6$ dB.
- Inter-SF collisions: occurs when the colliding packets are of different SFs ($\text{SF}_a \neq \text{SF}_b$). The packet is demodulated if the power difference is strictly greater than the inter-SF collision threshold which depends on the SF of the corresponding frame (see table 1): packet "a" is demodulated if: $\text{RSSI}_a - \text{RSSI}_b > \text{Thr}(\text{SF}_a)$.

Propagation Model: Propagation is modeled by the universal Okumura-Hata model, which is an accurate and widely used propagation model for predicting path loss in urban areas. Adaptations to rural and suburban areas are also added as recommended by ETSI for GSM 900 MHz [41]. This model takes into account the effects of diffraction, reflection and scattering caused by city structures. It is generally used for frequency ranges of 150 MHz to 1500 MHz, for a link distance varying from 1 km to 20 km and for antenna heights varying from 30 m to 200 m and from 1 m to 10 m for the transmitter and the base station antenna respectively [42]. Typical indoor penetration losses are considered (18 dB, 15 dB, 12 dB and 10 dB for dense urban, urban, suburban and rural environments respectively) along with additional 6 dB loss for deep indoor environments [43, 44].

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

5.4. Optimizing LoRa Communications using Massive Multi-Player Multi-Armed Bandit

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

$$\xi_{\alpha,q}(e^k) = (1 - \alpha e^k)^q. \tag{5}$$

$\xi_{\alpha,q}(e^k)$ is a decreasing function of the energy consumption e^k. The parameters $\alpha \in [0, 1)$ and $q \geq 1$ allow to shape it, depending on the energy consumption of arms (table 2).

Note that we focus here on the energy consumed due to transmissions rather than that due to policies computation, since the energy consumed by the transmission hardware, including power amplifiers and RF circuitry, far exceeds the energy required for such simple computations [45].

Table 2: The normalized energy consumption per arm e^k, where the colors from blue to red correspond to the values from low to high

TP (in dBm)	SF7	SF8	SF9	SF10	SF11	SF12
2	0.003	0.005	0.009	0.016	0.032	0.063
5	0.005	0.009	0.018	0.031	0.063	0.126
8	0.01	0.018	0.037	0.063	0.126	0.251
11	0.021	0.037	0.073	0.125	0.251	0.501
14	0.042	0.073	0.146	0.25	0.5	1

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

- $E^k \in \{0,1\}$ denotes the event *'no external collision occurs'* (intra-SF or inter-SF collision with an unknown node),
- $I_n^k \in \{0,1\}$ denotes the event *'no internal collision occurs for node n'*(intra-SF or inter-SF collision with an known node),
- $D_n^k \in \{0,1\}$ denotes the event *'no decoding error occurs'* (RSSI lower than the antenna sensitivity).

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

$$T_n^k = E^k I_n^k D_n^k. \tag{6}$$

To handle both energy consumption and packet delivery we combine equations (5) and (6) in the reward function of node n playing arm k below:

$$R_n^k(\alpha, q) = (1 - \alpha e^k)^q T_n^k. \tag{7}$$

To handle packet delivery, the used propagation model takes into account all conditions impacting it. Inter-SF or intra-SF collisions may occur even if the transmissions are not performed using the same parameters (SF, TP). Moreover, the propagation model introduces a decoding error, which depends on the topography, the position of the node, and the position of the gateway. Notice that this realistic propagation model violates two assumptions made by the theoretical model described in section 2: the channels are orthogonal, and the arms are the same for all players. Moreover, the re-transmissions are not taken into account in the utility function (equation 2), and hence in the target policies `DORG` and `DOFG`. Finally, we did not modify the LoRa protocol for including an optional 8 bytes overhead for exchanging messages between players. We simply consider the messages between players as a regular transmissions. Despite there is a significant gap between the theoretical model and the true model, in the next section we will see that Massively Multi-Player Multi-Armed Bandits is a competitive candidate for choosing the connection parameters of LoRa transmissions in order to minimize the energy consumption while ensuring high reliability.

5.5. Experimental results

Experimental setup: For our simulations, we consider a network operating in the LoRa European band $863 - 870$ MHz. We consider only one gateway and assume all transmissions are done on one frequency channel (868 MHz). The network configuration and input parameters are summarized in table 3. We consider the worst case of a deep indoor LoRa network in an urban city. The frame size is 11 bytes (4 bytes of payload for the consumption index and 7 bytes Zigbee Cluster Library application protocol overhead) [39] corresponding to a smart metering application. We consider a set of $N = 400$ end nodes where each node n has a fixed probability p_n to send a packet at the beginning of a time slot. The distribution of the nodes is uniformly chosen such that $\forall n, p_n \sim \mathcal{U}(7.10^{-4}, 5.10^{-3})$. We consider the maximum number of transmissions = 8. In case of a packet loss of any node n, it will increase its probability to send packets to $p_n^\dagger = p_n \times 8$ in order to be able to retransmit it before a new packet is needed to be sent. The communication parameters of the retransmissions are chosen according to the policy the nodes follow. In such settings, we compare the performances of ADR algorithm [32], *selfish* UCB [18], *selfish* Exp3 [46], which is a commonly-used algorithm in non-stochastic environments, CBAIMPB [19], and *collaborative exploration* followed by `DORG` or `DOFG`.

Due to the very slow increase of energy values near 0 and very fast increase near 1 as shown in table 2, we set the parameters of the penalty function (7) to $\alpha = 0.5$ and $q = 4$. Although `DORG` and `DOFG` assume that the mean rewards of the arms are the same for all the nodes which necessitates that all nodes be located at the same distance

Table 3: The network configuration and input parameters

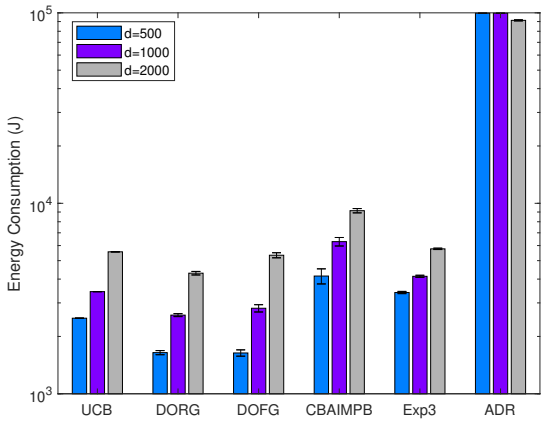
Channel Frequency	868 MHz
Bandwidth	125 kHz
Number of Gateways	1
Gateway noise figure	3 dB
Gateway antenna gain	5 dBi
Indoor penetration loss	15 dB
Additional deep indoor loss	6 dB
Gateway antenna height	30 m
End-device height	1.5 m
End-device antenna gain	0 dBi
Targeted C/N after despreading	6 dB

from the gateway, we consider here that the nodes are uniformly distributed in the hexagonal cell region centered by the gateway. We consider 3 different inter-sight distances $d = \{500, 1000, 2000\}$. For each trial, 5.10^5 packets are sent by the nodes. The figures present the averaged values over 40 trials with 95% confidence intervals. We perform two different experiments, each considering different external traffics.

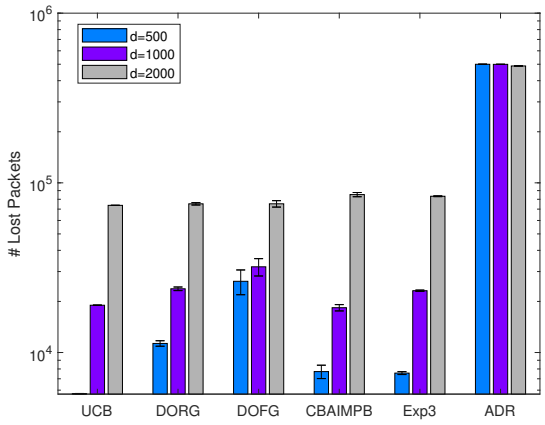
Experiment 1:

In the first experiment, to simulate external traffic, we consider $S = 200$ static devices located in the same area, each sends packets with a fixed probability $p = 0.01$. These external nodes elect an arm k for each transmission with a probability $l^k \sim \mathcal{U}(0, 1)$, such that $\sum_{k=1}^{K} l^k = 1$, which makes the environment stationary. Notice however that for selfish Exp3 or selfish UCB, which does not take into account other nodes, the environment cannot be considered as stationary, since the internal nodes can change arm and hence due to the collisions the reward function (7) evolves during time.

