JMLR: Workshop and Conference Proceedings 29:100–115, 2013 ACML 2013

Multi-armed Bandit Problem with Lock-up Periods

Junpei Komiyama JUNPEIKOMIYAMA@GMAIL.COM
Issei Sato SATO@R.DL.ITC.U-TOKYO.AC.JP
Hiroshi Nakagawa NAKAGAWA@DL.ITC.U-TOKYO.AC.JP
The University of Tokyo

Editor: Cheng Soon Ong and Tu Bao Ho

Abstract

We investigate a stochastic multi-armed bandit problem in which the forecaster's choice is restricted. In this problem, rounds are divided into lock-up periods and the forecaster must select the same arm throughout a period. While there has been much work on finding optimal algorithms for the stochastic multi-armed bandit problem, their use under restricted conditions is not obvious. We extend the application ranges of these algorithms by proposing their natural conversion from ones for the stochastic bandit problem (index-based algorithms and greedy algorithms) to ones for the multi-armed bandit problem with lock-up periods. We prove that the regret of the converted algorithms is $O(\log T + L_{max})$, where T is the total number of rounds and L_{max} is the maximum size of the lock-up periods. The regret is preferable, except for the case when the maximum size of the lock-up periods is large. For these cases, we propose a meta-algorithm that results in a smaller regret by using a empirical best arm for large periods. We empirically compare and discuss these algorithms.

Keywords: Multi-armed bandits, Online Learning, Stochastic Optimization

1. Introduction

A multi-armed bandit problem models many real-world sequential decision problems under uncertainty. In the problem, there are K bandit arms (options) from which to select. In each round, a forecaster pulls one of the arms and receives a reward. The reward distributions of the arms are initially unknown and the forecaster gradually acquires information through the game. The forecaster aims to maximize the sum of rewards, which is achieved by balancing *Exploration* and *Exploitation*. The performance of the algorithm is evaluated by a regret, or the difference between the total reward of the best arm and that of the policy. The multi-armed bandit problem has attracted attention of the Machine Learning community in particular and many extensions have been proposed, such as contextual recommendation (Langford and Zhang, 2007; Li et al., 2010), optimization (Dani et al., 2008), model selection (Maron and Moore, 1993; Mnih et al., 2008), and tree search (Kocsis and Szepesvári, 2006). Still, the base bandit problem itself is of great interest.

In studying the bandit problems, a forecaster has the freedom to select an arbitrary arm for each round. However, in real situations there are various restrictions for selecting arms. Many requirements, such as operation ease or resource constraints prevent the forecaster from free allocation. The examples below are typical scenarios.

Example 1 (A/B testings (Scott, 2010)) A/B testing is a well-known method when releasing new web page features. By comparing the user responses for multiple versions of web pages, administrators can estimate the effectiveness of the releases. There are many targets of A/B testing, e.g., ad placements, emails and top pages. Optimizing user attention is of great importance for most large-scale websites. However, there are many constraints preventing optimal allocation. For example, ad banners must be shown for a certain duration due to contracts with publishers.

Example 2 (Clinical trials (Gittins, 1989; Berry, 1978; Press, 2009)) Clinical trials are conducted in the final stages of drug development. The aim of such trials is to ensure the effectiveness and safety of newly developed drugs. There are many conditions necessary for this, e.g., amounts of drugs, placebo conditions, patients conditions. The trials are divided into many test phases. Between each test, the results of the previous test are reported. The next test is based on the information up to that and including that of the previous test. For the simplicity of operation, each test should be done with a single option. We would like to optimize the allocation even within these restrictions.

Essentially, these problems lie midway between sequential and batch problems. Forecasters are restricted to selecting the same option for certain rounds due to external constraints. Also, the sum of rewards is the quantity to optimize. To model these scenarios, we propose and study a multi-armed bandit problem with lock-up periods (lock-up bandit). The term "lock-up period" is a financial term meaning the predefined amount of time during which people concerned cannot sell shares. In the problem, we define the lock-up period as a set of successive rounds where the forecaster cannot change the arm to pull.

Structure of this paper: In Section 2 we formalize the proposed problem and discuss related works. In the following sections, we start from the stochastic multi-armed bandit (stochastic bandit) algorithms and prove they can keep small regrets in the existence of lock-up periods.

