URSI AP-RASC 2019, New Delhi, India; 09 - 15 March 2019

"Let Them Choose Their Strategies!" Expert Aggregation and Multi-Players Multi-Armed Bandits for Internet of Things Networks

Lilian Besson[(1)], Émilie Kaufmann[(2)] and Christophe Moy[(3)]
(1) CentraleSupélec, IETR–SCEE, Cesson-Sévigné, France, Lilian.Besson@CentraleSupelec.fr
(2) CNRS & CRIStAL, Univ. Lille, Inria SequeL, Lille, France, Emilie.Kaufmann@Univ-Lille1.fr
(3) Univ. Rennes, CNRS, IETR – UMR 6164, Rennes, France, Christophe.Moy@Univ-Rennes1.fr

Abstract

Decentralized access of several dynamic devices accessing a stationary wireless network can be modelled by the *multi-player multi-armed bandit* framework. It can model Internet of Things networks, and many different learning strategies have been proposed recently. Instead of choosing a specific algorithm offline, we propose to use an *online aggregation algorithm* to automatically let each object decide, on the fly, the best algorithm to use in a certain setting. Simulation results justify the interest of our proposal in different problems.

Keywords: Cognitive Radio, Reinforcement Learning, Multi-Player Bandits, Multi-Armed Bandits, Expert Aggregation.

1 Introduction

The model of Opportunistic Spectrum Access (OSA, [1]) for Cognitive Radio (CR, [2]), considers one Secondary User (SU) trying to use a licensed radio network, slotted both in time and frequency, and occupied by Primary Users (PU). The network usage from the PU determines the availability patterns of the radio channels, and the goal of the SU is to communicate as efficiently as possible, without interfering with the PU. Thus at each step, a SU first senses one channel, and only transmits if this channel is unoccupied by a PU. A common model in the literature is to describe the PU impact on the availability of the K channels in the following way: channels are independent and identically distributed (*i.i.d.*), and their qualities follow parametric distributions, *e.g.*, Bernoulli of means $\mu_1, \dots, \mu_K \in [0,1]$ for availabilities when dealing with binary sensing feedback. The SU has to select the best expected channel each time to maximize its throughput, or if successful communications are seen as rewards, the SU has to maximize its cumulative rewards, as in the Multi-Armed Bandit (MAB) problem [3].

MAB learning algorithms are known to be useful for the OSA setting [1], and UCB algorithms and other variants (e.g., kl-UCB or Bayes-UCB, [4, 3]) have been successfully applied to both numerically and physically simulated CR problems [5]. The performance of such learning algorithm $\mathscr{A}$ can be measured by different criteria. It is common in the bandit literature to study the regret [3], $\widetilde{R_T^{\mathscr{A}}} = \mu^* T - \sum_{t=1}^{T} r(t)$ which compares the reward in rewards between the algorithm $\mathscr{A}$ and the full-knowledge strategy which always picks the best arm, *i.e.*, the most available of mean μ^*. Good algorithms are expected to have slow-growing expected regret, but other criterion include the best arm pull frequency, or when applied to the CR problem, the throughput of the SU.

Many different learning algorithms have been proposed by the machine learning community [3], and most of them depend on several parameters, for instance $\alpha > 0$ for UCB, the prior for Thompson sampling, the kl function for kl-UCB etc. Every time a new MAB algorithm $\mathscr{A}$ is introduced, it is compared and benchmarked on some bandit instances, parameterized by $\boldsymbol{\mu} = (\mu_1, \dots, \mu_K)$, usually by focusing on its expected regret $R_T = \mathbb{E}_{\boldsymbol{\mu}}[\widetilde{R_T}]$. For a known and specific instance, simulations help to select the best algorithm in a pool of algorithms. But when one wants to tackle an *unknown* real-world problem, one expects to be efficient against *any* problem, of any kind, size and complexity. Ideally one would like to use an algorithm that can be applied identically against any problem, or at least any problem within a certain class. To choose the best algorithm, two approaches can be followed: either extensive benchmarks are done beforehand – if this is possible – to select the algorithm and its optimal parameters, or an adaptive algorithm is used to learn *on the fly* its parameters.

