JMLR: Workshop and Conference Proceedings vol 49:1–23, 2016

Maximin Action Identification: A New Bandit Framework for Games

Aurélien Garivier AURELIEN.GARIVIER@MATH.UNIV-TOULOUSE.FR
Institut de Mathématiques de Toulouse; UMR5219
Université de Toulouse; CNRS
UPS IMT, F-31062 Toulouse Cedex 9, France

Emilie Kaufmann EMILIE.KAUFMANN@INRIA.FR
Univ. Lille, CNRS, UMR 9189
CRIStAL - Centre de Recherche en Informatique Signal et Automatique de Lille
F-59000 Lille, France

Wouter M. Koolen WMKOOLEN@CWI.NL
Centrum Wiskunde & Informatica, Science Park 123, 1098 XG Amsterdam, The Netherlands

Abstract

We study an original problem of pure exploration in a strategic bandit model motivated by Monte Carlo Tree Search. It consists in identifying the best action in a game, when the player may sample random outcomes of sequentially chosen pairs of actions. We propose two strategies for the fixed-confidence setting: Maximin-LUCB, based on lower- and upper- confidence bounds; and Maximin-Racing, which operates by successively eliminating the sub-optimal actions. We discuss the sample complexity of both methods and compare their performance empirically. We sketch a lower bound analysis, and possible connections to an optimal algorithm.

Keywords: multi-armed bandit problems, games, best-arm identification, racing, LUCB

1. Setting: A Bandit Model for Two-Player Zero-Sum Random Games

We study a statistical learning problem inspired by the design of computer opponents for playing games. We are thinking about two-player zero sum full information games like Checkers, Chess, Go (Silver et al., 2016) . . . , and also games with randomness and hidden information like Scrabble or Poker (Bowling et al., 2015). At each step during game play, the agent is presented with the current game configuration, and is tasked with figuring out which of the available moves to play. In most interesting games, an exhaustive search of the game tree is completely out of the question, even with smart pruning.

Given that we cannot consider all states, the question is where and how to spend our computational effort. A popular approach is based on Monte Carlo Tree Search (MCTS) (Gelly et al., 2012; Browne et al., 2012). Very roughly, the idea of MCTS is to reason strategically about a tractable (say up to some depth) portion of the game tree rooted at the current configuration, and to use (randomized) heuristics to estimate values of states at the edge of the tractable area. One way to obtain such estimates is by 'rollouts': playing reasonable random policies for both players against each other until the game ends and seeing who wins.

MCTS methods are currently applied very successfully in the construction of game playing agents and we are interested in understanding and characterizing the fundamental complexity of such approaches. The existing picture is still rather incomplete. For example, there is no precise characterization of the number of rollouts required to identify a close to optimal action. Sometimes, cumulated regret minimizing algorithms (e.g. UCB derivatives) are used, whereas only the simple regret is relevant

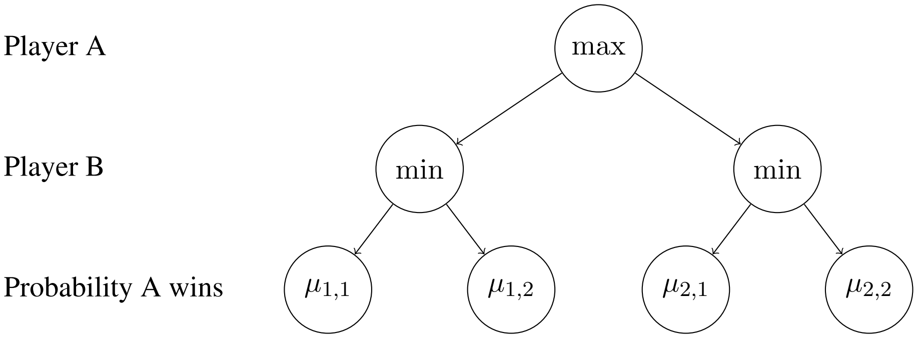

Figure 1: Game tree when there are two actions per player ($K = K_1 = K_2 = 2$).

here. As a first step in this direction, we investigate in this paper an idealized version of the MCTS problem for games, for which we develop a theory that leads to sample complexity guarantees.

More precisely, we study perhaps the simplest model incorporating both strategic reasoning and exploration (see Figure 1 for an example). We consider a two-player two-round zero-sum game, in which player A has K available actions. For each of these actions, indexed by i, player B can then choose among K_i possible actions, indexed by j. When player A chooses action $i \in \{1,\dots,K\}$ and then player B chooses action $j \in \{1,\dots,K_i\}$, the probability that player A wins is $\mu_{i,j}$. We investigate the situation from the perspective of player A, who wants to identify a maximin action

$$i^* \in \underset{i \in \{1,\dots,K\}}{\operatorname{argmax}} \; \min_{j \in \{1,\dots,K_i\}} \mu_{i,j}.$$

Assuming that player B is strategic and picks, whatever A's action i, the action j minimizing $\mu_{i,j}$, this is the best choice for A.

The parameters of the game are unknown to player A, but he can repeatedly choose a pair $P = (i,j)$ of actions for him *and player B*, and subsequently observe a sample from a Bernoulli distribution with mean $\mu_{i,j}$. At this point we imagine the sample could be generated e.g. by a single rollout estimate in an underlying longer game that we consider beyond tractable strategic consideration. Note that, in this learning phase, player A is not playing a game: he chooses actions for himself *and* for his adversary, and observes the random outcome.

The aim of this work is to propose a dynamic sampling strategy for player A in order to minimize the total number of samples (i.e. rollouts) needed to identify i^*. Denoting the possible move pairs by

$$\mathcal{P} = \left\{(i,j) : 1 \le i \le K, 1 \le j \le K_i\right\},$$

we formulate the problem as the search of a particular subset of arms in a stochastic bandit model with $\overline{K} = \sum_{i=1}^{K} K_i$ Bernoulli arms of respective expectations μ_P, $P \in \mathcal{P}$. In this bandit model, parametrized by $\boldsymbol{\mu} = (\mu_P)_{P \in \mathcal{P}}$, when the player chooses an arm (a pair of actions) P_t at round t, he observes a sample X_t drawn under a Bernoulli distribution with mean μ_{P_t}.

In contrast to best arm identification in bandit models (see, e.g., Even-Dar et al. (2006); Audibert et al. (2010)), where the goal is to identify the arm(s) with highest mean, $\operatorname{argmax}_P \mu_P$, here we want to identify as quickly as possible the maximin action i^* defined above. For this purpose, we adopt a sequential learning strategy (or algorithm) $(P_t, \tau, \hat{\imath})$. Denoting by $\mathcal{F}_t = \sigma(X_1,\dots,X_t)$ the sigma-field generated by the observations made up to time t, this strategy is made of

- a sampling rule $P_t \in \mathcal{P}$ indicating the arm chosen at round t, such that P_t is $\mathcal{F}_{t-1}$ measurable,
- a stopping rule τ after which a recommendation is to be made, which is a stopping time with respect to $\mathcal{F}_t$,
- a final guess $\hat{\imath}$ for the maximin action i^*.

For some fixed $\epsilon \geq 0$, the goal is to find as quickly as possible an ϵ-maximin action, with high probability. More specifically, given $\delta \in]0,1[$, the strategy should be *δ-PAC*, i.e. satisfy

$$\forall \boldsymbol{\mu},\ \mathbb{P}_{\boldsymbol{\mu}}\left(\min_{j\in\{1\dots K_{i^*}\}} \mu_{i^*,j} - \min_{j\in\{1\dots K_{\hat{\imath}}\}} \mu_{\hat{\imath},j} \leq \epsilon\right) \geq 1-\delta, \tag{1}$$

while keeping the total number of samples τ as small as possible. This is known, in the best-arm identification literature, as the *fixed-confidence* setting; alternatively, one may consider the *fixed-budget* setting where the total number of samples τ is fixed in advance, and where the goal is to minimize the probability that $\hat{\imath}$ is not an ϵ-maximin action.

Related Work. Tools from the bandit literature have been used in MCTS for around a decade (see Munos (2014) for a survey). Originally, MCTS was used to perform planning in a Markov Decision Process (MDP), which is a slightly different setting with no adversary: when an action is chosen, the transition towards a new state and the reward observed are generated by some (unknown) random process. A popular approach, UCT (Kocsis and Szepesvári, 2006) builds on Upper Confidence Bounds algorithms, which are useful tools for regret minimization in bandit models (e.g., Auer et al. (2002)). In this slightly different setup (see Bubeck and Cesa-Bianchi (2012) for a survey), the goal is to maximize the sum of the samples collected during the interaction with the bandit, which amounts in our setting to favor rollouts for which player A wins (which is not important in the learning phase). This situation is from a certain perspective a little puzzling and arguably confusing, because as shown by Bubeck et al. (2011), regret minimization and best arm identification are incompatible objectives in the sense that no algorithm can simultaneously be optimal for both.

More recently, tools from the best-arm identification literature have been used by Szorenyi et al. (2014) in the context of planning in a Markov Decision Process with a generative model. The proposed algorithm builds on the UGapE algorithm of Gabillon et al. (2012) to decide for which action new trajectories in the MDP starting from this action should be simulated. Just like a best arm identification algorithm is a building block for such more complex algorithms to perform planning in an MDP, we believe that understanding the maximin action identification problem is a key step towards more general algorithms in games, with provable sample complexity guarantees. For example, an algorithm for maximin action identification may be useful for planning in a competitive Markov Decision Processes (Filar and Vrieze, 1996) that models stochastic games.

Contributions. In this paper, we propose two algorithms for maximin action identification in the fixed-confidence setting, whose sample complexity is presented in Section 2. These algorithms are inspired by the two dominant approaches used in best arm identification algorithms. Our first algorithm, Maximin-LUCB, is described in Section 3: it relies on the use of Lower and Upper Confidence Bounds. The second, Maximin-Racing, is described in Section 4: it proceeds by successive elimination of sub-optimal arms. We prove that both algorithms are δ-PAC, and give upper bounds on their sample complexity. Our analysis contrasts with that of classical BAI procedures, and exhibits unexpected elements that are specific to the game setting. Along the way, we also propose some avenues of improvement that are illustrated empirically in Section 5. Finally, we propose in Section 6 for the two-action case a lower bound on the sample complexity of any δ-PAC algorithm, and sketch a strategy that may be optimal with respect to this lower bound. Most proofs are deferred to the Appendix.

2. Notation and Summary of the Results

To ease the notation, in the rest of the paper we assume that the actions of the two players are re-ordered so that for each i, $\mu_{i,j}$ is increasing in j, and $\mu_{i,1}$ is decreasing in i (so that $i^* = 1$ and $\mu^* = \mu_{1,1}$). These assumptions are illustrated in Figure 2. With this notation, the action $\hat{\imath}$ is an ϵ-maximin action if $\mu_{1,1} - \mu_{\hat{\imath},1} \le \epsilon$. We also introduce $\mathcal{P}_i = \{(i,j), j \in \{1,\dots,K_i\}\}$ as the group of arms related to the choice of action i for player A.

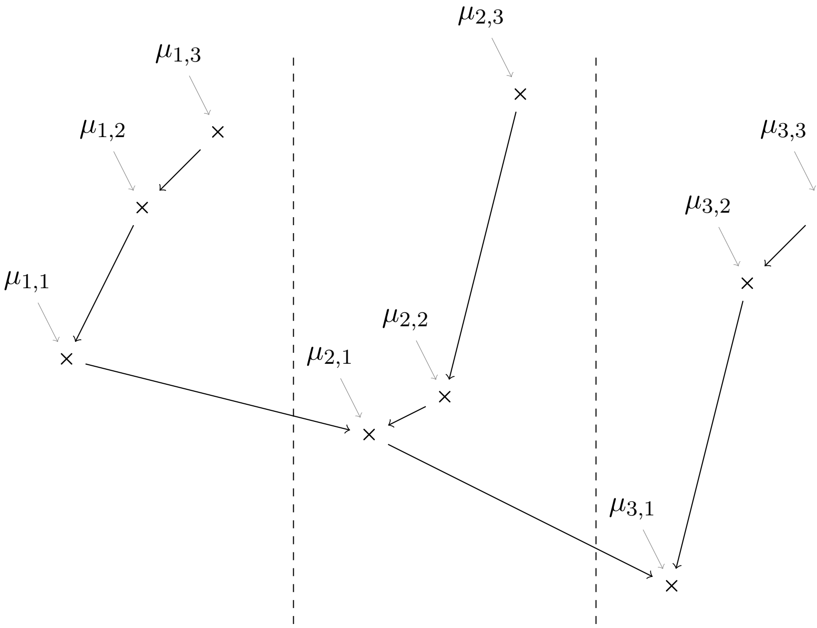

Figure 2: Example 'normal form' mean configuration. Arrows point to smaller values.

