



HAL
open science

New Algorithms for Multiplayer Bandits when Arm Means Vary Among Players

Emilie Kaufmann, Abbas Mehrabian

► **To cite this version:**

Emilie Kaufmann, Abbas Mehrabian. New Algorithms for Multiplayer Bandits when Arm Means Vary Among Players. 2019. hal-02006069v1

HAL Id: hal-02006069

<https://hal.science/hal-02006069v1>

Preprint submitted on 4 Feb 2019 (v1), last revised 3 Mar 2020 (v3)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

New Algorithms for Multiplayer Bandits when Arm Means Vary Among Players

Emilie Kaufmann

CNRS & Univ. Lille, CRISTAL UMR 9189, Inria SequeL

EMILIE.KAUFMANN@UNIV-LILLE.FR

Abbas Mehrabian

McGill University

ABBASMEHRABIAN@GMAIL.COM

Abstract

We study multiplayer stochastic multi-armed bandit problems in which the players cannot communicate, and if two or more players pull the same arm, a collision occurs and the involved players receive zero reward. Moreover, we assume each arm has a different mean for each player. Let T denote the number of rounds. An algorithm with regret $O((\log T)^{2+\kappa})$ for any constant κ was recently presented by Bistritz and Leshem (NeurIPS 2018), who left the existence of an algorithm with $O(\log T)$ regret as an open question. In this paper, we provide an affirmative answer to this question in the case when there is a unique optimal assignment of players to arms. For the general case we present an algorithm with expected regret $O((\log T)^{1+\kappa})$, for any $\kappa > 0$.

Keywords: Multi-armed bandit, Combinatorial semi-bandits, Distributed Learning

1. Introduction

Stochastic multi-armed bandit models have been studied extensively over the past years as they capture many sequential decision making problems of practical interest. In the simplest setup, an agent repeatedly chooses among several actions (referred to as “arms”) in each round of a game. To each action i is associated a real-valued parameter μ_i . Whenever the player performs the i -th action (“pulls arm i ”), she receives a random reward with mean μ_i . The player’s objective is to maximize the sum of rewards obtained during the game. If she knew the means associated to the actions before starting the game, she would play an action with the highest mean during all rounds. The problem is to design a strategy for the player to maximize her reward in the setting where the means are unknown. The *regret* of the strategy is the difference between the accumulated rewards in the two scenarios.

To minimize the regret the player is faced with an exploration/exploitation trade-off as she should try (explore) all actions to estimate their means accurately enough but she may want to exploit the action that look *probably* best given her current information. We refer the reader to [Bubeck and Cesa-Bianchi \(2012\)](#) for a survey on this problem. Multi-armed bandit (MAB) has been first studied as a simple model for sequential clinical trials, see, e.g., [Thompson \(1933\)](#); [Robbins \(1952\)](#), but has also found many modern applications to online content optimization, such as designing recommender systems, see [Li et al. \(2010\)](#). In the meantime, MAB models have been shown to be useful for modeling cognitive radio problems, see [Jouini et al. \(2009\)](#); [Anandkumar et al. \(2011\)](#). In this context, arms model different radio channels on which each device can communicate, and the reward associated to each arm may be a binary indicator of the success of the communication, or some measure of its quality.

The applications in cognitive radios have motivated studying the multi-player bandit problem, in which several agents (devices) play on the same bandit (communicate using the same channels). If two

or more agents pull the same arm, a *collision* occurs and all players pulling that arm receive zero reward. Without communicating, each agent should adopt a strategy aimed at maximizing the global reward obtained by all agents (so, we are considering a cooperative scenario rather than a competitive one). While a large body of the literature has studied the case where the mean of the arms are identical across players, we study in this paper the more challenging variant in which each user may have a different utility for each arm: if player m selects arm k , she receives a reward with mean μ_m^k . This variant is more realistic for applications to cognitive radio, as the quality of each channel may vary from one user (device) to another, depending for instance on its configuration or its geographical position.

More precisely, we study the model introduced by [Bistritz and Leshem \(2018\)](#), which has two main characteristics: first, each arm has a different mean for each player, and second, it assumes a fully distributed setting without allowing communication between players. [Bistritz and Leshem \(2018\)](#) propose an algorithm with a regret $O((\log T)^{2+\kappa})$ for any constant κ , where the constant in the big Oh depends on κ and the problem parameters. The regret of their proposed algorithm does not match their $\Omega(\log(T))$ lower bound, leading to the open question of whether there exists an algorithm with $O(\log T)$ regret. In this paper, we propose such an algorithm in the case when there is a unique optimal assignment of players to arms: the M-ETC algorithm, whose regret is upper-bounded in [Theorem 1](#). For the general case we present an algorithm whose dependence on T is $O((\log T)^{1+\kappa})$, for any constant κ : the M-ETC-Elim algorithm, whose regret is upper-bounded in [Theorem 2](#).

This paper is structured as follows. In [Section 2](#) we formally introduce the multi-player bandit model and give a detailed presentation of our contributions. Those results are put in perspective by a comparison with the state-of-the-art work in [Section 3](#). The general structure of our algorithms is presented in [Section 4](#). Then M-ETC and M-ETC-Elim are presented in detail in [Sections 5 and 6](#) respectively, where we also prove the claimed upper bounds on their regrets.

2. The model and Our Contributions

2.1. The General Multi-Player Bandit Problem

We consider a multi-player bandit problem where M players compete over K arms, with $M \leq K$. We denote by μ_k^m the mean reward (or expected utility) of arm k for player m . At each round $t = 1, 2, \dots, T$, player m selects arm $A^m(t)$ and receives a reward

$$R^m(t) = Y_{A^m(t),t}^m (1 - \mathbb{1}(\mathcal{C}_{A^m(t),t}))$$

where $(Y_{k,t}^m)_{t=1}^\infty$ is an i.i.d. sequence which is bounded in $[0, 1]$ and has mean μ_k^m , $\mathcal{C}_{k,t}$ is the event that at least two players have chosen arm k in round t , and $\mathbb{1}(\mathcal{C}_{k,t})$ is the corresponding indicator function.

We assume that the players know the number of arms, K , and have the same numbering for the arms. Moreover, we assume that they know the time horizon, T (total number of arm selections). However, as we shall see T is not used crucially in our algorithms; it is only used to set the failure probabilities, hence if the players only know an upper bound for T , all our results hold with T replaced by that upper bound. We assume that player m in round t observes her reward $R^m(t)$ and also both $Y_{A^m(t),t}^m$ and $\mathbb{1}(\mathcal{C}_{A^m(t),t})$. In the cognitive radio context, $Y_{A^m(t),t}^m$ is called the *sensing information* as it corresponds to the channel quality for player m if she were to use this channel in isolation. Note that in the special case in which the arm distributions satisfy $\mathbb{P}(Y_{k,t}^m = 0) = 0$ (e.g. if this distribution is continuous), then $Y_{A^m(t),t}^m$ and $\mathbb{1}(\mathcal{C}_{A^m(t),t})$ can both be reconstructed from the observation of $R^m(t)$. The decision of player m at round t can be based only on the observation that she has made in the past, that is, $A^m(t)$ is \mathcal{F}_{t-1}^m measurable,

where $\mathcal{F}_t^m = \sigma\left(A^m(1), Y_{A^m(1),1}^m, \mathbb{1}(\mathcal{C}_{A^m(1),1}), \dots, A^m(t), Y_{A^m(t),t}^m, \mathbb{1}(\mathcal{C}_{A^m(t),t})\right)$. Hence, our setting is fully distributed: a player cannot use extra information such as observations made by other players to make her decisions.

We use the shorthand $[n] := \{1, \dots, n\}$. A *matching* is an assignment of players to arms; formally, any one to one function $\pi : [M] \rightarrow [K]$ is a matching, and the utility (or weight) of a matching π is defined as

$$U(\pi) := \sum_{m=1}^M \mu_{\pi(m)}^m.$$

We denote by \mathcal{M} the set of all matchings. It is easily seen that $|\mathcal{M}| = K(K-1)\dots(K-M+1)$. Let $U^* := \max_{\pi \in \mathcal{M}} U(\pi)$ denote the maximum attainable utility. A *maximum matching* (or *optimal matching*) is a matching with utility U^* . For any matching π we denote its gap by $\Delta(\pi) := U^* - U(\pi)$. The strategy maximizing the social utility of the players (i.e. the sum of all their rewards) would be to play in each round according to a maximum matching, and the (*expected*) *regret* with respect to that oracle is defined as

$$R_T = TU^* - \mathbb{E} \left[\sum_{t=1}^T \sum_{m=1}^M R^m(t) \right].$$

Our goal is to design a strategy $(A^m(t))$ for each player m that attains a regret as small as possible. We expect the regret bound to scale also with Δ , defined as the gap between the utility of the best matching and the utility of that of the matching with second best utility:

$$\Delta := \min_{\pi: \Delta(\pi) > 0} \Delta(\pi).$$

Note that $\Delta > 0$ even in the presence of several optimal matchings.