(1) The state-of-the-art algorithms for the stochastic bandit problem are not directly applicable to restricted environments. In Section 3, we discuss the natural conversion from the standard stochastic bandit algorithms to lock-up bandit ones. We prove the upper-bound regret of converted UCB, which is optimal up to a constant factor when the periods are small compared with the total number of rounds.

(2) The regrets of the converted algorithms are upper bounded by the size of the largest lock-up period. In some cases, there are large lock-up periods and in these cases the regret is linear to that much sizes. For such a case, we want to minimize the regret during these the large periods. In Section 4, we propose the balancing and recommendation (BaR) meta-algorithm, which effectively reduces the regret losses in large periods. The regret of this meta-algorithm is represented using the cumulative and simple regrets of the base algorithm.

(3) In Section 5, we discuss two sets of experiments we conducted. The first was the empirical relation between the period size and the regret. The second set of experiments involved the before-after analysis of the BaR meta-algorithm.

Finally, we conclude the paper in Section 6.

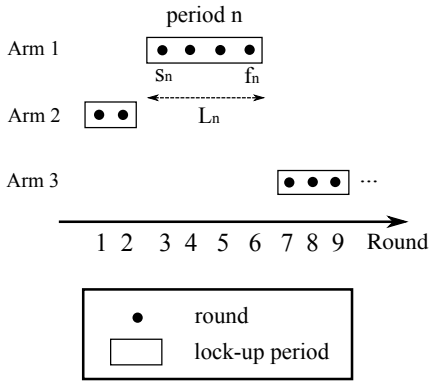


Figure 1: Lock-up bandit. Black dots represent rounds and rectangles represent lock-up periods.

2. Multi-armed bandit problem with lock-up periods

Our lock-up bandit is based on the stochastic bandit problem, in which the forecaster can select one arm for each round. However, in lock-up bandit, the rounds are divided into lock-up periods and the forecaster must select one arm for each lock-up period (Figure 1). The lock-up bandit problem is formally defined as follows. There are K arms associated with constant reward distributions $\{\nu_1, ..., \nu_K\}$. There are T rounds and rounds are divided into lock-up periods $L_1, ..., L_N$ where $\sum_{n=1}^{N} L_n = T$. We denote the start and end rounds of each periods as $(s_1, f_1), ..., (s_N, f_N)$. Note that $s_1 = 1, f_N = T$, $s_{n+1} - 1 = f_n$ and $L_n = f_n - s_n + 1$ hold for all periods $n \in [1, ..., N-1]$. Before the start of the first round, the forecaster is notified of K and $L_1, ..., L_N$. On each round $t = 1, ..., T$, if the round is the start of a period, the forecaster selects an arm. If not, he or she uses the same arm as the previous round. We denote the selected arm at round t as I_t. After selecting an arm, the forecaster receives the reward $X(t) \in [0, 1] \sim \nu_{I_t}$[1].

The goal of the forecaster is to minimize the (cumulative) regret

$$\mathbf{R}[T] = \mu_* T - \sum_{i=1}^{K} \mu_i T_i(T), \tag{1}$$

where $\mu_i = \mathbb{E}[\nu_i]$, μ_* is $\max_i \mu_i$, and $T_i(T)$ is the number of rounds arm i was selected in T rounds. We also use the gap $\Delta_i = \mu_* - \mu_i$ and the minimum nonzero gap $\Delta = \min_{i \in \{1,...,K\}, i \neq i^*} \Delta_i$. By the definition above, selecting suboptimal arm i increases the regret by Δ_i and that can be considered as a loss.

Remark 1 Multi-armed bandit with lock-up periods $L_1, ..., L_N, \sum_n L_n = T$ is more difficult than T-round stochastic bandit, where the forecaster can switch arms for every round. That is, a lower bound of the stochastic bandit problem also works as a lower bound of the lock-up bandit problem with the same number of rounds.