We propose in the paper two algorithms with a sample complexity of order $\mathbb{E}_{\boldsymbol{\mu}}[\tau] \simeq H^*(\boldsymbol{\mu}) \log(1/\delta)$, where

$$H^*(\boldsymbol{\mu}) := \sum_{(1,j)\in\mathcal{P}_1} \frac{1}{(\mu_{1,j} - \mu_{2,1})^2} + \sum_{(i,j)\in\mathcal{P}\setminus\mathcal{P}_1} \frac{1}{(\mu_{1,1} - \mu_{i,1})^2 \vee (\mu_{i,j} - \mu_{i,1})^2}.$$

This complexity term is easy to interpret:

- each arm $(1,j)$ in the optimal group $\mathcal{P}_1$ should be drawn enough to be discriminated from $\mu_{2,1}$ (roughly of order $(\mu_{1,j} - \mu_{2,1})^{-2} \log(1/\delta)$ times)
- each other arm (i,j) in $\mathcal{P} \setminus \mathcal{P}_1$ should be drawn enough to either be discriminated from the smallest arm in the same group, $\mu_{i,1}$, or so that $\mu_{i,1}$ is discriminated from $\mu_{1,1}$.

Our Theorems 4 and 8 respectively lead to (asymptotic) upper bounds featuring two complexity terms $H_1^*(\boldsymbol{\mu})$ and $H_2^*(\boldsymbol{\mu})$ that are very close to $H^*(\boldsymbol{\mu})$. When there are two actions per player, Theorem 5 shows that the sample complexity of M-LUCB is asymptotically upper bounded by $16H^*(\boldsymbol{\mu})\log(1/\delta)$. We discuss sample complexity lower bounds and their connection to future algorithms in Section 6.

It is to be noted that such a sample complexity cannot be attained by a naive use of algorithms for best arm identification. Indeed the most straightforward idea would be to follow the tree structure bottom-up; recursively identifying alternately best and worst arms. For two levels, we would first perform worst-arm identification on j for each i, followed by best-arm identification on the obtained set of worst arms. To see why such modular approaches fail, recall that the sample complexity of best and worst arm identification scales with the sum of the inverse squared gaps (see e.g. Kalyanakrishnan et al. (2012)). But many of these gaps, however small, are irrelevant for finding i^*. For a simple example consider $\mu_{1,1} = \mu_{1,2} \gg \mu_{2,1} = \mu_{2,2}$. Hierarchical algorithms find such equal means (zero gap) within actions arbitrarily hard, while our methods fare fine: only the large gap between actions appears in our complexity measure H^*. We now explain in Section 3 and 4 how to carefully use best arm identification tools to come up with algorithms whose sample complexity scales with $H^*(\boldsymbol{\mu})$.

3. First Approach: M-LUCB

We first describe a simple strategy based on confidence intervals, called Maximin-LUCB (M-LUCB). Confidence bounds have been successfully used for best-arm identification in the fixed-confidence setting (Kalyanakrishnan et al., 2012; Gabillon et al., 2012; Jamieson et al., 2014). The algorithm proposed in this section for maximin action identification is inspired by the LUCB algorithm of Kalyanakrishnan et al. (2012), based on Lower and Upper Confidence Bounds.

For every pair of actions $P \in \mathcal{P}$, let $\mathcal{I}_P(t) = [\mathrm{L}_P(t), \mathrm{U}_P(t)]$ be a confidence interval on μ_P built using observations from arm P gathered up to time t. Such a confidence interval can be obtained by using the number of draws $N_P(t) := \sum_{s=1}^{t} \mathbb{1}_{(P_s=P)}$ and the empirical mean of the observations for this pair $\hat{\mu}_P(t) := \sum_{s=1}^{t} X_s \mathbb{1}_{(P_s=P)}/N_P(t)$ (with the convention $\hat{\mu}_P(0) = 0$). The M-LUCB strategy aims at aligning the lower confidence bounds of arms that are in the same group $\mathcal{P}_i$. Arms to be drawn are chosen two by two: for any even time t, defining for every $i \in \{1, \dots, K\}$

$$c_i(t) = \operatorname*{argmin}_{1\le j\le K_i} \mathrm{L}_{(i,j)}(t) \quad \text{and} \quad \hat{\imath}(t) = \operatorname*{argmax}_{1\le i\le K} \min_{1\le j\le K_i} \hat{\mu}_{i,j}(t) ,$$

the algorithm draws at rounds $t+1$ and $t+2$ the arms $P_{t+1} = H_t$ and $P_{t+2} = S_t$ where

$$H_t = (\hat{\imath}(t), c_{\hat{\imath}(t)}(t)) \quad \text{and} \quad S_t = \operatorname*{argmax}_{P\in\{(i,c_i(t))\}_{i\neq\hat{\imath}}} \mathrm{U}_P(t).$$

This is indeed a regular LUCB sampling rule on a time-dependent set of arms each representing one action: $\{(i, c_i(t))\}_{i\in\{1,\dots,K\}}$. When there are two actions for each player, one may alternatively draw at each time t the arm $P_{t+1} = \operatorname{argmin}_{P\in\{H_t,S_t\}} N_P(t)$ only.

The stopping rule, which depends on the parameter $\epsilon \ge 0$ (that can be set to zero if $\mu_{1,1} > \mu_{2,1}$) is

$$\tau = \inf\left\{t \in 2\mathbb{N} : \mathrm{L}_{H_t}(t) > \mathrm{U}_{S_t}(t) - \epsilon\right\} . \tag{2}$$

Then arm $\hat{\imath} = \hat{\imath}(\tau)$, the empirical maximin action at that time, is recommended to player A. The stopping rule is illustrated in Figure 3. The choice of the representative arms $\{(i, c_i(t))_{i\in\{1,\dots,K\}}$ is crucial, and it can be understood from the proof of Lemma 11 in Appendix A why aligning the lower confidence bounds inside each action is a good idea.

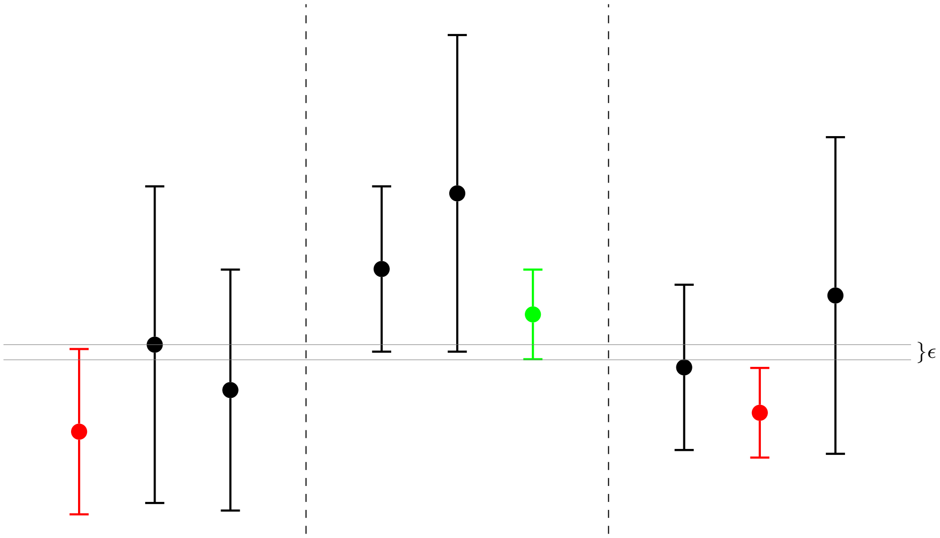

Figure 3: Stopping rule (2). The algorithm stops because the lower bound of the green arm beats up to slack ϵ the upper bound for at least one arm (marked red) in each other action, the one with smallest LCB. In this case action $\hat{\imath} = 2$ is recommended.

3.1. Analysis of the Algorithm

We analyze the algorithm under the assumptions $\mu_{1,1} > \mu_{2,1}$ and $\epsilon = 0$. We consider the Hoeffding-type confidence bounds

$$\mathrm{L}_P(t) = \hat{\mu}_P(t) - \sqrt{\frac{\beta(t,\delta)}{2N_P(t)}} \quad \text{and} \quad \mathrm{U}_P(t) = \hat{\mu}_P(t) + \sqrt{\frac{\beta(t,\delta)}{2N_P(t)}}\,, \tag{3}$$

where $\beta(t,\delta)$ is some exploration rate. A choice of $\beta(t,\delta)$ that ensures the δ-PAC property (1) is given below. In order to highlight the dependency of the stopping rule on the risk level δ, we denote it by τ_δ.

Theorem 1 *Let*

$$H_1^*(\boldsymbol{\mu}) = \sum_{j=1}^{K_1} \frac{1}{\left(\mu_{1,j} - \frac{\mu_{1,1}+\mu_{2,1}}{2}\right)^2} + \sum_{(i,j)\in\mathcal{P}\setminus\mathcal{P}_1} \frac{1}{\left(\frac{\mu_{1,1}+\mu_{2,1}}{2} - \mu_{i,1}\right)^2 \vee (\mu_{i,j} - \mu_{i,1})^2}.$$

On the event

$$\mathcal{E} = \bigcap_{P\in\mathcal{P}} \bigcap_{t\in 2\mathbb{N}} \left\{\mu_P \in [\mathrm{L}_P(t), \mathrm{U}_P(t)]\right\},$$

the M-LUCB strategy returns the maximin action and uses a total number of samples upper-bounded by

$$T(\boldsymbol{\mu},\delta) = \inf\left\{t \in \mathbb{N} : 4H_1^*(\boldsymbol{\mu})\beta(t,\delta) < t\right\}.$$

According to Theorem 1, the exploration rate should be large enough to control $\mathbb{P}_{\boldsymbol{\mu}}(\mathcal{E})$, and as small as possible so as to minimize $T(\boldsymbol{\mu},\delta)$. The self-normalized deviation bound of Cappé et al. (2013) gives a first solution (Corollary 2), whereas Lemma 7 of Kaufmann et al. (2015) yields Corollary 3. In both cases, explicit bounds on $T(\boldsymbol{\mu},\delta)$ are obtained using the technical Lemma 12 stated in Appendix A.

Corollary 2 *Let $\alpha > 0$ and $C = C_\alpha$ be such that*

$$e\overline{K} \sum_{t=1}^{\infty} \frac{(\log t)(\log(Ct^{1+\alpha}))}{t^{1+\alpha}} \leq C \,,$$

and let δ satisfy $4(1+\alpha)(C/\delta)^{1/(1+\alpha)} > 4.85$. With probability larger than $1-\delta$, the M-LUCB strategy using the exploration rate

$$\beta(t,\delta) = \log\left(\frac{Ct^{1+\alpha}}{\delta}\right) \,, \tag{4}$$

returns the maximin action within a number of steps upper-bounded as

$$\tau_\delta \leq 4(1+\alpha)H_1^*(\boldsymbol{\mu})\left[\log\left(\frac{1}{\delta}\right) + \log(C(4(1+\alpha)H_1^*(\boldsymbol{\mu}))^{1+\alpha}) + 2(1+\alpha)\log\log\left(\frac{4(1+\alpha)H_1^*(\boldsymbol{\mu})C^{\frac{1}{1+\alpha}}}{\delta^{\frac{1}{1+\alpha}}}\right)\right].$$

Corollary 3 *For b, c such that $c > 2$ and $b > c/2$, let the exploration rate be*

$$\beta(t,\delta) = \log\frac{1}{\delta} + b\log\log\frac{1}{\delta} + c\log\log(et)$$

and let

$$f_{b,c}(\delta) = \overline{K}\sqrt{e}\frac{\pi^2}{3}\frac{1}{8^{c/2}}\frac{(\sqrt{\log(1/\delta) + b\log\log(1/\delta)} + 2\sqrt{2})^c}{(\log(1/\delta))^b} \,.$$

With probability larger than $1 - f_{b,c}(\delta)\delta$, M-LUCB returns the maximin action and, for some positive constant C_c and for δ small enough,

$$\tau_\delta \leq 4H_1^*(\boldsymbol{\mu})\left[\log\left(\frac{1}{\delta}\right) + \log(8C_cH_1^*(\boldsymbol{\mu})) + 2\log\log\left(\frac{8C_cH_1^*(\boldsymbol{\mu})}{\delta}\right)\right].$$

Elaborating on the same ideas, it is possible to obtain results in expectation, at the price of a less explicit bound, that holds for a slightly larger exploration rate.