For the development of CR, a crucial step is actually to insert *multiple* $M \geq 2$ smart devices in the *same* background traffic. With the presence of a central controller that can assign the devices to separate channels, this amounts to choosing at each time step *several* arms of a MAB in order to maximize the global rewards, and can thus be viewed as an application of the multiple-play bandit. Due to the communication cost implied by a central controller, a more relevant model is the *decentralized multi-player* multi-armed bandit model [6], in which devices select arms individually and radio collisions may occur, which yield a zero reward. The goal for every player is to select one of the M best arms, as much as possible, without colliding too often with other devices. A first difficulty relies in the well-known trade-off between *exploration* and *exploitation*: devices need to explore all arms to estimate their means while trying to focus on the best arms to gain as much rewards as possible. The decentralized setting considers no exchange

of information between devices, that only know K and M, and to avoid collisions, devices should furthermore find orthogonal configurations (*i.e.*, the M devices use the M best arms without any collision), without communicating. Hence, in that case the trade-off is to be found between exploration, exploitation *and* low collisions.

Combining two of our recent works, we propose to use our aggregation algorithm **Aggregator** from [7], in combination with different multi-player MAB algorithms from [6] and [8]. We present below our mathematical model, then our contribution is stated as a three-part proposal, and the iconic UCB algorithm is presented. Simulation are presented for the two cases of $M < K$ and $M = K$ devices.

2 Multi-Player Stationary Bandit Model

We consider $K \geq 2$ radio channels, also called *arms*, of different characteristics, unknown to $M \geq 2$ identical objects that have to communicate to a fixed gateway, in a decentralized and autonomous manner. Following the classical OSA model [1], the radio protocol is slotted in both time and frequency. At each time step $t \in \mathbb{N}^*$, each device *tries to* communicate in a channel $A(t) \in \{1, \dots, K\}$. The device is a SU: it first senses (only) one channel k at a time, and can use it to communicate *only* if it was sensed free from any PU (*i.e.*, PU have full priority over the SU).

We choose to restrict to a *stochastic* model: after choosing the arm k, it is assumed that the sensing provides a *reward* $r_k(t)$, randomly drawn from a certain distribution depending on the arm index. Rewards are assumed to be bounded in $[0,1]$, and generally they follow one-parameter exponential families. We restrict to Bernoulli distributions, for sake of simplicity, meaning that arm k has a parameter $\mu_k \in [0,1]$. Rewards are drawn from $B(\mu_k)$, $r_k(t) \sim B(\mu_k)$, which can be simply interpreted by the SU: it is 1 if the channel k is not used by any PU during the time slot t, and is 0 otherwise.

The multi-player MAB setting considers $M \leq K$ devices [8], that have to make decisions at some pre-specified time instants. At time step $t \in \mathbb{N}^*$, device j selects an arm $A^j(t)$, independently from the other devices' selections. A *collision* occurs at time t if at least two devices choose the same arm. A collision occurs at time t for device j if $C^j(t) := \{\exists j' \neq j : A^{j'}(t) = A^j(t)\}$. Each device j then receives (and observes) the *binary rewards* $r^j(t) := Y_{A^j(t),t} \ \mathbb{1}(\overline{C^j(t)}) \in \{0,1\}$. In words, it receives the reward of the selected arm if it is the only one to select this arm, and a reward zero otherwise. Other models for rewards reward have been proposed, but we focus on full reward occlusion. This common setup is relevant to model the OSA problem: the device first checks for the presence of primary users in the chosen channel. If this channel is free ($Y_{A^j(t),t} = 1$), the transmission is successful ($r^j(t) = 1$) if no collision occurs with other smart devices ($\overline{C^j(t)}$).

