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

Etienne Boursier, Emilie Kaufmann, Abbas Mehrabian, Vianney Perchet

To cite this version:

Etienne Boursier, Emilie Kaufmann, Abbas Mehrabian, Vianney Perchet. A Practical Algorithm for Multiplayer Bandits when Arm Means Vary Among Players. 2019. hal-02006069v2

HAL Id: hal-02006069

https://hal.science/hal-02006069v2

Preprint submitted on 5 Jun 2019 (v2), 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.

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

Etienne Boursier
CMLA, ENS Paris-Saclay
boursier@cmla.ens-cachan.fr

Emilie Kaufmann
Univ. Lille, CNRS, Inria SequeL,
UMR 9189 - CRIStAL, Lille, France
emilie.kaufmann@univ-lille.fr

Abbas Mehrabian
School of Computer Science
McGill University
Montréal, Canada
abbas.mehrabian@gmail.com

Vianney Perchet
CMLA, ENS Paris-Saclay
Criteo AI Lab, Paris
vianney.perchet@normalesup.org

Abstract

We study a multiplayer stochastic multi-armed bandit problem in which players cannot communicate, and if two or more players pull the same arm, a collision occurs and the involved players receive zero reward. We consider the challenging *heterogeneous* setting, in which different arms may have different means for different players, and propose a new, efficient algorithm that combines the idea of leveraging forced collisions for implicit communication and that of performing matching eliminations. We give a finite-time analysis of our algorithm, bounding its regret by $O((\log T)^{1+\kappa})$ for any fixed $\kappa > 0$. If the optimal assignment of players to arms is unique, we further show that it attains the optimal $O(\log(T))$ regret, solving an open question raised at NeurIPS 2018 [7].

1 Introduction

Stochastic multi-armed bandit models have been studied extensively 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 ith 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 with the actions before starting the game, she would play an action with the largest mean reward 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 looks *probably* best given her current information. We refer the reader to [10, 18] for surveys on this problem. Multi-armed bandit (MAB) have been first studied as a simple model for sequential clinical trials, see [26, 24], but have also found many modern applications to online content optimization, such as designing recommender systems, see [19]. In the meantime, MAB models have also been used for cognitive radio problems, see [15, 1]. In this context, arms model different radio channels on which each device can communicate, and the reward associated to each arm is either a binary indicator of the success of the communication or some measure of its quality.

Preprint. Under review.

The applications to cognitive radios have motivated the *multiplayer* 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 agents pulling that arm receive zero reward. Without communicating, each agent must 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 most previous work on this problem focuses on the case where the mean of the arms are identical across players (the homogeneous variant), in this paper we study the more challenging heterogeneous 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 radios, as the quality of each channel may vary from one user (device) to another, depending for instance on its configuration and its location.

More precisely, we study the model introduced by [7], which has two main characteristics: first, each arm has a possibly different mean for each player, and second, we are in a fully distributed setting with no communication allowed between players. The authors of [7] proposed an algorithm with regret bounded by $O((\log T)^{2+\kappa})$ (for any constant κ), proved a lower bound of $\Omega(\log T)$, and asked if there is an algorithm matching this lower bound. In this paper, we propose a new algorithm for this model, called M-ETC-Elim, which depends on a hyperparameter c, and we upper bound its regret by $O(\log(T)^{1+1/c})$ for any $c > 1$. Moreover, if the optimal assignment of the players to the arms is unique, we prove that instantiating M-ETC-Elim with $c = 1$ yields regret at most $O(\log(T))$, which is optimal and answers affirmatively the open question mentioned above in this particular case. We present a non-asymptotic regret analysis of M-ETC-Elim leading to nearly optimal regret upper bounds, and also demonstrate the empirical efficiency of this new algorithm via simulations.

Paper organization. In Section 2 we formally introduce the heterogeneous multiplayer multi-armed bandit model and give a detailed presentation of our contributions. These results are put in perspective by comparison with the literature, given in Section 3. We describe the M-ETC-Elim algorithm in Section 4 and upper bound its regret in Section 5. Finally, we report in Section 6 results from an experimental study demonstrating the competitive practical performance of M-ETC-Elim.

2 The Model and Our Contributions

We study a multi-armed bandit model 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

$$R^m(t) = Y^m_{A^m(t),t}\left(1 - \mathbb{1}\left(\mathcal{C}_{A^m(t),t}\right)\right),$$

where $(Y^m_{k,t})_{t=1}^\infty$ is an i.i.d. sequence with mean μ_k^m taking values in $[0, 1]$, $\mathcal{C}_{k,t}$ is the event that at least two players have chosen arm k in round t, and $\mathbb{1}\left(\mathcal{C}_{k,t}\right)$ is the corresponding indicator function. In the cognitive radio context, $Y^m_{k,t}$ models the quality of channel k for player m if she were to use this channel in isolation in round t, but her actual rewards is set to zero if a collision occurs.

We assume that player m in round t observes her reward $R^m(t)$ and the collision indicator $\mathbb{1}\left(\mathcal{C}_{A^m(t),t}\right)$. Note that in the special case in which the reward distributions satisfy $\mathbb{P}(Y^m_{k,t} = 0) = 0$ (e.g., if the corresponding distribution is continuous), $\mathbb{1}\left(\mathcal{C}_{A^m(t),t}\right)$ can be reconstructed from the observation of $R^m(t)$. The decision of player m at round t can be based only on her past observations, that is, $A^m(t)$ is $\mathcal{F}^m_{t-1}$ measurable, where $\mathcal{F}^m_t = \sigma\left(A^m(1), R^m(1), \mathbb{1}\left(\mathcal{C}_{A^m(1),1}\right), \dots, A^m(t), R^m(t), \mathbb{1}\left(\mathcal{C}_{A^m(t),t}\right)\right)$. Hence, our setting is fully distributed: a player cannot use extra information such as observations made by others to make her decisions. Under this constraint, we aim at maximizing the global reward collected by all players.

We use the shorthand $[n] \coloneqq \{1, \dots, n\}$. A *matching* is a one-to-one assignment of players to arms; formally, any one-to-one function $\pi : [M] \to [K]$ is a matching, and the utility (or weight) of a matching π is defined as $U(\pi) \coloneqq \sum_{m=1}^M \mu^m_{\pi(m)}$. We denote by $\mathcal{M}$ the set of all matchings and let $U^* \coloneqq \max_{\pi \in \mathcal{M}} U(\pi)$ denote the maximum attainable utility. A *maximum matching* (or *optimal matching)* is a matching with utility U^*. 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 sequence of arm pulls) for each player m that attains the smallest possible regret. Our regret bounds will depend on the gap between the utility of the best matching and the utility of the matching with the second best utility, defined as $\Delta := \min_{\pi:\Delta(\pi)>0} \Delta(\pi)$, where $\Delta(\pi) := U^* - U(\pi)$. Note that $\Delta > 0$ even in the presence of several optimal matchings.

