Proceedings of Machine Learning Research 129:193–208, 2020 ACML 2020

Collaborative Exploration in Stochastic Multi-Player Bandits

Hiba Dakdouk HIBA.DAKDOUK@ORANGE.COM
Raphaël Féraud RAPHAEL.FERAUD@ORANGE.COM
Nadège Varsier NADEGE.VARSIER@ORANGE.COM
Orange Labs, France
Patrick Maillé PATRICK.MAILLE@IMT-ATLANTIQUE.FR
IMT Atlantique, France

Editors: Sinno Jialin Pan and Masashi Sugiyama

Abstract

Internet of Things (IoT) faces multiple challenges to achieve high reliability, low-latency and low power consumption. Its performance is affected by many factors such as external interference coming from other coexisting wireless communication technologies that are sharing the same spectrum. To address this problem, we introduce a general approach for the identification of poor-link quality channels. We formulate our problem as a multi-player multi-armed bandit problem, where the devices in an IoT network are the players, and the arms are the radio channels. For a realistic formulation, we do not assume that sensing information is available or that the number of players is below the number of arms. We develop and analyze a collaborative decentralized algorithm that aims to find a set of m (ϵ, m)-optimal arms using an Explore-m algorithm (as denoted by Kalyanakrishnan and Stone (2010)) as a subroutine, and hence blacklisting the suboptimal arms in order to improve the QoS of IoT networks while reducing their energy consumption. We prove analytically and experimentally that our algorithm outperforms selfish algorithms in terms of sample complexity with a low communication cost, and that although playing a smaller set of arms increases the collision rate, playing only the optimal arms improves the QoS of the network.

Keywords: Multi-player multi-armed bandits, IoT, collisions

1. Introduction

1.1. Background and Motivation

We consider the Explore-m problem in stationary multi-player multi-armed bandit settings, where the players' goal is to efficiently select a set of m arms of the highest mean rewards with as few samples as possible. We consider a set of players who share and play the same set of arms with different *active rates*, that are different probabilities to be active (a player being active means she is playing an arm). The arm rewards are independent Bernoulli-distributed random variables whose expected values are unknown to the players.

In this paper, we assume internal collisions occur if more than one player play the same arm at the same time, the reward of all colliding players then being 0. We do not consider any type of sensing, so the players cannot observe the collisions but instead only observe the outcome (1 or 0). In that case, collisions introduce biases in the observed average rewards of the arms to estimate the real averages (without collisions). This type of setting

corresponds to the problem of blacklisting radio channels in wireless communications, where the network devices aim to blacklist the radio channels with the worst qualities, and just send data through the optimal channels for high-quality communications Xue et al. (2018); Kotsiou et al. (2017); Dakdouk et al. (2018); Du and Roussos (2011). We seek to find the set of optimal channels while reducing the energy consumption of the devices.

Several Explore-m algorithms have been studied and investigated in the literature but they are applicable on single-player multi-armed bandits only. In this work, we use those algorithms as subroutines in a generalized algorithm where players collaborate to find m of the best-performing arms. We build on the ability of the players to infer the correct ranking of the arms according to their mean rewards even though they cannot observe the correct ones due to collisions. Our algorithm reduces the sample complexity compared to the selfish algorithms with a few exchanged messages between players.

1.2. Related Work

The problem of finding the best arms has been investigated thoroughly in the literature. Even-Dar et al. (2006); Audibert et al. (2010); Féraud et al. (2019); Allesiardo et al. (2017) study the case of Explore-1 that looks for one single optimal arm. In Even-Dar et al. (2006), the authors present several algorithms that aim to find the best arm with one player. With the *Naive* algorithm, the player plays each arm a specified number of times and takes the one with the highest empirical average as the optimal arm with a certain confidence level. Alternatively, the *Successive* and *Median Elimination* algorithms successively eliminate arms identified as suboptimal according to their empirical averages until only one is left, which is then labeled as the optimal one. The authors in Allesiardo et al. (2017) reformulate the multi-armed bandit problem by generalizing it to the stationary stochastic, piecewise stationary and adversarial bandit problems in order to take into account the cases where the best arm changes over time. The decentralized problem that we build on is presented in Féraud et al. (2019). In that work, a set of multiple players collaborate to find the optimal arm by asynchronously interacting with the same stochastic environment, while ensuring the privacy of players' shared information and controlling the communication cost. The authors' Decentralized Elimination algorithm uses any of the aforementioned or other Explore-1 algorithms as a subroutine, and the players share their decisions in a decentralized manner to reach a global decision regarding the optimal arm.