A multi-player MAB strategy is a tuple $\rho = (\rho^1, \dots, \rho^M)$ of arm selection strategies for each of the M devices, and the goal is to propose a strategy that maximizes the total reward of the system, under some constraints. First, each device j should adopt a sequential strategy ρ^j, that decides which arm to select at time t based on previous observations.

The performance of a multi-player strategy is measured using the mean *regret*, *i.e.*, the performance gap with respect to the best possible strategy. The regret of strategy ρ at horizon T is the difference between the cumulated reward of an oracle strategy, assigning in this case the M devices to the M best channels, and the cumulated reward of strategy ρ. Denote the best mean by μ_1^*, the second best μ_2^* etc. The regret is then defined as

$$R_T(\boldsymbol{\mu}, M, \rho) := \left(\sum_{k=1}^{M} \mu_k^* \right) T - \mathbb{E}_\mu \left[\sum_{t=1}^{T} \sum_{j=1}^{M} r^j(t) \right]. \quad (1)$$

Maximizing the expected sum of the global reward of the system is equivalent to minimizing the regret, and the simulation results below present regret plots of various decentralized multi-player algorithms. It is known that any policy in this setting cannot beat the (asymptotic) lower-bound $R_T = \Omega(\log(T))$ [6]. The algorithms we compare typically achieve the lower-bound up-to constant factors: they have logarithmic regret profiles as seen in the simulation plots, and **MCTopM**-kl-UCB actually achieves the best known regret upper-bound [8], in the form of $R_T = \mathcal{O}(\log(T))$ with a constant proportional to KM^2 and depending on the difficulty of the problem (*i.e.*, of the gaps between successive means $\mu_i^* - \mu_{i+1}^*$).

3 Three Blocks for our Proposal

Our proposal consist in using: 1) classical single-player index policies for channel selection, 2) collision avoidance strategies for the multi-player aspect, 3) and an aggregation algorithm that each device implements, to select on the fly either the best index-policy or the best collision-avoidance (or possibly both, but we do not include more simulations due to space constraints).

3.1 Single-Player Index Policies

Upper Confidence Bounds algorithms [3] use a *confidence interval* on the unknown mean μ_k of each arm, which can be viewed as adding a "bonus" exploration to the empirical mean. They follow the "*optimism-in-face-of-uncertainty*" principle: at each step, they play according to the best model, as the statistically best possible arm (*i.e.*, the highest UCB) is selected. UCB is called an *index policy*. In our model, every dynamic device implements its own UCB algorithm, *independently*. Algorithm 1 presents the pseudo-code of UCB.

More formally, for one device, let $N_k(t)$ be the number of times channel k was selected up-to time $t \geq 1$, $N_k(t) = \sum_{\tau=1}^{t} \mathbb{1}(A(\tau) = k)$. The empirical mean estimator $\widehat{\mu}_k(t)$ of channel k is defined as the mean reward obtained by selecting it up to time t, $\widehat{\mu}_k(t) = 1/N_k(t) \sum_{\tau=1}^{t} r_k(\tau) \mathbb{1}(A(\tau) = k)$.

For UCB, the *confidence* term is given by [9] $B_k(t) = \sqrt{\log(t)/(2N_k(t))}$, giving the upper confidence bound $U_k(t) = \widehat{\mu}_k(t) + B_k(t)$, which is used by the device to decide the channel for communicating at time step $t+1$: $A(t+1) = \arg\max_{1\leq k\leq K} U_k(t)$.

for $t = 1, \dots, T$ **do** `// At every time step`
  Compute $U_k(t) = \widehat{\mu_k}(t) + B_k(t)$;
  Transmit in channel $A(t) \sim \arg\max_k U_k(t)$;
  Observe reward $r_{A(t)}(t) \in \{0, 1\}$;
  Update internal data $N_k(t)$, $\widehat{\mu_k}(t)$, $B_k(t)$.
end

Algorithm 1: A base building block, the UCB algorithm.