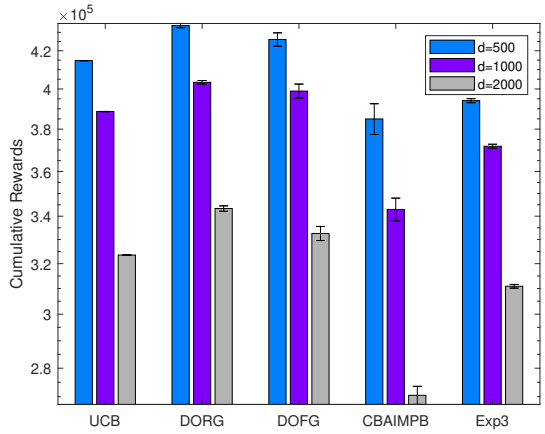
In figure 4 we present the average values of the total energy consumed by the end nodes, the total number of lost packets and the total sum of rewards gained by the end-devices. It clearly shows that the nodes when implementing the ADR algorithm suffer of very high energy consumption and packet loss compared to the learning methods with any inter-site distance. This directly leads to greater sum of rewards for all the learning methods, and implies that MAB algorithms guarantee better management of the trade-off between energy consumption and packet loss, and provides a better QoS.



(a) Energy consumption



(b) Total number of lost packets



(c) The total rewards of the network

Figure 4: **Experiment 1**: Performance of the LoRa network with end nodes distributed in hexagonal areas centered by the gateway with three different radii and external nodes following a fixed policy

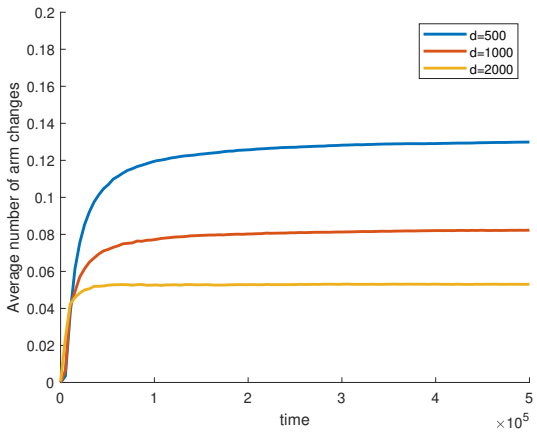


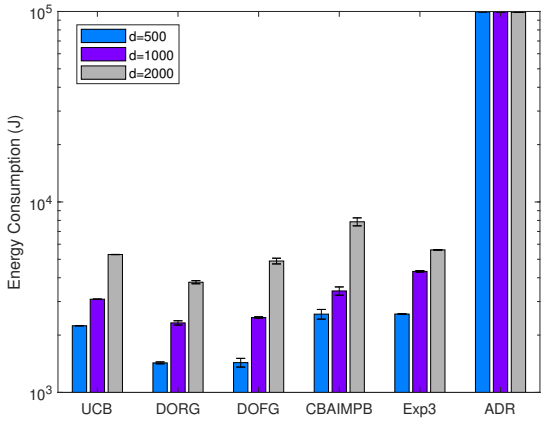
Figure 5: Average number of arm changes with respect to the number of plays

Despite there still being a gap between the theoretical model and the true model, `DORG` and `DOFG` largely outperform ADR in terms of energy consumption and packet losses and outperform UCB by compromising energy consumption and packet loss (see figure 4), while the latter shows to be highly robust against collisions. We also notice that *selfish* UCB outperform *selfish* Exp3 even though the stationary assumption is violated.

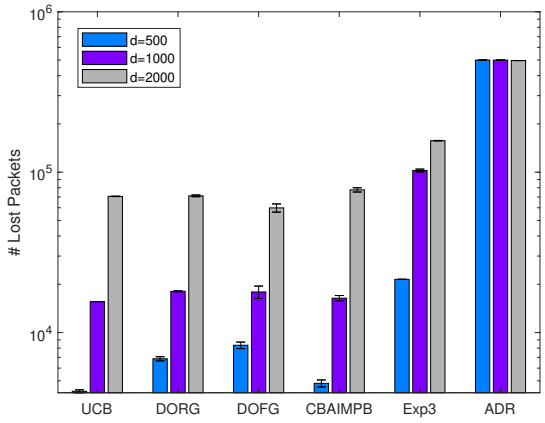
Experiment 2:

In this experiment, we consider that the external nodes are LoRa devices that follow the ADR mechanism. Due to ADR mechanism, the external nodes can change an arm at each time step. This introduces a non-stationarity even for the collaborative algorithms `DORG` and `DOFG`: the percentage of ADR nodes that change their arms tends to the order of 8% (figure 5).

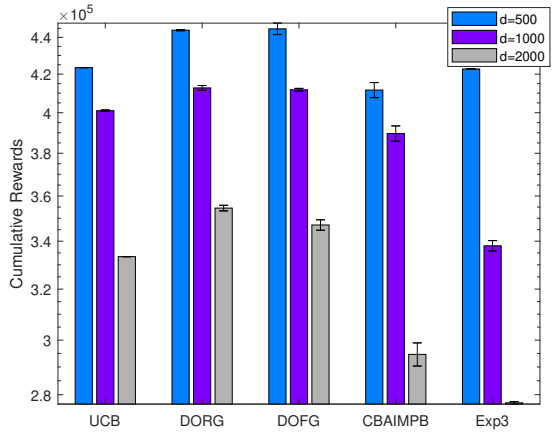
Notice that despite the non-stationary environment, the results are very similar to those in the previous experiment: all MAB algorithms outperform ADR, and our developed algorithms outperform other state-of-the-art MAB algorithms (figure 6). This experiment reveals that if there exist some nodes that does not follow our collaborative algorithms but ADR, they will lose in terms of delivery rate, while consuming more energy. Finally, notice that the *explore-then-exploit* algorithms `DORG` and `DOFG` are more appropriate for low-complexity devices (used in IoT networks) than classic selfish MAB algorithms, since after the exploration phase ends no computation takes place at the device side, while using MAB algorithms the devices keep com-



(a) Energy consumption



(b) Total number of lost packets



(c) The total rewards of the network

Figure 6: **Experiment 2**: Performance of the LoRa network with end nodes distributed in hexagonal areas centered by the gateway with three different radii and external nodes following ADR algorithm

puting confidence bounds or distributions to find the next arm to select.

6. Conclusion

We tackled the problem of optimizing transmissions in IoT networks. To do so, we modeled our problem as a massively multi-player multi-armed bandit problem, and proposed two policies `DORG` and `DOFG` that are efficient with any number of players, and can handle internal and external collisions without *sensing*. We then tested our algorithms on LoRa networks by replacing the ADR algorithm with our developed algorithms to manage the trade off between the energy consumption and the packet loss by selecting the spreading factor and the transmitting power of the transmissions. Using a LoRa simulator that meets the LoRaWAN standards, we experimentally showed that the multi-player MABs outperform the standard ADR algorithm by managing the trade off between the energy consumption and packet loss and achieving high reduction of both metrics at different distances from the gateway.

Regarding future research directions, we plan to adjust the two `DORG` and `DOFG` policies so they take into account the different mean rewards of the arms between the players and non-orthogonal channels, and consider the case of non-stationary environments. Also, in this work we considered a slotted-ALOHA transmission protocol where nodes send at the beginning of the fixed-duration time slots. But, since the time-on-air of packets varies (depending on the selected spreading factor), considering slotted-ALOHA necessitates long-duration time slots which decreases the performance by creating more collisions. Future works could overcome this by considering sub-slotting: one time slot can be divided into several sub-slots of durations that depend on the time-on-air of the transmission (1 sub-slot for SF12, 2 sub-slots for SF11, 4 for SF10,..etc.).

References

[1] D. Newman, "Return on iot: Dealing with the iot skills gap," 2019. Available online: https://www.forbes.com/sites/danielnewman/2019/07/30/return-on-iot-dealing-with-the-iot-skills-gap/27017efb7091 (accessed on 16 June 2021).

[2] B. S. Chaudhari, M. Zennaro, and S. Borkar, "LPWAN technologies: Emerging application characteristics, requirements, and design considerations," *Future Internet*, vol. 12, no. 3, p. 46, 2020.

[3] "LoRa alliance," Available online: https://www.lora-alliance.org (last accessed on 17 June 2021).

[4] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Machine learning*, vol. 47, no. 2-3, pp. 235–256, 2002.

[5] R. Bonnefoi, L. Besson, C. Moy, E. Kaufmann, and J. Palicot, "Multi-armed bandit learning in IoT networks: Learning helps even in non-stationary settings," in *International Conference on Cognitive Radio Oriented Wireless Networks*, pp. 173–185, Springer, 2017.

