Multi-Player Bandits – a Musical Chairs Approach

Jonathan Rosenski JONATHAN.ROSENSKI@WEIZMANN.AC.IL
Weizmann Institute of Science, Rehovot 7610001, Israel

Ohad Shamir OHAD.SHAMIR@WEIZMANN.AC.IL
Weizmann Institute of Science, Rehovot 7610001, Israel

Liran Szlak LIRAN.SZLAK@WEIZMANN.AC.IL
Weizmann Institute of Science, Rehovot 7610001, Israel

Abstract

We consider a variant of the stochastic multi-armed bandit problem, where multiple players simultaneously choose from the same set of arms and may collide, receiving no reward. This setting has been motivated by problems arising in cognitive radio networks, and is especially challenging under the realistic assumption that communication between players is limited. We provide a communication-free algorithm (Musical Chairs) which attains constant regret with high probability, as well as a sublinear-regret, communication-free algorithm (Dynamic Musical Chairs) for the more difficult setting of players dynamically entering and leaving throughout the game. Moreover, both algorithms do not require prior knowledge of the number of players. To the best of our knowledge, these are the first communication-free algorithms with these types of formal guarantees.

1. Introduction

The stochastic multi-armed bandit (MAB) problem is a classic and well-studied setting of sequential decision-making, which exemplifies the dilemma of exploration vs. exploitation (see (Bubeck & Cesa-Bianchi, 2012) for a comprehensive review). In this problem, a player sequentially chooses from a set of actions, denoted as 'arms'. At every round, each arm produces a reward sampled from some unknown distribution in $[0, 1]$, and the player receives that reward, but does not observe the reward of other arms. The player's goal, of course, is to maximize the cumulative reward. The dilemma of exploration vs. exploitation here is that the more the player 'explores' by trying different arms, she will have a better understanding of each machine's expected reward. The more the player 'exploits' the machine which she thinks is best, the less rounds are wasted on exploring bad machines.

In this work, we study a variant of this problem, where there are many players who choose from the same set of arms. If two or more choose the same arm then there is a 'collision' and no reward is provided by that arm. Moreover, we assume that players may not communicate. The goal is to find a distributed algorithm for players that will maximize the sum of their rewards. One motivation for this setting (as discussed in the Related Work section below) comes from the field of cognitive radio networks, where several users utilize the same set of channels, in a situation where the quality of the different channels varies, and direct coordination between the players is not possible. We use the standard notion of (expected) regret to measure our performance, namely the difference between the expected cumulative reward of the arm with highest mean reward, and the expected cumulative rewards of the players.

We focus on a particularly challenging situation, where the players cannot communicate, there is no central control, and the players cannot even know how many other players are also participating. At every round each player decides which arm to sample. After the round is over the player receives the reward associated with the chosen arm, or an indication that the arm was chosen by at least one other player, in which case they receive no reward. The event that more than one player chooses the same arm will be referred to as a collision.

We will consider two variants in this work - a *static* setting in which all players start the game simultaneously and play for T rounds, and a *dynamic* setting, in which players may enter and exit throughout the game. Our main results are

Proceedings of the 33^{rd} International Conference on Machine Learning, New York, NY, USA, 2016. JMLR: W&CP volume 48.

the following:

- For the static case we propose and analyze the Musical Chairs (MC) algorithm, which achieves, with high probability and assuming a fixed gap between the mean rewards, a constant regret independent of T.
- For the dynamic setting we propose the Dynamic Musical Chairs (DMC) algorithm, which achieves an $\tilde{O}(\sqrt{xT})$ regret (with high probability and assuming a fixed gap between rewards), where x is a bound on the total number of players entering and leaving.
- We study the behavior of previous algorithms for this problem, and show that in the dynamic setting, there are some reasonable scenarios leading to their regret being linear in T. For other scenarios, we show that our regret guarantees improve on previous ones.
- We present several experiments which validate our theoretical findings.

All guarantees hold assuming all players implement the algorithm, but do not require any communication or coordination during the game.

Related Work

Most previous work on multi-player multi-armed bandits assumed that players can communicate, and included elements such as a negotiation phase or exact knowledge of the number of players, which remains fixed throughout the game, e.g. (Liu & Zhao, 2009; Anandkumar et al., 2011; Kalathil et al., 2014). However, in modeling problems such as cognitive radio networks, where players may be unable or unwilling to coordinate, these are not always realistic assumptions. For example, the algorithm proposed in (Liu & Zhao, 2009) relies on the players agreeing on a time division schedule for sharing the best arms, and requires all players to know the number of players, which needs to be fixed. (Anandkumar et al., 2011) provide an algorithm which is communication and cooperation free, but the performance guarantees of this algorithm are rather vague, and do not hold for the dynamic setting. Another approach which requires communication is the algorithm proposed in (Kalathil et al., 2014), called dUCB$_4$, in which players negotiate using Bertsekas' auction algorithm, in order to reach an agreement where each player will have a unique arm. The paper also proposes an algorithm for stochastic rewards changing according to a Markov process.