Other algorithms The kl-UCB algorithm is similar, but instead it uses a Kullback-Leibler divergence function to compute a statistically better UCB [4]. The Thompson sampling (TS) [10] algorithm is Bayesian: it maintains a posterior distribution on each means (*e.g.*, Beta posteriors for Bernoulli arms), updated after each observation, and chooses an arm by sampling a random mean from each posterior and playing the arm with highest mean. Both UCB, kl-UCB and TS have been proved to be efficient for stationary bandit problems, and were used in the experiments in both of previous papers [7, 8].

3.2 Collision Avoidance Strategies

Each object uses an index policy on its own, using the sensing information to estimate the quality of each channel, and use a collision avoidance strategy to deal with other devices. We consider three strategies, ρ^{Rand} from [6], and **RandTopM** and **MCTopM** from our recent work [8]. For one object, the ρ^{Rand} strategy consists in using a dynamic *rank* in $u \in \{1, \dots, M\}$ and selecting the u-th best arm (as given by the index policy) instead of the best one. When facing a collision, a new uniform rank is selected.

Our algorithms work similarly but they are more direct: the object still plays always in a set of its estimate of the M best arms, but instead of relying on a rank to select its arm it simply plays one of the M best arms. As long as the chosen channel is still one of the best one, and no collision is observed, the device keeps using it. **MCTopM** is a modification of **RandTopM**: after a good transmission, a device can fix itself on a channel and ignore further collisions as long as its channel is still estimated as one of the best M channels. **MCTopM** performs theoretically better and usually empirically outperforms the two others.

3.3 Online Algorithms Selection

We assume to have $N \geq 2$ MAB algorithms, $\mathcal{A}_1, \dots, \mathcal{A}_N$, and let $\mathcal{A}_{\text{aggr}}$ be an aggregation algorithm, which runs the N algorithms in parallel (with the same slotted time), and use them to choose its channels based on a voting from their N decisions. $\mathcal{A}_{\text{aggr}}$ depends on a pool of algorithms and a set of parameters. A good aggregation algorithm $\mathcal{A}_{\text{aggr}}$ performs almost as well as the best of the $\mathcal{A}_a$, with a good choice of its parameters, independently of the MAB problem, and $\mathcal{A}_{\text{aggr}}$ performs similarly to the best of the $\mathcal{A}_a$. The aggregation algorithm maintains a probability distribution π^t on the N algorithms $\mathcal{A}_a$, starting from a uniform distribution: π^t_a is the probability of trusting the decision made by algorithm $\mathcal{A}_a$ at time t. $\mathcal{A}_{\text{aggr}}$ then simply performs a weighted vote on its algorithms: it decides whom to trust by sampling $a \in \{1, \dots, N\}$ from π^t, then follows $\mathcal{A}_a$'s decision.

Our proposal is called **Aggregator**, and it is detailed in Algorithm 1 in [7]. It is easy to implement, it requires no parameter to tune and it was showed to out-perform all the others state-of-the-art expert aggregation algorithm for stationary problems.

4 Experiments on Simulated Problems

We focus on an example of a *i.i.d.* MAB problems, with $K = 9$ channels divided in three groups: 2 very bad arms ($\mu = 0.01, 0.02$), 4 average arms ($\mu = 0.3$ to 0.6) and 3 very good arms ($\mu = 0.78, 0.8, 0.82$). Horizon is $T = 10000$ (but it is unknown by the objects), and simulations are repeated 1000 times, to estimate the expected regret, R_T is plotted below, as a function of T. We compare the performance of $M = 3$ and then $M = 9$ objects (resp. in Figures 1 and 2), using one index policy along with one collision-avoidance protocol. We also simulate objects that use aggregation with **Aggregator** applied on one of the two levels. As expected, the objects using **Aggregator** perform worst than the best combination of algorithms, but better than some combination, in the two cases. Without knowing before-hand which combination is the best, the aggregation approach appears as a robust solution.

We present in Figure 3 a different problem with $\boldsymbol{\mu} =$ and $M = 6$ devices. The conclusions are similar.