[6] K. Liu and Q. Zhao, "Distributed learning in multi-armed bandit with multiple players," *IEEE Transactions on Signal Processing*, vol. 58, no. 11, pp. 5667–5681, 2010.

[7] A. Anandkumar, N. Michael, and A. Tang, "Opportunistic spectrum access with multiple users: Learning under competition," in *Proceedings IEEE INFOCOM*, 2010.

[8] O. Avner and S. Mannor, "Concurrent bandits and cognitive radio networks," in *ECML PKDD*, (Berlin, Heidelberg), Springer-Verlag, 2014.

[9] N. Nayyar, D. Katathil, and R. Jain, "On regret-optimal learning in decentralized multi-player multi-armed bandits," in *EEE Transactions on Control of Network Systems*, 2015.

[10] J. Rosenski, O. Shamir, and L. Szlak, "Multi-player bandits–a musical chairs approach," in *International Conference on Machine Learning*, pp. 155–163, PMLR, 2016.

[11] M. K. Hanawal and S. J. Darak, "Multi-player bandits: A trekking approach," *arXiv preprint arXiv:1809.06040*, 2018.

[12] E. Boursier and V. Perchet, "Sic-mmab: Synchronisation involves communication in multiplayer multi-armed bandits," in *Advances in Neural Information Processing Systems 32*, pp. 12048–12057, 2019.

[13] P.-A. Wang, A. Proutiere, K. Ariu, Y. Jedra, and A. Russo, "Optimal algorithms for multiplayer multi-armed bandits," in *International Conference on Artificial Intelligence and Statistics*, pp. 4120–4129, PMLR, 2020.

[14] E. Boursier, V. Perchet, E. Kaufmann, and A. Mehrabian, "A practical algorithm for multiplayer bandits when arm means vary among players," in *AISTATS*, 2020.

[15] R. Kerkouche, R. Alami, R. Féraud, N. Varsier, and P. Maillé, "Node-based optimization of LoRa transmissions with multi-armed bandit algorithms," in *25th International Conference on Telecommunications (ICT)*, 2018.

[16] A. Valach and D. Macko, "Upper confidence bound based communication parameters selection to improve scalability of lora@fiit communication," *IEEE Sensors Journal*, vol. 22, no. 12, pp. 12415–12427, 2022.

[17] H. Dakdouk, E. Tarazona, R. Alami, R. Féraud, G. Z. Papadopoulos, and P. Maillé, "Reinforcement learning techniques for optimized channel hopping in ieee 802.15.4-tsch networks," in *Proceedings of the 21st ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, MSWIM '18, 2018.

[18] L. Besson and E. Kaufmann, "Multi-player bandits revisited," in *Proceedings of Algorithmic Learning Theory*, vol. 83, pp. 56–92, 2018.

[19] H. Dakdouk, R. Féraud, N. Varsier, and P. Maillé,

"Collaborative exploration in stochastic multi-player bandits," in *Asian Conference on Machine Learning*, pp. 193–208, PMLR, 2020.

[20] D. P. Bertsekas, R. G. Gallager, and P. Humblet, *Data networks*, vol. 2. Prentice-Hall International New Jersey, 1992.

[21] X. Wang and K. Kar, "Distributed algorithms for max-min fair rate allocation in aloha networks," in *Proceedings of the 42nd Annual Allerton Conference*, Citeseer, 2004.

[22] K. Cohen, A. Leshem, and E. Zehavi, "Game theoretic aspects of the multi-channel aloha protocol in cognitive radio networks," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 11, pp. 2276–2288, 2013.

[23] M. Anthony and P. L. Bartlett, *Neural Network Learning: Theoretical Foundations*. USA: Cambridge University Press, 1st ed., 1999.

[24] O. Perešíni and T. Krajčovič, "More efficient iot communication through lora network with lora@fiit and stiot protocols," in *2017 IEEE 11th International Conference on Application of Information and Communication Technologies (AICT)*, pp. 1–6, 2017.

[25] A. Augustin, J. Yi, T. Clausen, and W. M. Townsley, "A study of lora: Long range & low power networks for the internet of things," *Sensors*, vol. 16, no. 9, p. 1466, 2016.

[26] N. Sornin and A. Yegin, "Lorawan$^{\text{tm}}$ specification, v1.0.3," July 2018.

[27] S. Bubeck and N. Cesa-Bianchi, "Regret analysis of stochastic and nonstochastic multi-armed bandit problems," *arXiv preprint arXiv:1204.5721*, 2012.

[28] R. Féraud, R. Alami, and R. Laroche, "Decentralized exploration in multi-armed bandits," in *ICML*, 2019.

[29] C. Shi and C. Shen, "Federated multi-armed bandits," in *Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI)*, 2021.

[30] A. Garivier, T. Lattimore, and E. Kaufmann, "On explore-then-commit strategies," in *Advances in Neural Information Processing Systems*, 2016.

[31] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire, "The nonstochastic multiarmed bandit problem," *SIAM journal on computing*, vol. 32, no. 1, pp. 48–77, 2002.

[32] S. Ghoslya, "How does LoRaWAN adaptive data rate work?," http://www.sghoslya.com/p/how-does-lorawan-nodes-changes-their.html,last accessed June, 23rd 2021.

[33] Semtech, "Understanding the LoRa adaptive data rate," available on: semtech.com/LoRa, December 2019.

[34] A. Waret, M. Kaneko, A. Guitton, and N. El Rachkidy, "Lora throughput analysis with imperfect spreading factor orthogonality," *IEEE Wireless Communications Letters*, vol. 8, no. 2, pp. 408–411, 2018.

[35] F. Adelantado, X. Vilajosana, P. Tuset-Peiro, B. Martinez, J. Melia-Segui, and T. Watteyne, "Understanding the limits of LoRaWAN," *IEEE Communications magazine*, vol. 55, no. 9, pp. 34–40, 2017.

[36] R. Kufakunesu, G. P. Hancke, and A. M. Abu-Mahfouz, "A survey on adaptive data rate optimization in LoRaWAN: Recent solutions and major challenges," *Sensors*, vol. 20, no. 18, p. 5044, 2020.

[37] Semtech Corporation, "Lorawan – simple rate adaptation recommended algorithm." [Online; accessed 25-November-2021].

[38] R. Marini, W. Cerroni, and C. Buratti, "A novel collision-aware adaptive data rate algorithm for lorawan networks," *IEEE Internet of Things Journal*, vol. 8, no. 4, pp. 2670–2680, 2021.

[39] N. Varsier and J. Schwoerer, "Capacity limits of lorawan technology for smart metering applications," in *2017 IEEE international conference on communications (ICC)*, pp. 1–6, IEEE, 2017.

[40] Z. Ali, S. Henna, A. Akhunzada, M. Raza, and S. W. Kim, "Performance evaluation of LoRaWAN for green internet of things," *IEEE Access*, vol. 7, pp. 164102–164112, 2019.

[41] T. ETSI, "Digital cellular telecommunications system (phase 2+); radio network planning aspects (3gpp tr 03.30 version 8.4.0 release 1999)," *ETSI TR 101 362 V8.4.0*, June 2005.

[42] A. Deme, D. Dajab, M. Buba Bajoga, and D. Choji, "Hata-okumura model computer analysis for path loss determination at 900mhz for maiduguri, nigeria," *Mathematical Theory and Modeling*, vol. 3, no. 3, pp. 1–9, 2013.

[43] I. Rodriguez, H. C. Nguyen, N. T. Jørgensen, T. B. Sørensen, J. Elling, M. B. Gentsch, and P. Mogensen, "Path loss validation for urban micro cell scenarios at 3.5 ghz compared to 1.9 ghz," in *IEEE global communications conference (GLOBECOM)*, pp. 3942–3947, 2013.

[44] L. Ferreira, M. Kuipers, C. Rodrigues, and L. M. Correia, "Characterisation of signal penetration into buildings for GSM and UMTS," in *3rd International Symposium on Wireless Communication Systems*, pp. 63–67, IEEE, 2006.

[45] R. W. Leland, R. Murphy, B. A. Hendrickson, K. Yelick, J. Johnson, and J. Berry, "Large-scale data analytics and its relationship to simulation.," tech. rep., Sandia National Lab.(SNL-NM), Albuquerque, NM (United States), 2016.

[46] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire, "The nonstochastic multiarmed bandit problem," *SIAM journal on computing*, vol. 32, no. 1, pp. 48–77, 2002.