2.2. Our Contributions

We propose two new algorithms for the general multi-player bandit problem achieving (quasi) logarithmic regret. A crucial idea in our algorithms is to use forced collisions as a communication method between the players, an idea proposed by [Boursier and Perchet \(2018\)](#) for the setting when arms have the same means for all players. Our two algorithms rely on a common initialization phase, in which the players elect a leader in a distributed manner. Then a communication protocol that is described in [Section 4](#) is setup, in which the Leader and the Followers have different roles.

The first algorithm, designed for the case in which the optimal matching is unique, is presented in [Section 5](#). It proceeds in several epochs of increasing lengths. At the start of each epoch, each player pulls each arm several times and followers sends the empirical reward received from each of them to the leader. Based on all this information, the leader then computes the first and second empirical maximum matchings. If their difference is large enough, the leader decides that the actual maximum matching has been found. In this case, the leader sends a signal to each follower announcing this, and then sends to each of them the arm they should play. Then the exploitation phase starts, and each player pull this arm until the end of game. If the leader has not decided that the actual maximum matching has been found, she sends a one-bit signal to each follower asking them to proceed to the next epoch. We call this algorithm *Multiplayer Explore-Then-Commit* (M-ETC for short) as the player coordinate to explore all arm sufficiently before committing to jointly playing the best matching. We prove the following regret bound for this algorithm.

Theorem 1 *Assume that there is a unique maximum matching. Then the regret of M-ETC satisfies*

$$R_T = O\left(\frac{KM^3 \log(KT)}{\Delta^2} + M^2 K \log^2\left(\frac{M \log(KT)}{\Delta}\right)\right).$$

This algorithm does not achieve a logarithmic regret in the presence of multiple optimal matchings. Indeed, in this case the players never enter the exploitation phase, as the gap between the two best matchings remains close to zero. Hence, we propose another algorithm that achieves a regret which is close to logarithmic, regardless of whether the optimal matching is unique.

The idea of the second algorithm, presented in Section 6, is that the leader maintains a list of matchings which are candidates of being a maximum matching. This algorithm also uses epochs, and in each epoch the players are asked to explore these candidate matchings and send the results to the leader. The new information allows the leader to refine her estimate for the utilities of the candidate matchings, and so she may eliminate some of the candidates. If at some point the list of candidate matchings becomes a singleton, a unique maximum matching has been found, and the players pull that matching for the rest of the game. Note that the players may also never enter that exploitation phase but instead they would keep exploring several optimal matchings, which still ensures small regret. We call this algorithm *Multiplayer Explore-Than-Commit with matching Eliminations* (M-ETC-Elim for short). M-ETC-Elim depends on a parameter c that controls the size of the epochs. Its regret is upper bounded as follows.

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

$$R_T = O\left(\sum_{U(\pi) < U^*} \left(\frac{M^2 \ln(KT)}{\Delta(\pi)}\right)^{1+1/c} + M|\mathcal{M}|(\ln(KT))^{1+1/c}\right) = O\left(|\mathcal{M}| \times \left(\frac{M^2 \ln(KT)}{\Delta}\right)^{1+1/c}\right)$$

A consequence of this result is that for a fixed problem, for any (arbitrarily small) κ , there exists an algorithm (M-ETC-Elim with parameter $c = \lceil 1/\kappa \rceil$) with regret $R_T = O((\log(T))^{1+\kappa})$. Moreover, Theorem 8 stated in Section 6 shows that in the presence of a unique maximum matching, the M-ETC-Elim algorithm run with parameter $c = 1$ satisfies

$$R_T = O\left(\frac{|\mathcal{M}|M^2 \ln(K) \ln(KT)}{\Delta}\right). \quad (1)$$

To summarize, we provide two algorithms that attain logarithmic regret in the presence of a unique optimal matching. Among these the M-ETC-Elim algorithm attains quasi-logarithmic regret $O(\log(T)^{1+\kappa})$ in the general case. As we shall see, this flexibility comes at a higher computational cost (the leader has to compute the empirical utility of each matching). Moreover, in the presence of a unique optimal matching, the regret of M-ETC-Elim (1) has a worse scaling in M and K than that of M-ETC, but a better scaling in Δ .

3. Related Work

Our problem can be viewed as a challenging extension of two complex multi-armed bandit problems. First, relaxing the need for decentralization, i.e. when a central controller is requested to jointly select $A^1(t), \dots, A^M(t)$, the problem coincides with a combinatorial bandit problem with semi-bandit feedback, as explained in Section 3.1, where we review the achievable regret in the centralized setting.

Second, the particular case of a common utility for all players, i.e. $\mu_k^m = \mu_k$ for all $m \in \{1, \dots, M\}$, has been studied extensively, and we review in Section 3.2 the achievable regret in this multi-player bandit problem, presenting in particular some inspiring ideas to design algorithms for our more general problem. Finally in Section 3.3 we discuss the Game-of-Thrones algorithm of [Bistritz and Leshem \(2018\)](#), which is our only competitor in the fully distributed setting considered in this paper.

3.1. Stochastic Combinatorial Semi-Bandit

In the centralized setting of our problem, a central controller selects at each time step a matching that maps the M players to different arms in $\{1, \dots, K\}$. In other words, introducing $M \times K$ elementary arms with means μ_m^k for $m \in \{1, \dots, M\}$ and $k \in \{1, \dots, K\}$, the central controller selects a subset of M elementary arms whose indices form a matching from $\{1, \dots, M\}$ to $\{1, \dots, K\}$. Then, the utility of each elementary arm is observed and the reward is the sum of those. This is a particular combinatorial bandit problem with semi-bandit feedback, first studied by [Gai et al. \(2012\)](#).

One of the most famous algorithms for this setting is CUCB ([Wei Chen \(2013\)](#)), which computes an Upper Confidence Bound for the mean of each elementary arm and selects in our particular case the maximum weighted matching with the UCB as weights (which is easy to compute using, e.g. the celebrated Hungarian algorithm, see [Munkres \(1957\)](#)). The regret of CUCB is proved to satisfy $R_T = O((M^3 K / \Delta) \log(T))$. The dependency in M was further improved to M^2 by [Kveton et al. \(2015\)](#) for the same algorithm. [Combes et al. \(2015\)](#) propose the ESCB algorithm, which computes an Upper Confidence Bound on the utility of each matching and selects the one with the largest UCB. This alternative approach, more computationally demanding than CUCB, permits to further improve the dependency in M to $M \log^2(M)$ (see [Degenne and Perchet \(2016\)](#)). More recently a Thompson Sampling approach for this problem was analyzed by [Wang and Chen \(2018\)](#), with similar regret guarantees.

Various semi-distributed settings of our problem in which some kind of communication is allowed between players have been studied by [Avner and Mannor \(2016\)](#); [Kalathil et al. \(2014\)](#); [Nayyar et al. \(2018\)](#). In particular, [Nayyar et al. \(2018\)](#) present an algorithm with expected regret that is $O(\log T)$. Our algorithms are the first to reach a logarithmic regret in the fully-distributed setting. One can note however, that compared to the best centralized algorithm, the dependency in Δ , M and K obtained in [Theorems 1 and 2](#) are a bit worse, suggesting a price for decentralization.