5 Conclusion

Aggregation algorithms can be useful in the framework of multi-player MAB applied for decentralized Cognitive Radio models. For the real-world application of our model, tuning parameters before-hand is no longer possible when facing unknown problem instances, and thus an adaptive algorithm is preferable. Our algorithm **Aggregator** was proved to be useful to let each object decide automatically its preferred strategy.

Future works Our combined approach does not have theoretical guarantees yet, exploring the theoretical developments is left as a future work. We will also work on a hardware implementation of our proposal.

Acknowledgements This work is supported by the French National Research Agency (ANR) with the project BADASS (`N ANR-16-CE40-0002`), the French Ministry of Research (MENESR) and École Normale Supérieure de Paris-Saclay.

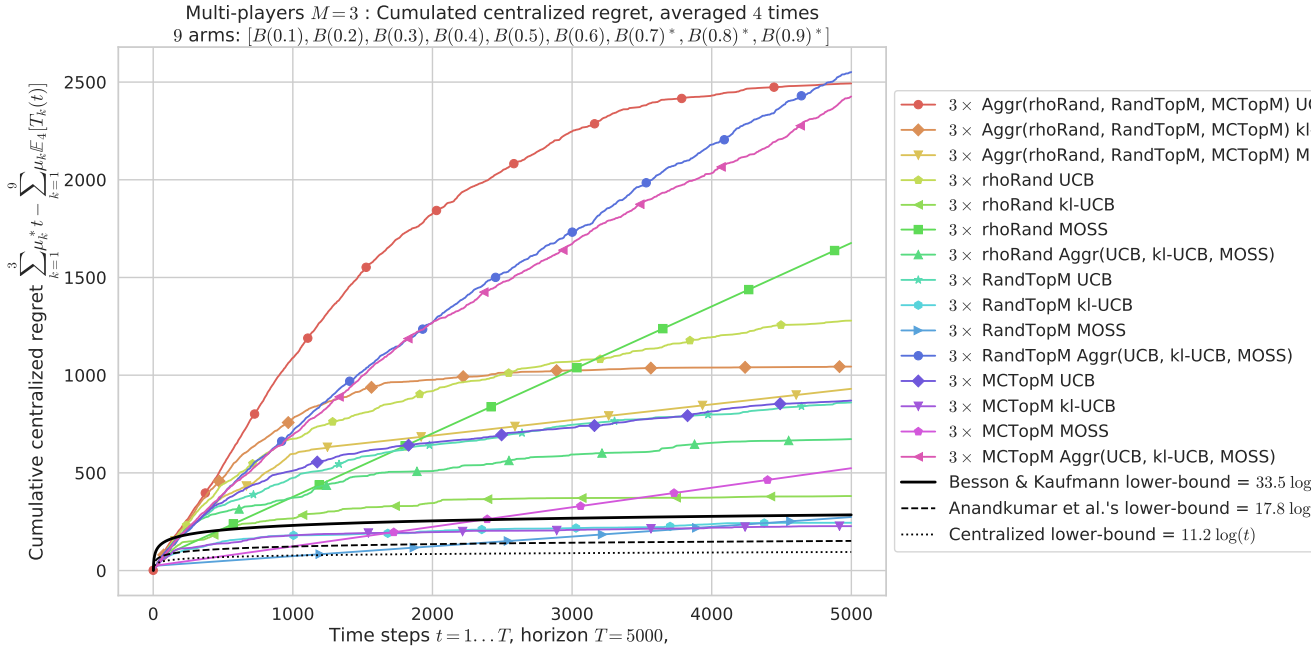

Figure 1. Comparing different strategies of $M = 3$ objects accessing $K = 9$ channels.