On the other hand, Kalyanakrishnan and Stone (2010); Kaufmann and Kalyanakrishnan (2013); Kalyanakrishnan et al. (2012); Jun and Nowak (2016) focus on the Explore-m problems. In Kalyanakrishnan and Stone (2010) the authors extend the Naive algorithm to find the m best arms and call it ***Direct*** algorithm. The ***Incremental*** algorithm uses the Median Elimination algorithm as a subroutine, and ***Halving*** modifies it to be suited to the Explore-m problem. A more powerful algorithm with a lower sample complexity called ***LUCB***, that relies on the comparison of the lower and upper confidence bounds of the empirical averages of the arms is presented in Kalyanakrishnan et al. (2012). Along with a similar algorithm (***Racing***) the authors in Kaufmann and Kalyanakrishnan (2013) propose the use of the KL-Divergence confidence bounds in these algorithms for a lower sample complexity. Although the sample complexity is lower, the computation of these confidence

bounds is complex (time and memory consuming) which makes them inappropriate for low-complexity devices (IoT devices).

Our work is built on the method presented in Féraud et al. (2019), where a collaborative, generic and decentralized algorithm is proposed to find an ϵ-approximation of the best arm, while protecting the privacy of players' information contained in their shared messages against any adversary and controlling the communication cost. However, the problem that we address in this paper is different, since we are looking for m of the best arms (up to some $\epsilon > 0$) instead of one, and privacy is not a requirement. We use this last constraint relaxation to improve the performance in terms of sample complexity. Finally here, collisions occur, which was not considered by Féraud et al. (2019).

The remainder of this paper is organized as follows. In Section 2 we introduce the m-best arm identification problem, and we present the collaborative setting we consider. Section 3 presents in details our collaborative algorithm that aims to find a set of m ϵ-optimal arms, and we provide a performance analysis by studying its sample complexity and communication cost in Section 4. We complete and illustrate the analysis of our proposed algorithm in Section 5 with some experiments, and we conclude the paper in Section 6 by suggesting directions for future work.

2. Collaborative Exploration Problem in Multi-Player Multi-Armed Bandits

Stochastic Multi-Player Multi-Armed Bandits. Let $\mathcal{N}$ be a set of N asynchronous players, such that at each time slot t each player n has a constant probability $p_n > 0$ to be active, i.e. to play an arm at time t. Equivalently, at each time step t, the set of active players $\mathcal{N}_t$ is sampled by N successive Bernoulli samples: $\mathcal{N}_t := \{n \in \mathcal{N} : a_n = 1 \text{ where } a_n \sim \mathcal{B}(p_n)\}$. Let $\mathcal{K}$ be the set of K arms, and 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 reward (without collisions) of each arm X_k is assumed to follow a Bernoulli distribution, $X_k \sim B(\theta_k)$, where θ_k is the mean reward of arm k (the quality of the arm). When two or more players play the same arm at the same time, a collision (internal collision) happens and the reward of all colliding players is 0. Let $Y_{n,k}$ be the outcome of player n after playing arm k, and C_k be a binary random variable, that takes the value 1 if a collision occurs on arm k. The assumptions we make in this paper are formalized below.

Assumption 1 ***Stationary Environment*** *The mean reward of each arm is constant over time.*

Assumption 2 ***Multi-Player Multi-Armed Bandits with Collisions*** *The reward of each player n is $Y_{n,k_n} = (1-C_{k_n})X_{k_n}$. In particular, when more than one player chooses the same arm k at the same time, a collision happens ($C_k = 1$), and the reward of all colliding players is zero.*

Assumption 3 ***No Sensing*** *Each player n can only observe her received rewards Y_{n,k_n}, but not collisions C_{k_n} nor the actual reward of the played arm X_{k_n}.*

Assumption 4 ***Large number of players*** *The number of players N can be greater than the number of arms K.*

Assumption 5 ***Socratic Players***[1] *Each player knows her own probability to be active* p_n.

Assumption 6 ***Active Players*** *All the players are potentially active:* $\forall n \in \mathcal{N}$, $p_n > 0$.

Assumption 2 corresponds to the multi-player multi-armed bandit problem with Bernoulli distributions. Assumption 3 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. Moreover, notice that Assumption 4 is unusual in multi-player bandits, where the players are generally assumed to be active at each time step and hence are assumed to be less numerous than there are channels, which is totally unrealistic for IoT networks. Assumption 5 is realistic for IoT, where the probability of sending a packet depends mainly on the type of the connected device. Assumption 6 defines a player as a device which may send a packet. Finally Assumption 1 restricts the scope of this paper to stationary stochastic multi-armed bandits.

The players' behavior will be described by a *policy*, denoted by $\pi = (\pi_1, ..., \pi_N)$, where $\pi_n = (\pi_n^1, ..., \pi_n^K)$ is the arm-choice policy of player n: π_n^k denotes the probability that player n plays arm k when active. The expected reward $\mu_n^k(\pi)$ of the active player n playing arm k, while the other players follow policy π, is then given by:

$$\mu_n^k(\pi) = \theta_k \prod_{n'=1, n' \neq n}^{N} (1 - p_{n'} \pi_{n'}^k). \tag{1}$$

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

Definition 1 (ϵ, m)***-best arms.*** *Considering that the arms are indexed in the decreasing order of their average rewards:* $\theta_1 \geq \theta_2 \geq .. \geq \theta_K$, *an arm* k *is an* (ϵ, m)*-best arm if:*

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

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

Definition 2 ***Sample Complexity.*** *For a given* $\delta \in (0, 1)$, *sample complexity is the total number of samples (or pulls) needed by all players to find a set of* m (ϵ, m)*-best arms with a confidence level* $1 - \delta$.