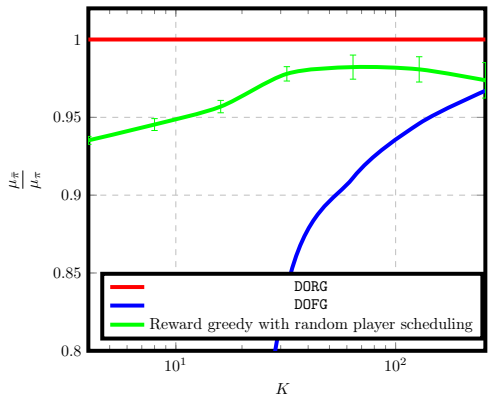
Appendix .1. Broader Impact

Optimizing the communications in IoT networks has a clear *positive environmental impact*. Indeed, when the number of collisions decreases, obviously the amount of wasted energy also decreases. Moreover, IoT devices often work on batteries, and minimizing the wasted energy increases the lifetime of batteries, which reduces the amount of batteries that need to be recycled. The decrease of the number of collisions is done thanks to the cooperation between players. In this work, we develop the concept of *fairness between players*, which is a necessary condition of cooperation. We believe that providing a mathematical framework to guarantee the fairness and then to favor cooperation is a necessity in our world where more and more automatic devices equipped with machine learning algorithms exchange information. This work is a first step in this direction.

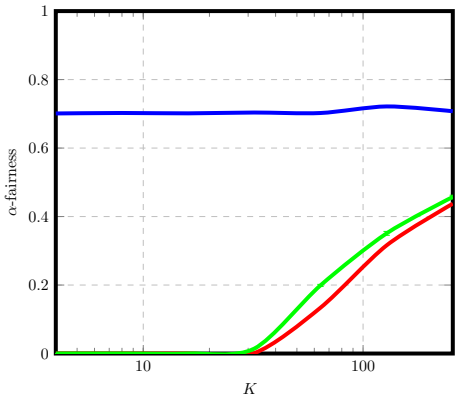
In a real life implementation of this work, to take care about ethical consideration, we will need also to take into account that the purposes of the devices is not the same. Some of them could have significant packets to transmit, for instance for heath care and emergency purpose. The fairness has to be weighted by the purpose of the devices. Finally, in a real life application the system has to be protected against malicious players that may lie about its probability of being active or about the rewards of a channel for bypassing the fairness constraint of algorithms. We believe that this issue can be fixed by the gateway that can check the consistency of the observed rewards and probabilities of each player's activity.

Appendix .2. Preliminary experiments

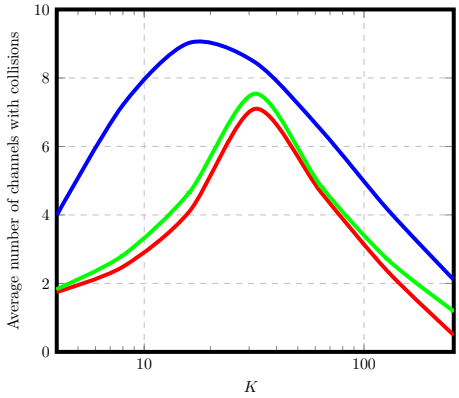
Similar to the experiments in section 3.3, figures with $N = 200$ and K ranging from 4 to 256 on a log scale are available below.



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

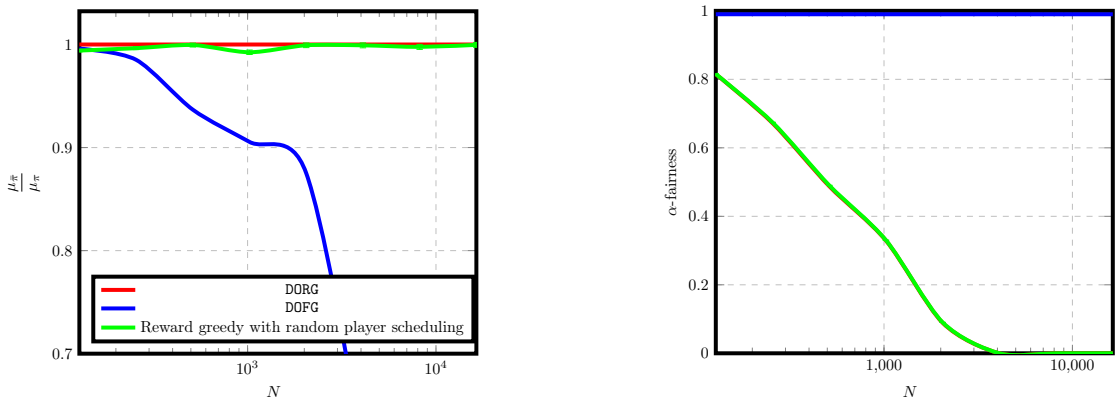


(b) Level of fairness between players



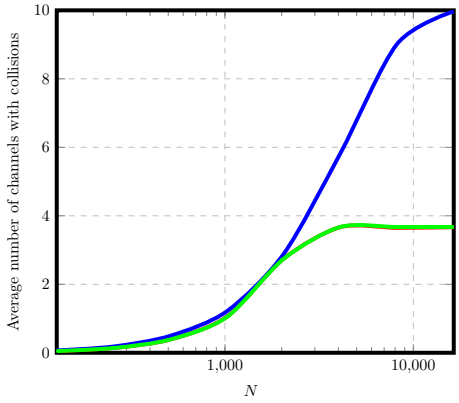
(c) Number of channels with internal collisions

Figure .7: With a fixed number of players $N = 200$, and for K values (ranging from 4 to 256 on a log scale), the performance of `DORG`, `DOFG`, and Reward Greedy (Algorithm 1) with random ordering is compared.



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

(b) Level of fairness between players



(c) Number of channels with internal collisions

Figure .8: With a fixed number of arms $K = 10$, and for N values (ranging from 128 to 16384 on a log scale), the performance of `DORG`, `DOFG`, and Reward Greedy (Algorithm 1) with random ordering is compared.

Appendix .3. Proofs

Appendix .3.1. Notations

For the sake of ease the reading of proofs, we provide below the notations.

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

Appendix .3.2. Proof of Theorem 1

There exists an optimal policy which is deterministic.

Proof. We may write the global objective as:

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

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

Appendix .3.3. Proof of Lemma 1

For a deterministic policy π, let $\mu_{\boldsymbol{\theta}}(\pi[n])$ denote the aggregated expected reward when only the players $1, ..., n$ are playing (all players $n' > n$ are deactivated). Then we have the recursive expression

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

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

Proof. We have:

$$\begin{aligned}
\mu_{\boldsymbol{\theta}}(\pi[n]) &= \mu_{\boldsymbol{\theta}}(\pi[n-1]) + \mu_{\boldsymbol{\theta}}(\pi[n]) - \mu_{\boldsymbol{\theta}}(\pi[n-1]) \\
&= \mu_{\boldsymbol{\theta}}(\pi[n-1]) + \sum_{k\in[K]} \theta^k z_{[n]}^k \ell_{[n]}^k - \sum_{k\in[K]} \theta^k z_{[n-1]}^k \ell_{[n-1]}^k \\
&= \mu_{\boldsymbol{\theta}}(\pi[n-1]) + \theta^{k_n} z_{[n]}^{k_n} \ell_{[n]}^{k_n} - \theta^{k_n} z_{[n-1]}^{k_n} \ell_{[n-1]}^{k_n} \qquad (.2) \\
&= \mu_{\boldsymbol{\theta}}(\pi[n-1]) + \theta^{k_n} \left(z_{[n]}^{k_n} \ell_{[n]}^{k_n} - z_{[n-1]}^{k_n} \ell_{[n-1]}^{k_n}\right) \\
&= \mu_{\boldsymbol{\theta}}(\pi[n-1]) + \theta^{k_n} \left((1-p_n) z_{[n-1]}^{k_n} \left(\ell_{[n-1]}^{k_n} + \frac{p_n}{1-p_n}\right) - z_{[n-1]}^{k_n} \ell_{[n-1]}^{k_n}\right) \\
&= \mu_{\boldsymbol{\theta}}(\pi[n-1]) + \theta^{k_n} \left(-p_n z_{[n-1]}^{k_n} \ell_{[n-1]}^{k_n} + p_n z_{[n-1]}^{k_n}\right) \\
&= \mu_{\boldsymbol{\theta}}(\pi[n-1]) + p_n \theta^{k_n} z_{[n-1]}^{k_n} \left(1 - \ell_{[n-1]}^{k_n}\right),
\end{aligned}$$

where the line (.2) comes from the fact that $z_{[n]}^k = z_{[n-1]}^k$ and $\ell_{[n]}^k = \ell_{[n-1]}^k$ for all $k \neq k_n$. □