1. We assume the reward is in $[0, 1]$. Generalization to any finite support $[a, b]$ is easy.

2.1. Round-wise notation and period-wise notation

Throughout this paper, we use t as a variable representing a round and n as a variable representing a period. We use i as a variable representing an arm. For example, the number of rounds the arm i was selected in T rounds is denoted as follows.

$$T_i(T) = \sum_{t=1}^{T} \mathbb{I}_{I_t=i}, \tag{2}$$

where $\mathbb{I}_A = 1$ if A is true and otherwise $= 0$. As the forecaster must select one arm during a period, we can denote I_n to represent the arm selected in n. Also, the same quality T_i can be expressed as follows.

$$T_i(L_1, ..., L_N) = \sum_{n=1}^{N} L_n \mathbb{I}_{I_n=i}. \tag{3}$$

2.2. Period Ordering

In lock-up bandit, the order of periods matters. Remember the length of first period is denoted as L_1 and that of the second is L_2 , etc. For example, the lock-up bandit problem with $L_1, ..., L_9 = 1, L_{10} = 10$ is much easier than the one with $L_1 = 10, L_2, ..., L_{10} = 1$. This is because in the former problem the forecaster can select the arm at period 10 based on the reward information in periods $1, ..., 9$ while in the latter one there is no information at the first period and no way to avoid 10 round losses (at least probability $1/K$). On the other hand, the size of the lock-up periods is also of great interest. We use parentheses to denote size-sorted periods: "(1)" indicates $\arg\max_n L_n$ and "(2)" indicates the second largest, etc. $L_{(1)}$ is also denoted as L_{max}.

2.3. Related Works

Multi-armed bandit problems have been extensively studied in the area of Machine Learning and Operations Research due to their simplicity and wide applications. The stochastic multi-armed bandit problem, in which the rewards of arms are drawn from some distributions are assumed, has attracted the most attention. UCB (Auer et al., 2002) is an efficient index-based algorithm and is widely used.

Interesting problems that pose restrictions on forecasters' selection have been investigated. The bandit problem with switching costs is extensively studied. In this problem, the switching of arms generates a certain amount of loss, and the forecaster is motivated to stay with the current decision. For further details, see (Jun, 2004; Mahajan and Teneketzis; Guha and Munagala, 2009).

Committing bandit (Bui et al., 2011) is a two phases bandit problem. The forecaster can select the arms freely in the experimentation phase but must commit to a single arm in the commitment phase. Three settings were investigated in the paper for the length of the experimentation phase. For two of the three settings, the forecaster can extend the experimentation phase with a certain amount of cost, and the main result of the paper is the algorithms for finding the optimal time to end the experimentation phase. For the

third setting[2], the experimentation phase has a fixed length N_e, and they showed optimal algorithm up to a constant factor. This setting is equal to lock-up bandit with periods $L_1, ..., L_{N_e} = 1$ and $L_{N_e+1} = T - N_e + 1$. Our lock-up bandit and committing bandit differ in two respects. First, we assume the restrictions are tight, namely the forecaster cannot extend or shorten the lock-up periods. Second, we do not separate the experimentation phase and commitment phase. Our theory is applicable to any sizes lock-up periods restriction.

Lock-up bandit is also related to the best arm identification problem with fixed budget (Audibert et al., 2010). In the best arm identification problem, the task of the forecaster is to find the best arm among K arms. There is a fixed test period, and immediately after the end of the test period the forecaster outputs a "recommendation" arm he believes is the best. In the test period, the forecaster can select the arm for each round freely and receives the rewards. The test period has a fixed length d and the forecaster is evaluated based on the probability that the recommendation arm he selects corresponds to the real best arm. This setting is equal to the lock-up bandit with $L_1, ..., L_d = 1$ and $L_{d+1} \to \infty$, because when the last period is sufficiently large, the regret in the test period is negligible.

3. Conversion from standard stochastic algorithms

There have been many studies on the stochastic bandit and many algorithms have been proposed. Stochastic bandit algorithms are based on the fact that at every round the forecaster can select an arm. However, once the choice is restricted, it is not clear how to determine the next arms when possible. In this section, we discuss the simple conversion from stochastic bandit algorithms into lock-up bandit algorithms. We also show that the converted UCB's regret is $O(\log T + L_{max})$.