The work most similar to ours is (Avner & Mannor, 2014), where communication is not allowed and there is no knowledge of the number of players. The proposed algorithm, named MEGA, is based on an elegant combination of the well-known ϵ- greedy MAB algorithm, together with a collision avoidance mechanism inspired by the classical ALOHA protocol.

We discuss in detail the MEGA algorithm in section 4, and show that it may perform poorly in some reasonable dynamic scenarios. Essentially, this is because collision frequency decreases as the game proceeds, but never reaches zero. Although the frequency can be tuned based on the algorithm's parameters, it is difficult to find a single combination of parameters that will work well in all scenarios.

2. Setting

In the standard (single-player) stochastic MAB setting there are K arms, with the rewards of each arm $i \in [K]$ sampled independently from some distribution on $[0,1]$, with expected reward μ^i. Every round, a player chooses an arm and would like to receive the highest cumulative reward possibly in T rounds overall. In this work, we focus for simplicity on the finite-horizon case, where T is fixed and known in advance.

The multi-player MAB setting is similar, but with several players instead of a single one. In fact, we consider two cases: one where the set of players, and therefore number of players N, is fixed and another where the number of players, N_t, can change at any round t. In our model we would like to minimize, or even eliminate, any central control and communication, and assume that players do not even possess knowledge of the value of N_t. Generally, we assume K, N and N_t are all much smaller than T. We will denote by "the top N arms" the set of N arms with the highest expected rewards.

The performance in the standard single-player MAB setting is usually measured by how small is the regret (where we take expectations over the rewards of the arms):

$$R := T \cdot \mu^* - \sum_{t=1}^{T} \mu\left(t\right)$$

where $\mu\left(t\right)$ is the expected reward of the arm chosen by the single player at round i, and $\mu^* = \max_i \mu^i$ is the expected reward of the arm with the highest expected reward. The regret is non-trivial if it is sub-linear in T.

In the multi-player setting, we generalize this notion, and define our regret with respect to the best static allocation of players to arms (in expectation over the rewards), as follows:

$$R := \sum_{t=1}^{T} \sum_{k \in K_t^*} \mu^k - \sum_{t=1}^{T} \sum_{j=1}^{N} \mu_j\left(t\right) \cdot \left(1 - \eta_j\left(t\right)\right)$$

where $\mu_j\left(t\right)$ is the expected reward of the arm chosen by player j at round t, N_t is the number of players at round t,

K_t^* is the set of the highest N_t ranked arms where the rank is taken over the expected rewards, and $\eta_j(t)$ is a *collision* indicator, which equals 1 if player j had a collision at round t, and 0 otherwise. We define a collision as the event where more than one player chose the same arm at a given round, and assume that no reward is obtained in that case.

Since achieving sublinear regret is trivially impossible when there are more players than arms, we assume throughout that the number of players is always less than the number of arms.

3. Algorithms and Analysis

3.1. The Musical Chairs (MC) Algorithm

We begin by considering the static case, where no players enter or leave. The MC algorithm, that we present below for this setting, is based on the idea that after a finite time of random exploration, all players learned a correct ranking of all the arms with high probability (assuming gaps between the mean rewards). If after this time all players could fix on one of the top N arms and never leave, then from this point onward, there would be no regret accumulating. The algorithm we present is composed of a learning phase, with enough rounds of random exploration for all players to learn the ranking of the arms and the number of players; a 'Musical Chairs' phase, in which the N players fix on the top N arms; and a 'fixed' phase where all players remain fixed on their arm.

Algorithm 1 MC

Input: Parameters T_0, T_1
$C_{T_0} = 0, \tilde{\mu}_i(t) = 0, o_i = 0, s_i(0) = 0 \; \forall i \in 1, ..., K$
for t = 0 to T_0 **do**
  sample arm $i \sim U(1, ..., K)$
  receive $\eta(t)$ and $r(t)$
  if $\eta(t) \neq 1$ **then**
    update $o_i = o_i + 1$
    $s_i(t) = s_i(t-1) + r(t)$
  else
    $C_{T_0} = C_{T_0} + 1$ // # collisions
  end if
end for
$\forall i \in 1, ..., K$ set $\tilde{\mu}_i = \frac{s_i(T_0)}{o_i}$ // Estimate of μ^i
Sort indices in $[K]$ according to empirical mean in an array. Call this array A.
$N^* := \min\left(\text{round}\left(\frac{\log\left(\frac{T_0 - C_{T_0}}{T_0}\right)}{\log\left(1-\frac{1}{K}\right)} + 1\right), K\right)$ and $N^* := K$ if $C_{T_0} = T_0$
$j =$ Musical Chairs(N^* , A)
Stay fixed on arm j for the remainder of the rounds (total rounds is T_1)