Appendix .3.4. Proof of Theorem 2

Lemma 3. *As long as $\ell_{n-1}^k \leq 2$, the reward-greedy criterion for Algorithm 1 decreases as we add a new player n:*

$$z_{[n]}^k \left(1 - \ell_{[n]}^k\right) \leq z_{[n-1]}^k \left(1 - \ell_{[n-1]}^k\right). \qquad (.3)$$

Proof. We look at the difference:

$$\begin{aligned}
\forall k \neq k_n, \quad z_{[n]}^k \left(1 - \ell_{[n]}^k\right) - z_{[n-1]}^k \left(1 - \ell_{[n-1]}^k\right) &= 0 \qquad (.4) \\
z_{[n]}^{k_n} \left(1 - \ell_{[n]}^{k_n}\right) - z_{[n-1]}^{k_n} \left(1 - \ell_{[n-1]}^{k_n}\right) &= (1-p_n) z_{[n-1]}^{k_n} \left(1 - \ell_{[n-1]}^{k_n} - \frac{p_n}{1-p_n}\right) \\
&\quad - z_{[n-1]}^{k_n} \left(1 - \ell_{[n-1]}^{k_n}\right) \qquad (.5) \\
&= (1-p_n) z_{[n-1]}^{k_n} \left(1 - \ell_{[n-1]}^{k_n}\right) - p_n z_{[n-1]}^{k_n} \\
&\quad - z_{[n-1]}^{k_n} \left(1 - \ell_{[n-1]}^{k_n}\right) \qquad (.6) \\
&= -p_n z_{[n-1]}^{k_n} \left(1 - \ell_{[n-1]}^{k_n}\right) - p_n z_{[n-1]}^{k_n} \qquad (.7) \\
&= -p_n z_{[n-1]}^{k_n} \left(2 - \ell_{[n-1]}^{k_n}\right) \qquad (.8)
\end{aligned}$$

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

□

Theorem 2: *If $\sum_{n\in[N]} \frac{p_n}{1-p_n} \leq K+1$, then, there exists an ordering over players $\sigma^* : [N] \to [N]$ such that Algorithm 1 returns an optimal policy.*

Proof. The proof makes use of Lemma 3 which states that, as long as $\ell_{n-1}^k \leq 2$, the reward-greedy criterion for Algorithm 1 decreases as we add a new player n.

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

Let an optimal policy π^* be given, and let us construct the player ordering σ^* such that Algorithm 1 applied on the σ^* ordering returns π^*.

Algorithm 4 Reconstruction of a player ordering that allows Algorithm 1 to return π^*

Inputs: $[K]$, $[N]$, $\{\theta^k\}_{k\in[K]}$, $\{p_n\}_{n\in[N]}$, π^*
Output: σ^* such that Algorithm 1 returns π^*
Init: per-arm inactivity probabilities: $z^k = 1$.
Init: per-arm activation odds sums: $\ell^k = 0$.
Init: Set of players remaining to be assigned: $\mathcal{N} = [N]$.
1: **for** $n = 1$ to N **do**
2: Let $\sigma^*(n)$ be an element of $\mathcal{N}$ such that $\pi^*(\sigma^*(n)) \in \arg\max_{k\in[K]} \theta^k z^k(1-\ell^k)$.
3: Update $\mathcal{N} \leftarrow \mathcal{N} - \{\sigma^*(n)\}$.
4: Update $z^{k_n} \leftarrow z^{k_n}\left(1-p_{\sigma^*(n)}\right)$.
5: Update $\ell^{k_n} \leftarrow \ell^{k_n} + \frac{p_{\sigma^*(n)}}{1-p_{\sigma^*(n)}}$.
6: **end for**

It is direct to understand that Algorithm 1 applied on a σ^* player ordering would retrieve π^*. Indeed, Algorithm 4 makes it so the players are ordered to be incrementally optimal. The last piece of the proof is to check the existence of a player $\sigma^*(n)$ assigned to a reward-greedy arm on line 2.

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

We may therefore conclude the proof by stating that Algorithm 4 will never fail to construct σ^* and that Algorithm 1 applied to the σ^* player ordering will return π^*. □

Appendix .3.5. Proof of Theorem 3

`DOFG` generates α-fair policies, with

$$\alpha \geq 1 - \max_{n\in[N]} p_n. \tag{.9}$$

Proof. Let $\pi^\dagger$ be the policy generated by `DOFG`. For every arm, we have the following equality:

$$\mu_{n,\boldsymbol{\theta}}(\pi^\dagger) = \theta^{k_n} \prod_{n'\neq n,\ \text{s.t. }\ k_{n'}=k_n} (1-p_{n'}) = \frac{\theta^{k_n} z^{k_n}}{1-p_n}. \tag{.10}$$

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

Case $k_N = k_{N+1}$, then from Equation .10, we have $\mu_{N+1,\boldsymbol{\theta}}(\pi^\dagger) = \frac{1-p_N}{1-p_{N+1}}\mu_{N,,\boldsymbol{\theta}}(\pi^\dagger)$. Since we know by construction that $p_{N+1} \leq p_N$, we may conclude that $\mu_{N+1,\boldsymbol{\theta}}(\pi^\dagger) \leq \mu_{N,\boldsymbol{\theta}}(\pi^\dagger)$.

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

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

$$\min_{n\in[N]} \mu_{n,\boldsymbol{\theta}}(\pi^\dagger) = \mu_{N,\boldsymbol{\theta}}(\pi^\dagger) \geq \max_{k\in[K]} \theta^k z^k. \tag{.11}$$

We also know that:

$$\max_{n\in[N]} \mu_{n,\boldsymbol{\theta}}(\pi^\dagger) = \max_{n\in[N]} \frac{\theta^{k_n} z^{k_n}}{1-p_n} \tag{.12}$$

$$\leq \frac{\max_{k\in[K]} \theta^k z^k}{1-\max_{n\in[N]} p_n} \tag{.13}$$

$$\leq \frac{1}{1-p_1} \min_{n\in[N]} \mu_{n,\boldsymbol{\theta}}(\pi^\dagger), \tag{.14}$$

which concludes the demonstration. □

Appendix .3.6. Proof of Lemma 2

By using Algorithm 3, in order to obtain with a probability $1-\delta$ an ϵ-approximation of the mean rewards of arms, player n needs to sample each arm at least

$$t^*_n = \left\lceil \frac{p_n \log(2K/\delta)}{2\epsilon^2(\prod_{n'\neq N}(1-p_{n'}/K))^2 \sum_{i=1}^N p_i} \right\rceil \text{ times.}$$

Proof. Due to equations 1 and 4, for a given probability of failure $\delta \in [0,1]$, and a given approximation factor ϵ, $\forall n \in [N]$, $\forall k \in [K]$ we have:

$$P(|\mu^k - \hat{\mu}^k_n| \geq \epsilon) \leq \frac{\delta}{K} \iff P(|\theta^k - \hat{\theta}^k_n| \geq \epsilon'_n) \leq \frac{\delta}{K}, \tag{.15}$$

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

Applying Hoeffding's inequality:

$$P(|\theta^k_n - \hat{\theta}^k_n| \geq \epsilon'_n) \leq 2e^{-2t^k_n {\epsilon'_n}^2}. \tag{.16}$$

Therefore for obtaining an ϵ-approximation of arm k on player n with a probability $1-\frac{\delta}{K}$:

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

Now, as Algorithm 3 shares the estimations of the N players for finding ϵ-approximation of arm k with high probability, we need $\sum_{n=1}^N t^*_n = t^\dagger$ samples.

Hence, if each player samples arm k at least $t^*_n \geq \lceil \frac{p_n \log(2K/\delta)}{2\epsilon^2(\prod_{n'\neq N}(1-p_{n'}/K))^2 \sum_{i=1}^N p_i} \rceil$ times, an ϵ-approximation of arm θ^k is obtained with a probability $1-\frac{\delta}{K}$.