Our contributions. We propose a new, efficient algorithm for the heterogeneous multiplayer bandit problem achieving (quasi) logarithmic regret. The algorithm, called Multiplayer Explore-Then-Commit with matching Elimination (M-ETC-Elim) is described in details in Section 4. It combines the idea of exploiting collisions for implicit communication, initially proposed by [9] for the homogeneous setting (which we have improved and adapted to our setting), with an efficient way to perform 'matching eliminations'. M-ETC-Elim consists of several epochs combining exploration and communication, and may end with an exploitation phase if a unique optimal matching has been found. The algorithm depends on a parameter $c \geq 1$ controlling the epochs sizes and enjoys the following regret guarantees.

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

$$R_T = O\left(MK\left(\frac{M^2\ln(KT)}{\Delta}\right)^{1+1/c}\right).$$

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

$$R_T = O\left(\frac{M^3K\ln(KT)}{\Delta}\right).$$

A consequence of part (a) is that for a fixed problem instance, for any (arbitrarily small) κ, there exists an algorithm (M-ETC-Elim with parameter $c = \lceil 1/\kappa \rceil$) with regret $R_T = O((\log(T))^{1+\kappa})$. We would like to emphasize that we carry out a non-asymptotic analysis of M-ETC-Elim. The regret bounds of Theorem 1 are stated with the $O(\cdot)$ notation for the ease of presentation and the hidden constants depend on the chosen parameter c only. In Theorems 3 and 8 we provide the counterparts of these results with explicit constants.

To summarize, we present a unified algorithm that can be used in the presence of either unique or multiple optimal matchings and get a nearly logarithmic regret in both cases, almost matching the known logarithmic lower bound. Moreover, our algorithm is easy to implement, performs well in practice as shown in Section 6 and does not need problem-dependent hyperparameter tuning.

3 Related Work

Centralized variant. Relaxing the need for decentralization, i.e., when a central controller is jointly selecting $A^1(t), \dots, A^M(t)$, our problem coincides with a combinatorial bandit problem with semi-bandit feedback, a setup first studied by [13]. More precisely, introducing $M \times K$ *elementary arms* with means μ_m^k for $m \in [M]$ and $k \in [K]$, the central controller selects at each time step M elementary arms whose indices form a matching. Then, the utility of each chosen elementary arm is observed and the obtained reward is their sum. A well-known algorithm for this setting is CUCB [30], whose regret satisfies $R_T = O\left((M^2K/\Delta)\log(T)\right)$ [17] (see also [29] for a Thompson sampling-based algorithm with similar regret). Improved dependency in M was obtained for the ESCB algorithm [11, 12], which is less numerically appealing as it requires to compute an upper confidence bound for each matching at every round. In this work, we propose an efficient algorithm with regret upper bounded by (roughly) $O\left((M^3K/\Delta)\log(T)\right)$ for the more challenging decentralized setting.

Homogeneous variant. Back to the decentralized setting, the particular case in which all players share a common utility for all arms, i.e. $\mu_k^m = \mu_k$ for all $m \in [M]$, has been studied extensively: the first line of work on this variant combines standard bandit algorithms with an orthogonalization mechanism [20, 1, 5], and obtains logarithmic regret, with a large multiplicative constant due to the number of collisions. [25] proposes 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. Drawing inspiration from this musical chairs protocol, [9] recently proposed an algorithm with an $O\left(((K-M)/\Delta + KM)\log(T)\right)$ regret, which relies on two other crucial ideas: *exploiting collisions for communication* and *performing arm eliminations*. Our algorithm also leverages these two ideas, with the following enhancements. The main advantage of our communication protocol over that of [9] is that the followers only send each piece of information once, to the leader, instead of sending it to the $M-1$ other players. Then, while [9] uses *arm eliminations* (coordinated between players) to reduce the regret, we cannot employ the same idea for our heterogeneous problem, as an arm that is bad for one player might be good for another player, and therefore cannot be eliminated. Our algorithm instead relies on *matching eliminations*.

Towards the fully distributed and heterogeneous setting. Various semi-distributed variants of our problem in which some kind of communication is allowed between players have been studied by [4, 16, 23]. In particular, the algorithms proposed by [16, 23] require a pre-determined channel dedicated to communications: in some phases of the algorithm, players in turn send information (sequences of bits) on this channel, and it is assumed that all other players can "listen" and have access to the sent information.

The fully distributed setting was first studied by [7], who proposed the Game-of-Thrones (GoT) algorithm and proved its regret is bounded by $O((\log T)^{2+\kappa})$ for any given constant $\kappa > 0$, if its parameters are *appropriately tuned*. In a recent preprint [8], the same authors provide an improved analysis, showing the same algorithm (with slightly modified phase lengths) enjoys quasi-logarithmic regret $O((\log T)^{1+\kappa})$. GoT is quite different from M-ETC-Elim: it proceeds in epochs, each consisting of an exploration phase, a so-called GoT phase and an exploitation phase. During the GoT phase, the players jointly run a Markov chain whose unique stochastically stable state corresponds to a maximum matching of the estimated means. A parameter $\varepsilon \in (0, 1)$ controls the accuracy of the estimated maximum matching obtained after a GoT phase. Letting c_1, c_2, c_3 be the constants parameterizing the lengths of the phases, the improved analysis of GoT [8] upper bounds its regret by $Mc_3 2^{k_0+1} + 2(c_1 + c_2)M \log_2^{1+\kappa}(T/c_3 + 2)$. This upper bound is asymptotic as it holds for T *large enough*, where 'how large' is not explicitly specified and *depends on* Δ.[1] Moreover, the upper bound is valid only when the parameter ε is chosen *small enough*: ε should satisfy some complicated constraints (Equation (66)-(67)) also featuring Δ. Hence, a valid tuning of the parameter ε would require prior knowledge on the arms utilities. In contrast, we provide in Theorem 3 a non-asymptotic regret upper bound for M-ETC-Elim, which holds for any choice of the parameter c controlling the epochs lengths. Also, we show that if the optimal assignment is unique, M-ETC-Elim has logarithmic regret. Besides, we also illustrate in Section 6 that M-ETC-Elim outperforms GoT in practice. Finally, GoT has several parameters to set ($\delta, \varepsilon, c_1, c_2, c_3$), while M-ETC-Elim has only one integral parameter c, and setting $c = 1$ works very well in all our experiments.

Finally, we would like to mention the recent independent preprint [27]. Although this work studies a slightly stronger feedback model,[2] the proposed algorithms share similarities with M-ETC-Elim: they also have exploration, communication and exploitation phases, yet without eliminations. We elaborate in Appendix E on our positioning with respect to this work.