3.2. Multi-player Bandits with Shared Utility Across Players

The version in which arms have the same means for all players has also been studied extensively. A first line of work combines standard bandit algorithms with an orthogonalization mechanism ([Liu and Zhao \(2010\)](#); [Anandkumar et al. \(2011\)](#); [Besson and Kaufmann \(2018\)](#)), and obtains logarithmic regret, with a large multiplicative constant due to the control of the number of collisions. [Rosenski et al. \(2016\)](#) propose an algorithm based on a uniform exploration phase in which each player identifies the top M arms, followed by a “musical chairs” protocol that allows each player to end up on a different arm quickly. This approach was later refined by [Lugosi and Mehrabian \(2018\)](#) who show that logarithmic regret can also be obtained in the absence of sensing, that is when each player only observes the rewards $R^m(t)$. Lower bounds on the regret were given by [Liu and Zhao \(2010\)](#); [Besson and Kaufmann \(2018\)](#). In particular, it was believed that each sub-optimal arm k needed to be drawn at least $\log(T)/(\mu_M^* - \mu_k)^2$ many times by each player where μ_M^* is the smallest mean among the top M arms. This lower bound was however disproved by [Boursier and Perchet \(2018\)](#), who propose an algorithm for which only the *sum of time spent by all players* on arm k is $\log(T)/(\mu_M^* - \mu_k)^2$. A discussion on the lower bound for

multi-player bandits can be found in Appendix E, supporting the claim of Boursier and Perchet (2018) that algorithms *exploiting collisions to obtain information* may indeed have a lower regret.

We borrow several ideas from these works, for example we crucially use the idea of ‘communication between the players using forced collisions,’ which was introduced in Boursier and Perchet (2018). We also use the musical chairs idea from Rosenski et al. (2016). Note that while Boursier and Perchet (2018) used *arm eliminations* (coordinated between players) to reduce the regret, we cannot rely on the same idea for our general problem, as an arm that is bad for one player can be good for another player, so it cannot be eliminated. Our second algorithm instead relies on *matching eliminations*.

3.3. A Comparison with the Game-of-Thrones Algorithm

The only previous paper on the setting we study here is by Bistritz and Leshem (2018), who use a very different approach from ours. The Game-of-Thrones algorithm proceeds in epochs, and each epoch is divided into three phases. The first is an exploration phase in which each player select arms at random in order to estimate the mean of each arm. In the second phase (called the GOT phase) the players jointly run a Markov chain (which is called the games of thrones dynamics) whose unique stochastically stable state corresponds to a maximum matching of the empirical means, on which the Markov chain will spend most of its time. The players then enter an exploitation phase in which they keep playing the arm they visited most often during the GOT phase.

The Game-of-Thrones algorithm is proved to have an expected regret of order $O((\log T)^{2+\kappa})$ for any given constant κ , where the constant in the $O()$ depends on κ and on the problem parameters $(K, M$ and $\Delta)$ in a way that is not very explicit in Theorem 3 of Bistritz and Leshem (2018). Unlike our algorithms, GOT does not need to know the horizon T . However, (a lower bound on) the value of Δ is needed to calibrate the length of the exploration phase. Moreover, the analysis of the Game-of-Thrones algorithm works ‘for small enough ε ,’ where ε is a parameter of the GOT phase, but it is not clear how the players must choose ε to achieve the guaranteed regret.

4. Common Structure of Our Algorithms

Our two algorithms share the same structure, presented as Algorithm 1. This is the algorithm that each player executes.

Algorithm 1: General Algorithm

Input: Time horizon T , number of arms K
 $R, M \leftarrow \text{INIT}(K, 1/KT)$
if $R = 1$ **then**
 | LEADERALGORITHM(M)
else
 | FOLLOWERALGORITHM(R, M)
end

Our algorithms share the same initialization phase, borrowed from Boursier and Perchet (2018), that outputs for each player a rank $R \in \{1, \dots, M\}$ as well as the value of M , that may be initially unknown to the players. This initialization phase, referred to as $\text{INIT}(K, \delta_0)$ depends on the number of arms and on a confidence parameter δ_0 . It relies on a ‘musical chair’ phase after which the players end up on distinct arms, followed by a so-called Sequential Hopping protocol that permit them to know their ordering. For

the sake of completeness, it is described in details in Appendix C, where we also prove the following. (This lemma was essentially proved in (Boursier and Perchet, 2018, Lemma 1).)

Lemma 3 *For any $\delta_0 > 0$, the procedure $\text{INIT}(K, \delta_0)$ takes $K \ln(K/\delta_0) + 2K - 2 < K \ln(e^2 K/\delta_0)$ many rounds and has the property that with probability at least $1 - \delta_0$, all players will learn M and obtain a distinct ranking from 1 to M .*

Once all players have learned their rank, player 1 becomes the *leader* and other players become the *followers*. The leader will execute additional computations internally, and will communicate with the followers individually, while each follower will communicate only to the leader. Note that this approach is different from the one introduced by Boursier and Perchet (2018), in which communications are decentralized, namely all players send messages to each other.

The leader and follower algorithms, described in the next sections, indeed rely on several *communication phases*, that start at the same time for every player. During communication phases, the default behavior of each player is to pull her *communication arm*. It is crucial that these communicating arms are distinct: a default choice that has this property, which we also use in our algorithms is for each player to use the arm whose index is the ranking of the player as its communicating arm. The leader has ranking 1, so she will use arm 1 as the communicating arm.

Suppose at a certain time, that is known to all players, the leader wants to send a sequence of b bits t_1, \dots, t_b , to the player with ranking i that has communicating arm i . During the next b rounds, for each $j = 1, 2, \dots, b$, if $t_j = 1$, the leader pulls arm i , otherwise, she pulls her own communicating arm 1, and all other followers stick to their communicating arms. Player i can thus reconstruct these b bits after these b rounds, by observing the collisions on arm i . Conversely, if a follower i with communicating arm i wants to communicate those b bits to the leader, the leader and all other followers keep pulling their communicating arms while player i pulls arm 1 if she wants to transmit a 1, and pulls i otherwise. The rankings are also useful to know *in which order communications should be performed*, as the leader will successively communicate messages to the $M - 1$ followers, and the $M - 1$ followers will successively communicate their messages to the leader.

Remark 4 *To obtain a smaller regret, it would be better for the players to use communicating arms that correspond to the current best empirical matching, rather than simply using arms 1 to R . This can be implemented as follows: at the end of each phase, the leader sends to each player the arm she should play in the current empirical best matching, as well as the arm the leader should play. The players then use these arms for communication. We expect this idea to significantly decrease the algorithms' regret in practice. However, it will not improve the asymptotic regret bounds, hence to avoid additional notation, we stick to simply using the arms 1 to R for communication.*

In our algorithms, the messages that are communicated to the players are the *values of empirical means* observed by each player. Note that the communication protocol only permits to transmit bits. If the rewards are binary as in the original setup of Boursier and Perchet (2018), this permits to reconstruct the empirical mean perfectly. To cover more general bounded distributions, our algorithms transmit *truncations* up to some precision of the value of the empirical means.

5. Regret Analysis of the M-ETC Algorithm

The M-ETC algorithm follows the general structure given in Algorithm 1, with the Leader and Follower algorithms specified below.

Procedure LeaderAlgorithm(M) for the M-ETC algorithm

Input: Number of players M
for $p=1,2,\dots$ **do**

 Pull each arm 2^p times (round robin from arm 1)

 $\tilde{\mu}_k^1 \leftarrow$ empirical utility of arm k

 Receive the values $\tilde{\mu}_1^m, \tilde{\mu}_2^m, \dots, \tilde{\mu}_K^m$ from each player m // (comm. protocol)

 Compute the maximum matching π_1 and the second maximum matching π_2 given the weights $\tilde{\mu}$
if $\sum_{m=1}^M \{\tilde{\mu}_{\pi_1(m)}^m - \tilde{\mu}_{\pi_2(m)}^m\} > 4M \times \sqrt{\ln(2M^2KT^2)/2^{p+1}}$ **then**

Send STOP signal to the players // (comm. protocol)

 Send the index of arm $\pi_1(m)$ to each player m // (comm. protocol)

 Pull arm $\pi_1(1)$ until the end of the game

else

Send CONTINUE signal to the players

end
end

Procedure FollowerAlgorithm(R,M) for the M-ETC algorithm

Input: Ranking R , number of players M
for $p=1,2,\dots$ **do**

 Pull each arm 2^p times (round robin starting from R)

for $k = 1, 2, \dots, K$ **do**
 $\hat{\mu}_k^R \leftarrow$ empirical utility of arm k

 Truncate $\hat{\mu}_k^R$ to $\tilde{\mu}_k^R$ using the $(p+1)/2$ most significant bits

end

 Send the values $\tilde{\mu}_1^R, \tilde{\mu}_2^R, \dots, \tilde{\mu}_K^R$ to the leader // (comm. protocol)

if the leader sends the STOP signal // (comm. protocol)

then

 Receive the index of arm $\pi_1(R)$ from leader and pull it until end of game

end
end

Let

$$\delta := (M^2KT^2)^{-1}, \quad \varepsilon_p := \sqrt{\ln(2/\delta)/2^{p+1}} \text{ for any integer } p. \quad (2)$$