□

Appendix .3.7. Proof of Theorem 4

Lemma 4. *In Algorithm 3, so that player n sends successfully m messages, with a probability $1-\delta$ player n needs to issue a number of transmissions $C(m)$, which is at most:*

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

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

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

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

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

□

Theorem 4 *When Algorithm 3 stops, the number of messages sent is, with probability $1-\delta$, less than $C(N(1+2K))$, where*

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

Proof. The required number of messages to send during Algorithm 3 is at most $N(1+2K)$. Using Lemma 4, the total number of transmissions done by all players to send successfully their messages is with probability $1-\delta$:

$$C(N(1+2K))\leq N(1+2K)\left\lceil\frac{\log\delta/(N(1+2K))}{\log(1-\sum_{k=1}^{K}\frac{(1-p_1/K)^{N-1}}{K}\theta^k)}+1\right\rceil \tag{.17}$$

□

Appendix .3.8. Proof of Theorem 5

With a probability at least $1-\delta$, when $N\geq 3$, Algorithm 3 stops while finding the ϵ-approximations of θ at:

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

where p_N is the lowest probability of sending a packet among the players, and $C(3)$ is the needed number of transmissions to successfully send 3 messages.

Proof. A player n stops, while finding its estimations with high probability, when it plays each arm k at least t_n^* times (Lemma 2). Let t_n^k be the number of plays of arm k by player n before the algorithm stops at time t^* with high probability. t_n^k is a binomial random variable with parameters t^* and p_n/K. Then we have:

$$\mathbb{E}[t_n^k] = \frac{p_n}{K}.t^* \tag{.18}$$

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

Applying Hoeffding's inequality we get:

$$\mathcal{P}(t_n^k - \frac{p_n}{K}.t^* < -\epsilon) \leq \exp^{-2\frac{\epsilon^2}{t^*}} = \frac{\delta}{NK}. \tag{.19}$$

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

$$t_n^* + C(3) - \frac{p_n}{K}.t^* < -\sqrt{\frac{t^*}{2}\log\frac{NK}{\delta}}, \tag{.20}$$

$$\Leftrightarrow -\frac{p_n}{K}.t^* + \sqrt{\frac{t^*}{2}\log\frac{NK}{\delta}} + t_n^* + C(3) < 0, \tag{.21}$$

$$\Leftrightarrow \sqrt{t^*} > \frac{K}{2p_n}\left(\sqrt{\frac{1}{2}\log\frac{NK}{\delta}} + \sqrt{\frac{1}{2}\log\frac{NK}{\delta} + 4\frac{p_n}{K}(t_n^* + C(3))}\right), \tag{.22}$$

$$\Leftrightarrow t^* > \frac{K^2}{4(p_n)^2}\left(\sqrt{\frac{1}{2}\log\frac{NK}{\delta}} + \sqrt{\frac{1}{2}\log\frac{NK}{\delta} + 4\frac{p_n}{K}(t_n^* + C(3))}\right)^2, \tag{.23}$$

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

$$t^* \leq \frac{K^2}{4(p_n)^2}\left(\sqrt{\frac{1}{2}\log\frac{NK}{\delta}} + \sqrt{\frac{1}{2}\log\frac{NK}{\delta} + 4\frac{p_n}{K}(t_n^* + C(3))}\right)^2, \tag{.24}$$

$$\Leftrightarrow t^* \leq \frac{K^2}{4(p_n)^2}\log\frac{NK}{\delta} + \frac{K(t_n^* + C(3))}{p_n} + \frac{K^2}{2(p_n)^2}\sqrt{\frac{1}{2}\log\frac{NK}{\delta}}\sqrt{\frac{1}{2}\log\frac{NK}{\delta} + 4\frac{p_n}{K}(t_n^* + C(3))}, \tag{.25}$$

$$\Rightarrow t^* \leq \frac{K^2}{4(p_n)^2}\log\frac{NK}{\delta} + \frac{K(t_n^* + C(3))}{p_n} + \frac{K^2}{4(p_n)^2}\log\frac{NK}{\delta} + \frac{K}{p_n}\sqrt{\frac{K}{2p_n}(t_n^* + C(3))\log\frac{NK}{\delta}}, \tag{.26}$$

$$\Rightarrow t^* \leq \frac{K}{p_n}\left(t_n^* + C(3) + \sqrt{\frac{K}{2p_n}(t_n^* + C(3))\log\frac{NK}{\delta}}\right) + \frac{K^2}{2(p_n)^2}\log\frac{NK}{\delta}. \tag{.27}$$

Then using Lemma 2, the following inequality holds with a probability at least $1-\delta$:

$$t^* \leq \frac{K}{p_n}\left(\frac{p_n \log(2K/\delta)}{2\epsilon^2(\prod_{n'\neq N}(1-p_{n'}/K))^2\sum_{i=1}^N p_i} + C(3) + \sqrt{\frac{K}{2p_n}\left(\frac{p_n \log(2K/\delta)}{2\epsilon^2(\prod_{n'\neq N}(1-p_{n'}/K))^2\sum_{i=1}^N p_i} + C(3)\right)\log\frac{NK}{\delta}}\right) + \frac{K^2}{2(p_n)^2}\log\frac{NK}{\delta}, \tag{.28}$$

$$t^* \leq \frac{K}{p_n}\left(\frac{p_n \log(NK/\delta)}{2\epsilon^2(\prod_{n'\neq N}(1-p_{n'}/K))^2\sum_{i=1}^N p_i} + C(3) + \sqrt{\frac{K}{2p_n}}\frac{p_n \log(NK/\delta)}{2\epsilon^2(\prod_{n'\neq N}(1-p_{n'}/K))^2\sum_{i=1}^N p_i} + \sqrt{\frac{K}{2p_n}C(3)\log(NK/\delta)}\right) + \frac{K^2}{2(p_n)^2}\log\frac{NK}{\delta}, \tag{.29}$$

$$t^* \leq \frac{K\log(NK/\delta)}{2\epsilon^2((1-p_1/K))^{2N-2}\sum_{i=1}^N p_i}\left(1+\sqrt{\frac{K}{2p_N}}\right) + \frac{K^2}{2(p_N)^2}\log\frac{NK}{\delta} + \left(\frac{K}{p_N}\right)^{3/2}\sqrt{\frac{C(3)}{2}\log\frac{NK}{\delta}} + \frac{KC(3)}{p_N} \tag{.30}$$

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

□

Appendix .3.9. Proof of Theorem 6

Lemma 5. *The expected instantaneous regret in the model $\boldsymbol{\theta}$ of the target policy $\pi^*_{\hat{\boldsymbol{\theta}}}$ using the estimated model $\hat{\boldsymbol{\theta}}$ with respect to the optimal policy $\pi^*_{\boldsymbol{\theta}}$ using the true model $\boldsymbol{\theta}$ is upper bounded by:*

$$\mu_{\boldsymbol{\theta}}(\pi^*_{\boldsymbol{\theta}}) - \mu_{\boldsymbol{\theta}}(\pi^*_{\hat{\boldsymbol{\theta}}}) \leq 2K\epsilon, \tag{.31}$$

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

Proof.

$$\mu_{\boldsymbol{\theta}}(\pi^*_{\boldsymbol{\theta}}) - \mu_{\boldsymbol{\theta}}(\pi^*_{\hat{\boldsymbol{\theta}}}) = \mu_{\boldsymbol{\theta}}(\pi^*) - \mu_{\hat{\boldsymbol{\theta}}}(\pi^*) + \mu_{\hat{\boldsymbol{\theta}}}(\pi^*) - \mu_{\hat{\boldsymbol{\theta}}}(\pi^*_{\hat{\boldsymbol{\theta}}}) + \mu_{\hat{\boldsymbol{\theta}}}(\pi^*_{\hat{\boldsymbol{\theta}}}) - \mu_{\boldsymbol{\theta}}(\pi^*_{\hat{\boldsymbol{\theta}}}) \tag{.32}$$

Then, we have:

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

□

Theorem 6: *When $N \geq 3$, and $\forall n \in [N], p_n = p$, the pseudo-regret with respect to the target policy π^* of Algorithm 3 followed by a policy $\pi^*_{\hat{\boldsymbol{\theta}}}$ is upper bounded by:*

$$R(T) \leq O\left(\frac{T^{2/3}\log NKT}{p^{3/2}(1-p/K)^{2N-2}N} + K^{9/4}T^{2/3}\right).$$

Proof. Let T be the time horizon, π_u be the uniform policy used in Algorithm 3, which outputs an ϵ-approximation with high probability of $\boldsymbol{\theta}$, and $\pi^*_{\boldsymbol{\theta}}$ be the optimal policy. Let t^* be stopping time of the

exploration phase. Then, the pseudo-regret with respect to a target policy $\pi^*_{\boldsymbol{\theta}}$ of Algorithm 3 is expressed as:

$$R(T) = t^*((\mu_{\boldsymbol{\theta}}(\pi^*_{\boldsymbol{\theta}}) - (\mu_{\boldsymbol{\theta}}(\pi_u)) + (T - t^*)((\mu_{\boldsymbol{\theta}}(\pi^*_{\boldsymbol{\theta}}) - \mu_{\boldsymbol{\theta}}(\pi^*_{\hat{\boldsymbol{\theta}}})), \tag{.33}$$

where $\mu_{\boldsymbol{\theta}}(\pi^*_{\hat{\boldsymbol{\theta}}})$ denotes the mean reward in the model $\boldsymbol{\theta}$ of the optimal policy using the estimated model $\hat{\boldsymbol{\theta}}$. The left term of equation .33 is the instantaneous pseudo-regret of the exploration policy $\pi_{\mathcal{U}}$, and the right term is the instantaneous pseudo-regret of the estimated optimal policy $\pi^*_{\hat{\boldsymbol{\theta}}}$.