Algorithm 2 Musical Chairs

Input: Parameter N^*, sorted array of arms A
loop for $T_1 - T_0$ **iterations**
  sample $i \sim U(1, ..., N^*)$ and choose arm $A[i]$
  receive $\eta(t)$
  if $\eta(t) == 0$ **then**
    output $A[i]$ and $return$
  end if
end loop

The Musical Chairs subroutine works by having each player randomly choose an arm in the top N arms, until she chooses one without experiencing a collision. From that point onwards, she chooses only that arm. It can be shown that if all players implement this subroutine, then after a bounded number of rounds (in expectation), all players will fix on different arms, and there will be no more added regret. The Musical Chairs subroutine's success depends on each player being able to accurately estimate a correct ranking of the machines (the ranking needs to be accurate enough to distinguish the best N machines from the rest) and to estimate the correct value of N.

3.2. Analysis of the MC algorithm

Let N be the number of players and let $i_1, ..., i_N$ denote the best N ranked arms. For each player we denote by $\tilde{\mu_j}$ to be that player's measured empirical mean reward of arm j. We use the following definition from (Avner & Mannor, 2014):

Definition 1. An ϵ-correct ranking of K arms is a sorted list of empirical mean rewards of arms such that $\forall i, j : \tilde{\mu_i}$ is listed before $\tilde{\mu_j}$ if $\mu_i - \mu_j > \epsilon$

Theorem 1. *Let $\Delta > 0$ be the gap between the expected reward of the Nth best arm and the $N+1$ best arm. Then for all $\epsilon < \Delta$ and $\delta \in (0,1)$, with probability $\geq 1 - \delta$, the expected regret of N players using the MC algorithm with K arms for T rounds, with parameter T_0 set to* $T_0 = \left\lceil \max\left(\frac{16K}{\epsilon^2}\ln\left(\frac{4K^2}{\delta}\right), \frac{K^2\log\left(\frac{4}{\delta}\right)}{0.02}\right)\right\rceil$ *is at most* $T_0 \cdot N + 2 \cdot \exp(2) \cdot N^2$.

Note that the bound we give is in expectation over the rewards and the algorithm's randomness, conditioned on the event (occurring with probability at least $1 - \delta$, where δ is a user-defined parameter) that players learn an ϵ-correct ranking and estimate the true number of players. Also, we assume that a lower bound on the reward gap is known. These types of guarantees/assumptions have also been used in previous work, e.g. (Avner & Mannor, 2014).

The proof of theorem 1 is composed of three arguments whose main idea is presented below. Formal proof appears in appendix A.

We begin by showing that that with high probability, all players will learn an ϵ-correct ranking after a time period independent of T.

We then show how estimating the number of players also requires a number of rounds independent of T with high probability. Knowing the value of N exactly is required in order for the players to run the Musical Chairs subroutine and choose an arm from the top N arms. To estimate N, each player keeps track of the number of collisions till time t, denoted as C_t, and after T_0 rounds, computes the estimate $N^* = \min\left(\text{round}\left(\log\left(\frac{T_0 - C_{T_0}}{T_0}\right) / \log\left(1 - \frac{1}{K}\right) + 1\right), K\right)$, where round$(\cdot)$ rounds to the nearest integer. We take the minimum between the two terms in order to handle the low probability event where the first term in the min expression is larger than K.

Finally, given that players were able to learn an ϵ-correct ranking and the number of players, we can upper bound the expected time (and hence the regret) for all the players to fix on different arms, and show that this bound is a constant.

3.3. The Dynamic Musical Chairs (DMC) Algorithm

In this subsection we consider the case when players can enter and leave. For the dynamic setting we suggest an extension of the MC algorithm, which simply runs the algorithm in epochs and restarts at the end of each epoch (see pseudocode below). We call this algorithm the Dynamic MC (DMC) algorithm and it requires the use of a shared clock between all players, to synchronize the epochs. We note that having a shared clock is a mild assumption which has been used previously in several works (See for example (Avner & Mannor, 2015), (Shukla & Ravimohan, 2014), (Nieminen et al., 2009)). This clock means that at any round t, players know what is $t \mod T_1$, where T_1 (a parameter of the algorithm) is the epoch length. However, communication between players is still not allowed, and the shared clock is not used for resource allocation or synchronization between players regarding which arm to choose. The DMC algorithm requires knowledge of the time horizon T. This is an assumption that will be important to lift in future work, possibly using doubling tricks.