After the initialization, the M-ETC algorithm proceeds in epochs $p = 1, 2, \dots$. At the start of epoch p , each player pulls each arm 2^p times, and records the empirical reward received from each of them. Player i starts from arm i , and this guarantees there is no collision during these $K2^p$ exploration rounds. Then a first communication phase begins, during which each player (except the leader) sends their observed empirical means for the arms to the leader. More precisely, for each arm, the player truncates the empirical mean (which is a number in $[0, 1]$) and sends only the $(p+1)/2$ most significant bits of this number to the leader. Based on all this information, the leader then computes the first and second empirical maximum matchings (which can be done in polynomial time, see e.g. [Chegireddy and Hamacher \(1987\)](#)). If their difference is larger than $4M\varepsilon_p$, the leader decides that the actual maximum matching has been found. In this case, the leader sends a one-bit signal to each player announcing this, and then sends to each of them

the arm they should play. Then the exploitation phase starts, and each player pulls her arm until the end of game. If the leader has not decided that the actual maximum matching has been found, she sends a one-bit signal to each player asking them to proceed to the next epoch.

The regret of M-ETC is upper bounded in the following Theorem, which yields Theorem 1 stated in Section 2. In the sequel, $\ln(\cdot)$ denotes the natural logarithm and $\lg(\cdot)$ denotes logarithm in base 2.

Theorem 5 *Assume that there is a unique maximum matching. Then the regret of M-ETC is upper bounded by*

$$2 + MK \ln(e^2 K^2 T) + \frac{64KM^3 \ln(2M^2 KT^2)}{\Delta^2} + 2M^2 K \lg^2 \left(\frac{32M^2 \ln(2M^2 KT^2)}{\Delta^2} \right) + M^2 \lg K.$$

Proof of Theorem 5. We first introduce useful notation. Denote by $\tilde{U}_p(\pi)$ the estimated utility for the matching π that the leader can compute based on the received information $\tilde{\mu}_k^m(p)$ in epoch p . We let \hat{p}_T the number of epochs before the start of the exploitation phase. Recall that a successful initialization means that all players identify M and that their ranks are distinct. We define the *good event*

$$\mathcal{E}_T = \left\{ \text{INIT}(K, 1/KT) \text{ is successful and } \forall p \leq \hat{p}_T, \forall \pi \in \mathcal{M}, |\tilde{U}_p(\pi) - U(\pi)| \leq 2M\epsilon_p \right\} \quad (3)$$

Using Hoeffding's equality and a union bound over at most $\lg(T)$ epochs (see details in Appendix D) together with Lemma 3, one can prove that \mathcal{E}_T holds with large probability.

Lemma 6 $\mathbb{P}(\mathcal{E}_T) \geq 1 - \frac{2}{MT}$.

If the good event does not hold, we may simply upper bound the regret by MT . Hence it suffices to bound the expected regret conditional on no bad event happening, and the unconditional expected regret is bounded by this value plus 2.

Suppose that \mathcal{E}_T holds. For any sub-optimal matching π and any $p \leq \hat{p}_T$, $\tilde{U}_\pi(p) - \tilde{U}_{\pi^*}(p) < 4M\epsilon_p$, hence whenever the player enter the exploitation phase, the optimal matching has been identified. Therefore M multiplied by the number of rounds until this happens is an upper bound for the regret (since the contribution of each round to the regret is at most M). Recall that Δ is the gap between the utility of the maximum matching and that of the second maximum matching. Let P be the smallest positive integer such that $8M\epsilon_P < \Delta$. Observe that by the end of epoch P , the leader will decide that she has found the maximum matching. Indeed, using the definition of \mathcal{E}_T , for all $\pi \neq \pi^*$,

$$\begin{aligned} \tilde{U}_P(\pi^*) - \tilde{U}_P(\pi) &\geq \Delta + \tilde{U}_P(\pi^*) - U(\pi^*) + U(\pi) - \tilde{U}_P(\pi) \\ &\geq \Delta - 4M\epsilon_P > 4M\epsilon_P, \end{aligned}$$

which triggers the beginning of the exploitation phase. Straightforward calculations using the definition of ϵ_p in (2) give

$$P \leq \lg(32M^2 \ln(2M^2 KT^2) / \Delta^2).$$

The number of rounds in the initialization phase is $K \ln(K^2 T) + 2K - 2$ by Lemma 3. For each epoch $p = 1, \dots, P$, first there are $K2^p$ exploration rounds, then $MK(p+1)/2$ rounds for players to send the leader their mean estimates, and an additional M rounds for the leader to send the players the one-bit signal. Finally, right before starting the exploitation phase, the leader sends the players the arm

each have to play, which consumes another $M \lg K$ rounds. Therefore, the total number of rounds before start of exploitation can be bounded by

$$\begin{aligned}
 & K \ln(e^2 K^2 T) + \sum_{p=1}^P \{K 2^p + MK(p+1)/2 + M\} + M \lg K \\
 & \leq K \ln(e^2 K^2 T) + K 2^{P+1} + \frac{MK(P+1)(P+2)}{4} + MP + M \lg K \\
 & \leq K \ln(e^2 K^2 T) + K 2^{P+1} + 2MKP^2 + M \lg K \\
 & \leq K \ln(e^2 K^2 T) + \frac{64KM^2 \ln(2M^2 KT^2)}{\Delta^2} + 2MK \lg^2(32M^2 \ln(2M^2 KT^2)/\Delta^2) + M \lg K,
 \end{aligned}$$

completing the proof of Theorem 5.

6. Regret Analysis of the M-ETC-Elim algorithm

The M-ETC-Elim algorithm follows the general structure given in Algorithm 1, with the Leader and Follower algorithms described in Appendix A. As can be seen from the description below, the main differences with the M-ETC algorithm are the lengths of the exploration phases, which are 2^{p^c} for some parameter c , and the fact that the leader computes and stores the values of each matching, and performs eliminations. M-ETC-Elim is thus a bit more costly to implement than M-ETC as it requires the leader to store and compute the utility of each matching. However, it enjoys (quasi) logarithmic regret regardless of whether there is a unique optimal matching, as can be seen below.

Let

$$\delta := (M^2 K T^2)^{-1}, \quad \varepsilon_p := \sqrt{\ln(2/\delta)/2^{1+p^c}} \text{ for any integer } p. \quad (4)$$

After the initialization phase, the algorithm proceeds in epochs $p = 1, 2, \dots$. The leader maintains a set of candidate matchings, which contains all matchings initially. In epoch p , first the leader communicates to each follower the list of candidate matchings \mathcal{C} , that is, the list of arms each follower should play in those matchings. If this list contains just one arm, each follower play that arm for the rest of the game. Otherwise, exploration starts. Each player is assigned to one arm in each matching $\pi \in \mathcal{C}$. For each candidate matching, the player pulls that arm 2^{p^c} times, and records the empirical reward received. Then another communication phase begins, during which each player sends their observed empirical means for the arms (which she has pulled) to the leader. More precisely, for each arm, the player truncates the empirical mean and sends only the p^c most significant bits of this number to the leader. Based on all this information, the leader then computes the empirical maximum utility, and eliminates (from the set of candidates) any matching whose empirical utility is smaller by at least $4M\varepsilon_p$. Then everyone proceeds to the next epoch.

We first prove the following regret upper bound for the M-ETC-Elim algorithm used with any positive integer c , which yields Theorem 2 stated in Section 2.

Theorem 7 *For any positive integer c , the M-ETC-Elim algorithm with parameter c , has its regret at time $T \geq \exp(2^{(4c^2)^c})$ upper bounded by*

$$2 + MK \ln(e^2 K^2 T) + 2M \lg(K)(\lg T)^{1/c} |\mathcal{M}| + 2eM^2 K (\lg T)^{1+1/c} + \sum_{U(\pi) < U^*} \left(\frac{64M^2 \ln(2M^2 KT^2)}{\Delta(\pi)} \right)^{1+1/c}.$$

We also provide in Theorem 8 a refined analysis of M-ETC-Elim with parameter $c = 1$, which notably shows that under the existence of a unique optimal matching, the regret is $O(\log(T))$ as for the M-ETC algorithm.