The goal of the collaborative exploration problem is to design an algorithm that minimizes the sample complexity to find a set of m (ϵ, m)-best arms, while controlling the number of exchanged messages between the players. Algorithm 1 formalizes the exploration problem. The players (i.e. the nodes in a network) are assumed to share some information through a single gateway (no direct node-to-node communication). By playing the arms and observing the corresponding rewards, the players decide what arms are optimal and eliminate the sub-optimal arms, then they share this information through the gateway. Although sharing information would add more cost on the players, this should help decrease the sample complexity. In the next section, we propose an algorithmic solution for exchanging information for collaborative best arms identification while reducing the sample complexity.

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

Algorithm 1 Collaborative Exploration in Multi-Player Multi-Armed Bandits

Inputs: a set of arms $\mathcal{K}$, a set of players $\mathcal{N}$, $\epsilon \in [0,1]$, $m \in \{1,...,K\}$
Output: a set $\tilde{\mathcal{K}}^n$ of m estimated (ϵ, m)-best arms for each player $n \in \mathcal{N}$
Initialization: $\forall n \in \mathcal{N}, \tilde{\mathcal{K}}^n := \mathcal{K}$
repeat
  each player $n \in \mathcal{N}$ receives messages from the gateway if any, and accordingly updates $\tilde{\mathcal{K}}^n$ (removing some arms)
  a set of players $\mathcal{N}_t$ is sampled from N successive Bernoulli samples: $\mathcal{N}_t := \{n \in \mathcal{N}, a_n \sim \mathcal{B}(p_n), a_n = 1\}$
  each player $n \in \mathcal{N}_t$ selects an arm $k_n \in \tilde{\mathcal{K}}^n$
  each player $n \in \mathcal{N}_t$ uses arm k_n to either share information about the arms with the gateway or to transmit data otherwise
  each player $n \in \mathcal{N}_t$ receives a reward $Y_{n,k_n} := (1 - C_{k_n})X_{k_n}$ (corresponding to the reception of an acknowledgment)
until $\forall n \in \mathcal{N}, |\tilde{\mathcal{K}}^n| = m$

3. Collaborative Best Arms Identification

The basic idea behind our approach is that in order to get a set of optimal arms with a low failure probability δ, each player finds a set of optimal arms but with a higher failure probability $\beta > \delta$ so the required number of samples by each player decreases. The players send to the gateway the set of arms they suggest to eliminate of the candidate m-best arms. However, the suboptimal arms are only really eliminated when at least a group of α players vote to eliminate them by sending "vote" messages, so an arm is really eliminated with the group probability of failure β^α that is needed to be equal to δ. Consequently, the required number of players to eliminate an arm should be at least $\alpha = \dfrac{\log \delta}{\log \beta}$.

3.1. Communication Protocol

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

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

The communication protocol that is used in the following is based on the same principle as ALOHA, i.e. when a collision occurs the message is resent the next time the player is active.

3.2. ArmSelection Subroutine

We will use in our proposed collaborative algorithm the Explore-m algorithms as subroutines. Those algorithms determine the players' sampling strategy of the arms, i.e. the

exploration policy. Since players cannot observe the real rewards of the played arms (because of internal collisions), we introduce a new constraint on the used subroutines.

Let $\rho_{n,k,\pi}$ be the probability that no collision happens on arm k for player n when all players follow policy π:

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

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

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

With that uniform exploration policy $\tilde{\pi}$, we have, for each player n, $\mu^k_n(\tilde{\pi}) = \theta_k \rho_{n,\tilde{\pi}}$, so from (1)

$$\theta_m - \theta_k \leq \epsilon \Leftrightarrow \mu^m_n(\tilde{\pi}) - \mu^k_n(\tilde{\pi}) \leq \rho_{n,\tilde{\pi}}.\epsilon \tag{2}$$

As (2) illustrates, each player n can use her observed values $Y_{n,k}$ to estimate μ^k_n, so as to find the set of (ϵ, m)-best arms by looking for $(\epsilon.\rho_{n,\tilde{\pi}}, m)$-best arms. But this requires the knowledge of $\rho_{n,\tilde{\pi}}$ and hence the values of the players' active rates. Therefore, we will impose that during a first phase, the players exchange their active rates by sending them to the gateway, and the latter calculates and sends the value $\epsilon' = \rho_{\tilde{\pi}}.\epsilon := \prod_{n \in \mathcal{N}} \left(1 - \frac{p_n}{K}\right).\epsilon$ to all players. When player n receives the value of ϵ', she calculates her value $\epsilon'_n = \rho_{n,\tilde{\pi}}.\epsilon := \prod_{n' \in \mathcal{N}/\{n\}} \left(1 - \frac{p_{n'}}{K}\right).\epsilon = \epsilon' / \left(1 - \frac{p_n}{K}\right)$.