4 Description of the M-ETC-Elim algorithm

Our algorithm relies on an initialization phase, in which the players elect a leader in a distributed manner. Then a communication protocol is set up, in which the leader and the followers have different roles: followers explore some arms and communicate to the leader estimates of the arm means, while the leader maintains a list of candidate optimal matchings, and communicates to the followers the list of arms that need exploration in order to refine the list, i.e. to eliminate some candidate matchings. The algorithm is called *Multiplayer Explore-Then-Commit with matching Eliminations* (M-ETC-Elim for short). Formally, each player executes the following Algorithm 1.

[1] [8, Theorem 4] requires $T \geq c_3(2^{k_0} - 2)$, where k_0 satisfies Equation (16) featuring κ and Δ.

[2] In their model, each player has the option of 'observing whether a given arm has been pulled by someone.'

Algorithm 1: M-ETC-Elim with parameter c

Input: Time horizon T, number of arms K

$R, M \longleftarrow \textsc{Init}(K, 1/KT)$

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

M-ETC-Elim requires as input the number of arms K (as well as a shared numbering of the arms across the players) and the time horizon T (the total number of arm selections). However, if the players only know an upper bound for T, our results hold with T replaced by that upper bound as well. If no upper bound for T is known, the players can employ a simple doubling trick [6]: we execute the algorithm assuming $T = 1$, then we execute it assuming $T = 2 \times 1$, and so on, until the actual time horizon is reached. If the expected regret of the algorithm for a known time horizon T is R_T, then the expected regret of the modified algorithm for unknown time horizon T would be $R'_T \leq \sum_{i=0}^{\log_2(T)} R_{2^i} \leq \log_2(T) \times R_T$.

Initialization. The initialization procedure, borrowed from [9], outputs for each player a rank $R \in [M]$ as well as the value of M, which is initially unknown to the players. This initialization phase relies on a "musical chairs" phase after which the players end up on distinct arms, followed by a so-called Sequential Hopping protocol that permits them to know their ordering. For the sake of completeness, it is described in detail in Appendix A, where we also prove the following.

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

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

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

In case of unreliable channels where some of the communicated bits may be lost, there are several options to make this communication protocol more robust, such as sending each bit multiple times or using the Bernoulli signaling protocol of [27]. Robustness has not been the focus of our work.

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

Both algorithms proceed in epochs whose length is parameterized by c. In epoch p, the leader weights the edges using the estimated weights. Then for every edge $(m, k) \in \mathcal{E}$, the leader computes the associated matching $\widetilde{\pi}_p^{m,k}$ defined as the maximum matching containing the edge (m, k). This computation can be done in polynomial time using, e.g., the Hungarian algorithm [22]. The leader then computes the utility of the maximum matching and eliminates from $\mathcal{E}$ any edge for which the weight of its associated matching is smaller by at least $4M\varepsilon_p$, where

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

The leader then forms the set of associated candidate matchings $\mathcal{C} := \{\widetilde{\pi}_p^{m,k}, (m,k) \in \mathcal{E}\}$ and communicates to each follower the list of arms to explore in these matchings. Then exploration begins, with each player assigned to her arm in each matching. For each of them, the player pulls that arm 2^{p^c} times and records the reward received. Then another communication phase begins, during which each follower sends her observed estimated mean for the arms to the leader. More precisely, for each explored arm, the follower truncates the estimated mean (a number in $[0,1]$) and sends only the $\frac{p^c+1}{2}$ most significant bits of this number to the leader. The leader updates the estimated weights and everyone proceeds to the next epoch. If at some point the list of candidate matchings $\mathcal{C}$ becomes a singleton, it means that the real maximum matching is unique and has been found; so all players jointly pull that matching for the rest of the game (the exploitation phase). The pseudocode for the leader's algorithm is presented below, while the corresponding follower algorithm appears in Appendix A. In the pseudocodes, `(comm.)` refers to a call to the communication protocol.

Note that in the presence of several optimal matchings, the players will not enter the exploitation phase but will keep exploring several optimal matchings, which still ensures small regret. Also, observe that $\mathcal{C}$ does not necessarily contain *all* potentially optimal matchings, but all the edges in those matchings remain in $\mathcal{E}$ and are guaranteed to be explored.

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

---

Input: Number of players M

$\mathcal{E} \longleftarrow [M] \times [K]$ `// list of candidate edges`
$\widetilde{\mu}_k^m \longleftarrow 0$ for all $(m,k) \in [M] \times [K]$ `// empirical estimates for utilities`
for $p = 1, 2, \dots$ **do**
  $\mathcal{C} \longleftarrow \emptyset$ `// list of associated matchings`
  $\pi_1 \longleftarrow \text{argmax} \left\{ \sum_{m=1}^{M} \widetilde{\mu}_{\pi(m)}^m : \pi \in \mathcal{M} \right\}$ `// using Hungarian algorithm`
  for $(m,k) \in \mathcal{E}$ **do**
    $\pi \longleftarrow \text{argmax} \left\{ \sum_{n=1}^{M} \widetilde{\mu}_{\pi(n)}^n : \pi(m) = k \right\}$ `// using Hungarian algorithm`
    if $\sum_{n=1}^{M} \left\{ \widetilde{\mu}_{\pi_1(n)}^n - \widetilde{\mu}_{\pi(n)}^n \right\} \leq 4M \times \sqrt{\ln(2M^2KT^2)/2^{1+p^c}}$ **then** add π to $\mathcal{C}$
    else remove (m,k) from $\mathcal{E}$
  end
  for *each player* $m = 2, \dots, M$ **do**
    Send to player m the value of size($\mathcal{C}$) `// (comm.)`
    for $i = 1, 2, \dots, \text{size}(\mathcal{C})$ **do**
      Send to player m the arm associated to player m in $\mathcal{C}[i]$ `// (comm.)`
    end
    Send to player m the communication arm of the leader, $\pi_1(1)$
  end
  if $size(\mathcal{C}) = 1$ **then** pull for the rest of the game the arm associated to player 1 in the unique matching in $\mathcal{C}$ `// enter the exploitation phase`
  for $i = 1, 2, \dots, \text{size}(\mathcal{C})$ **do**
    pull 2^{p^c} times the arm associated to player 1 in the matching $\mathcal{C}[i]$ `// exploration`
  end
  for $k = 1, 2, \dots, K$ **do**
    $\widetilde{\mu}_k^1 \longleftarrow$ empirically estimated utility of arm k if it was pulled in this epoch, 0 otherwise
  end
  Receive the values $\widetilde{\mu}_1^m, \widetilde{\mu}_2^m, \dots, \widetilde{\mu}_K^m$ from each player m `// (comm.)`
end

---

5 General Finite-Time Analysis of M-ETC-Elim

In the sequel, $\ln(\cdot)$ denotes the natural logarithm and $\lg(\cdot)$ denotes logarithm in base 2. Theorem 3 below provides a non-asymptotic upper bound on the regret of M-ETC-Elim.