Theorem 4 *The M-LUCB algorithm using $\beta(t,\delta)$ defined by (4), with $\alpha > 1$, is δ-PAC and satisfies*

$$\limsup_{\delta\to 0}\frac{\mathbb{E}_{\boldsymbol{\mu}}[\tau_\delta]}{\log(1/\delta)} \leq 4(1+\alpha)H_1^*(\boldsymbol{\mu}).$$

$H_1^*(\boldsymbol{\mu})$ is very close to the quantity $H^*(\boldsymbol{\mu})$ introduced in Section 2. The difference is that in $H_1^*(\boldsymbol{\mu})$ $\mu_{i,1}$ and $\mu_{1,1}$ need to be discriminated from a 'virtual arm' with mean $(\mu_{1,1}+\mu_{2,1})/2$ in place of $\mu_{1,1}$ and $\mu_{2,1}$ respectively. We view this virtual arm (that corresponds to the choice of a parameter c in Appendix A) as an artifact of our proof. In the particular case of two actions per player, we propose the following finer result, that holds for the variant of M-LUCB that samples the least drawn arm among H_t and S_t at round $t+1$.

Theorem 5 *Assume $K = K_1 = K_2 = 2$. The M-LUCB algorithm using $\beta(t,\delta)$ defined by (4) with $\alpha > 1$ is δ-PAC and satisfies*

$$\limsup_{\delta\to 0}\frac{\mathbb{E}_{\boldsymbol{\mu}}[\tau_\delta]}{\log(1/\delta)} \leq 8(1+\alpha)H^*(\boldsymbol{\mu}).$$

3.2. Improved Intervals and Stopping Rule

The symmetry and the simple form of the sub-gaussian confidence intervals (3) are convenient for the analysis, but they can be greatly improved thanks to better deviation bounds for Bernoulli distributions. A simple improvement (see Kaufmann and Kalyanakrishnan (2013)) is to use Chernoff confidence intervals, based on the binary relative entropy function $d(x,y) = x\log(x/y) + (1-x)\log((1-x)/(1-y))$. Moreover, the use of a better stopping rule based on generalized likelihood ratio tests (GLRT) has been proposed recently for best-arm identification, leading to significant improvements. We propose here an adaptation of the Chernoff stopping rule of Garivier and Kaufmann (2016), valid for the case $\epsilon = 0$.

Our stopping rule is based on the statistic:

$$Z_{P,Q}(t) := \log \frac{\max_{\mu'_P \geq \mu'_Q} p_{\mu'_P}\left(\underline{X}^P_{N_P(t)}\right) p_{\mu'_Q}\left(\underline{X}^Q_{N_Q(t)}\right)}{\max_{\mu'_P \leq \mu'_Q} p_{\mu'_P}\left(\underline{X}^P_{N_P(t)}\right) p_{\mu'_Q}\left(\underline{X}^Q_{N_Q(t)}\right)},$$

where $\underline{X}^P_s$ is a vector that contains the first s observations of arm P and $p_\mu(Z_1,\dots,Z_s)$ is the likelihood of the given s observations from an i.i.d. Bernoulli distribution with mean μ. Introducing the empirical mean of the combined samples from arms P and Q,

$$\hat{\mu}_{P,Q}(t) \quad := \quad \frac{N_P(t)}{N_P(t)+N_Q(t)}\hat{\mu}_P(t) + \frac{N_Q(t)}{N_P(t)+N_Q(t)}\hat{\mu}_Q(t),$$

it appears that for $\hat{\mu}_P(t) \geq \hat{\mu}_Q(t)$,

$$Z_{P,Q}(t) = N_P(t)d\left(\hat{\mu}_P(t), \hat{\mu}_{P,Q}(t)\right) + N_Q(t)d\left(\hat{\mu}_Q(t), \hat{\mu}_{P,Q}(t)\right),$$

and $Z_{P,Q}(t) = -Z_{Q,P}(t)$. The stopping rule is defined as

$$\begin{aligned}\tau &= \inf\left\{t \in \mathbb{N} : \exists i \in \{1,\dots,K\} : \forall i' \neq i, \exists j' \in \{1,\dots,K_{i'}\} : \forall j \in \{1,\dots,K_i\}, Z_{(i,j),(i',j')}(t) > \beta(t,\delta)\right\} \\ &= \inf\left\{t \in \mathbb{N} : \max_{i\in\{1,\dots,K\}} \min_{i'\neq i} \max_{j'\in\{1,\dots,K_{i'}\}} \min_{j\in\{1,\dots,K_i\}} Z_{(i,j),(i',j')}(t) > \beta(t,\delta)\right\}. \qquad (5)\end{aligned}$$

Proposition 6 *Using the stopping rule (5) with the exploration rate $\beta(t,\delta) = \log\left(\frac{2K_1(K-1)t}{\delta}\right)$, whatever the sampling rule, if τ is a.s. finite, the recommendation is correct with probability $\mathbb{P}_{\boldsymbol{\mu}}\left(\hat{\imath} = i^*\right) \geq 1-\delta$.*

Sketch of Proof. Recall that in our notation the optimal action is $i^* = 1$.

$$\begin{aligned}\mathbb{P}_{\boldsymbol{\mu}}\left(\hat{\imath} \neq 1\right) &\leq \mathbb{P}_{\boldsymbol{\mu}}\left(\exists t \in \mathbb{N}, \exists i \in \{2,\dots,K\}, \exists j' \in \{1,\dots,K_1\}, Z_{(i,1),(1,j')}(t) > \beta(t,\delta)\right) \\ &\leq \sum_{i=2}^{K}\sum_{j'=1}^{K_1} \mathbb{P}_{\boldsymbol{\mu}}\left(\exists t \in \mathbb{N}, Z_{(i,1),(1,j')}(t) > \beta(t,\delta)\right).\end{aligned}$$

Note that for $i \neq 1$, $\mu_{i,1} < \mu_{1,j'}$ for all $j' \in \{1,\dots,K_1\}$. The result follows from the following bound proved in Garivier and Kaufmann (2016): whenever $\mu_P < \mu_Q$, for any sampling strategy,

$$\mathbb{P}_{\boldsymbol{\mu}}\left(\exists t \in \mathbb{N} : Z_{P,Q}(t) \geq \log\left(\frac{2t}{\delta}\right)\right) \leq \delta. \qquad (6)$$

4. Second Approach: A Racing Algorithm

We now propose a Racing-type algorithm for the maximin action identification problem, inspired by another line of algorithms for best arm identification (Maron and Moore, 1997; Even-Dar et al., 2006; Kaufmann and Kalyanakrishnan, 2013). Racing algorithms are simple and powerful methods that progressively concentrate on the best actions. We give in this section an analysis of a Maximin-Racing algorithm that relies on the refined information-theoretic tools introduced in the previous section.

4.1. A Generic Maximin-Racing Algorithm

The Maximin Racing algorithm maintains a set of active arms $\mathcal{R}$ and proceeds in rounds, in which all the active arms are sampled. At the end of round r, all active arms have been sampled r times and some arms may be eliminated according to some *elimination rule*. We denote by $\hat{\mu}_P(r)$ the average of the r observations on arm P. The elimination rule relies on an *elimination function* $f(x,y)$ ($f(x,y)$ is large if x is significantly larger than y), and on a *threshold function* $\beta(r,\delta)$.

The Maximin-Racing algorithm presented below performs two kinds of eliminations: the largest arm in each set $\mathcal{R}_i$ may be eliminated if it appears to be significantly larger than the smallest arm in $\mathcal{R}_i$ (*high arm elimination*), and the group of arms $\mathcal{R}_i$ containing the smallest arm may be eliminated (all the arms in $\mathcal{R}_i$ are removed from the active set) if it contains one arm that appears significantly smaller than all the arms of another group $\mathcal{R}_j$ (*action elimination*).

Maximin Racing Algorithm.
Parameters. Elimination function f, threshold function β
Initialization. For each $i \in \{1,\dots,K\}$, $\mathcal{R}_i = \mathcal{P}_i$, and $\mathcal{R} := \mathcal{R}_1 \cup \dots \cup \mathcal{R}_K$.
Main Loop. At round r:

- all arms in $\mathcal{R}$ are drawn, empirical means $\hat{\mu}_P(r)$, $P \in \mathcal{R}$ are updated
- *High arms elimination step*: for each action $i = 1 \dots K$, if $|\mathcal{R}_i| \geq 2$ and

$$rf\left(\max_{P\in\mathcal{R}_i}\hat{\mu}_P(r), \min_{P\in\mathcal{R}_i}\hat{\mu}_P(r)\right) \geq \beta(r,\delta)\ , \tag{7}$$

  then remove $P_m = \underset{P\in\mathcal{R}_i}{\operatorname{argmax}}\, \hat{\mu}_P(r)$ from the active set : $\mathcal{R}_i = \mathcal{R}_i \backslash \{P_m\}$, $\mathcal{R} = \mathcal{R}\backslash\{P_m\}$.
- *Action elimination step*: if $(\tilde{\imath},\tilde{\jmath}) = \underset{P\in\mathcal{R}}{\operatorname{argmin}}\, \hat{\mu}_P(r)$ and if

$$rf\left(\max_{i\neq\tilde{\imath}}\min_{P\in\mathcal{R}_i}\hat{\mu}_P(r), \hat{\mu}_{(\tilde{\imath},\tilde{\jmath})}(r)\right) \geq \beta(r,\delta)\ ,$$

  then remove $\tilde{\imath}$ from the possible maximin actions: $\mathcal{R} = \mathcal{R}\backslash\mathcal{R}_{\tilde{\imath}}$ and $\mathcal{R}_{\tilde{\imath}} = \varnothing$.

The algorithm stops when all but one of the $\mathcal{R}_i$ are empty, and outputs the index of the remaining set as the maximin action. If the stopping condition is not met at the end of round

$$r = r_0 := \frac{2}{\epsilon^2}\log\left(\frac{4\overline{K}}{\delta}\right),$$

then the algorithm stops and returns one of the empirical maximin actions.

4.2. Tuning the Elimination and Threshold Functions

In the best-arm identification literature, several elimination functions have been studied. The first idea, presented in the Successive Elimination algorithm of Even-Dar et al. (2006), is to use the simple difference $f(x,y) = (x-y)^2\mathbb{1}_{(x\geq y)}$; in order to take into account possible differences in the deviations of the arms, the KL-Racing algorithm of Kaufmann and Kalyanakrishnan (2013) uses an elimination function equivalent to $f(x,y) = d_*(x,y)\mathbb{1}_{(x\geq y)}$, where $d_*(x,y)$ is defined as the common value of $d(x,z)$ and $d(y,z)$ for the unique z satisfying $d(x,z) = d(y,z)$. In this paper, we use the divergence function

$$f(x,y) = I(x,y) := \left[d\left(x, \frac{x+y}{2}\right) + d\left(y, \frac{x+y}{2}\right)\right]\mathbb{1}_{(x\geq y)} \tag{8}$$

inspired by the deviation bounds of Section 3.2. In particular, using again Inequality (6) for the uniform sampling rule yields, whenever $\mu_P < \mu_Q$,

$$\mathbb{P}_{\boldsymbol{\mu}}\left(\exists r \in \mathbb{N} : rI(\hat{\mu}_P(r), \hat{\mu}_Q(r)) \geq \log\frac{2r}{\delta}\right) \leq \delta. \tag{9}$$

Using this bound, Proposition 7 (proved in Appendix B.1) proposes a choice of the threshold function for which the Maximin-Racing algorithm is δ-PAC.