Proposition 2 (Conversion of the stochastic bandit algorithms into lock-up bandit algorithms)
We call a stochastic bandit algorithm $\mathcal{A}$. We define an algorithm $\mathcal{A}'$ for the lock-up bandit that uses $\mathcal{A}$ as an internal algorithm. If each round is the start of the lock-up period, $\mathcal{A}'$ invokes $\mathcal{A}$ and receives an arm. Then uses the arm as $\mathcal{A}$'s selection. When the round is not the start of the lock-up period, $\mathcal{A}'$ must select the same arm as the last round. In this case, invoke $\mathcal{A}$ and discard its selection. After receiving a reward, $\mathcal{A}'$ feeds $\mathcal{A}$ the selection and reward tuple $(I_t, X(t))$. $\mathcal{A}$ learns from the reward tuple as if it were selected by itself.

It is true that the conversion above is not promised to be applicable for all algorithms in stochastic bandit[3]. However, most algorithms, including index-based algorithms (UCB, UCB-Tuned (Auer et al., 2002), UCB-E (Audibert et al., 2010), UCB-V (Audibert et al., 2008), MOSS (Audibert and Bubeck, 2009), KL-UCB (Garivier and Cappé, 2011), etc.), and ϵ_n-greedy can be converted into lock-up bandit algorithms with the above procedure.

We denote converted algorithms using primes. For example, UCB and ϵ_n-greedy converted are UCB$'$ and ϵ_n-greedy$'$. Remember that our main concern is the regret in lock-up bandit.

2. The authors called it a "hard experimentation deadline setting."
3. For example, for algorithms that maintain lists and select the next arms from the lists, the conversion above is not directly applicable.

Theorem 3 (Regret upper bound of UCB′) *The regret of UCB′ in lock-up bandit is upper bounded as follows.*

$$\mathbb{E}[\mathbf{R}(L_1, ..., L_N)] \leq \sum_{i \neq i^*} \left\{ \frac{8 \log T}{\Delta_i} + L_{max} \Delta_i \left(1 + \frac{\pi^2}{3} \right) \right\}. \tag{4}$$

Proof sketch: The proof is the extension of (Auer et al., 2002) to the lock-up bandit. The base theorem relies on the fact that the probability of suboptimal arm i played after $T_i(t) \geq \lceil (8 \log T)/\Delta_i^2 \rceil$ is sufficiently low and its sum is loosely bounded by $\pi^2/3$. In lock-up bandit, there are two main changes,

(1) $(8 \log T)/\Delta_i^2$ is replaced with $(8 \log T)/\Delta_i^2 + (L_{max} - 1)$. The number of arms selected before $T_i(t) \geq (8 \log T)/\Delta_i^2$ is upper bounded by this quantity.

(2) $\pi^2/3$ is multiplied by L_{max}.

The full proof is presented in Appendix.

Theorem 3 indicates that the regret of UCB′ is bounded by $O(\log T + L_{max})$ for any list of lock-up periods $L_1, ..., L_N, \sum_n L_n = T$. When L_{max} is small compared with T, UCB′ achieves $O(\log T)$ regret. A logarithmic bound is optimal up to a constant factor in stochastic bandit. Since lock-up bandit is more difficult than stochastic bandit (c.f. Remark 1), the bound is optimal up to a constant factor. However, when there are some periods that are bigger than the order of $\log T$, the regret in the periods matters. In the next section, we propose a meta-algorithm to reduce the regrets in large periods.

4. How to reduce regrets in large periods

In this section, we propose BaR, a general meta-algorithm for reducing regrets in large periods.

4.1. Minimizing regret in large periods

In lock-up bandit, an algorithm cannot change the arm during a lock-up period. If an algorithm selects a suboptimal arm i at the start of a round n, the regret is increased by $\Delta_i L_n$. For this reason, we want to avoid choosing a suboptimal arm at the start of large periods. The notion of simple regret introduced by Bubeck et al. (2009) describes the minimum possible regret in a specific round. They proposed a pure exploration bandit problem. In this problem, for each round the algorithm selects an arm and receives a reward. After receiving the reward, the algorithm outputs an additional arm: the recommendation arm. At a certain round the game ends, and the algorithm is evaluated based on the quality of the recommendation arm. The goal with the algorithm is to minimize the simple regret, or the one-time regret of the recommendation arm. In summary, the pure exploration bandit problem is the same framework as the stochastic bandit problem except for the existence of the recommendation arm and the goal. In terms of the *Exploration* and *Exploitation* tradeoff, recommendation arm is an exploitation-only arm. The simple regret describes the best possible accuracy of the recommendation arm. In contrast with the simple regret, the sum of rewards during the game is called a cumulative regret, which is the quantity to

optimize in the stochastic bandit. We denote the cumulative regrets as $\mathbf{R}(T)$ and the simple regret as $\mathbf{r}(T)$ (in the period-wise notation, $\mathbf{R}(L_1, ..., L_N)$ and $\mathbf{r}(L_1, ..., L_N)$). Interestingly, there is a tradeoff between the two regrets.