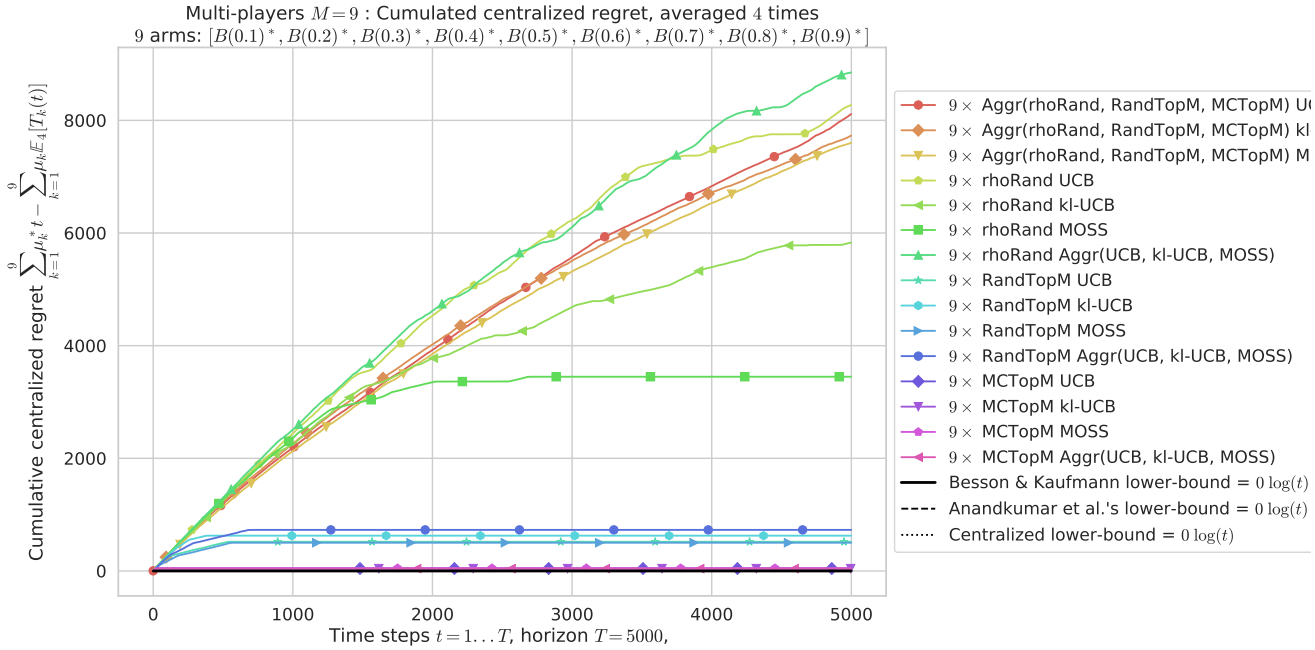

Figure 2. The simpler setting with $M = 9$ objects.

Open Source The code in Python 3 used for the simulations and the figures [11], is open-sourced at `https://GitHub.com/SMPyBandits/SMPyBandits` and documented at `https://SMPyBandits.GitHub.io`.

References

[1] Q. Zhao and B. M. Sadler, "A Survey of Dynamic Spectrum Access," *IEEE Signal Processing magazine*, vol. 24, no. 3, pp. 79–89, 2007.

[2] J. Mitola and G. Q. Maguire, "Cognitive Radio: making software radios more personal," *IEEE Personal*