Theorem 8 *If the maximum matching is unique, the regret of the M-ETC-Elim algorithm with parameter $c = 1$ is upper bounded by*

$$2 + MK \ln(e^2 K^2 T) + M \lg(K) |\mathcal{M}| \lg \left(\frac{64M^2 \ln(2M^2 KT^2)}{\Delta^2} \right) + M^2 K \lg^2 \left(\frac{64M^2 \ln(2M^2 KT^2)}{\Delta^2} \right) \\ + \sum_{U(\pi) < U^*} \frac{128M^2 \ln(2M^2 KT^2)}{\Delta(\pi)} = O \left(|\mathcal{M}| \frac{M^2 \ln(KT) \ln(K)}{\Delta} \right)$$

We now proceed to proving Theorems 7 and 8. Letting \mathcal{E}_T be the same good event as (3), where \hat{p}_T denotes this time the number of epochs used by M-ETC-Elim. Using similar arguments as for Lemma 6, notably the fact that during epoch p , for each pair $(m, k) \in [M] \times [K]$ that is contained in some candidate matching, player m has pulled arm k at least 2^{p^c} times, and $(1/2)^{p^c} \leq \varepsilon_p$, one can prove the following.

Lemma 9 $\mathbb{P}(\mathcal{E}_T) \geq 1 - \frac{2}{MT}$.

As before, if \mathcal{E}_T doesn't hold, we may upper bound the regret by MT . Hence it suffices to bound the expected regret conditional on \mathcal{E}_T and the unconditional expected regret is bounded by this value plus 2.

Suppose that \mathcal{E}_T holds. We first bound the number of rounds during the initialization and communications. The number of rounds in the initialization phase is $< K \ln(e^2 K^2 T)$ by Lemma 3. For each epoch p , first the leader communicates to each player the list of candidate matchings. There can be up to $|\mathcal{M}|$ such matchings, and for each of them the leader must communicate to the player the arm she has to pull (there is no need to communicate to her the whole matching) which requires $\lg K$ bits, and there are a total of M players, so this takes at most $M \lg(K) |\mathcal{M}|$ many rounds. At the end of epoch each player sends the leader the empirical estimates for the arms she has pulled, which requires at most MKp^c many rounds. Finally, with \hat{p}_T denoting the number of epoch before the (possible) start of the exploitation, the total regret incurred due to the initialization and communication phases can be bounded by

$$M \times \left\{ K \ln(e^2 K^2 T) + \sum_{p=1}^{\hat{p}_T} (M \lg(K) |\mathcal{M}| + MKp^c) \right\} \leq MK \ln(e^2 K^2 T) + M(\hat{p}_T) \lg(K) |\mathcal{M}| + M^2 K (\hat{p}_T)^{c+1}.$$

During any other round, the players are jointly pulling a matching. We individually compute the contribution of each matching to the regret. For a matching π , denote by $\Delta(\pi) := U^* - U(\pi)$ its gap. Fix a matching π . On the one hand, its contribution to the regret can be bounded by zero if $\Delta(\pi) = 0$. Otherwise, let $P(\pi)$ be the smallest positive integer such that $8M\varepsilon_{P(\pi)} < \Delta(\pi)$ and observe that by the end of epoch $P(\pi)$, the leader will eliminate π from the set of candidate matchings, and it will not be pulled again. Hence the contribution of any sub-optimal π to the regret can be bounded by

$$\sum_{p=1}^{P(\pi)} \Delta(\pi) 2^{p^c} \leq 2\Delta(\pi) 2^{P(\pi)^c},$$

where we have used the inequality $\sum_{n=1}^p 2^{n^c} \leq 2 \times 2^{p^c}$ valid for any positive integers p and c , which can be proved by induction. Straightforward calculations using the definition of ε_p in (4) give

$$P(\pi) \leq 1 + L(\pi)^{1/c}, \text{ where } L(\pi) := \lg \left(\frac{32M^2 \ln(2M^2 KT^2)}{\Delta(\pi)^2} \right).$$

Consequently, the total expected regret of the algorithm can be bounded by

$$MK \ln(e^2 K^2 T) + M(\hat{p}_T) \lg(K) |\mathcal{M}| + M^2 K (\hat{p}_T)^{c+1} + \sum_{U(\pi) < U^*} 2\Delta(\pi) 2^{P(\pi)^c} \quad (5)$$

Proof of Theorem 7. Note that the assumption $T \geq \exp(2^{(4c^2)^c})$ gives $(\lg \ln T)^{1/c} \geq 4c^2 \geq \frac{c}{\ln(1+1/2c)}$, where we have used that $\ln(1+x) \geq x/2$ for any $0 < x \leq 1/2$. In particular, $(\lg T)^{1/c} \geq c$. We will also use the inequality

$$(x+1)^c \leq e^{c/x} x^c \quad (6)$$

which holds for all positive x , since $(x+1)^c/x^c = (1+1/x)^c \leq \exp(1/x)^c = \exp(c/x)$.

Using the a crude upper bound on the number of epochs that can fit within T samples, we get $\hat{p}_T \leq 1 + (\lg T)^{1/c}$. As $(\lg T)^{1/c} \geq c \geq 1$ one gets $\hat{p}_T \leq 2(\lg T)^{1/c}$. Also (6) gives $(\hat{p}_T)^c \leq e \lg T$, hence we can upper bound the two middle terms of (5) as

$$M(\hat{p}_T) \lg(K) |\mathcal{M}| + M^2 K (\hat{p}_T)^{c+1} \leq 2M \lg(K) (\lg T)^{1/c} |\mathcal{M}| + 2eM^2 K (\lg T)^{1+1/c}. \quad (7)$$

Now we claim that we have

$$P(\pi)^c \leq \left(1 + \frac{1}{2c}\right) L(\pi). \quad (8)$$

Indeed, since $\Delta(\pi) \leq M$, we have $L(\pi)^{1/c} > (\lg \ln T)^{1/c} \geq \frac{c}{\ln(1+1/2c)}$ and so (6) with $x = L(\pi)^{1/c}$ gives (8). Hence,

$$2\Delta(\pi) 2^{P(\pi)^c} \leq 2\Delta(\pi) \left(\frac{32M^2 \ln(2M^2 K T^2)}{\Delta(\pi)^2} \right)^{1+1/2c} \leq \left(\frac{64M^2 \ln(2M^2 K T^2)}{\Delta(\pi)} \right)^{1+1/c}$$

Plugging this bound and (7) in (5) yields the bound in Theorem 7.

Proof of Theorem 8. When there is a unique optimal matching, if the good event holds, the M-ETC-Elim algorithm will eventually enter the exploitation phase. That is \hat{p}_T can be much smaller than the crude upper bound used in the previous proof.

More specifically, introducing π' as the second maximum matching, so that $\Delta(\pi') = \Delta$, it can be shown that, on the event \mathcal{E}_T ,

$$\hat{p}_T \leq P(\pi') \leq \lg \left(\frac{64M^2 \ln(2M^2 K T^2)}{\Delta^2} \right).$$

Plugging this bound in (5) and using the specific value $c = 1$ yields the bound in Theorem 8.

7. Conclusion

In this paper we answered an open question by [Bistriz and Leshem \(2018\)](#), showing it is possible to design an algorithm with a regret strictly smaller than $O(\log^2(T))$ for the general multi-player bandit problem in which arms may have different utility for each player. More precisely, we proved that $O(\log^{1+\kappa}(T))$ regret can be attained for any constant $\kappa > 0$. Moreover, in the presence of a unique optimal of assignment of player to arms, one can obtain $O(\log(T))$ regret. Our algorithms are pushing further the communication trick initially introduced by [Boursier and Perchet \(2018\)](#), using centralized communications between a leader (who performs computations) and followers.

It can be noted that the algorithm heavily relies on the fact that both the sensing information and the collision indicators are observed at each round. In future work, we will investigate whether algorithms with logarithmic regret can be proposed for the general multi-player bandit problem when the players learn from their rewards only. So far such algorithm have only been proposed when all players share the same utilities for arms, see [Lugosi and Mehrabian \(2018\)](#); [Boursier and Perchet \(2018\)](#).