Theorem 3. *For any positive integer c, the M-ETC-Elim algorithm with parameter c, has its regret at time $T \geq T_0(c) := \exp(2^{\frac{c^c}{\ln^c(1+\frac{1}{2c})}})$ upper bounded by*

$$2 + MK\ln(e^2K^2T) + 6M^2K\lg(K)(\lg T)^{1/c} + e^2MK(\lg T)^{1+1/c} + \frac{2M^3K\lg(K)}{\sqrt{2}-1}\sqrt{\ln(2M^2KT^2)}$$
$$+ \frac{2\sqrt{2}}{3-2\sqrt{2}}M^2K\sqrt{\ln(2M^2KT^2)}\lg(\ln(T)) + \frac{2\sqrt{2}-1}{\sqrt{2}-1}\sum_{(m,k)\in[M]\times[K]}\left(\frac{32M^2\ln(2M^2KT^2)}{\Delta(\pi^{m,k})}\right)^{1+1/c},$$

where $\pi^{m,k}$ is the best sub-optimal matching attributing arm k to player m, namely $\pi^{m,k} \in$ argmax $\{U(\pi) : \pi(m) = k$ and $U(\pi) < U^*\}$.

Statement (a) in Theorem 1 easily follows by lower bounding all gaps by Δ. Statement (b) in Theorem 1 similarly follows from Theorem 8, stated and proved in Appendix C. The constant $T_0(c)$ in Theorem 3 is equal to 252 for $c = 1$ but becomes significantly larger when c increases. Still, the condition on T is explicit and independent of the problem parameters.

In the rest of this section, we give a proof of Theorem 3, which has several intermediate lemmas, whose proofs are delayed to Appendix D. We first introduce useful notations. Let $\mathcal{C}_p$ denote the set of candidate matchings used in epoch p, and for each matching π let $\tilde{U}_p(\pi)$ be the utility of π that the leader can estimate based on the information received at the end of epoch p. Let $\hat{p}_T$ be the total number of epochs before the (possible) start of the exploitation phase. As $2^{\hat{p}_T^c} \leq T$, we have $\hat{p}_T \leq \lg(T)$. Recall that a successful initialization means that all players identify M, and that their ranks are distinct. We introduce the *good event*

$$\mathcal{G}_T = \left\{\textsc{Init}(K, 1/KT) \text{ is successful and } \forall p \leq \hat{p}_T, \forall \pi \in \mathcal{C}_{p+1}, |\tilde{U}_p(\pi) - U(\pi)| \leq 2M\varepsilon_p\right\}. \quad (2)$$

During epoch p, for each candidate edge (m, k), player m has pulled arm k at least 2^{p^c} times and the quantization error is smaller than ε_p. Hoeffding's inequality and a union bound over at most $\lg(T)$ epochs (see Appendix D.1) together with Lemma 2 yield that $\mathcal{G}_T$ holds with large probability.

Lemma 4. $\mathbb{P}(\mathcal{G}_T) \geq 1 - \frac{2}{MT}$.

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

Suppose that $\mathcal{G}_T$ happens. First, the regret incurred during the initialization phase is upper bounded by $MK\ln(e^2K^2T)$ by Lemma 2. Then, using the best estimated matching as communication arms yields regret at most $2 + 2M\varepsilon_{p-1}$ in each communication round of epoch p, which is used to prove Lemma 5 below, which upper bounds the regret incurred during all communication phases.

Lemma 5. *The regret due to communications is bounded by:*

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

For large horizons, Lemma 6 bounds some terms such as $\hat{p}_T$ and $(\hat{p}_T)^c$.

Lemma 6. *For any sub-optimal matching π, let $P(\pi)$ be the smallest positive integer such that $8M\varepsilon_{P(\pi)} < \Delta(\pi)$. The assumption $T \geq T_0(c)$ implies for any matching π that $\Delta(\pi)2^{P(\pi)^c} \leq \left(\frac{32M^2\ln(2M^2KT^2)}{\Delta(\pi)}\right)^{1+\frac{1}{c}}$. Also, $2^c \leq 2\lg(\ln(T))$, $\hat{p}_T \leq 2(\lg T)^{1/c}$ and $(\hat{p}_T)^c \leq e\lg T$.*

Hence for $T \geq T_0(c)$, we can further upper bound the first three terms of the sum in Lemma 5 by

$$6M^2K\lg(K)(\lg T)^{1/c} + e^2MK(\lg T)^{1+1/c} + \frac{2\sqrt{2}}{3-2\sqrt{2}}M^2K\sqrt{\ln(2/\delta)}\lg(\ln(T)). \quad (3)$$

It then remains to upper bound the regret incurred during exploration and exploitation phases. On $\mathcal{G}_T$, the players are jointly pulling a matching during those phases, and no regret is incurred during the exploitation phase. For an edge (m,k), let $\widetilde{\Delta}_p^{m,k} := U^* - U(\widetilde{\pi}_p^{m,k})$ be the gap of its associated matching at epoch p. During any epoch p, the incurred regret is then $\sum_{\pi\in\mathcal{C}_p}\Delta(\pi)2^{p^c} = \sum_{(m,k)\in\mathcal{E}}\widetilde{\Delta}_p^{m,k}2^{p^c}$. We recall that $\pi^{m,k}$ is the best sub-optimal matching attributing the arm k to the player m. Observe that for any epoch $p > P(\pi^{m,k})$, since $\mathcal{G}_T$ happens, $\pi^{m,k}$ (and any worse matching) is not added to $\mathcal{C}_p$; thus during any epoch $p > P(\pi^{m,k})$, the edge (m,k) is either eliminated from the set of candidate edges, or it is contained in some optimal matching and satisfies $\widetilde{\Delta}_p^{m,k} = 0$. Hence, the total regret incurred during exploration phases is bounded by

$$\sum_{(m,k)\in[M]\times[K]}\sum_{p=1}^{P(\pi^{m,k})}\widetilde{\Delta}_p^{m,k}2^{p^c}. \tag{4}$$

The difficulty for bounding this sum is that $\widetilde{\Delta}_p^{m,k}$ depends on p and is random, since $\widetilde{\pi}_p^{m,k}$ is random. Yet, a convexity argument allows us to overcome this and relate $\widetilde{\Delta}_p^{m,k}$ to $\Delta(\pi^{m,k})$:

Lemma 7. *For any edge (m,k), if $p \le P(\pi^{m,k}) - 1$, then $\widetilde{\Delta}_p^{m,k}2^{p^c} \le \Delta(\pi^{m,k})\frac{2^{P(\pi^{m,k})^c}}{\sqrt{2}^{P(\pi^{m,k})-(p+1)}}$.*

From Lemma 7, we have $\displaystyle\sum_{p=1}^{P(\pi^{m,k})}\widetilde{\Delta}_p^{m,k}2^{p^c} \le \left(\sum_{p=0}^{\infty}\frac{1}{\sqrt{2}^p}\right)\Delta(\pi^{m,k})2^{P(\pi^{m,k})^c} + \widetilde{\Delta}^{m,k}_{P(\pi^{m,k})}2^{P(\pi^{m,k})^c}$.
As $\widetilde{\pi}^{m,k}_{P(\pi^{m,k})}$ is either optimal or its gap is larger than $\Delta(\pi^{m,k})$, from Lemma 6 we have that $\widetilde{\Delta}^{m,k}_{P(\pi^{m,k})}2^{P(\pi^{m,k})^c} \le \left(\frac{32M^2\ln(2M^2KT^2)}{\Delta(\pi^{m,k})}\right)^{1+1/c}$ in both cases. Therefore, we find that

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

Plugging this bound in (4), the bound (3) in Lemma 5 and summing up all terms yields Theorem 3.

6 Numerical Experiments

We executed the following algorithms: M-ETC-Elim with $c = 1$ and $c = 2$, GoT (the latest version in [8]) with parameters $\delta = 0, \varepsilon = 0.01, c_1 = 500, c_2 = c_3 = 6000$ (these parameters and the reward matrix U_1 below are from the simulations section of [8]), and Selfish-UCB, a heuristic studied by [5, 9] in the homogeneous case in which each player runs the UCB algorithm [3] on the reward sequence $(R^m(t))_{t=1}^{\infty}$ (which is not i.i.d.). We experiment with Bernoulli rewards and the reward matrices

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

where the entry (m,k) gives the value of μ_k^m. Figure 1 reports the algorithms' regrets for various time horizons T, averaged over 100 independent replications. The first instance (matrix U_1, left plot) has a unique optimal matching and we observe that M-ETC-Elim has logarithmic regret (as promised by Theorem 1) and largely outperforms all competitors. The second instance (matrix U_2, right plot) is a more challenging instance, with more arms and players, two optimal matchings and several near-optimal matchings. It can be observed that M-ETC-Elim with $c = 1$ performs the best for large T as well, though Selfish-UCB is also competitive. Yet there is very little theoretical understanding of Selfish-UCB. Appendix B contains the results of additional experiments corroborating our findings, where we also discuss some practical aspects of the implementation of M-ETC-Elim.

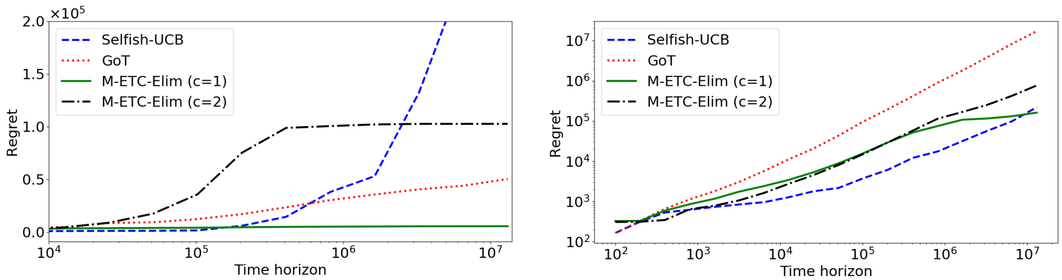

Figure 1: R_T as a function of T with reward matrices U_1 (left) and U_2 (right) and Bernoulli rewards.

7 Conclusion

In this paper, we presented a practical algorithm for the heterogeneous multiplayer bandit problem that can be used in the presence of either unique or multiple optimal matchings and get a nearly logarithmic regret in both cases, thus answering an open question of [7]. M-ETC-Elim crucially relies on the assumption that the collision indicators are observed in each round. In future work, we will investigate whether algorithms with logarithmic regret can be proposed when the players observe their rewards $R^m(t)$ only. So far, such algorithms only exist in the homogeneous setting [21, 9].

References

[1] 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.

[2] V. Anantharam, P. Varaya, and J. Walrand. Asymptotically efficient allocation rules for the multiarmed bandit problem with multiple plays-part I: I.I.D. rewards. *IEEE Transactions on Automatic Control*, 32(11):968–976, 1987.

[3] Auer. Using Confidence bounds for Exploration Exploitation trade-offs. *Journal of Machine Learning Research*, 3:397–422, 2002.

[4] 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.

[5] 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.

[6] Lilian Besson and Emilie Kaufmann. What doubling tricks can and can't do for multi-armed bandits. *arXiv.org:1803.06971*, 2018.

[7] Ilai Bistritz and Amir Leshem. Distributed multi-player bandits - a game of thrones approach. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[8] Ilai Bistritz and Amir Leshem. Game of thrones: Fully distributed learning for multi-player bandits. *arXiv.org:1810.11162v3*, 2019.

[9] Etienne Boursier and Vianney Perchet. SIC-MMAB: synchronisation involves communication in multiplayer multi-armed bandits. *arXiv.org:1809.08151*, 2018.

[10] 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.

[11] R. Combes, S. Talebi, A. Proutière, and M. Lelarge. Combinatorial bandits revisited. In *Advances in Neural Information Processing Systems (NIPS)*, 2015.

[12] Rémy Degenne and Vianney Perchet. Combinatorial semi-bandit with known covariance. In *Advances in Neural Information Processing Systems (NIPS)*, 2016.

[13] 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.

[14] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *J. Amer. Statist. Assoc.*, 58:13–30, 1963.

[15] 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.

[16] D. Kalathil, N. Nayyar, and R. Jain. Decentralized learning for multiplayer multiarmed bandits. *IEEE Transactions on Information Theory*, 60(4):2331–2345, April 2014.

[17] 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.

[18] Tor Lattimore and Csaba Szepesvári. *Bandit algorithms*. 2019. Book draft, available at `https://tor-lattimore.com/downloads/book/book.pdf`.

[19] 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.

[20] Keqin Liu and Qing Zhao. Distributed learning in multi-armed bandit with multiple players. *IEEE Trans. Signal Process.*, 58(11):5667–5681, 2010.

[21] Gábor Lugosi and Abbas Mehrabian. Multiplayer bandits without observing collision information. *arXiv preprint arXiv:1808.08416*, 2018.