Theorem 4 (Cumulative regret and simple regret tradeoff (Bubeck et al., 2009)) *For all allocation strategies ϕ and all function ξ, if there exists some constant C and the allocation policy satisfies*

$$\mathbb{E}[\mathbf{R}(T)] \leq C\xi(T), \tag{5}$$

for all Bernoulli reward distributions $\{\nu_1, ..., \nu_K\}$, then the simple regret of any recommendation strategies based on the allocation strategies ϕ has the following lower bound: For all sets of $K \geq 3$ distinct Bernoulli reward distributions, all different from Dirac distributions centered at 1, there exists a constant D and ordering $\{\nu_1, ..., \nu_K\}$ of the considered distributions with

$$\mathbb{E}[\mathbf{r}(T)] \geq \frac{\Delta}{2} \exp\left(-D\xi(T)\right). \tag{6}$$

An intuitive explanation of Theorem 4 is as follows: the minimum possible simple regret for a round is determined by the cumulative regret to that point. Bubeck et al. (2009) proposed three natural recommendation policies: Empirical Best Arm (EBA), Most Played Arm (MPA) and Empirical Distribution of Plays (EDP). We use EBA, which recommends the arm of the best empirical mean, throughout this paper.

4.2. BaR meta-algorithm

Algorithm 1 BaR meta-algorithm

Require: K arms, $L_1, ..., L_N$, N_r, and base algorithm $\mathcal{A}$

for $n \in 1, ..., N$ **do**
  if $n \in \{(1), ..., (N_r)\}$ **then**
    invokes $\mathcal{A}$ to query for recommendation arm ψ
    select arm $I_n = \psi$
    receive reward X until the period ends. The reward information is discarded.
  else
    invokes $\mathcal{A}$ to query for the arm selection ϕ
    select arm $I_n = \phi$
    receive reward X and feed $\mathcal{A}$ with the reward tuple (I_n, X) until the period ends.
  end if
end for

Good algorithms of the multi-armed bandit problem balance exploration and exploitation and result in $O(\log T)$ expected cumulative regret. If there are lock-up periods, this balance is perturbed by $O(L_{max})$. If the value of exploration becomes large (i.e., a suboptimal arm is chosen at the start of the largest period ($L_{max} \gg \log T$)), it is difficult to restore the optimal balance of exploration and exploitation. The main idea of the BaR meta-algorithm (Algorithm 1) is using the recommendation arms as its selection at large periods to avoid choosing suboptimal arms. This meta-algorithm uses a base lock-up bandit

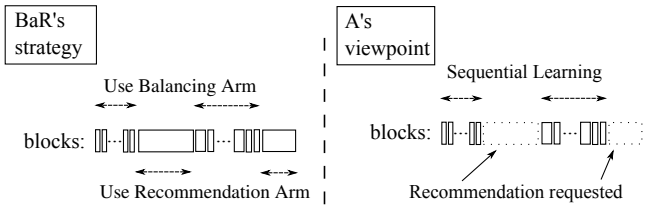


Figure 2: BaR meta-algorithm. Two large periods are assigned to the recommendation set.

algorithm, which we denote as $\mathcal{A}$. Before the start of BaR, we decide the recommendation set $\{(1), ..., (N_r)\}$, or the Top-N_r subset of the lock-up periods sorted by size. If each period is in the recommendation set, the algorithm queries $\mathcal{A}$ for the recommendation arm and uses it as the selection. $\mathcal{A}$ is not notified of the reward information. Otherwise, the algorithm works as a wrapper of $\mathcal{A}$. From the viewpoint of $\mathcal{A}$, it seems as if the periods in the recommendation set were banished (Figure 2). The regret of BaR can be derived from $\mathcal{A}$'s cumulative and simple regrets.