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

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

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

Definition 3 ***ArmSelection subroutine*** $\mathcal{A}$. *An ArmSelection subroutine $\mathcal{A}$ is an (ϵ, m)-best arms identification algorithm that takes an approximation factor $\epsilon > 0$, a confidence level $1 - \beta < 1$ and a set of remaining arms $\mathcal{K}(l)$ as inputs. It is run by every player n: at every time slot it selects a remaining (not globally eliminated in $\mathcal{K}(l)$) arm to be played. Under specific conditions (depending on the subroutine used) it returns a set of suboptimal arms $\overline{\mathcal{K}}^n(l^n)$ locally eliminated by player n, so player n votes to eliminate them and her epoch l^n ends.*

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

Property 1 ***(remaining*** (m,ϵ)***-optimal arms)*** *For each player n, at each local elimination epoch l^n the probability that there remain less than m of the (ϵ,m)-best arms (the arms in $\mathcal{K}_{m,\epsilon}$) in $\mathcal{K}^n(l^n)$ is small. More specifically,*

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

with β the probability of failure of the used subroutine.

Property 2 ***(finite sample complexity)*** *For any confidence level $1-\beta < 1$ and approximation factor $\epsilon > 0$, the ArmSelection subroutine finds in a finite time a set of (m,ϵ)-optimal arms. Formally,*

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

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

- **The fixed-design algorithms** use uniform sampling during a predetermined number of samples, such as *Direct* algorithm in Kalyanakrishnan and Stone (2010) ($L=1$ and $f=1$) that eliminates the $k-m$ sub-optimal arms at the end of the sampling phase.
- **The successive elimination algorithms** are based on uniform sampling and arm eliminations. The arm, which cannot be an (ϵ,m)-optimal arm with a high probability, is discarded from $\mathcal{K}^n(l^n)$. *Racing* in Kaufmann and Kalyanakrishnan (2013) is a successive elimination algorithm ($L=K-m$ and $f=1/(K-m)$).
- **The explore-then-commit algorithms** are based on adaptive sampling and a stopping rule. We focus on those of uniform sampling strategies. The stopping rule simply tests if the difference, between the maximum of upper confidence bound of suboptimal arms and the lower confidence bound of the empirical best arm, is higher than the approximation factor ϵ. When the algorithm stops it eliminates the set of sub-optimal arms. *LUCB* in Kalyanakrishnan et al. (2012) is an explore-then-commit algorithm ($L=1$ and $f=1$).

3.3. Collaborative Best Arms Identification in Multi-Player Bandits

The Collaborative Best Arms Identification algorithm (see Algorithm 2) works as follows: it takes as inputs, the approximation factor ϵ, the global failure probability δ, the ArmSelection subroutine failure probability β, and the number of nearly-optimal arms to find m. Every player n will run the ArmSelection subroutine $\mathcal{A}$ with an approximation factor $\epsilon'_{n,l} = \rho_{n,l}.\epsilon = \prod_{n' \in \mathcal{N}/\{n\}} \left(1 - \frac{p_{n'}}{|\mathcal{K}(l)|}\right).\epsilon$ at global elimination epoch l in order to end up with a set of (ϵ,m)-optimal arms. It outputs a common set of m (ϵ,m)-best arms for all players. The step $ack_n := \text{send}(s,k_n)$ used in our algorithm 2 means that the message s is sent on channel k_n to the gateway, and that a binary acknowledgement is waited for a given duration. It returns the value of the acknowledgement to player n ($ack_n = 1$ if the message has been sent successfully and 0 otherwise).
The main steps of Algorithm 2 are the following:

- The players receive the gateway's messages even if they are not active and update their current sets of arms (line 2).

Algorithm 2 Collaborative Best Arms Identification in Multi-Player Bandits: CBAIMPB($\mathcal{K}$,$\mathcal{N}$,$\mathcal{A}$,ϵ,δ,β,m)

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

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

- The first time a player is active she sends her active rate to the gateway, and keeps sending it by selecting channels uniformly whenever she is active until she receives an acknowledgment (lines 10,12).
- Whenever a player n receives the value of ϵ' from the gateway, she calculates her value of $\epsilon'_{n,l}$ (line 5).
- If an active player n has sent her active rate successfully and has no new information to share with the gateway, she runs an ArmSelection subroutine with a failure probability β, and her approximation factor $\epsilon'_{n,l}$ when she is active. (line 19).
- If $\overline{\mathcal{K}}^n(l^n) \neq \emptyset$, player n keeps trying to send the indexes of the arms in $\overline{\mathcal{K}}^n(l^n)$ to the gateway until she succeeds (lines 13-17).
- If enough players want to eliminate an arm, it is eliminated from the global set of arms $\mathcal{K}(l)$ with a low probability of failure δ, and the gateway sends the updated set $\mathcal{K}(l)$ to all players (lines 24-28).
- When a player has found her set of m optimal arms while the global set of optimal has not been found yet, she is reset and restarts exploring the arms again so she can then vote as a new player (line 31).
- When there are only m arms left in $\mathcal{K}(l)$, they are (ϵ,m)-optimal arms with a high probability $1-\delta$, and the algorithm terminates (line 34).