[22] James Munkres. Algorithms for the assignment and transportation problems. *J. Soc. Indust. Appl. Math.*, 5:32–38, 1957.

[23] 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.

[24] H. Robbins. Some aspects of the sequential design of experiments. *Bulletin of the American Mathematical Society*, 58(5):527–535, 1952.

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

[26] 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.

[27] Harshvardhan Tibrewal, Sravan Patchala, Manjesh K. Hanawal, and Sumit J. Darak. Multiplayer multi-armed bandits for optimal assignment in heterogeneous networks. *arXiv.org:1901.03868v2*, 2019.

[28] Martin J Wainwright. *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge University Press, 2019.

[29] Siwei Wang and Wei Chen. Thompson sampling for combinatorial semi-bandits. In *International Conference on Machine Learning (ICML)*, 2018.

[30] Yang Yuan Wei Chen, Yajun Wang. Combinatorial multi-armed bandit: General framework and applications. In *International Conference on Machine Learning (ICML)*, 2013.

A Description of the Initialization Procedure and Followers' Pseudocode

The pseudocode of the $\textsc{Init}(K, \delta_0)$ procedure, first introduced by [9], is presented in Algorithm 2 for the sake of completeness. We now provide a proof of Lemma 2.

Algorithm 2: $\textsc{Init}$, the initialization algorithm

Input: number of arms K, failure probability δ_0
Output: Ranking R, number of players M
`// first, occupy a distinct arm using the musical chairs algorithm`
$k \longleftarrow 0$
for $T_0 := K \ln(K/\delta_0)$ *rounds* **do** `// rounds` $1, \ldots, T_0$
    if $k = 0$ **then**
        pull a uniformly random arm $i \in [K]$
        if *no collision occurred* **then** $k \longleftarrow i$ `// arm` k `is occupied`
    else
        pull arm k
    end
end
`// next, learn` M `and identify your ranking`
$R \longleftarrow 1$
$M \longleftarrow 1$
for $2k - 2$ *rounds* **do** `// rounds` $T_0 + 1, \ldots, T_0 + 2k - 2$
    pull arm k
    if *collision occurred* **then**
        $R \longleftarrow R + 1$
        $M \longleftarrow M + 1$
    end
end
for $i = 1, 2, \ldots, K - k$ **do** `// rounds` $T_0 + 2k - 1, \ldots, T_0 + K + k - 2$
    pull arm $k + i$
    if *collision occurred* **then**
        $M \longleftarrow M + 1$
    end
end
for $K - k$ *rounds* **do** `// rounds` $T_0 + K + k - 1, \ldots, T_0 + 2K - 2$
    pull arm 1
end

Let $T_0 := K \ln(K/\delta_0)$. During the first T_0 rounds, each player tries to occupy a distinct arm using the so-called musical chairs algorithm (first introduced in [25]): 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

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

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

Once each player has 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 $k + 1, k + 2, \ldots, K$, and can learn the 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 $1, \ldots, k - 1$ by counting the collisions during the waiting period; see Algorithm 2 for details. The crucial observation to verify the correctness of the algorithm is that two players occupying arms k_1 and k_2 will collide exactly once, and that happens at round $T_0 + k_1 + k_2 - 2$.

Next, we describe the pseudocode that the followers execute in M-ETC-Elim.

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

---

Input: Ranking R, number of players M

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

end

B Practical considerations and additional experiments

B.1 Implementation enhancements for M-ETC-Elim

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

First, to estimate the means, the players are better off taking into account all pulls of the arms, rather than just the last epoch. Note that after the exploration phase of epoch p, each candidate edge has been pulled $N_p := \sum_{i=1}^{p} 2^{i^c}$ times. Thus, with probability at least $1 - 2\lg(T)/(MT)$, each edge has been estimated within additive error $\leq \varepsilon'_p = \sqrt{\ln(M^2TK)/2N_p}$ by Hoeffding's inequality. The players then truncate these estimates using $b := \lceil -\lg(0.1\varepsilon'_p)\rceil$ bits, adding up to $0.1\varepsilon'_p$ additive error due to quantization. They then send these b bits to the leader. Now, the threshold for eliminating a matching would be $2.2M\varepsilon'_p$ rather than $4M \times \sqrt{\ln(2M^2KT^2)/2^{1+p^c}}$ (see line 8 of the LeaderAlgorithm presented on page 6).

Second, we choose the set $\mathcal{C}$ of 'matchings to explore' more carefully. Recall that a matching is a candidate if its estimated gap is at most $2.2M\varepsilon'_p$, and an edge is candidate (lies in $\mathcal{E}$) if it is part of some candidate matching. There are at most MK candidate edges, and we need only estimate those in the next epoch. Now, for each candidate edge, we can choose any good matching containing it, and add that to $\mathcal{C}$. This guarantees that $|\mathcal{C}| \leq MK$, which gives the bound in Theorem 1. But to reduce the size of $\mathcal{C}$ in practice, we do the following: initially, all edges are candidate. After each exploration phase, we do the following: we mark all edges as 'uncovered.' For each candidate uncovered edge e, we compute the maximum matching π' containing that edge (using estimated means). If this matching π' has gap larger than $2.2M\varepsilon'_p$, we remove it from the set of candidate edges. Otherwise, we add π' to $\mathcal{C}$, and moreover, we mark all of its edges as 'covered.' We then look at the next uncovered candidate edge, and continue similarly, until all candidate edges are covered. This guarantees that all the candidate edges are explored, while the number of explored matchings could be much smaller than the number of candidate edges.

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

B.2 Other reward distributions.

In our model and analysis, we have assumed that $Y_{k,t}^m \in [0,1]$ for simplicity (this is a standard assumption in online learning), but it is immediate to generalize the algorithm and its analysis to reward distributions bounded in any known interval. Also, we can adapt our algorithm and analysis to subgaussian distributions with mean lying in a known interval. A random variable X is σ-subgaussian if for all $\lambda \in \mathbb{R}$ we have $\mathbb{E}[e^{\lambda(X-\mathbb{E}X)}] \leq e^{\sigma^2\lambda^2/2}$. This includes in particular Gaussian distributions and distributions with bounded support. Suppose for simplicity that the means lie in $[0,1]$. Then the algorithm need only change in two places: first, when the followers are sending the estimated means to the leader, they must send 0 and 1 if the empirically estimated mean is < 0 and > 1, respectively. Second, the definition of ε_p must be changed to $\varepsilon_p := \sqrt{\sigma^2 \ln(2/\delta)/2^{p^c-1}}$. The only change in the analysis is that instead of using Hoeffding's inequality which requires a bounded distribution, one has to use a concentration inequality for sums of subgaussian distributions (see, e.g., [28, Proposition 2.5]). We executed the same algorithms as in Section 6 with the same reward matrices but with Gaussian rewards with variance 0.05. The results are somewhat similar to the Bernoulli case and can be found in Figure 2.