Theorem 5 allows us to upper-bound the stopping time of Algorithm 3 with t^* on an event of high probability $1 - \delta$:

$$t^* \leq \frac{K \log(NK/\delta)}{2\epsilon^2((1 - p_1/K))^{2N-2} \sum_{i=1}^N p_i} \left(1 + \sqrt{\frac{K}{2p_N}}\right) + \frac{K^2}{2(p_N)^2} \log \frac{NK}{\delta} + \left(\frac{K}{p_N}\right)^{3/2} \sqrt{\frac{C(3)}{2} \log \frac{NK}{\delta}} + \frac{KC(3)}{p_N}. \tag{.34}$$

When $\forall n \in [N], p_n = p$, with a probability $1 - \delta$, we have:

$$t^* \leq \frac{K \log(NK/\delta)}{2\epsilon^2(1 - p/K)^{2N-2} Np} \left(1 + \sqrt{\frac{K}{2p}}\right) + \frac{K^2}{2p^2} \log \frac{NK}{\delta} + \left(\frac{K}{p}\right)^{3/2} \sqrt{\frac{C(3)}{2} \log \frac{NK}{\delta}} + \frac{KC(3)}{p}. \tag{.35}$$

The instantaneous pseudo-regret of uniform policy with respect to the optimal policy $\pi^*_{\boldsymbol{\theta}}$ is upper bounded by:

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

and on the other hand we know by Lemma 5 that:

$$\mu_{\boldsymbol{\theta}}(\pi^*_{\boldsymbol{\theta}}) - \mu_{\boldsymbol{\theta}}(\pi^*_{\hat{\boldsymbol{\theta}}}) \leq 2K\epsilon \tag{.36}$$

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

$$R(T) \leq t^*(\mu_\theta(\pi^*_{\boldsymbol{\theta}}) - \mu_{\boldsymbol{\theta}}(\pi_u)) + (T - t^*)(\mu_{\boldsymbol{\theta}}(\pi^*_{\boldsymbol{\theta}}) - \mu_{\boldsymbol{\theta}}(\pi^*_{\hat{\boldsymbol{\theta}}})) + \delta KT \tag{.37}$$

Then, by setting $\delta = 1/T$, the pseudo-regret of Algorithm 3 followed by a policy $\pi^*_{\hat{\boldsymbol{\theta}}}$ is:

$$\begin{aligned} R(T) &\leq Kt^* + (T - t^*) \times 2K\epsilon + K, && \text{(.38)} \\ &\leq Kt^* + 2K\epsilon T + K, && \text{(.39)} \\ &\leq O\left(\frac{K^{5/2} \log NKT}{p^{3/2}\epsilon^2(1 - p/K)^{2N-2}N} + \frac{K^2}{2p^2} \log NKT + KT\epsilon\right). && \text{(.40)} \end{aligned}$$

Finally, by setting $\epsilon = K^{5/4}/\sqrt[3]{T}$, we conclude the proof:

$$R(T) \leq O\left(\frac{T^{2/3} \log NKT}{p^{3/2}(1 - p/K)^{2N-2}N} + K^{9/4}T^{2/3}\right). \tag{.41}$$

□

Appendix .3.10. Proof of Theorem 7

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

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

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

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

Choice of a class of problems.. The most difficult point for evaluating a regret lower bound is that in the general case, the optimal policy, which maximizes the mean reward (see equation (3)), is unknown. For handling this point we choose a particular class of problems, where $N = K + 1$. Then, we assume that the distribution of players and the mean rewards of arms are such that:

$$\begin{cases} \forall k \in [K-1] \quad \theta^k = \theta^{k+1} + \epsilon, \\ p_1 > p_2 = ... = p_K > p_{K+1}, \\ p_1(1 - p_{K+1}) + p_{K+1}(1 - p_1) = p_2, \\ p_2(1 - p_{K+1}) + p_{K+1}(1 - p_2) > p_2, \\ \forall k \in [K] \quad \frac{\epsilon}{2p_k} < \theta^k. \end{cases} \tag{.42}$$

The optimal policy.. When $\frac{\epsilon}{2p_k} < \theta^k$ (equation (.42)), superposing players on any arm provides less reward than spreading players on the arms. Indeed, let Δ_s be the gap between the mean reward of two players $k_1, k_2, k_1 < k_2 \leq K$ assigned on different arms, and the mean reward of two players assigned on the same arm:

$$\begin{aligned} \Delta_s &= p_{k_1}\theta^{k_1} + p_{k_2}\theta^{k_2} - p_{k_1}\theta^{k_1}(1 - p_{k_2}) - p_{k_2}\theta^{k_1}(1 - p_{k_1}), && (.43) \\ &= p_{k_2}(\theta^{k_2} - \theta^{k_1}) + 2p_{k_1}p_{k_2}\theta^{k_1}, && (.44) \\ &= -p_{k_2}\epsilon + 2p_{k_1}p_{k_2}\theta^{k_1} > 0. && (.45) \end{aligned}$$

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

$$\begin{aligned} \Delta_{1,2} &= (p_1(1 - p_{K+1}) + p_{K+1}(1 - p_1))\theta^1 + p_2\theta^2 - p_1\theta^1 - (p_2(1 - p_{K+1}) + p_{K+1}(1 - p_2))\theta^2 && (.46) \\ &= p_2\theta^1 - p_1\theta^1 + p_2\theta^2 - (p_2(1 - p_{K+1}) + p_{K+1}(1 - p_2))\theta^2 < 0 && (.47) \end{aligned}$$

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

$$\begin{aligned} \Delta_{2,k} &= (p_2(1 - p_{K+1}) + p_{K+1}(1 - p_2))\theta^2 + p_2\theta^k - p_2\theta^2 - (p_2(1 - p_{K+1}) + p_{K+1}(1 - p_2))\theta^k && (.48) \\ &= (p_2(1 - p_{K+1}) + p_{K+1}(1 - p_2))(\theta^2 - \theta^k) - p_2(\theta^2 - \theta^k) > 0 && (.49) \end{aligned}$$

Hence, when equation (.42) holds, the optimal assignment of players over arms is:

$$\pi^*_{\boldsymbol{\theta}} = (p_1, \theta^1), (p_2, p_{K+1}, \theta^2), ..., (p_{K-1}, \theta^{K-1}), (p_K, \theta^K). \tag{.50}$$

The optimal exploration policy.. As an ϵ-approximation of each arm is needed to compute the optimal policy. The optimal exploration policy plays each arm the same expected (with respect to the distribution of players $\boldsymbol{p}$) number of times. When equation (.42) holds, any optimal exploration policy belongs to the following set:

$$\pi_E^* \in \{m \in [K], \forall n \in [K] \setminus \{1\}, k \in [K] \setminus \{m\} : (p_n, \theta^k), (p_1, p_{K+1}, \theta^m)\}. \tag{.51}$$

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

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

$$R(T) = t^*(\mu_{\boldsymbol{\theta}}(\pi_{\boldsymbol{\theta}}^*) - \mu_{\boldsymbol{\theta}}(\pi_E^*)) + (T - t^*)(\mu_{\boldsymbol{\theta}}(\pi_{\boldsymbol{\theta}}^*) - \mu_{\boldsymbol{\theta}}(\pi_{\hat{\boldsymbol{\theta}}}^*)), \tag{.52}$$