References

- Animashree Anandkumar, Nithin Michael, Ao Kevin Tang, and Ananthram Swami. Distributed algorithms for learning and cognitive medium access with logarithmic regret. *IEEE Journal on Selected Areas in Communications*, 29(4):731–745, 2011.
- O. Avner and S. Mannor. Multi-user lax communications: A multi-armed bandit approach. In *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*, pages 1–9, April 2016. doi: 10.1109/INFOCOM.2016.7524557.
- Lilian Besson and Emilie Kaufmann. Multi-player bandits revisited. In *Proceedings of Algorithmic Learning Theory*, volume 83 of *Proceedings of Machine Learning Research*, pages 56–92. PMLR, 2018.
- Itai Bistriz and Amir Leshem. Distributed multi-player bandits - a game of thrones approach. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.
- Etienne Boursier and Vianney Perchet. SIC-MMAB: synchronisation involves communication in multi-player multi-armed bandits. *arXiv.org:1809.08151*, 2018.
- S. Bubeck and N. Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends in Machine Learning*, 5(1):1–122, 2012.
- Chandra R. Chegireddy and Horst W. Hamacher. Algorithms for finding K -best perfect matchings. *Discrete Appl. Math.*, 18(2):155–165, 1987. Rio conference on combinatorial optimization (Rio de Janeiro, 1985).
- R. Combes, S. Talebi, A. Proutière, and M. Lelarge. Combinatorial bandits revisited. In *Advances in Neural Information Processing Systems (NIPS)*, 2015.
- Rémy Degenne and Vianney Perchet. Combinatorial semi-bandit with known covariance. In *Advances in Neural Information Processing Systems (NIPS)*, 2016.
- Yi Gai, Bhaskar Krishnamachari, and Rahul Jain. Combinatorial network optimization with unknown variables: Multi-armed bandits with linear rewards and individual observations. *IEEE/ACM Trans. Netw.*, 20(5):1466–1478, 2012.
- Aurélien Garivier, Pierre Ménard, and Gilles Stoltz. Explore first, exploit next: The true shape of regret in bandit problems. *arXiv preprint arXiv:1602.07182*, 2016.
- Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *J. Amer. Statist. Assoc.*, 58:13–30, 1963.

- W. Jouini, D. Ernst, C. Moy, and J. Palicot. Multi-Armed Bandit Based Policies for Cognitive Radio's Decision Making Issues. In *International Conference Signals, Circuits and Systems*. IEEE, 2009.
- D. Kalathil, N. Nayyar, and R. Jain. Decentralized learning for multiplayer multiarmed bandits. *IEEE Transactions on Information Theory*, 60(4):2331–2345, April 2014. ISSN 0018-9448. doi: 10.1109/TIT.2014.2302471.
- Branislav Kveton, Zheng Wen, Azin Ashkan, and Csaba Szepesvári. Tight regret bounds for stochastic combinatorial semi-bandits. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics, AISTATS*, 2015.
- L. Li, W. Chu, J. Langford, and R. E. Schapire. A Contextual-Bandit Approach to Personalized News Article Recommendation. In *International Conference on World Wide Web*, pages 661–670. ACM, 2010.
- Keqin Liu and Qing Zhao. Distributed learning in multi-armed bandit with multiple players. *IEEE Trans. Signal Process.*, 58(11):5667–5681, 2010.
- Gábor Lugosi and Abbas Mehrabian. Multiplayer bandits without observing collision information. *arXiv preprint arXiv:1808.08416*, 2018.
- James Munkres. Algorithms for the assignment and transportation problems. *J. Soc. Indust. Appl. Math.*, 5:32–38, 1957.
- N. Nayyar, D. Kalathil, and R. Jain. On regret-optimal learning in decentralized multiplayer multiarmed bandits. *IEEE Transactions on Control of Network Systems*, 5(1):597–606, March 2018. ISSN 2325-5870. doi: 10.1109/TCNS.2016.2635380.
- H. Robbins. Some aspects of the sequential design of experiments. *Bulletin of the American Mathematical Society*, 58(5):527–535, 1952.
- Jonathan Rosenski, Ohad Shamir, and Liran Szlak. Multi-player bandits—a musical chairs approach. In *International Conference on Machine Learning*, pages 155–163, 2016.
- W.R. Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25:285–294, 1933.
- Siwei Wang and Wei Chen. Thompson sampling for combinatorial semi-bandits. In *International Conference on Machine Learning (ICML)*, 2018.
- Yang Yuan Wei Chen, Yajun Wang. Combinatorial multi-armed bandit: General framework and applications. In *International Conference on Machine Learning (ICML)*, 2013.

Appendix A. Leader Algorithm for M-ETC-Elim

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

Input: Number of players M

$\mathcal{C} \leftarrow \mathcal{M}$ // list of candidate matchings

for $p = 1, 2, \dots$ **do**

for each player $m = 1, \dots, M$ **do**

 Send to player m the value of $\text{size}(\mathcal{C})$ // (comm. protocol)

for $i = 1, 2, \dots, \text{size}(\mathcal{C})$ **do**

 Send to player m the arm associated to player m in the matching $\mathcal{C}[i]$

 // (comm. protocol)

end

end

if $\text{size}(\mathcal{C}) = 1$ // (enter exploitation phase)

then

 pull for the rest of the game the arm associated to player 1 in the unique matching in \mathcal{C}

end

for $i = 1, 2, \dots, \text{size}(\mathcal{C})$ **do**

 pull 2^{p^c} times the arm associated to player 1 in the matching $\mathcal{C}[i]$

end

for $k = 1, 2, \dots, K$ **do**

$\tilde{\mu}_k^1 \leftarrow$ empirical utility of arm k

 // if arm k has not been pulled in this epoch, let $\tilde{\mu}_k^1 \leftarrow 0$

end

 Receive the values $\tilde{\mu}_1^m, \tilde{\mu}_2^m, \dots, \tilde{\mu}_K^m$ from each player m // (comm. protocol)

$\pi_1 \leftarrow \operatorname{argmax} \left\{ \sum_{m=1}^M \tilde{\mu}_{\pi(m)}^m : \pi \in \mathcal{C} \right\}$

for $\pi \in \mathcal{C}$ **do**

if $\sum_{m=1}^M \left\{ \tilde{\mu}_{\pi_1(m)}^m - \tilde{\mu}_{\pi(m)}^m \right\} > 4M \times \sqrt{\ln(2M^2KT^2)/2^{1+p^c}}$ **then** remove π from \mathcal{C}

end

end

Appendix B. Follower Algorithm for M-ETC-Elim

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

Input: Ranking R , number of players M

for $p = 1, 2, \dots$ **do**

Receive the value of $\text{size}(\mathcal{C})$ // (comm. protocol)

for $i = 1, 2, \dots, \text{size}(\mathcal{C})$ **do**

Receive the arm associated to this player in the matching $\mathcal{C}[i]$ // (comm. protocol)

end

if $\text{size}(\mathcal{C}) = 1$ // (enter exploitation phase)

then

pull for the rest of the game the arm associated to this player in the unique matching in \mathcal{C}

end

for $i = 1, 2, \dots, \text{size}(\mathcal{C})$ **do**

pull 2^{p^c} times the arm associated to this player in the matching $\mathcal{C}[i]$

end

for $k = 1, 2, \dots, K$ **do**

$\hat{\mu}_k^R \leftarrow$ empirical utility of arm k

// if arm k has not been pulled in this epoch, let $\hat{\mu}_k^R \leftarrow 0$

Truncate $\hat{\mu}_k^R$ to $\tilde{\mu}_k^R$ using the p^c most significant bits

end

Send the values $\tilde{\mu}_1^R, \tilde{\mu}_2^R, \dots, \tilde{\mu}_K^R$ to the leader // (comm. protocol)

end

Appendix C. Detailed Description of the Initialization Procedure

The pseudo-code of the $\text{INIT}(K, \delta_0)$ procedure, first introduced by [Boursier and Perchet \(2018\)](#), is presented in [Algorithm 2](#) for the sake of completeness. We now provide a proof of [Lemma 3](#).

Let $T_0 := K \ln(K/\delta_0)$. During the first T_0 rounds, each player tries to occupy a distinct arm using a so-called musical chairs algorithm (first introduced in [Rosenski et al. \(2016\)](#)): she repeatedly pulls a random arm until she gets no collision, and then sticks to that arm. We claim that after T_0 rounds, with probability $1 - \delta_0$ all players have succeeded in occupying some arm. Indeed, the probability that a given player \mathcal{A} that has not occupied an arm so far, does not succeed in the next round is at most $1 - 1/K$, since there exists at least 1 arm that is not pulled in that round, and this arm is chosen by \mathcal{A} with probability $1/K$. Hence, the probability that \mathcal{A} does not succeed in occupying an arm during these T_0 rounds is not more than

$$(1 - 1/K)^{T_0} < \exp(-T_0/K) = \delta_0/K \leq \delta_0/M,$$

and a union bound over the M players proves the claim.