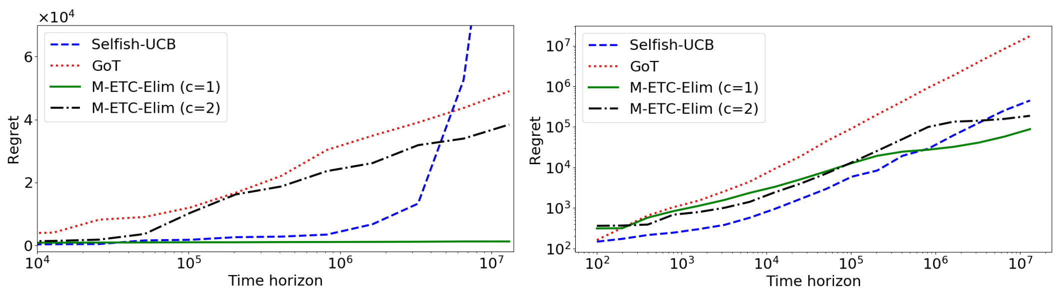

Figure 2: Numerical comparison of M-ETC-Elim, GoT and Selfish-UCB on reward matrices U_1 (left) and U_2 (right) with Gaussian rewards and variance 0.05. The x-axis has logarithmic scale in both plots. The y-axis has logarithmic scale in the right plot.

The reason we performed these Gaussian experiments is to have a more fair comparison against GoT. Indeed the numerical experiments of [8] rely on the same reward matrix U_1 and Gaussian rewards.

C Regret Analysis in the Presence of a Unique Maximum Matching

In Theorem 8 below we provide a refined analysis of M-ETC-Elim with parameter $c = 1$ if the maximum matching is unique. It notably justifies the $O(\frac{KM^3}{\Delta}\log(T))$ regret upper bound stated in Theorem 1(b). Its proof, given below, follows essentially the same line as the finite-time analysis given in Section 5, except for the last part. In the sequel, $\ln(\cdot)$ denotes the natural logarithm and $\lg(\cdot)$ denotes logarithm in base 2.

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

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

Proof. The good event and the regret incurred during the initialization phase are the same as in the finite-time analysis given in Section 5. When there is a unique optimal matching, if the good event happens, 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 Lemma 6 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 $\mathcal{G}_T$,

$$\hat{p}_T \leq P(\pi') \leq \lg\left(\frac{64M^2\ln(2M^2KT^2)}{\Delta^2}\right).$$

Plugging this bound in Lemma 5 yields that the regret incurred during communications is bounded by

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

Also, for $c = 1$ and any matching π, it can be shown as in Appendix D.3 that

$$P(\pi) \leq 1 + \lg\left(\frac{32M^2\ln(2M^2KT^2)}{\Delta(\pi)^2}\right).$$

In particular, $\Delta(\pi)2^{P(\pi)} \leq \frac{64M^2\ln(2M^2KT^2)}{\Delta(\pi)}$. Using the same argument as in Section 5, the regret incurred during exploration and exploitation phases is then bounded by

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

Summing up the regret bounds for all phases then proves Theorem 8.

D Proofs of Auxiliary Lemmas for Theorems 3 and 8

D.1 Proof of Lemma 4

We first recall Hoeffding's inequality.

Proposition 9 (Hoeffding's inequality [14, Theorem 2]). *Let $X_1, \ldots, X_n$ 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

$$\mathcal{G}_T = \left\{\textsc{Init}(K, 1/KT) \text{ is successful and } \forall p \leq \hat{p}_T, \forall \pi \in \mathcal{C}_{p+1}, |\tilde{U}_p(\pi) - U(\pi)| \leq 2M\varepsilon_p\right\}.$$

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

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

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

Fix an epoch p and a candidate edge (m, k). We denote by $\widehat{\mu}_k^m(p)$ the estimated mean of arm k for player m at the end of epoch p and by $\widetilde{\mu}_k^m(p)$ the truncated estimated mean sent to the leader by this player at the end of epoch p.

By Hoeffding's inequality and since this estimated mean is based on at least 2^{p^c} pulls, we have

$$\mathbb{P}\left(|\widehat{\mu}_k^m(p) - \mu_k^m| > \varepsilon_p\right) < \delta.$$

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

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

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

$$\begin{aligned}
&\mathbb{P}\left(\exists \pi \in \mathcal{M} \text{ with candidate edges} : |\tilde{U}_p(\pi) - U(\pi)| > 2M\varepsilon_p | \mathcal{H}\right) \\
&\quad \leq \mathbb{P}\left(\exists \text{ candidate edge } (m,k) : |\widetilde{\mu}_k^m(\pi) - \mu_k^m| > 2\varepsilon_p\right) \\
&\quad \leq KM\delta.
\end{aligned}$$

Finally, a union bound on p yields

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

D.2 Proof of Lemma 5

For each epoch p, the leader first communicates to each player the list of candidate matchings. There can be up to MK candidate matchings, and for each of them the leader communicates to the player the arm she has to pull (there is no need to communicate to her the whole matching) which requires $\lg K$ bits, and there are a total of M players, so this takes at most $M^2 K \lg(K)$ many rounds[3].

At the end of the epoch, each player sends the leader the empirical estimates for the arms she has pulled, which requires at most $MK(1 + p^c)/2$ many rounds. As players use the best estimated matching as communication arms for the communication phases, a single communication round incurs regret at most $2 + 2M\varepsilon_{p-1}$, since the gap between the best estimated matching of the previous phase and the best matching is at most $2M\varepsilon_{p-1}$, where we define $\varepsilon_0 := \sqrt{\frac{\ln(2/\delta)}{2}} \geq \frac{1}{2}$. The first term is for the two players colliding, while the term $2M\varepsilon_{p-1}$ is due to the other players who are pulling the best estimated matching instead of the real best one. With $\hat{p}_T$ denoting the number of epochs before the (possible) start of the exploitation, the total regret due to communication phases can be bounded by

$$\begin{aligned}
R_c &\leq \sum_{p=1}^{\hat{p}_T} \left(2M^2 K \lg(K) + MK(1 + p^c)\right)(1 + M\varepsilon_{p-1}) \\
&\leq 3M^2 K \lg(K) \hat{p}_T + MK(\hat{p}_T)^{c+1} + M^2 K \sum_{p=1}^{\hat{p}_T} \left(2M \lg(K) + (1 + p^c)\right)\varepsilon_{p-1}.
\end{aligned}$$