4. Analysis of the Algorithm

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

Theorem 1 ***Low Communication Cost.*** *Using an ArmSelection subroutine with a uniform sampling exploration strategy, the total number of sent messages by algorithm CBAIMPB to find a set of (ϵ, m)-optimal arms is with a probability of failure η less than:*

$$\alpha \left\lceil \frac{\log(\frac{1-\eta/\alpha}{\sum_{k=1}^{K} \frac{\rho_n(\pi_u)}{K}\theta_k})}{\log(1-\sum_{k=1}^{K} \frac{\rho_n(\pi_u)}{K}\theta_k)} + 1 \right\rceil \text{ messages.} \tag{3}$$

For the analysis of the sample complexity of our algorithm, let $T_{\mathcal{A}}$ be the number of samples needed by the ArmSelection subroutine $\mathcal{A}$ to find a set of (ϵ', m)-best arms with probability of failure β, and T^* be the total number of samples at stopping time. Let $\mathcal{N}_S$

be the set of the $S = \lfloor \frac{\log \delta}{\log \beta} \rfloor$ most likely players, and let $p^* = \min_{n \in \mathcal{N}_S} p_n$. Theorem 2 provides the sample complexity of the algorithm CBAIMPB. This value depends on the ArmSelection subroutine used. Corollary 1 states the sample complexity of CBAIMPB when **Direct** Kalyanakrishnan and Stone (2010) algorithm is used as an ArmSelection subroutine.

Theorem 2 (Sample Complexity) *Using an ArmSelection($\beta, \delta, m, \epsilon$) subroutine with a uniform sampling exploration strategy, with a probability at least* $(1-\delta)(1 - I_{1-p*}(T^* - T_{\mathcal{A}}, 1 + T_{\mathcal{A}}))^{\lfloor \frac{\log \delta}{\log \beta} \rfloor}$ *CBAIMPB terminates after:*

$$\mathcal{O}\left(\frac{1}{p^*}\left(T_{\mathcal{A}} + \sqrt{\frac{1}{2}\log\frac{S}{\delta}}\right)\right) \textit{samples} \tag{4}$$

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

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

$$\mathcal{O}\left(\frac{1}{p^*}\left(\frac{K}{{\epsilon'_{n^\dagger}}^2}\log(\frac{K}{\beta}) + \sqrt{\frac{1}{2}\log\frac{S}{\delta}}\right)\right) \textit{samples} \tag{5}$$

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

5. Simulation Results

In order to illustrate and complete the analysis of our algorithm *CBAIMPB*, we compare its performance using the Explore-m algorithms *Direct*, *LUCB* and *Racing* as subroutines with their selfish versions. We run the algorithms with different values of N and $K = 10$, such that $\forall k, \theta^k \sim \mathcal{U}(0, 1)$. Each player n has a probability to be active p_n equal to $1/N$. We consider $\delta = 0.1$, $\beta = 0.9$, $\epsilon = 0.2$ and $m = 4$. We study the sample complexity as well as the communication cost of our algorithm with different ArmSelection subroutines. The results are averaged over 30 experiments and the figures show 95% confidence intervals.

Figure 1 (a) clearly shows that our cooperative algorithm with any ArmSelection subroutine outperforms the selfish versions of them in terms of sample complexity. Regarding the subroutines, Racing outperforms LUCB and the latter has a lower sample complexity than Direct algorithm in either the cooperative or the selfish versions. On the other hand, Racing has the highest communication cost among the three algorithms as shown in Figure 1 (b). This is because it is a successive elimination algorithm where the players eliminate one arm successively and they send one message after each elimination, while LUCB and Direct algorithms are of the explore-then-commit and fixed-design algorithms respectively and they eliminate all the suboptimal arms when the stopping condition is provided so one message is then sent.