Once each player have occupied some arm, the next goal is to determine the number of players and their ranking. This part of the procedure is deterministic. The ranking of the players will be determined by the indices of the arms they have occupied: a player with a smaller index will have a smaller ranking. To implement this, a player that has occupied arm $k \in [K]$, will pull this arm for $2k - 2$ more rounds (the waiting period), and will then sweep through the arms $k + 1, k + 2, \dots, K$, and can learn the

Algorithm 2: INIT, the initialization algorithm

Input: number of arms K , failure probability δ_0

Output: Ranking R , number of players M

```

 $k \leftarrow 0$ 
for  $T_0 := K \ln(K/\delta_0)$  rounds do // rounds  $1, \dots, T_0$ 
    if  $k = 0$  then
        | pull a uniformly random arm  $i \in [K]$ 
        | if no collision occurred then  $k \leftarrow i$  // arm  $k$  is occupied
    else
        | pull arm  $k$ 
    end
end
 $R \leftarrow 1$ 
 $M \leftarrow 1$ 
for  $2k - 2$  rounds do // rounds  $T_0 + 1, \dots, T_0 + 2k - 2$ 
    | pull arm  $k$ 
    | if collision occurred then
        | |  $R \leftarrow R + 1$ 
        | |  $M \leftarrow M + 1$ 
    | end
end
for  $i = 1, 2, \dots, K - k$  do // rounds  $T_0 + 2k - 1, \dots, T_0 + K + k - 2$ 
    | pull arm  $k + i$ 
    | if collision occurred then
        | |  $M \leftarrow M + 1$ 
    | end
end
for  $K - k$  rounds do // rounds  $T_0 + K + k - 1, \dots, T_0 + 2K - 2$ 
    | pull arm 1
end

```

number of players who have occupied arms in this range by counting the number of collisions she gets. Moreover, she can learn the number of players occupying arms $1, \dots, k - 1$ by counting the collisions during the waiting period; see Algorithm 2 for details. The crucial observation to prove correctness of the algorithm is that two players occupying arms k_1 and k_2 will collide exactly once, and that happens in round $T_0 + k_1 + k_2 - 2$.

Appendix D. Proof of Lemma 6

We first recall Hoeffding's inequality.

Proposition 10 (Hoeffding's inequality (Hoeffding, 1963, Theorem 2)) *Let X_1, \dots, X_n be independent random variables taking values in $[0, 1]$. Then for any $t \geq 0$ we have*

$$\mathbb{P} \left(\left| \frac{1}{n} \sum X_i - \mathbb{E} \left[\frac{1}{n} \sum X_i \right] \right| > t \right) < 2 \exp(-2nt^2).$$

Recall the definition of the good event

$$\mathcal{E}_T = \left\{ \text{INIT}(K, 1/KT) \text{ is successful for all players and for all } p \leq \hat{p}_T, \forall \pi \in \mathcal{M}, |\tilde{U}_p(\pi) - U(\pi)| \leq 2M\epsilon_p \right\}$$

and recall $\epsilon_p := \sqrt{\ln(2/\delta)/2^{p+1}}$. Let \mathcal{F} be the event that $\text{INIT}(K, 1/KT)$ is successful for all players. One has

$$\begin{aligned} \mathbb{P}(\mathcal{E}_T^c) &\leq \mathbb{P}(\mathcal{F}^c) + \mathbb{P}\left(\exists p \leq \hat{p}_T, \exists \pi \in \mathcal{M} : |\tilde{U}_p(\pi) - U(\pi)| > 2M\epsilon_p | \mathcal{F}\right) \\ &\leq \frac{1}{KT} + \mathbb{P}\left(\exists p \leq \lg(T), \exists \pi \in \mathcal{M} : |\tilde{U}_p(\pi) - U(\pi)| > 2M\epsilon_p | \mathcal{F}\right), \end{aligned}$$

where we use that $\hat{p}_T \leq \lg(T)$ deterministically.

Fix an epoch p , a player m and an arm k . We denote by $\hat{\mu}_k^m(p)$ the empirical mean of arm k for player m at the end of epoch p . By Hoeffding's inequality and since this empirical mean is based on at least 2^p pulls, we have

$$\mathbb{P}(|\hat{\mu}_k^m(p) - \mu_k^m| > \epsilon_p) < \delta.$$

Now the value of $\hat{\mu}_k^m(p) \in [0, 1]$ that is sent to the leader uses the $(p+1)/2$ most significant bits. The truncation error is thus at most $2^{-(p+1)/2} < \epsilon_p$, and we have

$$\mathbb{P}(|\tilde{\mu}_k^m(p) - \mu_k^m| > 2\epsilon_p) < \delta.$$

Conditionally to the event \mathcal{F} that the initialization is successful, the quantity $\tilde{U}_p(\pi)$ is a sum of M values $\tilde{\mu}_k^m(p)$ for M different arms k . Hence, it follows that

$$\mathbb{P}\left(\exists \pi \in \mathcal{M} : |\tilde{U}_p(\pi) - U(\pi)| > 2M\epsilon_p | \mathcal{F}\right) < \mathbb{P}(\exists k \leq K, m \leq M : |\tilde{\mu}_k^m(\pi) - \mu_k^m| > 2\epsilon_p) < KM\delta.$$

Finally, a union bound on p yields

$$\mathbb{P}(\mathcal{E}_T^c) \leq \frac{1}{KT} + \lg(T)KM\delta \leq \frac{1}{MT} + \frac{1}{MT}.$$

Appendix E. Lower Bounds or New Insights on Collisions

In the case of shared utilities across players (standard multi-player bandit problem, discussed in Section 3.2), Liu and Zhao (2010) and Besson and Kaufmann (2018) gave lower bounds on the regret by trying to push standard change-of-distribution arguments to the multi-player setting. However, Boursier and Perchet (2018) pointed out that those lower bound are wrong as they *cannot apply to every algorithm*. Here we clarify what is wrong or right about the lower bound in Besson and Kaufmann (2018). In particular, we provide a sufficient condition on the algorithm, that says in spirit “the collisions do not bring too much information on the arm means”, under which the existing lower bound is valid.

Consider Bernoulli arms with mean utilities μ_1, \dots, μ_K (shared by all players) and assume, to simplify the presentation, that $\mu_1 \geq \mu_2 \geq \dots \geq \mu_K$, with $\mu_M > \mu_{M+1}$. Letting $N_k^m(T)$ be the number of selections of arm k by player m , Besson and Kaufmann (2018) claim that, for each player m and each arm $k > M$, any uniformly efficient algorithm \mathcal{A} satisfies

$$\liminf_{T \rightarrow \infty} \frac{\mathbb{E}_\mu[N_k^m(T)]}{\log(T)} \geq \frac{1}{\text{kl}(\mu_k, \mu_M)}. \quad (9)$$

Fix a player m and consider the observations \mathcal{O}_t gathered by this player after t round of an algorithm:

$$\mathcal{O}_t = (U_1, Y_1, C_1, \dots, U_t, Y_t, C_t),$$

where $Y_t := Y_{A^m(t),t}$ denotes the sensing information, $C_t := \mathbb{1}_{C_{A^m(t),t}}$ denotes the collision information and U_t denotes some external source of randomness useful to select $A^m(t+1)$. By definition of a sequential strategy, $A^m(t+1)$ is $\sigma(\mathcal{O}_t)$ -measurable. The (wrong) lower bound (9) is obtained by combining two steps, among which we highlight that the first is correct.