We now bound the sum as:

$$\begin{aligned}
\sum_{p=1}^{\hat{p}_T} \left(2M \lg(K) + (1+p^c)\right)\varepsilon_{p-1} &= 2M\lg(K)\sqrt{\ln(2/\delta)} \sum_{p=0}^{\hat{p}_T - 1} \frac{1}{\sqrt{2}^{1+p^c}} + \sqrt{\ln(2/\delta)} \sum_{p=0}^{\hat{p}_T - 1} \frac{1 + (p+1)^c}{\sqrt{2}^{1+p^c}} \\
&\leq 2M\lg(K)\sqrt{\ln(2/\delta)} \sum_{n=1}^{\infty} \frac{1}{\sqrt{2}^n} + \sqrt{\ln(2/\delta)} \sum_{n=1}^{\infty} \frac{n2^c}{\sqrt{2}^n} \\
&\leq 2M\lg(K)\sqrt{\ln(2/\delta)} \frac{1}{\sqrt{2}-1} + \sqrt{\ln(2/\delta)} \frac{2^c\sqrt{2}}{(\sqrt{2}-1)^2},
\end{aligned}$$

completing the proof of Lemma 5.

D.3 Proof of Lemma 6

The assumption $T \geq \exp(2^{\frac{c^c}{\ln^c(1+\frac{1}{2c})}})$ gives $\lg(\ln T)^{1/c} \geq \frac{c}{\ln(1+1/2c)}$. In particular, $(\lg T)^{1/c} \geq c$. We will also use the inequality

$$(x+1)^c \leq e^{c/x} x^c, \tag{5}$$

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

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

[3] Actually the leader also sends her communication arm and the size of the list she is sending, but there are actually at most $MK - M + 1$ candidate matchings as the best one is repeated M times. So it still takes at most $M^2 K \lg K$ many rounds.

Also, $2\lg(\ln(T)) \geq 2c^c \geq 2^c$. It remains to show the first inequality of Lemma 6.

Straightforward calculations using the definition of ε_p in (1) give

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

We claim that we have

$$P(\pi)^c \leq \left(1 + \frac{1}{2c}\right)L(\pi). \tag{6}$$

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 (5) with $x = L(\pi)^{1/c}$ gives (6). Hence,

$$\Delta(\pi)2^{P(\pi)^c} \leq \Delta(\pi)\left(\frac{32M^2\ln(2M^2KT^2)}{\Delta(\pi)^2}\right)^{1+1/2c} \leq \left(\frac{32M^2\ln(2M^2KT^2)}{\Delta(\pi)}\right)^{1+1/c}. \tag{7}$$

D.4 Proof of Lemma 7.

For the sake of clarity, we define for this proof $\Delta \coloneqq \Delta(\pi^{m,k})$, $P \coloneqq P(\pi^{m,k})$ and $\Delta_p \coloneqq \widetilde{\Delta}_p^{m,k}$. First, $\Delta > 8M\varepsilon_P$ by definition of P. Also, $\Delta_p \leq 8M\varepsilon_{p-1}$ for any $p \leq P-1$, otherwise the edge (m,k) would have been eliminated before epoch p. It then holds

$$\Delta_p \leq \frac{\varepsilon_{p-1}}{\varepsilon_P}\Delta = \sqrt{2}^{P^c-(p-1)^c}\Delta. \tag{8}$$

It comes from the convexity of $x \mapsto x^c$ that $(p+1)^c + (p-1)^c - 2p^c \geq 0$ and thus $P^c + (p-1)^c - 2p^c \geq P^c - (p+1)^c \geq P - (p+1)$. It then follows

$$p^c + \frac{P^c - (p-1)^c}{2} \leq P^c + \frac{p+1-P}{2}.$$

Plugging this in (8) finally bounds:

$$2^{p^c}\Delta_p \leq \frac{2^{P^c}}{\sqrt{2}^{P-(p+1)}}\Delta.$$

E Positioning with respect to [27]

The recent independent preprint [27] studies a slightly stronger feedback model than ours: they assume each player in each round has the option of 'observing whether a given arm has been pulled by someone,' without actually pulling that arm (thus avoiding collision due to this "observation"), an operation that is called "sensing." Due to the stronger feedback, communications do not need to be implicitly done through collisions and bits can be broadcast to other players via sensing. Note that it is actually possible to send a single bit of information from one player to all other players in a single round in their model, an action that requires $M-1$ rounds in our model. Still, the algorithms proposed by [27] can be modified to obtain algorithms for our setting, and M-ETC-Elim can also be adapted to their setting.

The two algorithms proposed by [27] share similarities with M-ETC-Elim: they also have exploration, communication and exploitation phases, but they do not use eliminations. Regarding the theoretical guarantees obtained, a first remark is that those claimed in [27] only hold in the presence of a unique optimal matching, whereas our analysis of M-ETC-Elim applies in the general case. Moreover, we believe that the current statements of their regret upper bounds are imprecise. Indeed, the regret upper bound $O(M^2K\log(T))$ claimed in their Theorem 4 does not feature any dependency in the gap parameter Δ, which contradicts the (asymptotic) $\Omega(((K-M)/\Delta)\log(T))$ lower bound of [2] for the easier centralized homogeneous variant. In fact, checking the proofs more carefully, one observes that both their regret bounds for the heterogeneous setting (Theorems 3 and 4) indeed depends exponentially on $1/\Delta$ while this is not stated in their theorem statements.

Here we explain why the dependence of their regret bounds on the gap Δ is exponential, and in particular, it is at least $2^{\Theta(\Delta^{-2})}$. First, note that in the presence of a unique optimal matching, Δ_{min}

in their notation is exactly Δ in ours. In their proof of Theorem 3 in their Appendix B, the exploitation regret is written as $R_1^E(T) + R_2^E(T)$. Then the term $R_1^E(T)$ is bounded by $2^{l'+1}(1 + \Delta_{max})$. If we look at the definition of l', we see it is the smallest l such that $\log^{-\beta/2}(t_l) =: \varepsilon(l) \leq \Delta_{min}$. Note that t_l is the start of the lth exploration phase, so it is roughly of order 2^l, and to be more precise, we have $\log(t_l) = \Theta(l)$. This gives $l' = \Theta(\Delta_{min}^{-2/\beta})$, hence the upper bound for $R_1^E(T)$, i.e., $(1 + \Delta_{max}) \times 2^{l'+1}$, is at least $2^{l'} = 2^{\Theta(\Delta_{min}^{-2/\beta})} \geq 2^{\Theta(\Delta_{min}^{-2})}$ because $\beta < 1$. This dependency is ignored in their proof and it is written $(1 + \Delta_{max}) \times 2^{l'+1} = C_2$, implying C_2 is a constant.

A similar situation holds for the proof of their Theorem 4. There, l_1 plays the role of l'.