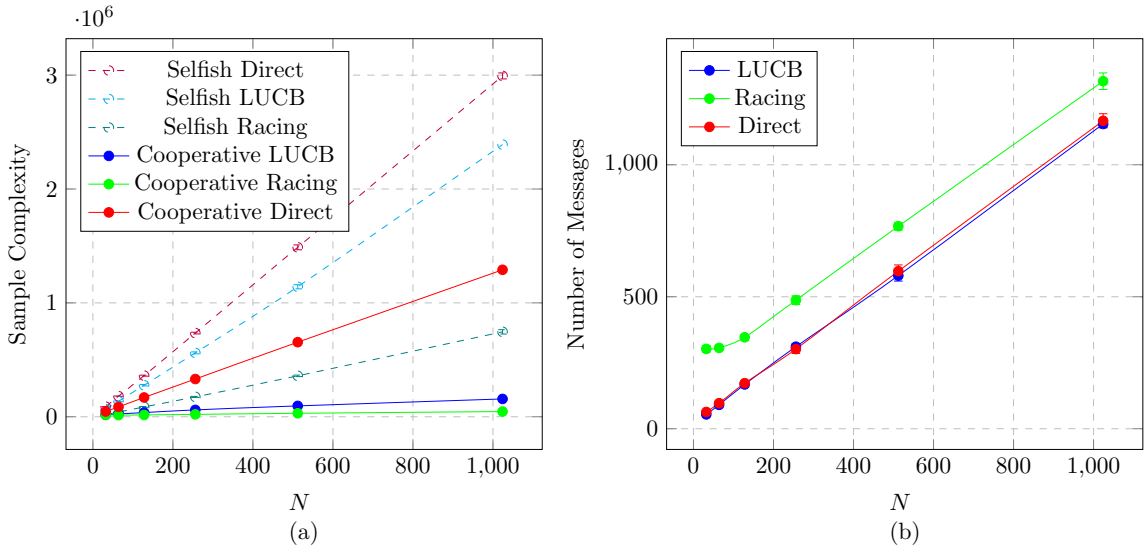

Figure 1: (a) Sample Complexity (Cooperation vs Selfishness), (b) Communication Cost as a function of the number of players N

On the other hand, after the players find the set of optimal arms, they need to exploit this set so that they increase their successful communication rate, i.e. the fraction of the successfully sent messages with respect to the total number of messages. The change in the successful communication rate depends on the value of m that should be carefully tuned and the exploitation policy the players follow. In order to study the advantage of playing a set of optimal arms instead of playing all the arms (that would increase the collision rate), we compare the successful communication rate and the collision rate of all the players achieved by the two scenarios. For simplicity, the exploitation policy we use is the uniform policy. We run the exploitation phase with various values of N, such that the distribution of players is uniform and the upper bound of the distribution is chosen such that the internal collision rate does not exceed 0.2 when the number of players reaches 1300 and play the arms uniformly, so $\forall n, p_n \sim \mathcal{U}(5.4.10^{-4}, 3.8.10^{-3})$. In **scenario 1**, the players share a set of $K = 10$ arms, such that $\forall k, \theta^k \sim \mathcal{U}(0, 1)$. In **scenario 2**, the players play a set of $(\epsilon = 0.1, m = 4)$-optimal arms of the 10 previously played arms. The exploitation phase lasts for a time horizon $T = 10^6$ time slots. The results are averaged over 30 trials and the figures show 95% confidence intervals.

Figure 2 (a) clearly shows the advantage of playing a set of optimal arms instead of playing all available arms. Although with a smaller set of arms the internal collision rate increases as shown in Figure 2 (b), but playing less number of arms of the highest qualities significantly increases the successful communication rate.

6. Conclusions and Perspectives

For the problem of *Decentralized Exploration in Multi-Player Multi-Armed Bandits*, we have designed and analyzed a new approach that aims to find a set of m optimal arms

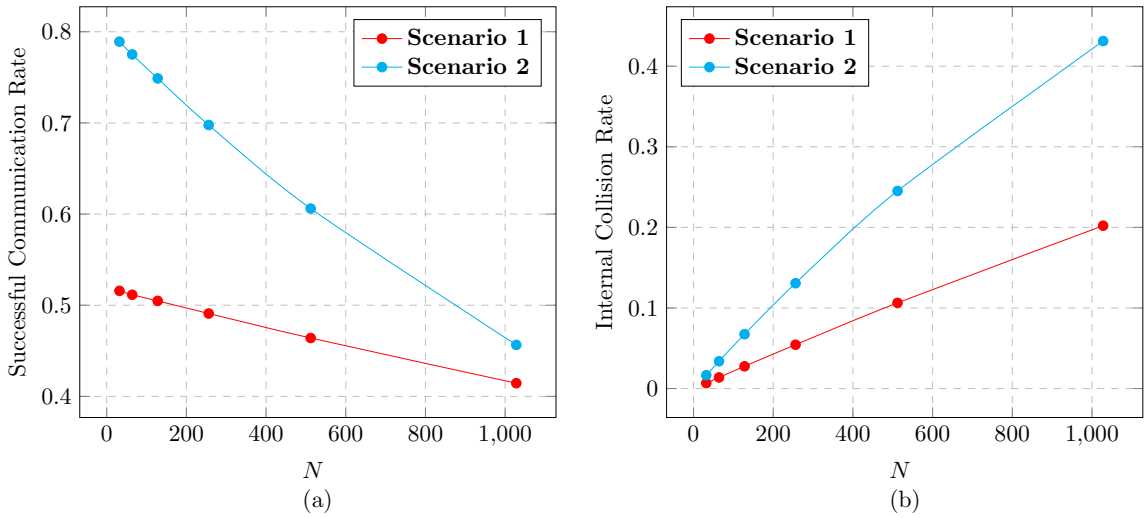

Figure 2: (a) Successful Communication Rate, (b) Internal Collision Rate as a function of the number of players N