The first step is to introduce, for $\epsilon > 0$, the alternative model parameterized by λ such that

$$\begin{cases} \lambda_\ell = \mu_\ell & \text{for all } \ell \neq k, \\ \lambda_k = \mu_{M^*} + \epsilon. \end{cases}$$

This model is such that $M\text{-best}_\mu = \{1, \dots, M\}$ and $M\text{-best}_\lambda = \{1, \dots, (M-1), k\}$. Using information theoretic arguments given in [Garivier et al. \(2016\)](#) shows that for any event \mathcal{E}_T that is $\sigma(\mathcal{O}_T)$ measurable,

$$\text{KL} \left(\mathbb{P}_\mu^{\mathcal{O}_T}, \mathbb{P}_\lambda^{\mathcal{O}_T} \right) \geq \text{kl} \left(\mathbb{P}_\mu(\mathcal{E}_T), \mathbb{P}_\lambda(\mathcal{E}_T) \right),$$

where $\mathbb{P}_\mu^{\mathcal{O}_T}$ (resp. $\mathbb{P}_\lambda^{\mathcal{O}_T}$) is the distribution of the vector \mathcal{O}_T under the model μ (resp. λ) when algorithm \mathcal{A} is applied and KL denotes the Kullback-Leibler divergence. Now the event

$$\mathcal{E}_T = \left(N_k^m(T) > \frac{T}{2M} \right)$$

is supposed to have a small probability under μ (under which k is sub-optimal) and a large probability under λ (under which k is one of the optimal arms, and is likely to be drawn a lot). More precise arguments detailed in [Besson and Kaufmann \(2018\)](#) permit to show that

$$\liminf_{T \rightarrow \infty} \frac{\text{kl} \left(\mathbb{P}_\mu(\mathcal{E}_T), \mathbb{P}_\lambda(\mathcal{E}_T) \right)}{\log(T)} \geq 1,$$

which yields

$$\liminf_{T \rightarrow \infty} \frac{\text{KL} \left(\mathbb{P}_\mu^{\mathcal{O}_T}, \mathbb{P}_\lambda^{\mathcal{O}_T} \right)}{\log(T)} \geq 1. \quad (10)$$

We emphasize here that the statement (10) may still be useful to derive a lower bound. The wrong conclusion in [Besson and Kaufmann \(2018\)](#) came from the second step, which is the computation of the Kullback-Leibler divergence $\text{KL} \left(\mathbb{P}_\mu^{\mathcal{O}_t}, \mathbb{P}_\lambda^{\mathcal{O}_t} \right)$. We detail (correct) computation of this quantity now, that rely on the chain rule for KL-divergence, computing some terms and using induction (more details can

be found in [Besson and Kaufmann \(2018\)](#)):

$$\begin{aligned}
 \text{KL}(\mathbb{P}_{\boldsymbol{\mu}}^{\mathcal{O}_T}, \mathbb{P}_{\boldsymbol{\lambda}}^{\mathcal{O}_T}) &= \text{KL}(\mathbb{P}_{\boldsymbol{\mu}}^{\mathcal{O}_{T-1}}, \mathbb{P}_{\boldsymbol{\lambda}}^{\mathcal{O}_{T-1}}) + \text{KL}(\mathbb{P}_{\boldsymbol{\mu}}^{Y_T, C_T, U_T | \mathcal{O}_{T-1}}, \mathbb{P}_{\boldsymbol{\lambda}}^{Y_T, C_T, U_T | \mathcal{O}_{T-1}}) \\
 &= \text{KL}(\mathbb{P}_{\boldsymbol{\mu}}^{\mathcal{O}_{T-1}}, \mathbb{P}_{\boldsymbol{\lambda}}^{\mathcal{O}_{T-1}}) + \text{KL}(\mathbb{P}_{\boldsymbol{\mu}}^{Y_T | \mathcal{O}_{T-1}}, \mathbb{P}_{\boldsymbol{\lambda}}^{Y_T | \mathcal{O}_{T-1}}) + \text{KL}(\mathbb{P}_{\boldsymbol{\mu}}^{C_T | \mathcal{O}_{T-1}}, \mathbb{P}_{\boldsymbol{\lambda}}^{C_T | \mathcal{O}_{T-1}}) \\
 &= \text{KL}(\mathbb{P}_{\boldsymbol{\mu}}^{\mathcal{O}_{T-1}}, \mathbb{P}_{\boldsymbol{\lambda}}^{\mathcal{O}_{T-1}}) + \mathbb{E}_{\boldsymbol{\mu}} \left[\sum_{\ell=1}^K \mathbb{1}_{(A^m(T)=\ell)} \text{kl}(\mu_{\ell}, \lambda_{\ell}) \right] + \text{KL}(\mathbb{P}_{\boldsymbol{\mu}}^{C_T | \mathcal{O}_{T-1}}, \mathbb{P}_{\boldsymbol{\lambda}}^{C_T | \mathcal{O}_{T-1}}) \\
 &= \sum_{\ell=1}^K \mathbb{E}[N_{\ell}^m(T)] \text{kl}(\mu_{\ell}, \lambda_{\ell}) + \sum_{t=1}^T \text{KL}(\mathbb{P}_{\boldsymbol{\mu}}^{C_t | \mathcal{O}_{t-1}}, \mathbb{P}_{\boldsymbol{\lambda}}^{C_t | \mathcal{O}_{t-1}}) \\
 &= \mathbb{E}[N_k^m(T)] \text{kl}(\mu_k, \mu_M + \epsilon) + \sum_{t=1}^T \text{KL}(\mathbb{P}_{\boldsymbol{\mu}}^{C_t | \mathcal{O}_{t-1}}, \mathbb{P}_{\boldsymbol{\lambda}}^{C_t | \mathcal{O}_{t-1}}),
 \end{aligned}$$

where the last step uses that in the alternative model $\boldsymbol{\lambda}$ there is only a single arm that is modified, hence $\text{kl}(\mu_{\ell}, \lambda_{\ell}) = 0$ except for $\ell = k$.

We introduce the collision information term for algorithm \mathcal{A} as

$$\mathcal{I}_{\boldsymbol{\mu}, \boldsymbol{\lambda}}(\mathcal{A}, T) := \sum_{t=1}^T \text{KL}(\mathbb{P}_{\boldsymbol{\mu}}^{C_t | \mathcal{O}_{t-1}}, \mathbb{P}_{\boldsymbol{\lambda}}^{C_t | \mathcal{O}_{t-1}}).$$

The wrong claim in [Besson and Kaufmann \(2018\)](#) is that the collision information term is zero for any algorithm. However, Lemma 11 below provides a sufficient condition on the algorithm \mathcal{A} for the lower bound (9) to still be correct. This condition is that the collision information term is negligible with respect to $\log(T)$. As the lower bound (9) is violated by the SIC-MMAB algorithm of [Boursier and Perchet \(2018\)](#), this algorithm doesn't have a negligible information term. This is expected as under SIC-MMAB the collisions depend on the empirical means of other arms, which vary between two models $\boldsymbol{\mu}$ and $\boldsymbol{\lambda}$. However, the condition in Lemma 11 may be true for other algorithms, like the Rand-TopM algorithm of [Besson and Kaufmann \(2018\)](#).

Lemma 11 *Any uniformly efficient algorithm that satisfies $\mathcal{I}_{\boldsymbol{\mu}, \boldsymbol{\lambda}}(\mathcal{A}, T) = o(\log(T))$ satisfies that for any player m and sub-optimal arm k*

$$\liminf_{T \rightarrow \infty} \frac{\mathbb{E}_{\boldsymbol{\mu}}[N_k^m(T)]}{\log(T)} \geq \frac{1}{\text{kl}(\mu_k, \mu_M)}.$$

Proof From the above, for all $\epsilon > 0$ we have

$$\begin{aligned}
 \liminf_{T \rightarrow \infty} \frac{\mathbb{E}_{\boldsymbol{\mu}}[N_k^m(T)] \text{kl}(\mu_k, \mu_M + \epsilon)}{\log(T)} &= \liminf_{T \rightarrow \infty} \frac{\mathbb{E}_{\boldsymbol{\mu}}[N_k^m(T)] \text{kl}(\mu_k, \mu_M + \epsilon) + \mathcal{I}_{\boldsymbol{\mu}, \boldsymbol{\lambda}}(\mathcal{A}, T)}{\log(T)} \\
 &= \liminf_{T \rightarrow \infty} \frac{\text{KL}(\mathbb{P}_{\boldsymbol{\mu}}^{\mathcal{O}_T}, \mathbb{P}_{\boldsymbol{\lambda}}^{\mathcal{O}_T})}{\log(T)} \geq 1
 \end{aligned}$$

Hence, for all $\epsilon > 0$,

$$\liminf_{T \rightarrow \infty} \frac{\mathbb{E}_{\boldsymbol{\mu}}[N_k^m(T)]}{\log(T)} \geq \frac{1}{\text{kl}(\mu_k, \mu_M + \epsilon)}$$

and the conclusion follows by letting ϵ go to zero. ■