We emphasize that in the dynamic setting, some restriction on the frequency at which players enter or leave is necessary for any algorithm to obtain a sub-linear regret bound. This is because if players may enter or leave at every round, then it is possible that no player stays long enough to even learn the true ranking of any arm, in which case any algorithm will result in linear regret. For this reason, we assume that the overall number of players entering and leaving is sublinear in T. Moreover, since time periods are synchronized, we will allow ourselves to assume that players can only enter and leave after the learning period in each epoch. We note that according to our analysis, the proportion of rounds belonging to learning period is a vanishing portion of the total number of rounds T, and therefore this assumption is not overly restrictive. Moreover, under some conditions, this assumption can be weakened to cover only leaving players, without significantly changing our regret bounds[1].

Algorithm 3 Dynamic MC

Input: Parameters T_0, T_1, T
explore/learn until $t \mod T_1 = 0$
run in loop MC (T_0, T_1)

3.4. Analysis of DMC Algorithm

The main result here is the following theorem:

Theorem 2. *Let $N_m \leq K$ be an upper bound on the number of active players at any time point; $\Delta_{min} = \min_{i=1,\ldots,N_m} \mu^i - \mu^{i+1}$ the minimal gap between the best $N_m + 1$ arms, with a known lower bound $\epsilon > 0$; and x be an upper bound on the total number of players entering and leaving during T rounds. Then with arbitrarily high probability, the expected regret of the DMC algorithm (over the rewards), using $\Theta(\sqrt{xT})$ epochs with $\tilde{O}(1)$ learning rounds at the beginning of each epoch, is at most:*

$$\tilde{O}\left(\sqrt{xT}\right)$$

where the $\tilde{O}$ hides factors logarithmic in T, δ, and polynomial in Δ_{min}, K, N_m.

As in theorem 1, the bound is in expectation over rewards and the algorithm's randomness, conditioned on the high-probability event that in each epoch, the players learn the correct ranking and the number of players.

The bound in the theorem hides several factors to simplify the presentation. More specifically, the bound is based on the following lemma, and taking $T_1 = \left\lceil \sqrt{\frac{T \cdot (T_0 + 2 \cdot T_f)}{x}} \right\rceil$:

Lemma 1. *Let e be the total number of players entering, l the total number of players leaving, T_f is the expected time for any player to fix on an arm (at most $\exp(2) \cdot N_m$ by theorem 1), and Δ_{min} be a lower bound on $\min_i \mu^i - \mu^{i+1}$ where μ^i is the expected reward of the i^{th} best arm, for any $i \leq N_m$.*

Then $\forall \delta \in (0,1)$ and $\epsilon < \Delta_{\min}$, w.p. $\geq 1 - \delta$, the expected regret of the Dynamic MC algorithm played for T rounds,

[1] For example, if the entering players can refrain from picking arms during the learning phase, and accumulate regret. Since the total length of the learning phase is less than our regret bounds, this won't affect the bounds by more than a small constant.

with parameters: $T_0 = \left\lceil \max\left(\frac{16K}{\epsilon^2} \ln\left(\frac{4K^2}{\frac{\delta}{2T}} \right), \frac{K^2 \log\left(\frac{4T}{\delta}\right)}{0.02} \right) \right\rceil$ *and* T_1 *chosen such that* $T_1 > T_0$*, is at most* $\frac{T}{T_1} \cdot (N_m \cdot (T_0 + 2 \cdot T_f)) + e \cdot 2\,(T_1 - T_0) + l\,(T_1 - T_0)$.

Note that N_m is not known to the players, however, it is always possible to upper bound it since we are in the setting where the number of players does not exceed the number of arms, i.e., $N_m \leq K$ and thus we can calculate a sufficient time for learning, T_0, by replacing N_m by K.

The lemma is proven by using the proof of theorem 1 with the confidence parameter set to $\frac{\delta}{2 \cdot T}$, and taking the union bound over all epochs. This ensures that, with high probability, the players learn the true rankings and estimate the number of players correctly at each epoch. For this reason T_0 includes a $\log(T)$ factor as stated above. We then separately bound the regret arising from the learning phase and fixing on an arm, as well as regret due to entering and leaving players. The formal proof appears in appendix A.

4. Comparison to the MEGA Algorithm

As discussed in the introduction, the most relevant existing algorithm for our setting (at least, that we are aware of) is the MEGA algorithm presented in (Avner & Mannor, 2014). In terms of formal guarantees, the algorithm attains $O(T^{2/3})$ in the static setting. A full analysis of the algorithm in the dynamic setting is lacking, but it is shown that if a single player leaves at some time point, the system re-stabilizes at an optimal configuration, after essentially $O(T^{2/3})$ rounds. The algorithm is clever, based on well-established techniques, allows players to enter and leave at any round, and compared to our approach, is not based on repeatedly restarting the algorithm, which can be wasteful in practice (an issue we shall return to later on). On the flip side, our algorithms have fewer parameters, attain considerably better performance in the static setting, and can provably cope with the general dynamic setting. In this section, we show that this is not just a matter of analysis, and that the approach taken by the MEGA algorithm indeed has some deficiencies in the dynamic setting. We begin by outlining the MEGA algorithm at a level sufficient to understand our analysis, and then demonstrate how it may perform poorly in some natural dynamic scenarios.