Proposition 7 *With the elimination function $I(x,y)$ of Equation (8) and with the threshold function $\beta(t,\delta) = \log(4C_K t/\delta)$, the Maximin-Racing algorithm satisfies*

$$\mathbb{P}_{\boldsymbol{\mu}}\left(\mu_{1,1} - \mu_{\hat{\imath},1} \leq \epsilon\right) \geq 1-\delta,$$

with $C_K \leq (\overline{K})^2$. If $\mu_{1,1} > \mu_{1,2}$ and if $\forall i, \mu_{i,1} < \mu_{i,2}$, then we may take $C_K = K \times \max_i K_i$.

4.3. Sample Complexity Analysis

We propose here an asymptotic analysis of the number of draws of each arm (i,j) under the Maximin-Racing algorithm, denoted by $\tau_\delta(i,j)$. These bounds are expressed with the deviation function I, and hold for $\epsilon > 0$. For $\epsilon = 0$, one can provide similar bounds under the additional assumption that all arms are pairwise distinct.

Theorem 8 *Assume $\mu_{1,1} > \mu_{2,1}$. For every $\epsilon > 0$, and for $\beta(t,\delta)$ chosen as in Proposition 7, the Maximin-Racing algorithm satisfies, for every $j \in \{1,\dots,K_1\}$,*

$$\limsup_{\delta\to 0}\frac{\mathbb{E}_{\boldsymbol{\mu}}[\tau_\delta(1,j)]}{\log(1/\delta)} \leq \frac{1}{\max\left(\epsilon^2/2, I(\mu_{1,1},\mu_{2,1}), I(\mu_{1,j},\mu_{1,1})\right)}$$

and, for any $(i,j) \in \mathcal{P}\setminus\mathcal{P}_1$,

$$\limsup_{\delta\to 0}\frac{\mathbb{E}_{\boldsymbol{\mu}}[\tau_\delta(i,j)]}{\log(1/\delta)} \leq \frac{1}{\max\left(\epsilon^2/2, I(\mu_{i,1},\mu_{1,1}), I(\mu_{i,j},\mu_{i,1})\right)}.$$

It follows from Pinsker's inequality that $I(x,y) > (x-y)^2$, and hence Theorem 8 implies in particular that for the M-Racing algorithm (for a sufficiently small ϵ) $\limsup_{\delta\to 0}\mathbb{E}_{\boldsymbol{\mu}}[\tau_\delta]/\log(1/\delta) \leq H_2^*(\boldsymbol{\mu})$, where

$$H_2^*(\boldsymbol{\mu}) \quad := \quad \sum_{j=1}^{K_1}\frac{1}{(\mu_{1,j}-\mu_{1,1})^2 \vee (\mu_{1,1}-\mu_{2,1})^2} + \sum_{(i,j)\in\mathcal{P}\setminus\mathcal{P}_1}\frac{1}{(\mu_{1,1}-\mu_{i,1})^2 \vee (\mu_{i,j}-\mu_{i,1})^2}.$$

The complexity term $H_2^*(\boldsymbol{\mu})$ is very close to the quantities $H^*(\boldsymbol{\mu})$ and $H_1^*(\boldsymbol{\mu})$ respectively introduced in Section 2 and Theorem 1. The terms corresponding to arms in $\mathcal{P} \setminus \mathcal{P}_1$ are identical to those in $H^*(\boldsymbol{\mu})$ and strictly smaller than those in $H_1^*(\boldsymbol{\mu})$ (since no 'virtual arm' $(\mu_{1,1} + \mu_{2,1})/2$ has been introduced in the analysis of M-Racing). However, the terms corresponding to the arms $(1, j), j \geq 2$ are strictly larger than those in $H^*(\boldsymbol{\mu})$ (as $(\mu_{1,j} - \mu_{1,1}) \vee (\mu_{1,1} - \mu_{2,1})$ is strictly smaller than $(\mu_{1,j} - \mu_{2,1})$), and can even be smaller than those in $H_1^*(\boldsymbol{\mu})$. But this is mitigated by the fact that there is no multiplicative constant in front of the complexity term $H_2^*(\boldsymbol{\mu})$. Besides, as Theorem 8 involves the deviation function $I(x, y) = d(x, (x + y)/2) + d(y, (x + y)/2$ and not a subgaussian approximation, the resulting upper bound on the sample complexity of M-Racing can indeed be significantly better.

5. Numerical Experiments and Discussion

In the previous sections, we have proposed two different algorithms for the maximin action identification problem. The analysis that we have given does not clearly advocate the superiority of one or the other. The goal of this section is to propose a brief numerical comparison in different settings, and to compare with other possible strategies.

We will notably study empirically two interesting variants of M-LUCB. The first improvement that we propose is the M-KL-LUCB strategy, based on KL-based confidence bounds (Kaufmann and Kalyanakrishnan (2013)). The second variant, M-Chernoff, additionally improves the stopping rule as presented in Section 3.2. Whereas Proposition 6 justifies the use of the exploration rate $\beta(t, \delta) = \log(4\overline{K}^2 t/\delta)$, which is over-conservative in practice, we use $\beta(t, \delta) = \log((\log(t) + 1)/\delta)$ in all our experiments, as suggested by Corollary 3 (this appears to be already quite a conservative choice in practice). In the experiments, we set $\delta = 0.1$, $\epsilon = 0$.

To simplify the discussion and the comparison, we first focus on the particular case in which there are two actions for each player. As an element of comparison, one can observe that finding i^* is at most as hard as finding the worst arm (or the three best) among the four arms $(\mu_{i,j})_{1 \leq i,j \leq 2}$. Thus, one could use standard best-arm identification strategies like the (original) LUCB algorithm. For the latter, the complexity is of order

$$\frac{2}{(\mu_{1,1} - \mu_{2,1})^2} + \frac{1}{(\mu_{1,2} - \mu_{2,1})^2} + \frac{1}{(\mu_{2,2} - \mu_{2,1})^2},$$

which is much worse than the complexity term obtained for M-LUCB in Theorem 5 when $\mu_{2,2}$ and $\mu_{2,1}$ are close to one another. This is because a best arm identification algorithm does not only find the maximin action, but additionally figures out which of the arms in the other action is worst. Our algorithm does not need to discriminate between $\mu_{2,1}$ and $\mu_{2,2}$, it only tries to assess that one of these two arms is smaller than $\mu_{1,1}$. However, for specific instances in which the gap between $\mu_{2,2}$ and $\mu_{2,1}$ is very large, the difference vanishes. This is illustrated in the numerical experiments of Table 1, which involve the following three sets of parameters (the entry (i, j) in each matrix is the mean $\mu_{i,j}$):

$$\boldsymbol{\mu}_1 = \begin{bmatrix} 0.4 & 0.5 \\ 0.3 & 0.35 \end{bmatrix} \qquad \boldsymbol{\mu}_2 = \begin{bmatrix} 0.4 & 0.5 \\ 0.3 & 0.45 \end{bmatrix} \qquad \boldsymbol{\mu}_3 = \begin{bmatrix} 0.4 & 0.5 \\ 0.3 & 0.6 \end{bmatrix} \tag{10}$$

We also perform experiments in a model with 3 × 3 actions with parameters:

$$\boldsymbol{\mu} = \begin{bmatrix} 0.45 & 0.5 & 0.55 \\ 0.35 & 0.4 & 0.6 \\ 0.3 & 0.47 & 0.52 \end{bmatrix} \tag{11}$$

	$\tau_{1,1}$	$\tau_{1,2}$	$\tau_{2,1}$	$\tau_{2,2}$	$\tau_{1,1}$	$\tau_{1,2}$	$\tau_{2,1}$	$\tau_{2,2}$	$\tau_{1,1}$	$\tau_{1,2}$	$\tau_{2,1}$	$\tau_{2,2}$
M-LUCB	1762	198	1761	462	1761	197	1760	110	1755	197	1755	36
M-KL-LUCB	762	92	733	237	743	92	743	54	735	93	740	16
M-Chernoff	**315**	**59**	**291**	**136**	**325**	**61**	327	**41**	**321**	**61**	326	**13**
M-Racing	324	152	301	298	329	161	**318**	137	322	159	**323**	35
KL-LUCB	351	64	3074	2768	627	83	841	187	684	88	774	32

Table 1: Number of draws of the different arms under the models (10) parameterized by $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \boldsymbol{\mu}_3$ (from left to right), averaged over $N = 10000$ repetitions

Figure 4 shows that the best three algorithms in the previous experiments behave as expected: the number of draws of the arms are ordered exactly as suggested by the bounds given in the analysis. These experiments tend to show that, in practice, the best two algorithms are M-Racing and M-Chernoff, with a slight advantage for the latter. However, we did not provide theoretical sample complexity bounds for M-Chernoff, and it is to be noted that the use of Hoeffding bounds in the M-LUCB algorithm (that has been analyzed) is a cause of sub-optimality. Among the algorithms for which we provide theoretical sample complexity guarantees, the M-Racing algorithm appears to perform best.

$$\tau_{\text{M-KL-LUCB}} = \begin{bmatrix} 798 & 212 & 92 \\ 752 & 248 & 22 \\ 210 & 44 & 21 \end{bmatrix} \quad \tau_{\text{M-Ch.}} = \begin{bmatrix} 367 & 131 & 67 \\ 333 & 156 & 18 \\ 129 & 31 & 17 \end{bmatrix} \quad \tau_{\text{M-Racing}} = \begin{bmatrix} 472 & 291 & 173 \\ 337 & 337 & 42 \\ 161 & 185 & 71 \end{bmatrix}$$

Figure 4: Number of draws of each arm under the bandit model (11), averaged of $N = 10000$ repetitions

6. Perspectives

To finish, let us sketch the (still speculative) perspective of an important improvement. For simplicity, we focus on the case where each player chooses among only two possible actions, and we change our notation, using: $\mu_1 := \mu_{1,1}, \mu_2 := \mu_{1,2}, \mu_3 := \mu_{2,1}, \mu_4 := \mu_{2,2}$. As we will see below, the optimal strategy is going to depend a lot on the position of μ_4 relatively to μ_1 and μ_2. Given $w = (w_1, \dots, w_4) \in \Sigma_K = \{w \in \mathbb{R}_+^4 : w_1 + \dots + w_4 = 1\}$, we define for a, b, c in $\{1, \dots, 4\}$:

$$\mu_{a,b}(w) = \frac{w_a \mu_a + w_b \mu_b}{w_a + w_b} \quad \text{and} \quad \mu_{a,b,c}(w) = \frac{w_a \mu_a + w_b \mu_b + w_c \mu_c}{w_a + w_b + w_c} .$$

Using a similar argument than the one of Garivier and Kaufmann (2016) in the context of best-arm identification, one can prove the following (non explicit) lower bound on the sample complexity.