Remark 5 (Regret of [BaR, $\mathcal{A}$]) *If BaR is run with the recommendation set* $\{(1), ..., (N_r)\}$, *the regret is denoted by the base algorithm's cumulative and simple regret as,*

$$\begin{aligned}&\mathbf{R}(L_1, ..., L_N) = \mathbf{R}_{base}(L_1, ..., L_N \setminus L_{(1)}, ..., L_{(N_r)})\\ &+\sum_{n=1}^{N_r} L_{(n)}\mathbf{r}_{base}([L_{n'}|n' < (n), L_{n'} \notin \{(1), ..., (N_r)\}]),\end{aligned} \tag{7}$$

where, the first term of RHS is the cumulative regret of $\mathcal{A}$ run in the environment where $\{(1), ..., (N_r)\}$ are removed. Also, in the second term of RHS, $[L_{n'}|n' < (n), n' \notin \{(1), ..., (N_r)\}]$ means the list of periods before the period (n) and not in $\{(1), ..., (N_r)\}$. For example, suppose $N = 100$ and the recommendation set $\{(1), ..., (N_r)\}$ is $\{(1), (2)\} = \{50, 100\}$. The cumulative regret is defined as the regret of the base algorithm run at the lock-up periods $1, ..., 49, 51, ..., 99$. The sum of simple regret is 50's simple regret after periods $1, ..., 49$ and period 100's simple regret after periods $1, ..., 49, 51, ..., 99$. The BaR meta-algorithm decomposes the regrets into the cumulative and the simple. The cumulative regret is dependent upon the maximum size of the periods (Theorem 3). By removing large periods, we can reduce the maximum size of the periods. Also, the recommendation is the best method for selecting the optimal arm. Therefore, it can minimize the regret generated from the simple regret part.

Our next concern is how to estimate the cumulative and simple regrets of base algorithms. In Section 3, we defined the uniform upper bound of a cumulative regret of UCB$'$. However, we have not introduced any simple regret so far. In the next subsection, we describe UCB-E and discuss its regret.

4.3. UCB-E

UCB-E was introduced by Audibert et al. (2010) as an explorative algorithm for stochastic bandit. It uses $\sqrt{a/T_i(t)}$ as the confidence bound. In the fixed horizon bandit game (i.e. T is known), the algorithm is flexible. When we set $a = 2\log T$, we obtain exactly the same

the cumulative regret upper bound as UCB and can choose a large value to obtain a better simple regret bound. We convert UCB-E by using the Proposition procedure 2 to obtain UCB-E′.

Theorem 6 (Uniform cumulative regret upper bound of UCB-E′) *If UCB-E′ is run with parameter* $a \geq 2\log T$*, it satisfies*

$$\mathbb{E}[\mathbf{R}(L_1, ..., L_N)] \leq \sum_{i \neq i^*} \left\{ \frac{4a}{\Delta_i} + \Delta_i L_{max} \left(1 + \frac{\pi^2}{3}\right) \right\}. \tag{8}$$

The proof is very similar to Theorem 3. The proof is presented in Appendix.

Theorem 7 (Uniform simple regret upper bound of UCB-E′) *If UCB-E′ is run with parameter* $0 < a \leq \frac{25}{36}\frac{T - KL_{max}}{H_1}$ *then it satisfies*

$$\mathbb{E}[\mathbf{r}(L_1, ..., L_N)] \leq 2TK \exp\left(-\frac{2a}{25}\right), \tag{9}$$

where, $H_1 = \sum_{i \neq i^*} 1/\Delta_i^2 + 1/\Delta^2$.

Proof Sketch: The proof relies on the fact that the empirical mean never deviates from the thin confidence bound $1/5\sqrt{a/T_i(t)}$ with high-probability. It holds in all $a \leq 25(T - KL_{max})/(36H_1)$, even in the existence of lock-up periods. The full proof is presented in Appendix.