4.1. Outline of the algorithm

The MEGA algorithm uses a well known ϵ-greedy MAB approach, augmented with a collision avoidance mechanism. Initially, players mostly explore arms in order to learn their ranking, and then gradually move to exploiting the best arms, while trying to avoid arms they have collided on. Specifically, each player has an exploration probability which scales like $O\left(\frac{1}{t}\right)$, where t is the current round. The exploration probability also depends on two input parameters, c and d, where d is a lower bound on the gap of the N^{th} and $N+1^{th}$ best arms. Each player has a persistence probability, p_t, whose initial value, p_0, is another input parameter. p_t is increased to $p_{t-1} \cdot \alpha + (1-\alpha)$ for every round in which the player picks the same arm consecutively, where α is another input parameter. Otherwise, if the player switches arms, p_t is set to p_0. In the case of a collision, the colliding players indefinitely flip a coin with their own respective probabilities, p_t, for deciding whether to persist on the arm on which they collided. In case a player does not persist after a collision, she marks this arm unavailable until a time point sampled uniformly at random from $\{t, ..., t + t^\beta\}$ where β is another input parameter of the algorithm.

Note that both our algorithm and the MEGA algorithm require a lower bound on the gap between the N^{th} best arm and the $N+1^{th}$ best arm.

One issue with the MEGA algorithm approach is that players never entirely stop colliding, even when $T \to \infty$. At least in the static case, it seems advantageous to fix player's choice after a while, hence avoiding all future collisions and additional regret. The motivation for the MC algorithm is to create a procedure which guarantees that once learning completes, all players will choose one of the N best arms for the rest of the game.

In the dynamic setting, however, the issue is quite the reverse: The ϵ-greedy mechanism, which the MEGA algorithm is based on, is not good at adapting to changing circumstances. In the next subsection, we illustrate two problematic ramifications: One is that players entering late in the game are not able to learn the ranking of the best arm, and the other is that when players leave, the best arms may stay vacant for a long period of time before being sampled by other players. A third issue is that if the reward distributions change over time, a rapidly decreasing exploration probability is problematic. The DMC algorithm can address this as it runs in epochs, hence any mistake in one epoch is undone in the next.

4.2. Problematic Scenarios for the MEGA algorithm

Below, we study the realistic situation where players both enter and leave, and demonstrate that the regret of the MEGA algorithm can be substantially worse than our regret guarantees (both in terms of regret guarantees as well as in terms of actual regret obtained), sometimes even linear in T. For the proofs of the theorems presented in this section we refer the reader to appendix A.

The first scenario we wish to discuss is the simple setting of two players and two arms, where the second player enters

at some round in the game and the first player then leaves at some later round. We will describe what will happen, intuitively, if the players are following the MEGA algorithm, with a formal theorem presented below. In the scenario we described, the first player will learn a correct ranking of the two arms with high probability, and will proceed to exploit the highest ranked arm, thus making his persistence probability very high and exploration probability very low. If a second player enters late in the game, then any attempt to sample the highest ranked arm will cause a collision in which the first player will stay, and the second player will fail to sample the arm, since the first players' persistence probability is so high and the new player's persistence is set to a lower p_0. This means that the second player will not be able to learn the true ranking of the two arms. Thus, if the first player leaves after a period of time, such that the second player is not likely to explore, then the second player will exploit the second ranked arm, causing linear regret. This scenario can be extended to multiple players and arms, by adding players one by one in time intervals that ensure that players who entered late will not succeed in learning the true ranking, due to the collision avoidance mechanism. The formal result regarding this scenario is:

Theorem 3. *Consider a multi-player MAB setting as described above, where the second player enters at round* $\lceil \frac{T}{2} \rceil$*, and the first player leaves at round* $\lceil \frac{T}{2} + f \cdot T \rceil$*) for some parameter* f*. Then for all choices of the MEGA algorithm parameters* c, d, β, p_0*, if* α *(the parameter that controls the collision avoidance mechanism) is chosen such that* $\alpha \leq 1 - \frac{4\log(4fT)}{T}$*, and* f *is chosen such that* $\frac{\frac{c \cdot K^2}{d^2 \cdot (K-1)}}{T} \leq f \leq \frac{d^2 \cdot (K-1)}{8 \cdot c \cdot K^2}$*, then:*

- *The expected regret of the MEGA algorithm is* $\Omega(T)$.
- *The conditional expected regret of the DMC algorithm (using* $\Theta(\sqrt{T})$ *epochs) is* $\tilde{O}\left(\sqrt{T}\right)$.