Theorem 9 *Any δ-PAC algorithm satisfies*

$$\mathbb{E}_{\boldsymbol{\mu}}[\tau_\delta] \geq T^*(\boldsymbol{\mu})\, d(\delta, 1 - \delta),$$

where

$$\begin{aligned} T^*(\boldsymbol{\mu})^{-1} &:= \sup_{w\in\Sigma_K} \inf_{\boldsymbol{\mu}':\mu_1'\wedge\mu_2'<\mu_3'\wedge\mu_4'} \left(\sum_{a=1}^K w_a\, d(\mu_a,\mu_a')\right) \\ &= \sup_{w\in\Sigma_K} F(\boldsymbol{\mu},w), \end{aligned} \tag{12}$$

where $F(\boldsymbol{\mu},w) = \min[F_1(\boldsymbol{\mu},w),F_2(\boldsymbol{\mu},w)]$, *with*

$$F_a(\boldsymbol{\mu},w) = \begin{cases} w_a\, d\big(\mu_a,\mu_{a,3}(w)\big) + w_3\, d\big(\mu_3,\mu_{a,3}(w)\big) & \textit{if } \mu_4 \geq \mu_{a3}(w)\ , \\ w_a\, d\big(\mu_a,\mu_{a,3,4}(w)\big) + w_3\, d\big(\mu_3,\mu_{a,3,4}(w)\big) + w_4\, d\big(\mu_4,\mu_{a,3,4}(w)\big) & \textit{otherwise.} \end{cases}$$

A particular case. When $\mu_4 > \mu_2$, for any $w \in \Sigma_K$ it holds that $\mu_4 \geq \mu_{1,3}(w)$ and $\mu_4 \geq \mu_{2,3}(w)$. Hence the complexity term can be rewritten to

$$T^*(\boldsymbol{\mu})^{-1} = \sup_{w\in\Sigma_K} \min_{a=1,2} w_a\, d\big(\mu_a,\mu_{a,3}(w)\big) + w_3\, d\big(\mu_3,\mu_{a,3}(w)\big)\ .$$

In that case it is possible to show that the following quantity,

$$w^*(\boldsymbol{\mu}) = \operatorname*{argmax}_{w\in\Sigma_K} \min_{a=1,2} w_a\, d\big(\mu_a,\mu_{a,3}(w)\ \big) + w_3\, d\big(\mu_3,\mu_{a,3}(w)\big)$$

is unique and to give an explicit expression. This quantity is to be interpreted as the vector of proportions of draws of the arms by a strategy matching the lower bound. In this particular case, one finds $w_4^*(\boldsymbol{\mu}) = 0$, showing that an optimal strategy could draw arm 4 only an asymptotically vanishing proportion of times as δ and ϵ go to 0.

Towards an Asymptotically Optimal Algorithm. Assume that the solution of the general optimization problem (12) is well-behaved (unicity of the solution, continuity in the parameters, …) and that we can find an efficient algorithm to compute, for any given $\boldsymbol{\mu}$,

$$w^*(\boldsymbol{\mu}) = \operatorname*{argmax}_{w\in\Sigma_K} \min[F_1(\boldsymbol{\mu},w),F_2(\boldsymbol{\mu},w)].$$

Letting $\mathrm{Alt}(\boldsymbol{\mu}) = \{\boldsymbol{\mu}' : i^*(\boldsymbol{\mu}) \neq i^*(\boldsymbol{\mu}')\}$, one can introduce

$$\begin{aligned} \hat{Z}(t) &:= \inf_{\boldsymbol{\mu}'\in\mathrm{Alt}(\hat{\boldsymbol{\mu}}(t))} \sum_{a=1}^K N_a(t) d(\hat{\mu}_a(t),\mu_a') = tF\Big(\big(N_a(t)/t\big)_{a=1\ldots4}, \hat{\boldsymbol{\mu}}(t)\Big) \\ &= \log \frac{\max_{\boldsymbol{\mu}'\notin\mathrm{Alt}(\hat{\boldsymbol{\mu}}(t))} p_{\boldsymbol{\mu}'}(X_1,\ldots,X_t)}{\max_{\boldsymbol{\mu}'\in\mathrm{Alt}(\hat{\boldsymbol{\mu}}(t))} p_{\boldsymbol{\mu}'}(X_1,\ldots,X_t)}, \end{aligned}$$

where $p_{\boldsymbol{\mu}'}(X_1,\ldots,X_t)$ is the likelihood of the observations up to time t under the model parametrized by $\boldsymbol{\mu}' \in [0,1]^4$. Then, if we can design a sampling rule ensuring that for all a, $N_a(t)/t$ tends to $w_a^*(\boldsymbol{\mu})$, and if we combine it with the stopping rule

$$\tau_\delta = \inf\Big\{t \in \mathbb{N} : \hat{Z}(t) > \log(Ct/\delta)\Big\}$$

for some positive constant C, then one can prove the following:

$$\limsup_{\delta\to 0} \frac{\mathbb{E}_{\boldsymbol{\mu}}[\tau_\delta]}{\log(1/\delta)} \leq T^*(\boldsymbol{\mu}).$$

But proving that this stopping rule does ensures a δ-PAC algorithm is not straightforward, and the analysis remains to be done.

Acknowledgments

This work was partially supported by the CIMI (Centre International de Mathématiques et d'Informatique) Excellence program while Emilie Kaufmann visited Toulouse in November 2015. Garivier and Kaufmann acknowledge the support of the French Agence Nationale de la Recherche (ANR), under grants ANR-13-BS01-0005 (project SPADRO) and ANR-13-CORD-0020 (project ALICIA). Koolen acknowledges support from the Netherlands Organization for Scientific Research (NWO) under Veni grant 639.021.439.

References

J-Y. Audibert, S. Bubeck, and R. Munos. Best Arm Identification in Multi-armed Bandits. In *Proceedings of the 23rd Conference on Learning Theory*, 2010.

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

M. Bowling, N. Burch, M. Johanson, and O. Tammelin. Heads-up limit hold'em poker is solved. *Science*, 347(6218):145–149, January 2015.

C. Browne, E. Powley, D. Whitehouse, S. Lucas, P. Cowling, P. Rohlfshagen, S. Tavener, D. Perez, S. Samothrakis, and S. Colton. A survey of monte carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in games,*, 4(1):1–49, 2012.

S. Bubeck and N. Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Fondations and Trends in Machine Learning*, 5(1):1–122, 2012.

S. Bubeck, R. Munos, and G. Stoltz. Pure Exploration in Finitely Armed and Continuous Armed Bandits. *Theoretical Computer Science 412, 1832-1852*, 412:1832–1852, 2011.

O. Cappé, A. Garivier, O-A. Maillard, R. Munos, and G. Stoltz. Kullback-Leibler upper confidence bounds for optimal sequential allocation. *Annals of Statistics*, 41(3):1516–1541, 2013.

E. Even-Dar, S. Mannor, and Y. Mansour. Action Elimination and Stopping Conditions for the Multi-Armed Bandit and Reinforcement Learning Problems. *Journal of Machine Learning Research*, 7: 1079–1105, 2006.

J. Filar and K. Vrieze. *Competitive Markov Decision Processes*. Springer, 1996.

V. Gabillon, M. Ghavamzadeh, and A. Lazaric. Best Arm Identification: A Unified Approach to Fixed Budget and Fixed Confidence. In *Advances in Neural Information Processing Systems*, 2012.

Aurélien Garivier and Emilie Kaufmann. Optimal best arm identification with fixed confidence. In *Proceedings of the 29th Conference On Learning Theory (to appear)*, 2016.

S. Gelly, L. Kocsis, M. Schoenauer, M. Sebag, D. Silver, C. Szepesvári, and O. Teytaud. The grand challenge of computer go: Monte carlo tree search and extensions. *Commun. ACM*, 55(3):106–113, 2012.

K. Jamieson, M. Malloy, R. Nowak, and S. Bubeck. lil'UCB: an Optimal Exploration Algorithm for Multi-Armed Bandits. In *Proceedings of the 27th Conference on Learning Theory*, 2014.

S. Kalyanakrishnan, A. Tewari, P. Auer, and P. Stone. PAC subset selection in stochastic multi-armed bandits. In *International Conference on Machine Learning (ICML)*, 2012.

E. Kaufmann and S. Kalyanakrishnan. Information complexity in bandit subset selection. In *Proceeding of the 26th Conference On Learning Theory.*, 2013.

E. Kaufmann, O. Cappé, and A. Garivier. On the Complexity of Best Arm Identification in Multi-Armed Bandit Models. *Journal of Machine Learning Research (to appear)*, 2015.

L. Kocsis and C. Szepesvári. Bandit based monte-carlo planning. In *Proceedings of the 17th European Conference on Machine Learning*, ECML'06, pages 282–293, Berlin, Heidelberg, 2006. Springer-Verlag. ISBN 3-540-45375-X, 978-3-540-45375-8.

O. Maron and A. Moore. The Racing algorithm: Model selection for Lazy learners. *Artificial Intelligence Review*, 11(1-5):113–131, 1997.

R. Munos. *From bandits to Monte-Carlo Tree Search: The optimistic principle applied to optimization and planning.*, volume 7(1). Foundations and Trends in Machine Learning, 2014.

D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis. Mastering the game of go with deep neural networks and tree search. *Nature*, 529:484–489, 2016.

B. Szorenyi, G. Kedenburg, and R. Munos. Optimistic planning in markov decision processes using a generative model. In *Advances in Neural Information Processing Systems*, 2014.

Appendix A. Analysis of the Maximin-LUCB Algorithm

We define the event

$$\mathcal{E}_t = \bigcap_{P\in\mathcal{P}} (\mu_P \in [\mathrm{L}_P(t), \mathrm{U}_P(t)]),$$

so that the event $\mathcal{E}$ defined in Theorem 1 rewrites $\mathcal{E} = \bigcap_{t\in 2\mathbb{N}} \mathcal{E}_t$.

Assume that the event $\mathcal{E}$ holds. The arm $\hat{\imath}$ recommended satisfies, by definition of the stopping rule (and by definition of S_t) that, for all $i \neq \hat{\imath}$

$$\min_{j\in K_{\hat{\imath}}} \mathrm{L}_{(\hat{\imath},j)}(\tau_\delta) > \min_{j\in K_i} \mathrm{U}_{(i,j)}(\tau_\delta) - \epsilon.$$

Using that $\mathrm{L}_P(\tau_\delta) \leq \mu_P \leq \mathrm{U}_P(\tau_\delta)$ for all $P \in \mathcal{P}$ (by definition of $\mathcal{E}$) yields for all i

$$\mu_{\hat{\imath},1} = \min_{j\in K_{\hat{\imath}}} \mu_{\hat{\imath},j} > \min_{j\in K_i} \mu_{i,j} - \epsilon = \mu_{i,1} - \epsilon,$$

hence $\max_{i\neq\hat{\imath}} \mu_{i,1} - \mu_{\hat{\imath},1} < \epsilon$. Thus, either $\hat{\imath} = 1$ or $\hat{\imath}$ satisfies $\mu_{1,1} - \mu_{\hat{\imath},1} < \epsilon$. In both case, $\hat{\imath}$ is ϵ-optimal, which proves that M-LUCB is correct on $\mathcal{E}$.

Now we analyze M-LUCB with $\epsilon = 0$. Our analysis is based on the following two key lemmas, whose proof is given below.

Lemma 10 *Let $c \in [\mu_{2,1}, \mu_{1,1}]$ and $t \in 2\mathbb{N}$. On $\mathcal{E}_t$, if $(\tau_\delta > t)$, there exists $P \in \{H_t, S_t\}$ such that*

$$(c \in [\mathrm{L}_P(t), \mathrm{U}_P(t)])\,.$$

Lemma 11 *Let $c \in [\mu_{2,1}, \mu_{1,1}]$ and $t \in 2\mathbb{N}$. On $\mathcal{E}_t$, for every $(i,j) \in \{H_t, S_t\}$,*

$$c \in [\mathrm{L}_{(i,j)}(t), \mathrm{U}_{(i,j)}(t)] \;\;\Rightarrow\;\; N_{(i,j)}(t) \le \min\left(\frac{2}{(\mu_{i,1} - c)^2}, \frac{2}{(\mu_{i,j} - \mu_{i,1})^2}\right)\beta(t,\delta).$$

Moreover, for $j \in \{1, \dots, K_1\}$, one has

$$c \in [\mathrm{L}_{(1,j)}(t), \mathrm{U}_{(1,j)}(t)] \;\;\Rightarrow\;\; N_{(1,j)}(t) \le \frac{2}{(\mu_{1,j} - c)^2}\beta(t,\delta).$$

Defining, for every arm $P = (i,j) \in \mathcal{P}$ the constant

$$c_P = \frac{1}{\max\left[\left(\mu_{i,1} - \frac{\mu_{1,1}+\mu_{2,1}}{2}\right)^2, (\mu_{i,j} - \mu_{i,1})^2\right]} \text{ if } P \in \mathcal{P} \setminus \mathcal{P}_1, \;\; c_P = \frac{1}{\left(\mu_{1,j} - \frac{\mu_{1,1}+\mu_{2,1}}{2}\right)^2} \text{ if } P = (1,j) \in \mathcal{P}_1$$

combining the two lemmas (for the particular choice $c = \frac{\mu_{1,1}+\mu_{2,1}}{2}$) yields the following key statement:

$$\mathcal{E}_t \cap (\tau_\delta > t) \;\;\Rightarrow\;\; \exists P \in \{H_t, S_t\} : N_P(t) \le 2c_P\beta(t,\delta). \tag{13}$$

Note that $H_1^*(\boldsymbol{\mu}) = \sum_{P \in \mathcal{P}} c_P$, from its definition in Theorem 1.