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

Lower bound of the right term.. The right term equation (.52) is the instantaneous regret of the estimated optimal policy $\pi_{\hat{\boldsymbol{\theta}}}^*$. For stating a lower bound on this term, we lower bound it by the minimal gap between the optimal policy and the estimated optimal policy when a mistake in the ranking of two arms is done. As the probability of making a mistake in the estimation the model $\boldsymbol{\theta}$ is not null, it exists $c \in (0, \delta)$ such that:

$$\mu_{\boldsymbol{\theta}}(\pi_{\boldsymbol{\theta}}^*) - \mu_{\boldsymbol{\theta}}(\pi_{\hat{\boldsymbol{\theta}}}^*) \geq c \min_{k \in [K], \hat{\theta}^{k+1} > \hat{\theta}^k} \left(\mu_{\boldsymbol{\theta}}(\pi_{\boldsymbol{\theta}}^*) - \mu_{\boldsymbol{\theta}}(\pi_{\hat{\boldsymbol{\theta}}}^*)\right). \tag{.53}$$

The minimal gap, between the mean reward of the optimal policy (see equation (.50)) and a policy where an arm is not well ranked, is obtained when the ranks of arms 2 and 3 are inverted.

$$\begin{aligned}\min_{k \in [K], \hat{\theta}^{k+1} > \hat{\theta}^k} \left(\mu_{\boldsymbol{\theta}}(\pi_{\boldsymbol{\theta}}^*) - \mu_{\boldsymbol{\theta}}(\pi_{\hat{\boldsymbol{\theta}}}^*)\right) \geq (p_2(1 - p_{K+1}) + p_{K+1}(1 - p_2))\theta^2 + p_2\theta^3 \\ - p_2\theta^2 - p_2(1 - p_{K+1} + p_{K+1}(1 - p_2))\theta^3\end{aligned} \tag{.54}$$

Hence we have:

$$\mu_{\boldsymbol{\theta}}(\pi_{\boldsymbol{\theta}}^*) - \mu_{\boldsymbol{\theta}}(\pi_{\hat{\boldsymbol{\theta}}}^*) \geq c_{\boldsymbol{p}}\epsilon, \text{ where } c_{\boldsymbol{p}} > 0. \tag{.55}$$

Lower bound of the left term.. The left term of equation (.52) is the instantaneous regret of the optimal exploration policy π_E^*. The optimal exploration policy cannot be the optimal policy since estimating ϵ-approximations of arms necessitates to play the same expected number of times the arms, and hence assigning p_1 and p_{K+1} on the same arm, which is not optimal. There are three possibilities:

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

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

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

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

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

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

Hence we have:

$$\mu_{\boldsymbol{\theta}}(\pi^*_{\boldsymbol{\theta}}) - \mu_{\boldsymbol{\theta}}(\pi^*_E) \geq c_{\boldsymbol{\theta},\boldsymbol{p}}, \tag{.56}$$

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

Lower bound of the regret.. Now, injecting the lower bound of $\mu_{\boldsymbol{\theta}}(\pi^*_{\boldsymbol{\theta}}) - \mu_{\boldsymbol{\theta}}(\pi^*_E)$ (equation (.56)) and the lower bound of $\mu_{\boldsymbol{\theta}}(\pi^*_{\boldsymbol{\theta}}) - \mu_{\boldsymbol{\theta}}(\pi^*_{\hat{\boldsymbol{\theta}}})$ (equation (.55)) in the pseudo-regret decomposition (equation (.52)), we obtain:

$$\begin{aligned}R(T) &\geq t^* c_{\boldsymbol{\theta},\boldsymbol{p}} + (T - t^*)c_{\boldsymbol{p}}\epsilon, &(.57)\\ &\geq t^* c_{\boldsymbol{\theta},\boldsymbol{p}} + T\epsilon\Delta_{\boldsymbol{p}} - t^* c_{\boldsymbol{p}}\epsilon. &(.58)\end{aligned}$$

The lower bound of number of samples for finding a bias ϵ of a coin is $\Omega\left(1/\epsilon^2 \log 1/\delta\right)$ [23]. At each time step, a maximum of N players are sampled. Hence, the time t^* where π^*_E finds exactly an ϵ-approximation of each arm θ^k is at least:

$$\Omega\left(\frac{K}{N\epsilon^2}\log\frac{1}{\delta}\right) \Leftrightarrow \exists c_1 > 0, t^* = c_1\frac{K}{N\epsilon^2}\log\frac{1}{\delta}. \tag{.59}$$

We have:

$$R(T) \geq c_1 c_{\boldsymbol{\theta},\boldsymbol{p}}\frac{K}{N\epsilon^2}\log\frac{1}{\delta} + Tc_{\boldsymbol{p}}\epsilon - c_1 c_{\boldsymbol{p}}\epsilon\frac{K}{N\epsilon}\log\frac{1}{\delta}. \tag{.60}$$

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

$$E[R(T)] \geq \Omega\left(T^{2/3}\frac{\log T}{N} + T^{2/3} - \frac{K^{1/2}}{N}T^{1/3}\log T\right). \tag{.61}$$

Hence, we have:

$$E[R(T)] \geq \Omega\left(T^{2/3}\frac{\log T}{N}\right). \tag{.62}$$

□

Appendix .3.11. Proof of Theorem 8

Applying Algorithm 2 on a model estimate $\hat{\boldsymbol{\theta}}$ returns with a probability $1-\delta$ an α-fair policy in the true model $\boldsymbol{\theta}$:

$$\alpha \geq 1 - \max_{n\in[N]} p_n - \frac{2\|\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}\|_\infty}{\max_{n\in[N]}\frac{\hat{\theta}^{k_n} z^{k_n}}{1-p_n}} \tag{.63}$$

Proof. Theorem 3 states that the policy returned by Algorithm 2, denoted as $\pi^\dagger$ has the following fairness guarantees:

$$\hat{\alpha} = \frac{\min_{n\in[N]} \mu_{n,\hat{\boldsymbol{\theta}}}(\pi^\dagger)}{\max_{n\in[N]} \mu_{n,\hat{\boldsymbol{\theta}}}(\pi^\dagger)} \geq 1 - \max_{n\in[N]} p_n, \tag{.64}$$

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

$$\mu_{n,\hat{\boldsymbol{\theta}}}(\pi^\dagger) = \hat{\theta}^{k_n} \prod_{n',\text{ s.t. } k_{n'}=k_n} (1-p_{n'}) = \frac{\hat{\theta}^{k_n} z^{k_n}}{1-p_n}. \tag{.65}$$

We therefore get:

$$\begin{aligned}
\alpha &= \frac{\min_{n\in[N]} \mu_{n,\boldsymbol{\theta}}(\pi^\dagger)}{\max_{n\in[N]} \mu_{n,\boldsymbol{\theta}}(\pi^\dagger)} && (.66)\\
&= \frac{\min_{n\in[N]} \frac{\theta^{k_n} z^{k_n}}{1-p_n}}{\max_{n\in[N]} \frac{\theta^{k_n} z^{k_n}}{1-p_n}} && (.67)\\
&\geq \frac{\min_{n\in[N]} \frac{\hat{\theta}^{k_n} z^{k_n}}{1-p_n} - \|\boldsymbol{\theta}-\hat{\boldsymbol{\theta}}\|_\infty}{\max_{n\in[N]} \frac{\hat{\theta}^{k_n} z^{k_n}}{1-p_n} + \|\boldsymbol{\theta}-\hat{\boldsymbol{\theta}}\|_\infty} \qquad \text{since } \frac{z^{k_n}}{1-p_n} \leq 1, \forall n && (.68)\\
&= \hat{\alpha} - \frac{2\|\boldsymbol{\theta}-\hat{\boldsymbol{\theta}}\|_\infty}{\max_{n\in[N]} \frac{\hat{\theta}^{k_n} z^{k_n}}{1-p_n} + \|\boldsymbol{\theta}-\hat{\boldsymbol{\theta}}\|_\infty} && (.69)\\
&\geq 1 - \max_{n\in[N]} p_n - \frac{2\|\boldsymbol{\theta}-\hat{\boldsymbol{\theta}}\|_\infty}{\max_{n\in[N]} \frac{\hat{\theta}^{k_n} z^{k_n}}{1-p_n}} && (.70)
\end{aligned}$$

Now, Theorem 5 states that with a probability $1-\delta$ Algorithm 3 stops while finding ϵ-approximations of model $\boldsymbol{\theta}$. Finally, we get:

$$\alpha \geq 1 - \max_{n\in[N]} p_n - \frac{2\|\boldsymbol{\theta}-\hat{\boldsymbol{\theta}}\|_\infty}{\max_{n\in[N]} \frac{(\theta^{k_n}-\epsilon) z^{k_n}}{1-p_n}} \tag{.71}$$

□