Notice that for the DMC algorithm we have an upper bound on the expected regret conditioned on the event that all players learn an ϵ-correct ranking, which happens with arbitrarily high probability.

In particular, if we choose f to be a constant in the required range, we get a scenario where the regret bounds above hold for any $\alpha \leq 1 - O(log(T)/T)$ (and any possible values of the other parameters of the MEGA algorithm). We note that when α is larger than $1 - O(log(T)/T)$, the persistence probability p will hardly deviate from p_0, which makes the persistence mechanism non-functional and can easily lead to large regret, even in the static setting.

We now turn to discuss a second reasonable scenario, in which players alternate between entering and exiting, at intervals of T^λ rounds. We will show in this scenario that

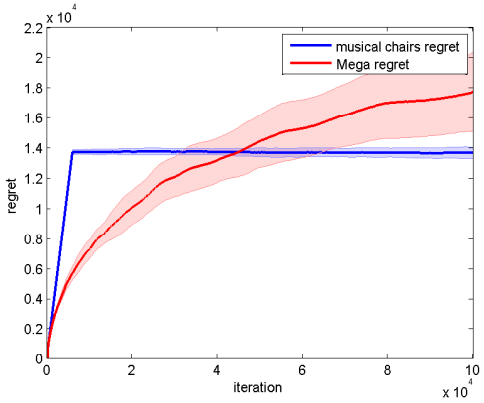


Figure 1. total accumulated regret after 50,000 iterations

however λ is chosen, the regret bound of the MEGA algorithm (as given in (Avner & Mannor, 2014), using recommended parameter values, and even just counting regret due to players leaving) is worse than the regret bound of the DMC algorithm (which incorporate regret due to both players leaving, entering, learning or fixing). Note that unlike Theorem 3, here we compare the available regret upper bounds, rather than proving a regret lower bound.

The setting is defined as follows: one player exits (or enters, alternating) every T^λ rounds, for some $\lambda < 1$. In the worst case, the player who left was occupying the highest ranked arm. In the analysis of (Avner & Mannor, 2014), players following the MEGA algorithm might take up to t^β rounds before being able to access this arm, due to the collision avoidance mechanism (where t is the round at which the player exited, and β is a parameter of the algorithm, whose recommended value based on the static setting analysis is $2/3$). For players following the DMC algorithm a player leaving can affect the regret of the current epoch only. Intuitively, if we set the epoch length compatible to the rate of exiting players, we can achieve a better regret bound than what is indicated by the MEGA algorithm analysis. Formally, we have the following:

Theorem 4. *In the multi-player MAB setting with one player leaving every* $(2 \cdot r) \cdot \lceil T^\lambda \rceil$ *rounds, and one player entering every* $(2 \cdot r + 1) \cdot \lceil T^\lambda \rceil$ *rounds, for* $r = 0, 1, 2, ...$*, and* $\lambda > \beta$*, we have that*

1. *The regret upper bound of the MEGA algorithm is at least* $O\left(T^{1-(\lambda-\beta)}\right)$
2. *The expected regret upper bound of the DMC algorithm is* $\tilde{O}\left(T^{1-\frac{\lambda}{2}}\right)$

As in the previous theorem, the expected regret of the DMC algorithm is conditioned on the event that all players learn an ϵ-correct ranking, which happens with arbitrarily high

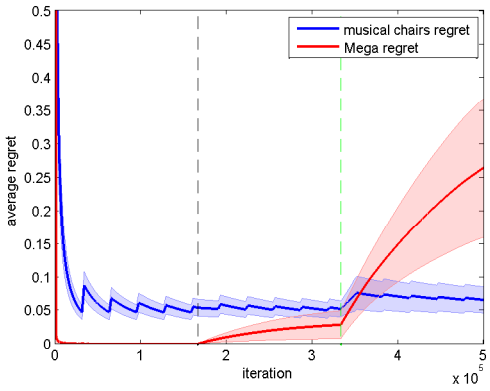


Figure 2. case 1; The black dashed lines represents players entering and the green dashed line shows when the first player exits.

probability. The assumption $\lambda > \beta$ is required in order to apply the existing MEGA analysis. When we pick the recommended value $\beta = \frac{2}{3}$, the exponent in the regret bound is uniformly superior for our algorithm. Note that if β is chosen differently then the regret bound of (Avner & Mannor, 2014) will increase, even in the static setting. A graphic illustration of the theorem appears in Figure 4 in appendix B.

5. Experiments

For our experiments, we implemented the DMC algorithm for the dynamic case and the MC algorithm for the static case. For comparison, we implemented the MEGA algorithm of (Avner & Mannor, 2014), which is the current state-of-the-art for our problem setting. Besides the experiments presented here, additional experiments and figures appear in appendix B.