A.1. Proof of Theorem 1

Let T be a deterministic time. On the event $\mathcal{E} = \bigcap_{t \in 2\mathbb{N}} \mathcal{E}_t$, using (13) and the fact that for every even t, $(\tau_\delta > t) = (\tau_\delta > t+1)$ by definition of the algorithm, one has

$$\begin{aligned}
\min(\tau_\delta, T) &= \sum_{t=1}^{T} \mathbb{1}_{(\tau_\delta > t)} = 2\sum_{\substack{t \in 2\mathbb{N} \\ t \le T}} \mathbb{1}_{(\tau_\delta > t)} = 2\sum_{\substack{t \in 2\mathbb{N} \\ t \le T}} \mathbb{1}_{(\exists P \in \{H_t, S_t\} : N_P(t) \le 2c_p\beta(t,\delta))} \\
&\le 2\sum_{\substack{t \in 2\mathbb{N} \\ t \le T}} \sum_{P \in \mathcal{P}} \mathbb{1}_{(P_{t+1} = P) \cup (P_{t+2} = P)} \mathbb{1}_{(N_P(t) \le 2c_P\beta(T,\delta))} \\
&\le 4\sum_{P \in \mathcal{P}} c_P\beta(T,\delta) = 4H_1^*(\boldsymbol{\mu})\beta(T,\delta).
\end{aligned}$$

For any T such that $4H_1^*(\boldsymbol{\mu})\beta(T,\delta) < T$, one has $\min(\tau_\delta, T) < T$, which implies $\tau_\delta < T$. Therefore $\tau_\delta \le T(\boldsymbol{\mu}, \delta)$ for $T(\boldsymbol{\mu}, \delta)$ defined in Theorem 1.

A.2. Proof of Theorem 4

Let $\gamma > 0$. Let T be a deterministic time. On the event $\mathcal{G}_T = \bigcap_{\substack{t \in 2\mathbb{N} \\ \lfloor \gamma T \rfloor \le t \le T}} \mathcal{E}_t$, one can write

$$\begin{aligned}
\min(\tau_\delta, T) &= 2\gamma T + 2\sum_{\substack{t \in 2\mathbb{N} \\ \lfloor \gamma T \rfloor \le t \le T}} \mathbb{1}_{(\tau_\delta > t)} = 2\gamma T + 2\sum_{\substack{t \in 2\mathbb{N} \\ \lfloor \gamma T \rfloor \le t \le T}} \mathbb{1}_{(\exists P \in \{H_t, S_t\} : N_P(t) \le 2c_p\beta(t,\delta))} \\
&\le 2\gamma T + 2\sum_{\substack{t \in 2\mathbb{N} \\ \lfloor \gamma T \rfloor \le t \le T}} \sum_{P \in \mathcal{P}} \mathbb{1}_{(P_{t+1} = P) \cup (P_{t+1} = P)} \mathbb{1}_{(N_P(t) \le 2c_P\beta(T,\delta))} \\
&\le 2\gamma T + 4H_1^*(\boldsymbol{\mu})\beta(T,\delta).
\end{aligned}$$

Introducing $T_\gamma(\boldsymbol{\mu},\delta) := \inf\{T \in \mathbb{N} : 4H_1^*(\boldsymbol{\mu})\beta(T,\delta) < (1-2\gamma)T\}$, for all $T \geq T_\gamma(\boldsymbol{\mu},\delta)$, $\mathcal{G}_T \subseteq (\tau_\delta \leq T)$. One can bound the expectation of τ_δ in the following way (using notably the self-normalized deviation inequality of Cappé et al. (2013)):

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{\mu}}[\tau_\delta] &= \sum_{T=1}^{\infty} \mathbb{P}_{\boldsymbol{\mu}}(\tau_\delta > T) \leq T_\gamma + \sum_{T=T_\gamma}^{\infty} \mathbb{P}_{\boldsymbol{\mu}}\left(\tau_\delta > T\right) \leq T_\gamma + \sum_{T=T_\gamma}^{\infty} \mathbb{P}_{\boldsymbol{\mu}}\left(\mathcal{G}_T^c\right) \\
&\leq T_\gamma + \sum_{T=1}^{\infty}\sum_{t=\gamma T}^{T}\sum_{P\in\mathcal{P}}\left[\mathbb{P}_{\boldsymbol{\mu}}\left(\mu_P > \hat{\mu}_P(t) + \sqrt{\frac{\beta(t,\delta)}{2N_P(t)}}\right) + \mathbb{P}_{\boldsymbol{\mu}}\left(\mu_P < \hat{\mu}_P(t) - \sqrt{\frac{\beta(t,\delta)}{2N_P(t)}}\right)\right] \\
&\leq T_\gamma + \sum_{T=1}^{\infty}\sum_{t=\gamma T}^{T} 2\overline{K}\mathbb{P}_{\boldsymbol{\mu}}\left(\mu_P > \hat{\mu}_P(t) + \sqrt{\frac{\beta(t,1)}{2N_P(t)}}\right) \\
&\leq T_\gamma + \sum_{T=1}^{\infty}\sum_{t=\gamma T}^{T} 2\overline{K}e\log(t)\beta(t,1)\exp(-\beta(t,1)) \\
&\leq T_\gamma + \sum_{T=1}^{\infty} 2\overline{K}eT\log(T)\beta(T,1)\exp(-\beta(\gamma T,1)) \\
&= T_\gamma + \sum_{T=1}^{\infty} \frac{2\overline{K}eT\log(T)\log(CT^{1+\alpha})}{C\gamma^{1+\alpha}T^{1+\alpha}},
\end{aligned}$$

where the series is convergent for $\alpha > 1$. One has

$$T_\gamma(\boldsymbol{\mu},\delta) = \inf\left\{T \in \mathbb{N} : \log\left(\frac{CT^{1+\alpha}}{\delta}\right) < \frac{(1-2\gamma)T}{4H_1^*(\boldsymbol{\mu})}\right\}.$$

The technical Lemma 12 below permits to give an upper bound on $T_\gamma(\boldsymbol{\mu},\delta)$ for small values of δ, that implies in particular

$$\limsup_{\delta\to 0}\frac{\mathbb{E}_{\boldsymbol{\mu}}[\tau_\delta]}{\log(1/\delta)} \leq \frac{4(1+\alpha)H_1^*(\boldsymbol{\mu})}{1-2\gamma}.$$

Letting γ go to zero yields the result.

Lemma 12 *If $\alpha, c_1, c_2 > 0$ are such that $a = (1+\alpha)c_2^{1/(1+\alpha)}/c_1 > 4.85$, then*

$$x = \frac{1+\alpha}{c_1}\big(\log(a) + 2\log(\log(a))\big)$$

is such that $c_1 x \geq \log(c_2 x^{1+\alpha})$.

Proof. One can check that if $a \geq 4.85$, then $\log^2(a) > \log(a) + 2\log(\log(a))$. Thus, $y = \log(a) + 2\log(\log(a))$ is such that $y \geq \log(ay)$. Using $y = c_1 x/(1+\alpha)$ and $a = (1+\alpha)c_2^{1/(1+\alpha)}/c_1$, one obtains the result.

□

A.3. Proof of Lemma 10

We show that on $\mathcal{E}_t \cap (\tau_\delta > t)$, the following four statements cannot occur, which yields that the threshold c is contained in one of the intervals $\mathcal{I}_{H_t}(t)$ or $\mathcal{I}_{S_t}(t)$:

1. $(\mathrm{L}_{H_t}(t) > c) \cap (\mathrm{L}_{S_t}(t) > c)$
2. $(\mathrm{U}_{H_t}(t) < c) \cap (\mathrm{U}_{S_t}(t) < c)$
3. $(\mathrm{U}_{H_t}(t) < c) \cap (\mathrm{L}_{S_t}(t) > c)$
4. $(\mathrm{L}_{H_t}(t) > c) \cap (\mathrm{U}_{S_t}(t) < c)$

1. implies that there exists two actions i and i' such that $\forall j \leq K_i, \mathrm{L}_{i,j}(t) \geq c$ and $\forall j' \leq K_{i'}, \mathrm{L}_{i',j'}(t) \geq c$. Because $\mathcal{E}_t$ holds, one has in particular $\mu_{i,1} > c$ and $\mu_{j,1} > c$, which is excluded since $\mu_{1,1}$ is the only such arm that is larger than c.

2. implies that for all $i \in \{1, K\}$, $\mathrm{U}_{(i,c_i(t))}(t) \leq c$. Thus, in particular $\mathrm{U}_{(1,c_1(t))} \leq c$ and, as $\mathcal{E}_t$ holds, there exists $j \leq K_1$ such that $\mu_{1,j} < c$, which is excluded.

3. implies that there exists $i \neq \hat{\imath}(t)$ such that $\min_j \hat{\mu}_{i,j}(t) > \hat{\mu}_{H_t}(t) \geq \min_j \hat{\mu}_{(\hat{\imath}(t),j)}(t)$, which contradicts the definition of $\hat{\imath}(t)$.

4. implies that $\mathrm{L}_{H_t}(t) > \mathrm{U}_{S_t}(t)$, thus the algorithm must have stopped before the t-th round, which is excluded since $\tau_\delta > t$.

We proved that there exists $P \in \{H_t, S_t\}$ such that $c \in \mathcal{I}_P(t)$.

A.4. Proof of Lemma 11

Assume that $\mathcal{E}_t$ holds and that $c \in [\mathrm{L}_{(i,j)}(t), \mathrm{U}_{(i,j)}(t)]$. We first show that $\mu_{i,1}$ is also contained in $[\mathrm{L}_{(i,j)}(t), \mathrm{U}_{(i,j)}(t)]$. First, by definition of the algorithm, if $(i,j) = H_t$ or S_t, one has $(i,j) = (i, c_i(t))$, hence

$$\mathrm{L}_{(i,j)}(t) \leq \mathrm{L}_{(i,1)}(t) \leq \mu_{i,1},$$

using that $\mathcal{E}_t$ holds. Now, if we assume that $\mu_{i,1} > \mathrm{U}_{(i,j)}(t)$, because $\mathcal{E}_t$ holds, one has $\mu_{i,1} > \mu_{i,j}$, which is a contradiction. Thus, $\mu_{i,1} \leq \mathrm{U}_{(i,j)}(t)$.

As c and $\mu_{i,1}$ are both contained in $[\mathrm{L}_{(i,j)}(t), \mathrm{U}_{(i,j)}(t)]$, whose diameter is $2\sqrt{\beta(t,\delta)/(2N_{(i,j)}(t))}$, one has

$$|c - \mu_{i,1}| < 2\sqrt{\frac{\beta(t,\delta)}{2N_{(i,j)}(t)}} \quad \Leftrightarrow \quad N_{(i,j)}(t) \leq \frac{2\beta(t,\delta)}{(\mu_{i,1} - c)^2}.$$

Moreover, one can use again that $\mathrm{L}_{(i,j)}(t) \leq \mathrm{L}_{(i,1)}(t)$ to write

$$\begin{aligned} \mathrm{U}_{(i,j)}(t) - 2\sqrt{\frac{\beta(t,\delta)}{2N_{(i,j)}(t)}} &\leq \mathrm{L}_{(i,1)}(t) \\ \mu_{i,j} - 2\sqrt{\frac{\beta(t,\delta)}{2N_{(i,j)}(t)}} &\leq \mu_{i,1}, \end{aligned}$$

which yields $N_{(i,j)}(t) \leq \frac{2\beta(t,\delta)}{(\mu_{i,j} - \mu_{i,1})^2}$.