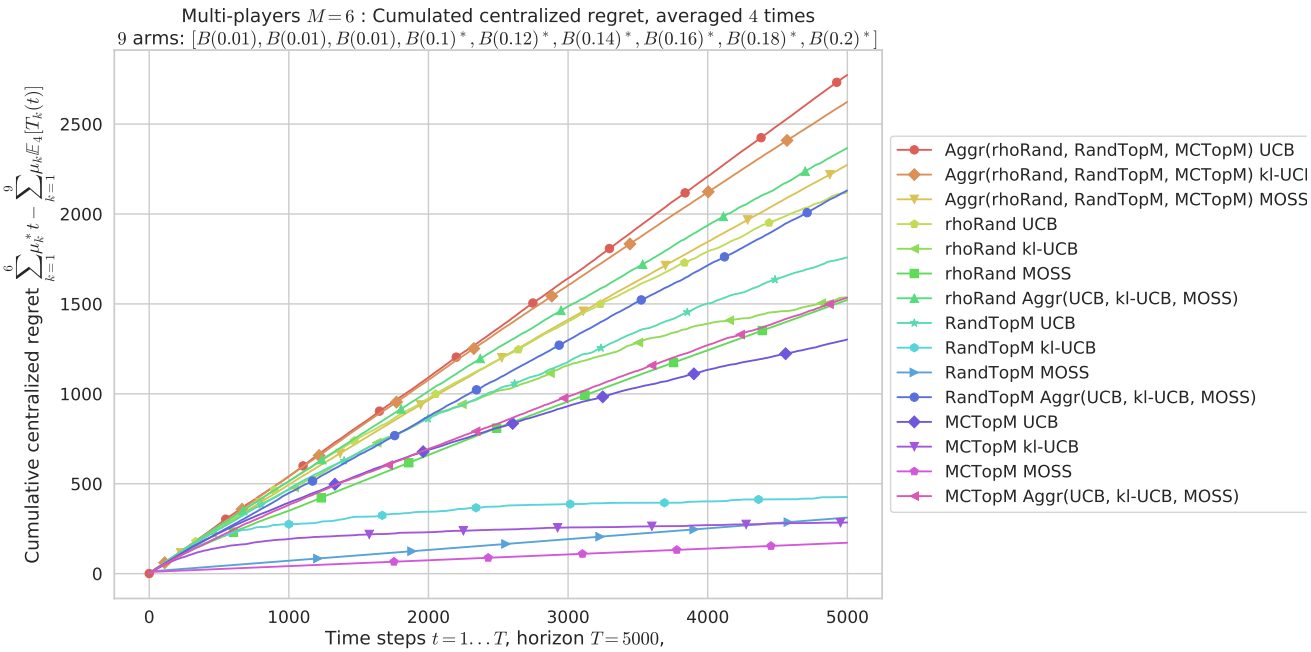

Figure 3. Another problem with $M = 6$ objects.

Communications, vol. 6, no. 4, pp. 13–18, 1999.

[3] S. Bubeck, N. Cesa-Bianchi, *et al.*, "Regret analysis of stochastic and nonstochastic multi-armed bandit problems," *Foundations and Trends® in Machine Learning*, vol. 5, no. 1, pp. 1–122, 2012.

[4] A. Garivier and O. Cappé, "The KL-UCB Algorithm for Bounded Stochastic Bandits and Beyond," in *COLT*, pp. 359–376, 2011.

[5] W. Jouini, D. Ernst, C. Moy, and J. Palicot, "Upper Confidence Bound based decision making strategies and Dynamic Spectrum Access," in *IEEE International Conference on Communications (ICC)*, pp. 1–5, IEEE, 2010.

[6] A. Anandkumar, N. Michael, and A. K. Tang, "Opportunistic Spectrum Access with multiple users: Learning under competition," in *IEEE INFOCOM*, 2010.

[7] L. Besson, E. Kaufmann, and C. Moy, "Aggregation of Multi-Armed Bandits Learning Algorithms for Opportunistic Spectrum Access," in *IEEE Wireless Communications and Networking Conference*, (Barcelona, Spain), 2018.

[8] L. Besson and E. Kaufmann, "Multi-Player Bandits Revisited," in *Algorithmic Learning Theory*, (Lanzarote, Spain), Mehryar Mohri and Karthik Sridharan, 2018.

[9] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time Analysis of the Multi-armed Bandit Problem," *Machine Learning*, vol. 47, no. 2, pp. 235–256, 2002.

[10] W. R. Thompson, "On the Likelihood that One Unknown Probability Exceeds Another in View of the Evidence of Two Samples," *Biometrika*, vol. 25, 1933.

[11] L. Besson, "SMPyBandits: an Experimental Framework for Single and Multi-Players Multi-Arms Bandits Algorithms in Python." Presentation paper, at hal.inria.fr/hal-01840022, 2018.