For each experimental setup and algorithm, we repeated the experiment 20 times, and plotted the average and standard deviation of the resulting regret (the standard deviation is shown with a shaded region encompassing the average regret). In scenarios that are dynamic we mark the time that a player enters or leaves with a dashed line. In most figures, we plot the average per-round regret, as a function of the number of rounds so far.

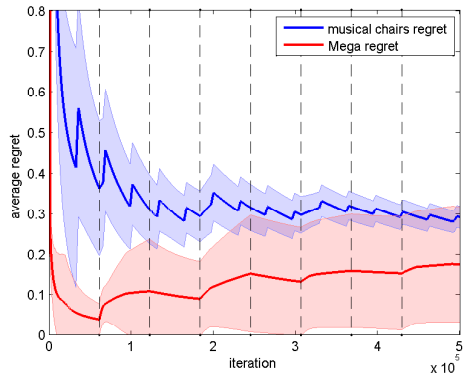
For the parameters of the MEGA algorithm, we used the empirical values suggested in (Avner & Mannor, 2014) (rather than the theoretical values which are overly conservative). The only exception is the gap between the mean rewards of the N^{th} and $N+1^{th}$ best arms, which was taken as the actual gap rather than a rough lower bound. Note that this only gives the MEGA algorithm more power. Moreover, in all experiments, the gap is at least 0.05, which is the heuristic value suggested to be used as the lower bound in (Avner & Mannor, 2014). For the dynamic scenarios, where the players and the number of players change, we use the minimum gap between the N^{th} and $N+1^{th}$ best arms over all rounds. For example, if at the beginning there are 2 players and the gap between the second and third arm is 0.3, but by the end there are 4 players and the gap between the fourth and fifth arm is 0.01, then we use 0.01 as the value of this gap.

For the MC and DMC algorithm, we set T_0 to be 3000 in all experiments. For the DMC parameter, T_1, we use either the theoretically optimal value presented in this work or that value scaled by a small constant (see details below for the specific value in each experiment).

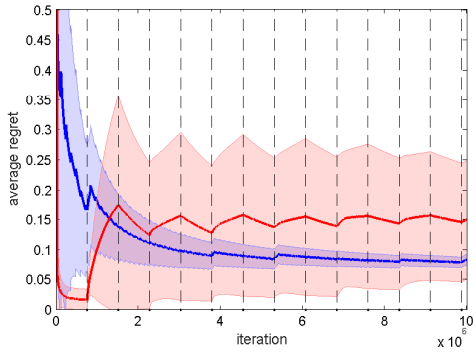
In the DMC algorithm, a potential source of waste is that newly entering players can accumulate linear regret until the next epoch begins. Therefore, in the DMC experiments, we added the following heuristic: When a player enters during the middle of the epoch, she chooses an arm with probability proportional to the empirical mean of its rewards (as observed by her so far, initially set to 1), multiplied by the empirical probability of not colliding on that arm (initially set to 1). After the epoch is over, she chooses arms by following the DMC algorithm. Intuitively this would quickly stop a large amount of collisions with players who are already fixed, and would encourage more players to exploit rather than only explore. This happens because any arm that has a player 'fixed' on it would always give newly entered players an empirical probability for colliding of 1.

We begin with a simple scenario corresponding to the static setting. There is an initial set of 6 players, which remains fixed throughout the game, and 10 arms. The mean rewards of the arms are chosen uniformly at random in $[0, 1]$ (with a gap of at least 0.05 between the N^{th} and $N+1^{th}$ arm). At every round of the game the rewards of each arm are chosen to be 1 with probability equal to the mean reward, and zero otherwise.

In this scenario we can see the short time period where players running the MC algorithm are learning, and then, with high probability, they all know which are the best N arms and never make any more mistakes or collisions. The added regret at every round after learning is zero, while in the MEGA algorithm, even though the exploration probability goes down with time, it is never zero. Also, in the MEGA algorithm every time a player has the best arm become 'available' that player will try to exploit it, probably colliding with other players who also want to exploit that arm. Therefore, in the MEGA algorithm there will always be collisions, even though they happen less frequently with time. This is further exemplified in figure 1 where we can see that after the learning stage there is no further accumulated regret for the MC algorithm while the MEGA algorithm never stops accumulating regret.



(a) case 2



(b) case 2 with 10 million iterations

Figure 3. Players alternate between entering and leaving (black dashed lines).

Another scenario we simulate is that of section 4 in theorem 3. The game starts with one player. At round $\lceil \frac{T}{3} \rceil$, a second player enters and after another $\lceil \frac{T}{3} \rceil$ rounds the first player leaves. There are 4 arms, with a lower bound on the gap of 0.8 between the expected reward of the N^{th} and $N+1^{th}$ best arm, T_1 set to $34,757$, and rewards are chosen deterministically. The results can be seen in figure 2.