by running Explore-m algorithms that sample the arms uniformly as subroutines. Our approach takes into account collisions between players and does not assume any type of sensing or constraints on the number of players. The players collaborate by sharing some information, and we show that the communication cost is relatively low. We have also proved experimentally that our algorithm outperforms the selfish versions in terms of sample complexity, and that although playing a smaller set of arms increases the collision rate, playing only the optimal arms increases the successful communication rate.

Our algorithm is to be tested and evaluated on IoT networks using a LoRa network simulator presented by Varsier and Schwoerer (2017), and real-world experimentation by implementing our algorithm in LoRa devices and study how to improve their performance while minimizing energy consumption. In the future, this work can be extended to focus on user-dependent best arms, where the players do not experience the same qualities of the arms using linear bandits. It may be also extended to cover non-stationary environments where the arms' qualities could change with time.

References

Robin Allesiardo, Raphaël Féraud, and Odalric-Ambrym Maillard. The non-stationary stochastic multi-armed bandit problem. *International Journal of Data Science and Analytics*, 3(4):267–283, 2017.

Jean-Yves Audibert, Sébastien Bubeck, and Remi Munos. Best arm identification in multi-armed bandits. In *COLT - The 23rd Conference on Learning Theory*, pages 41–53, 11 2010.

Hiba Dakdouk, Erika Tarazona, Reda Alami, Raphaël Féraud, Georgios Z Papadopoulos, and Patrick Maillé. Reinforcement learning techniques for optimized channel hopping in ieee 802.15. 4-TSCH networks. In *Proceedings of the 21st ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, pages 99–107, 2018.

Peng Du and George Roussos. Adaptive channel hopping for wireless sensor networks. In *International Conference on Selected Topics in Mobile and Wireless Networking (iCOST)*, pages 19–23. IEEE, 2011.

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

Raphaël Féraud, Réda Alami, and Romain Laroche. Decentralized exploration in multi-armed bandits. In *ICML*, 2019.

Kwang-Sung Jun and Robert D Nowak. Anytime exploration for multi-armed bandits using confidence information. In *ICML*, pages 974–982, 2016.

Shivaram Kalyanakrishnan and Peter Stone. Efficient selection of multiple bandit arms: Theory and practice. In *ICML*, volume 10, pages 511–518, 2010.

Shivaram Kalyanakrishnan, Ambuj Tewari, Peter Auer, and Peter Stone. Pac subset selection in stochastic multi-armed bandits. In *ICML*, volume 12, pages 655–662, 2012.

Emilie Kaufmann and Shivaram Kalyanakrishnan. Information complexity in bandit subset selection. In *Conference on Learning Theory*, pages 228–251, 2013.

Vasileios Kotsiou, Georgios Z Papadopoulos, Periklis Chatzimisios, and Fabrice Theoleyre. Label: Link-based adaptive blacklisting technique for 6tisch wireless industrial networks. In *Proceedings of the 20th ACM International Conference on Modelling, Analysis and Simulation of Wireless and Mobile Systems*, pages 25–33, 2017.

Nadège Varsier and Jean Schwoerer. Capacity limits of lorawan technology for smart metering applications. In *2017 IEEE international conference on communications (ICC)*, pages 1–6. IEEE, 2017.

Y. Xue, P. Zhou, S. Mao, D. Wu, and Y. Zhou. Pure-exploration bandits for channel selection in mission-critical wireless communications. *IEEE Transactions on Vehicular Technology*, 67(11):10995–11007, 2018.

Appendix A. Proofs

A.1. Proof of Theorem 1

Low Communication Cost. Using an ArmSelection subroutine with a uniform sampling exploration strategy, the total number of sent messages by algorithm CBAIMPB to find a

set of (ϵ, m)-optimal arms is with a probability of failure η less than:

$$\alpha \left\lceil \frac{\log(\frac{1-\eta/\alpha}{\sum_{k=1}^{K} \frac{\rho_n(\pi_u)}{K}\theta_k})}{\log(1-\sum_{k=1}^{K} \frac{\rho_n(\pi_u)}{K}\theta_k)} + 1 \right\rceil messages \tag{6}$$

where $\alpha = \lfloor \frac{\log \delta}{\log \beta} \rfloor .K - m + N$.