If $i = 1$, note that $|\mu_{1,j} - c| > \max(|\mu_{1,1} - c|, |\mu_{1,j} - \mu_{1,1}|)$. Using that c and $\mu_{1,j}$ both belong to $\mathcal{I}_{(1,j)}(t)$), whose diameter is $2\sqrt{\beta(t,\delta)/(2N_{(1,j)}(t))}$ yield, using a similar argument as above, that for all $j \in \{1, \dots, K_1\}$,

$$N_{(1,j)}(t) \leq \frac{2\beta(t,\delta)}{(\mu_{1,j} - c)^2}.$$

A.5. Proof of Theorem 5

In the particular case of two actions by player, we analyze the version of LUCB that draws only one arm per round. More precisely, in this particular case, letting

$$X_t = \underset{j=1,2}{\text{argmin}}\ \mathrm{L}_{(1,j)}(t) \quad \text{and} \quad Y_t = \underset{j=1,2}{\text{argmin}}\ \mathrm{L}_{(2,j)}(t),$$

one has $P_{t+1} = \underset{P \in \{X_t, Y_t\}}{\text{argmax}}\ N_P(t)$.

The analysis follows the same lines as that of Theorem 4. First, we notice that the algorithm outputs the maximin action on the event $\mathcal{E} = \cap_{t \in \mathbb{N}} \mathcal{E}_t$, and thus the exploration rate defined in Corollary 2 guarantees a δ-PAC algorithm. Then, the sample complexity analysis relies on a specific characterization of the draw of each of the arms given in Lemma 13 below (which is a counterpart of Lemma 11). This result justifies the new complexity term that appears in Theorem 5, as $\sum_{1 \leq i,j \leq 2} c_{(i,j)} \leq H^*(\boldsymbol{\mu})$.

Lemma 13 *On the event $\mathcal{E}$, for all $P \in \mathcal{P}$, one has*

$$(P_{t+1} = P) \cap (\tau_\delta > t) \subseteq (N_P(t) \leq 8 c_P \beta(t,\delta)),$$

with

$$c_{(1,1)} = \frac{1}{(\mu_{1,1} - \mu_{2,1})^2}, \quad c_{(1,2)} = \frac{1}{(\mu_{1,2} - \mu_{2,1})^2}, \quad c_{(2,1)} = \frac{1}{(\mu_{1,1} - \mu_{2,1})^2},$$

and

$$c_{(2,2)} = \frac{1}{\max(4(\mu_{2,2} - \mu_{2,1})^2, (\mu_{1,1} - \mu_{2,1})^2)}.$$

Proof of Lemma 13. The proof of this result uses extensively the fact that the confidence intervals in (3) are symmetric:

$$\mathrm{U}_P(t) = \mathrm{L}_P(t) + 2\sqrt{\frac{\beta(t,\delta)}{2N_P(t)}}.$$

Assume that $(P_{t+1} = (1,1))$. By definition of the sampling strategy, one has $\mathrm{L}_{(1,1)}(t) \leq \mathrm{L}_{(1,2)}(t)$ and $N_{(1,1)}(t) \leq N_{Y_t}(t)$. If $(\tau_\delta > t)$, one has

$$\begin{aligned} \mathrm{L}_{(1,1)}(t) &\leq \mathrm{U}_{Y_t}(t) \\ \mathrm{U}_{(1,1)}(t) - 2\sqrt{\frac{\beta(t,\delta)}{2N_{(1,1)}(t)}} &\leq \mathrm{L}_{Y_t}(t) + 2\sqrt{\frac{\beta(t,\delta)}{2N_{Y_t}(t)}}. \end{aligned}$$

On $\mathcal{E}$, $\mu_{1,1} \leq \mathrm{U}_{(1,1)}(t)$ and $\mathrm{L}_{Y_t}(t) = \min(\mathrm{L}_{(2,1)}(t), \mathrm{L}_{(2,2)}(t)) \leq \min(\mu_{2,1}, \mu_{2,2}) = \mu_{2,1}$. Thus

$$\mu_{1,1} - \mu_{2,1} \leq 2\sqrt{\frac{\beta(t,\delta)}{2N_{Y_t}(t)}} + 2\sqrt{\frac{\beta(t,\delta)}{2N_{(1,1)}(t)}} \leq 4\sqrt{\frac{\beta(t,\delta)}{2N_{(1,1)}(t)}},$$

using that $N_{(1,1)}(t) \le N_{Y_t}(t)$. This proves that

$$(P_{t+1} = (1,1)) \cap (\tau_\delta > t) \subseteq \left(N_{(1,1)}(t) \le \frac{8\beta(t,\delta)}{(\mu_{1,1} - \mu_{2,1})^2} \right).$$

A very similar reasoning shows that

$$(P_{t+1} = (1,2)) \cap (\tau_\delta > t) \subseteq \left(N_{(1,2)}(t) \le \frac{8\beta(t,\delta)}{(\mu_{1,2} - \mu_{2,1})^2} \right).$$

Assume that $(P_{t+1} = (2,1))$. If $(\tau_\delta > t)$, one has

$$\begin{aligned} \mathrm{L}_{X_t}(t) &\le \mathrm{U}_{(2,1)}(t) \\ \mathrm{U}_{X_t}(t) - 2\sqrt{\frac{\beta(t,\delta)}{2N_{X_t}(t)}} &\le \mathrm{L}_{(2,1)}(t) + 2\sqrt{\frac{\beta(t,\delta)}{2N_{(2,1)}(t)}}. \end{aligned}$$

On $\mathcal{E}$, $\mu_{1,1} \le \mu_{X_t} \le \mathrm{U}_{X_t}(t)$ and $\mathrm{L}_{(2,1)}(t) \le \mu_{2,1}$. Thus

$$\mu_{1,1} - \mu_{2,1} \le 2\sqrt{\frac{\beta(t,\delta)}{2N_{X_t}(t)}} + 2\sqrt{\frac{\beta(t,\delta)}{2N_{(2,1)}(t)}} \le 4\sqrt{\frac{\beta(t,\delta)}{2N_{(2,1)}(t)}},$$

using that $N_{(2,1)}(t) \le N_{X_t}(t)$. This proves that

$$(P_{t+1} = (2,1)) \cap (\tau_\delta > t) \subseteq \left(N_{(2,1)}(t) \le \frac{8\beta(t,\delta)}{(\mu_{1,1} - \mu_{2,1})^2} \right).$$

Assume that $(P_{t+1} = (2,2))$. First, using the fact that $\mathrm{L}_{(2,2)}(t) \le \mathrm{L}_{(2,1)}(t)$ yields, on $\mathcal{E}$,

$$\begin{aligned} \mathrm{U}_{(2,2)}(t) - 2\sqrt{\frac{\beta(t,\delta)}{2N_{(2,2)}(t)}} &\le \mu_{2,1} \\ \mu_{2,2} - \mu_{2,1} &\le 2\sqrt{\frac{\beta(t,\delta)}{2N_{(2,2)}(t)}}, \end{aligned}$$

which leads to $N_{(2,2)}(t) \le 2\beta(t,\delta)/(\mu_{2,2} - \mu_{2,1})^2$. Then, if $(\tau_\delta > t)$, on $\mathcal{E}$ (using also that $\mathrm{L}_{(2,2)}(t) \le \mathrm{L}_{(2,1)}(t)$),

$$\begin{aligned} \mathrm{L}_{X_t}(t) &\le \mathrm{U}_{(2,2)}(t) \\ \mathrm{U}_{X_t}(t) - 2\sqrt{\frac{\beta(t,\delta)}{2N_{X_t}(t)}} &\le \mathrm{L}_{(2,2)}(t) + 2\sqrt{\frac{\beta(t,\delta)}{2N_{(2,2)}(t)}} \\ \mathrm{U}_{X_t}(t) - 2\sqrt{\frac{\beta(t,\delta)}{2N_{X_t}(t)}} &\le \mathrm{L}_{(2,1)}(t) + 2\sqrt{\frac{\beta(t,\delta)}{2N_{(2,2)}(t)}} \\ \mu_{1,1} - 2\sqrt{\frac{\beta(t,\delta)}{2N_{X_t}(t)}} &\le \mu_{2,1} + 2\sqrt{\frac{\beta(t,\delta)}{2N_{(2,2)}(t)}} \\ \mu_{1,1} - \mu_{2,1} &\le 4\sqrt{\frac{\beta(t,\delta)}{2N_{(2,2)}(t)}}. \end{aligned}$$

Thus, if $\mu_{2,2} < \mu_{1,1}$, one also has $N_{(2,2)}(t) \le 8\beta(t,\delta)/(\mu_{1,1} - \mu_{2,1})^2$. Combining the two bounds yield

$$(P_{t+1} = (2,2)) \cap (\tau_\delta > t) \subseteq \left(N_{(2,2)}(t) \le \frac{8\beta(t,\delta)}{\max\left(4(\mu_{2,2} - \mu_{2,1})^2, (\mu_{1,1} - \mu_{2,1})^2\right)} \right).$$

Appendix B. Analysis of the Maximin-Racing Algorithm

B.1. Proof of Proposition 7.

First note that for every $P \in \mathcal{P}$, introducing an i.i.d. sequence of successive observations from arm P, the sequence of associated empirical means $(\hat{\mu}_P(r))_{r\in\mathbb{N}}$ is defined independently of the arm being active.

We introduce the event $\mathcal{E} = \mathcal{E}_1 \cap \mathcal{E}_2$ with

$$\begin{aligned}
\mathcal{E}_1 &= \bigcap_{i=1}^{K} \bigcap_{\substack{(i,j)\in\mathcal{P}_i: \\ \mu_{i,j}=\mu_{i,1}}} \bigcap_{\substack{(i,j')\in\mathcal{P}_i: \\ \mu_{i,j'}>\mu_{i,1}}} \left(\forall r \in \mathbb{N}, rf(\hat{\mu}_{i,j}(r), \hat{\mu}_{i,j'}(r)) \le \beta(r,\delta)\right) \\
\mathcal{E}_2 &= \bigcap_{\substack{i\in\{1,\dots,K\}: \\ \mu_{i,1}<\mu_{1,1}}} \bigcap_{\substack{(i,j)\in A_i: \\ \mu_{i,j}=\mu_{i,1}}} \bigcap_{\substack{i'\in\{1,\dots,K\}: \\ \mu_{i',1}=\mu_{1,1}}} \bigcap_{(i',j')\in A_{i'}} \left(\forall r \in \mathbb{N}, rf(\hat{\mu}_{i,j}(r), \hat{\mu}_{i',j'}(r)) \le \beta(r,\delta)\right)
\end{aligned}$$

and the event

$$\mathcal{F} = \bigcap_{P\in\mathcal{P}} \left(|\hat{\mu}_P(r_0) - \mu_P| \le \frac{\epsilon}{2} \right).$$

From (9) and a union bound, $\mathbb{P}(\mathcal{E}^c) \le \delta/2$. From Hoeffding inequality and a union bound, using also the definition of r_0, one has $\mathbb{P}(\mathcal{F}^c) \le \delta/2$. Finally, $\mathbb{P}_{\boldsymbol{\mu}}(\mathcal{E} \cap \mathcal{F}) \ge 1 - \delta$.

We now show that on $\mathcal{E} \cap \mathcal{F}$, the algorithm outputs an ϵ-optimal arm. On the event $\mathcal{E}$, the following two statements are true for any round $r \le r_0$:

1. For all i, if $\mathcal{R}_i \ne \emptyset$, then there exists $(i,j) \in \mathcal{R}_i$ such that $\mu_{i,j} = \mu_{i,1}$
2. If there exists i such that $\mathcal{R}_i \ne \emptyset$, then there exists $i' : \mu_{i',1} = \mu_{1,1}$ such that $\mathcal{R}_{i'} \ne \emptyset$.