As discussed in section 4, in the scenario of figure 2 a second player who enters late is not able to learn the best arm, because the first player always exploits the best arm and has a very high persistence probability. Therefore, once the first player leaves the game, and allows the best arm to be free for the second player to use, it will take the second player time proportional to the number of rounds she has played to explore this arm. Since her exploration probability will be very low, exploring this arm will take a very long time. The DMC algorithm runs in epochs and therefore a problem of inflexibility, or an inability to change for dynamic settings, does not arise. This phenomenon can be seen in figure 2: When the first player leaves (marked by a dashed green line) the average regret of the MEGA algorithm increases dramatically, while for the DMC algorithm the decreasing trend continues. This suggests that the MEGA algorithm may be susceptible to large regret in some natural dynamic scenarios.

Another dynamic player scenario we simulate (demonstrating theorem 4) is where the game starts with a set of five players and 10 arms and every $T^{0.84}$ rounds, we alternate between a player leaving and a player entering. The leaving player is chosen at random from the set of current players. Figure 3(a) shows the outcome for $T = 5 * 10^5$, and 3(b) shows the outcome for $T = 6 * 10^6$. In these scenarios T_1 is chosen to be $32,482$ in figure 3(a) and $119,921$ in figure 3(b). Although our algorithm performs better when T is large enough (confirming the theoretical evidence in theorem 2), we note that this is not the case for smaller values of T. We believe that this is due to the epoch-based nature of the DMC algorithm, which can be wasteful when T is moderate. However, when T is sufficiently large (as in figure 3(b)), the DMC algorithm outperforms MEGA.

6. Discussion

In this work we propose new algorithms for the stochastic multi-player multi-armed bandit problem, with no communication or central control. We provide an analysis for the static setting, showing that the proposed MC algorithm achieves a better upper bound on the regret (as a function of the number of rounds) than the current state of the art. We also provide the DMC algorithm, which is the first (to the best of our knowledge) with formal guarantees which copes with the general dynamic setting. We also study some natural dynamic scenarios, in which the behavior of previous approaches can be problematic.

This work leaves several questions open. For example, as noted earlier, both the DMC algorithm and the earlier MEGA algorithm require knowing a lower bound on the gap between the N-th and $N+1$-th best arm, and it would be interesting to remove this assumption while attaining similar guarantees. Another issue with the DMC algorithm is its epoch-based nature, which considerably degrades practical performance (especially if the total number of rounds T is not too large). Can we develop other algorithms with provable guarantees for the dynamic setting?

More generally, it would be quite interesting to develop algorithms in the adversarial setting, where the rewards are arbitrary. In the adversarial case, one cannot rely on high-reward arms to remain such in the future, and it is not clear at all what algorithmic mechanism can work here. Another interesting direction is to remove the assumption that players faithfully execute a given algorithm: In practice, players may be non-cooperative and greedy, and it would be interesting to devise algorithms which are also incentive-compatible, and study related game-theoretic questions.

Acknowledgments

This research is partially supported by Israel Science Foundation grant 425/13, and an FP7 Marie Curie CIG grant. We thank Nicolò Cesa-Bianchi and Yishay Mansour for several discussions which helped initiate this line of work.

References

Anandkumar, Animashree, Michael, Nithin, Tang, Ao Kevin, and Swami, Ananthram. Distributed algorithms for learning and cognitive medium access with logarithmic regret. *Selected Areas in Communications, IEEE Journal on*, 29(4):731–745, 2011.

Avner, Orly and Mannor, Shie. Concurrent bandits and cognitive radio networks. In *Machine Learning and Knowledge Discovery in Databases*, pp. 66–81. Springer, 2014.

Avner, Orly and Mannor, Shie. Learning to coordinate without communication in multi-user multi-armed bandit problems. *arXiv preprint arXiv:1504.08167*, 2015.

Bubeck, Sébastien and Cesa-Bianchi, Nicolò. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *arXiv preprint 1204.5721*, 2012.

Kalathil, Dileep, Nayyar, Naumaan, and Jain, Rahul. Decentralized learning for multiplayer multiarmed bandits. *Information Theory, IEEE Transactions on*, 60(4):2331–2345, 2014.

Liu, Keqin and Zhao, Qing. Distributed learning in multi-armed bandit with multiple players. *arXiv preprint arXiv:0910.2065*, 2009.

Nieminen, Jari, Jäntti, Riku, and Qian, Lijun. Time synchronization of cognitive radio networks. In *Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE*, pp. 1–6. IEEE, 2009.

Shukla, Annpurna and Ravimohan. Synchronization in cognitive radio systems: A survey. *International Journal of Advanced Research in Computer and Communication Engineering Vol. 3, Issue 7, July 2014*, 2014.

Sutton, Richard S and Barto, Andrew G. *Reinforcement learning: An introduction*. MIT press Cambridge, 1998.