Proof An arm is eliminated when $\lfloor \frac{log\delta}{log\beta} \rfloor$ players vote to eliminate it. Hence, the number of sent messages to eliminate $K - m$ arms is at least $\lfloor \frac{log\delta}{log\beta} \rfloor (K - m)$. Considering the settings of no collisions at most $(\lfloor \frac{log\delta}{log\beta} \rfloor - 1).m$ messages are sent to vote to eliminate the remaining m arms (but they are not globally eliminated) and one extra message per player to share the active rates . Consequently at most $\lfloor \frac{log\delta}{log\beta} \rfloor .K - m + N$ messages are sent by all players using any ArmSelection subroutine if no collisions are taken into account.

On the other hand, considering the settings of collisions and re-transmissions, let $C(\alpha)$ be the random variable corresponding to the number of trials of player n to send α messages. $C(1)$ 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 $q = 1 - p$.

We know that $\mathbb{P}(C(1) = F + 1) = q^F.p$, where F is the number of failures before the success. Then with $\mathbb{P}(C(1) = F + 1) = 1 - \eta$, we know with a probability $1 - \eta$ that the number of trials of player n to send a message is:

$$C(1) \leq \left\lceil \frac{\log \frac{1-\eta}{p}}{\log q} + 1 \right\rceil \tag{7}$$

Similarly, we know with a probability $1 - \eta/\alpha$ that the number of trials of player n to send a message is:

$$C(1) \leq \left\lceil \frac{\log \frac{1-\eta/\alpha}{p}}{\log q} + 1 \right\rceil \tag{8}$$

Sending one message is independent from others, so using the addition rule of the union of independent events and knowing the values of p and q, we get that for sending α messages, with a probability $1 - \eta$ player n needs at most :

$$C(\alpha) \leq \alpha \left\lceil \frac{\log(\frac{1-\eta/\alpha}{\sum_{k=1}^{K} \frac{\rho_n(\pi_u)}{K}\theta_k})}{\log(1-\sum_{k=1}^{K} \frac{\rho_n(\pi_u)}{K}\theta_k)} + 1 \right\rceil \tag{9}$$

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

A.2. Proof of Theorem 2

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

$$\mathcal{O}\left(\frac{1}{p^*}\left(T_{\mathcal{A}} + \sqrt{\frac{1}{2}\log\frac{S}{\delta}}\right)\right) samples \tag{10}$$

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

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

$$\mathbb{E}[T_n] = p_n.T^* \tag{11}$$

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

Applying Hoeffding's inequality, we get:

$$\mathcal{P}(T_n - p_n.T^* \leq -\epsilon) \leq \exp^{-2\epsilon^2} = \frac{\delta}{S} \tag{12}$$

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

$$T_{\mathcal{A}} - p_n.T^* \leq -\sqrt{\frac{1}{2}\log\frac{S}{\delta}} \tag{13}$$

Then, when E_1 does not occur we have with a probability at most δ:

$$T^* \geq \frac{1}{p_{\delta,\beta}}.(T_{\mathcal{A}} + \sqrt{\frac{1}{2}\log\frac{S}{\delta}}) \tag{14}$$

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

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

$$T^* \leq \frac{1}{p_{\delta,\beta}}.(T_{\mathcal{A}} + \sqrt{\frac{1}{2}\log\frac{S}{\delta}}) \tag{15}$$

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

$$\mathbb{P}(T_{n^*} < T_{\mathcal{A}}) = I_{1-p*}(T^* - T_{\mathcal{A}}, 1 + T_{\mathcal{A}}), \tag{16}$$

where $I_a(b, c)$ denotes the incomplete beta function evaluated at a with parameters b and c. Equation (16) comes from the relation between the incomplete beta function and the cumulative binomial distribution.
We have, $\mathbb{P}(p_{\delta,\beta} = p^*) = \mathbb{P}(\forall n \in \mathcal{N}_S, \mathbb{P}(T_n \geq T_{\mathcal{A}}))$.
Finally, knowing $|\mathcal{N}_S| = S = \lfloor \frac{log\delta}{log\beta} \rfloor$, with a probability at least $(1 - I_{1-p*}(T^* - T_{\mathcal{A}}, 1 + T_{\mathcal{A}}))^{\lfloor \frac{log\delta}{log\beta} \rfloor}$, we have $p_{\delta,\beta} = p^*$. ■

A.3. Proof of Corollary 1

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

$$\mathcal{O}\left(\frac{1}{p^*}\left(\frac{K}{{\epsilon'_{n^\dagger}}^2}\log(\frac{K}{\beta}) + \sqrt{\frac{1}{2}\log\frac{S}{\delta}}\right)\right) samples \tag{17}$$

where $n^\dagger = \text{argmin}_{n\in\mathcal{N}} p_n$.
Proof The Direct algorithm in Kalyanakrishnan and Stone (2010) finds with a probability at least $1 - \beta$ a set of m (ϵ', m)-optimal arms with:

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

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

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

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

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

Then, by substitution Theorem 2 completes the proof. ■