Indeed, if 1. is not true, there is a non empty set $\mathcal{R}_i$ in which all the arms in the set $\{(i,j) \in \mathcal{P}_i : \mu_{i,j} = \mu_{i,1}\}$ have been discarded. Hence, in a previous round at least one of these arms must have appeared strictly larger than one of the arms in the set $\{(i,j') \in \mathcal{P}_i : \mu_{i,j'} > \mu_{i,1}\}$ (in the sense of our elimination rule), which is not possible from the definition of $\mathcal{E}_1$. Now if 2. is not true, there exists $i' : \mu_{i',1} = \mu_{1,1}$, such that $\mathcal{R}_{i'}$ has been discarded at a previous round by some non-empty set $\mathcal{R}_i$, with $\mu_{i,1} < \mu_{1,1}$. Hence, there exists $(i',j') \in A_{i'}$ that appears significantly smaller than all arms in $\mathcal{R}_i$ (in the sense of our elimination rule). As $\mathcal{R}_i$ contains by 1. some arm $\mu_{i,j}$ with $\mu_{i,j} = \mu_{i,1}$, there exists r such that $rd(\mu_{(i,j)}(r), \mu_{(i',j')}(r)) > \beta(r,\delta)$, which contradicts the definition of $\mathcal{E}_2$.

From the statements 1. and 2., on $\mathcal{E} \cap \mathcal{F}$ if the algorithm terminates before r_0, using that the last set in the race $\mathcal{R}_i$ must satisfy $\mu_{i,1} = \mu_{1,1}$, the action $\hat{\imath}$ is in particular ϵ-optimal. If the algorithm has not stopped at r_0, the arm $\hat{\imath}$ recommended is the empirical maximin action. Letting $\mathcal{R}_i$ some set still in the race with $\mu_{i,1} = \mu_{1,1}$, one has,

$$\min_{P\in\mathcal{R}_{\hat{\imath}}} \hat{\mu}_P(r_0) \ge \min_{P\in\mathcal{R}_i} \hat{\mu}_P(r_0).$$

As $\mathcal{F}$ holds and because there exists $(\hat{\imath},\hat{\jmath}) \in \mathcal{R}_{\hat{\imath}}$ with $\mu_{\hat{\imath},\hat{\jmath}} = \mu_{\hat{\imath},1}$, and $(i,j) \in \mathcal{R}_i$ with $\mu_{i,j} = \mu_{1,1}$, one has

$$\begin{aligned}
\min_{P\in\mathcal{R}_i} \hat{\mu}_P(r_0) &\ge \min_{P\in\mathcal{R}_i} (\mu_P - \epsilon/2) = \mu_{i,j} - \epsilon/2 = \mu_{1,1} - \epsilon/2. \\
\min_{P\in\mathcal{R}_{\hat{\imath}}} \hat{\mu}_P(r_0) &\le \min_{P\in\mathcal{R}_{\hat{\imath}}} (\mu_P + \epsilon/2) = \mu_{\hat{\imath},\hat{\jmath}} + \epsilon/2 = \mu_{\hat{\imath},1} + \epsilon/2.
\end{aligned}$$

and thus $\hat{\imath}$ is ϵ-optimal, since

$$\mu_{\hat{\imath},1} + \frac{\epsilon}{2} \geq \mu_{1,1} - \frac{\epsilon}{2} \quad \Leftrightarrow \quad \mu_{1,1} - \mu_{\hat{\imath},1} \leq \epsilon.$$

□

B.2. Proof of Theorem 8

Recall $\mu_{1,1} > \mu_{2,1}$. We present the proof assuming additionally that for all $i \in \{1, K\}$, $\mu_{i,1} < \mu_{i,2}$ (an assumption that can be relaxed, at the cost of more complex notations).

Let $\alpha > 0$. The function f defined in (8) is uniformly continuous on $[0,1]^2$, thus there exists η^α such that

$$||(x,y) - (x',y')||_\infty \leq \eta^\alpha \quad \Rightarrow \quad |f(x,y) - f(x',y')| \leq \alpha.$$

We introduce the event

$$\mathcal{G}_{\alpha,r} = \bigcap_{P \in \mathcal{P}} (|\hat{\mu}_P(r) - \mu_P| \leq \eta^\alpha)$$

and let $\mathcal{E}$ be the event defined in the proof of Lemma 7, which rewrites in a simpler way with our assumptions on the arms :

$$\mathcal{E} = \bigcap_{i=2}^{K} \bigcap_{j=1}^{K_1} (\forall r \in \mathbb{N}, r f(\hat{\mu}_{i,1}(r), \hat{\mu}_{1,j}(r)) \leq \beta(r,\delta)) \bigcap_{i=1}^{K} \bigcap_{j=2}^{K_i} (\forall r \in \mathbb{N}, r f(\hat{\mu}_{i,1}(r), \hat{\mu}_{i,j}(r)) \leq \beta(r,\delta))$$

Recall that on this event, arm (1,1) is never eliminated before the algorithm stops and whenever an arm $(i,j) \in \mathcal{R}$, we know that the corresponding minimal arm $(i,1) \in \mathcal{R}$.

Let $(i,j) \neq (1,1)$ and recall that $\tau_\delta(i,j)$ is the number of rounds during which arm (i,j) is drawn. One has

$$\mathbb{E}_{\boldsymbol{\mu}}[\tau_\delta(i,j)] = \mathbb{E}_{\boldsymbol{\mu}}[\tau_\delta(i,j)\mathbb{1}_{\mathcal{E}}] + \mathbb{E}_{\boldsymbol{\mu}}[\tau_\delta(i,j)\mathbb{1}_{\mathcal{E}^c}] \leq \mathbb{E}_{\boldsymbol{\mu}}[\tau_\delta(i,j)\mathbb{1}_{\mathcal{E}}] + \frac{r_0\delta}{2}. \tag{14}$$

On the event $\mathcal{E}$, if arm (i,j) is still in the race at the end of round r,

- it cannot be significantly larger than $(i,1)$: $r f(\hat{\mu}_{i,j}(r), \hat{\mu}_{i,1}(r)) \leq \beta(r,\delta)$
- arm $(i,1)$ cannot be significantly smaller than $(1,1)$ (otherwise all arms in $\mathcal{R}_i$, including (i,j), are eliminated): $r f(\hat{\mu}_{i,1}(r), \hat{\mu}_{1,1}(r)) \leq \beta(r,\delta)$

Finally, one can write

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{\mu}}[\tau_\delta(i,j)\mathbb{1}_{\mathcal{E}}] \quad & \leq \quad \mathbb{E}_{\boldsymbol{\mu}}\left[\mathbb{1}_{\mathcal{E}} \sum_{r=1}^{r_0} \mathbb{1}_{((i,j)\in\mathcal{R} \text{ at round r})}\right] \\
& \leq \quad \mathbb{E}_{\boldsymbol{\mu}}\left[\sum_{r=1}^{r_0} \mathbb{1}_{(r \max[f(\hat{\mu}_{i,j}(r),\hat{\mu}_{i,1}(r)), f(\hat{\mu}_{i,1}(r),\hat{\mu}_{1,1}(r))] \leq \beta(r,\delta))}\right] \\
& \leq \quad \mathbb{E}_{\boldsymbol{\mu}}\left[\sum_{r=1}^{r_0} \mathbb{1}_{(r \max[f(\hat{\mu}_{i,j}(r),\hat{\mu}_{i,1}(r)), f(\hat{\mu}_{i,1}(r),\hat{\mu}_{1,1}(r))] \leq \beta(r,\delta))} \mathbb{1}_{\mathcal{G}_{\alpha,r}}\right] + \sum_{r=1}^{r_0} \mathbb{P}_{\boldsymbol{\mu}}(\mathcal{G}_{\alpha,r}^c) \\
& \leq \quad \sum_{r=1}^{r_0} \mathbb{1}_{(r(\max[f(\mu_{i,j},\mu_{i,1}), f(\mu_{i,1},\mu_{1,1})] - \alpha) \leq \log(4C_K r/\delta)} + \sum_{r=1}^{\infty} \mathbb{P}_{\boldsymbol{\mu}}(\mathcal{G}_{\alpha,r}^c) \\
& \leq \quad T_{(i,j)}(\delta,\alpha) + \sum_{r=1}^{\infty} 2\overline{K} \exp(-2(\eta^\alpha)^2 r),
\end{aligned}$$

using Hoeffding inequality and introducing

$$T_{(i,j)}(\delta,\alpha) := \inf\left\{r \in \mathbb{N} : r\left(\max\left[f(\mu_{i,j},\mu_{i,1}), f(\mu_{i,1},\mu_{1,1})\right] - \alpha\right) > \log\left(\frac{4C_K r}{\delta}\right)\right\}$$

Some algebra (Lemma 12) shows that $T_{(i,j)}(\delta,\alpha) = \frac{1}{\max[f(\mu_{i,j},\mu_{i,1}),f(\mu_{i,1},\mu_{1,1})]-\alpha}\log\left(\frac{4C_K}{\delta}\right)+o_{\delta\to 0}\left(\log\frac{1}{\delta}\right)$ and finally, for all $\alpha > 0$,

$$\mathbb{E}_{\boldsymbol{\mu}}\left[\tau_\delta(i,j)\right] \le \frac{1}{\max[f(\mu_{i,j},\mu_{i,1}), f(\mu_{i,1},\mu_{1,1})] - \alpha}\log\left(\frac{4C_K}{\delta}\right) + o\left(\log\frac{1}{\delta}\right).$$

As this holds for all α, and keeping in mind the trivial bound $\mathbb{E}_{\boldsymbol{\mu}}\left[\tau_\delta(i,j)\right] \le r_0 = \frac{2}{\epsilon^2}\log\left(\frac{4K}{\delta}\right)$, one obtains

$$\limsup_{\delta\to 0}\frac{\mathbb{E}_{\boldsymbol{\mu}}[\tau_\delta(i,j)]}{\log(1/\delta)} \le \frac{1}{\max\left[\epsilon^2/2, I(\mu_{i,j},\mu_{i,1}), I(\mu_{i,1},\mu_{1,1})\right]}.$$

To upper bound the number of draws of the arm $(1,1)$, one can proceed similarly and write that, for all $\alpha > 0$,

$$\begin{aligned}
\tau_\delta(1,1)\mathbb{1}_{\mathcal{E}} &= \sup_{(i,j)\in\mathcal{P}\backslash\{(1,1)\}} \tau_\delta(i,j)\mathbb{1}_{\mathcal{E}} \\
&\le \sup_{(i,j)\in\mathcal{P}\backslash\{(1,1)\}} \sum_{r=1}^{r_0} \mathbb{1}_{(r\max[f(\hat\mu_{i,j}(r),\hat\mu_{i,1}(r)),f(\hat\mu_{i,1}(r),\hat\mu_{1,1}(r))]\le\beta(r,\delta))} \\
&\le \sup_{(i,j)\in\mathcal{P}\backslash\{(1,1)\}} \sum_{r=1}^{r_0} \mathbb{1}_{(r(f(\mu_{i,j},\mu_{i,1})\wedge f(\mu_{i,1},\mu_{1,1})-\alpha)\le\beta(r,\delta))} + \sum_{r=1}^{\infty}\mathbb{1}_{\mathcal{G}^c_{\alpha,r}} \\
&\le \sup_{(i,j)\in\mathcal{P}\backslash\{(1,1)\}} T_{(i,j)}(\delta,\alpha) + \sum_{r=1}^{\infty}\mathbb{1}_{\mathcal{G}^c_{\alpha,r}}.
\end{aligned}$$

Taking the expectation and using the more explicit expression of the $T_{(i,j)}$ yields

$$\limsup_{\delta\to 0}\frac{\mathbb{E}_{\boldsymbol{\mu}}[\tau_\delta(1,1)]}{\log(1/\delta)} \le \frac{1}{\max\left[\epsilon^2/2, I(\mu_{2,1},\mu_{1,1})\right]}.$$

For all $j \in \{2,\ldots,K_1\}$, using that on $\mathcal{E}$, $\tau_{(1,j)} \le \tau_{(1,1)}$ also yields, using (14),

$$\mathbb{E}_{\boldsymbol{\mu}}[\tau_\delta(1,j)] \le \mathbb{E}_{\boldsymbol{\mu}}[\tau_\delta(1,1)\mathbb{1}_{\mathcal{E}}] + \frac{r_0\delta}{2},$$

and from the above, it follows that

$$\limsup_{\delta\to 0}\frac{\mathbb{E}_{\boldsymbol{\mu}}[\tau_\delta(1,j)]}{\log(1/\delta)} \le \frac{1}{I(\mu_{2,1},\mu_{1,1})},$$

which concludes the proof.