5. Experiments

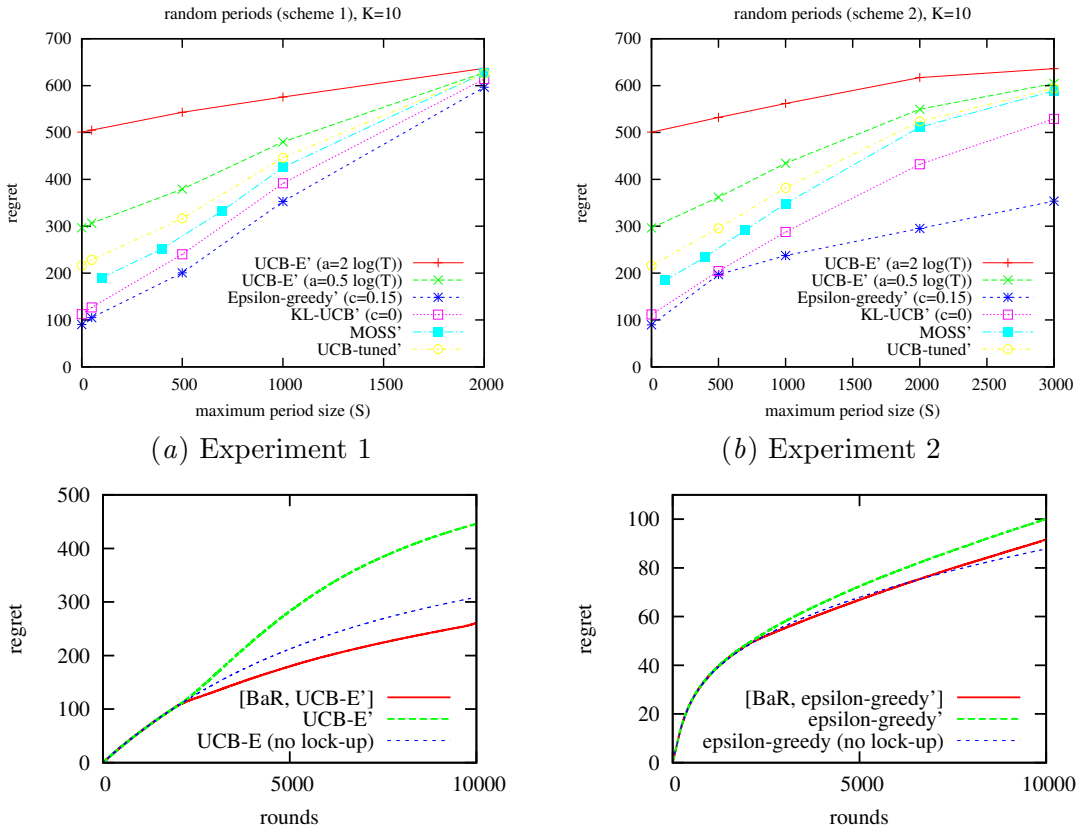
We conducted two sets of experiments to support the theoretical results in the previous two sections.

(1) In Section 3 we proposed a simple conversion of stochastic bandit algorithms to obtain lock-up bandit algorithms. The converted algorithms' regrets are linearly dependent upon the maximum period size. In the first set of experiments (Experiments 1 and 2), we studied the dependency between the maximum period size and a regret.

(2) In Section 4, we proposed the BaR meta-algorithm, which reduces the regret in large periods. In the second set of experiments (Experiments 3 and 4), we conducted a before/after analysis of BaR.

5.1. Experimental settings

All experiments involved ten-armed lock-up bandits with $T = 10000$. The rewards of arms were Bernoulli distributions with means
$(\mu_1, ..., \mu_{10}) = (0.1, 0.05, 0.05, 0.05, 0.02, 0.02, 0.02, 0.01, 0.01, 0.01)$.



(a) Experiment 1 (b) Experiment 2

(c) Experiment 3 (d) Experiment 4

Figure 3: Experimental results. Experiments 1 and 2 show regret as function of maximum period size. Experiments 3 and 4 show regret before/after application of BaR.

5.1.1. Settings of Experiments 1 and 2

The algorithms we used were UCB-E$'$ (Audibert et al., 2010) with parameter $a = 2\log T$, $a = 1/2\log T$, ϵ_n-greedy$'$ (Auer et al., 2002) with parameters $(c, d) = (0.15, 0.1)$, MOSS$'$ (Audibert and Bubeck, 2009), KL-UCB$'$ (Garivier and Cappé, 2011) with parameter $c = 0$, and UCB-Tuned$'$ [4] (Auer et al., 2002). We do not intend to argue which algorithm is better[5].

We showed the regret as a function of maximum period size S ([6]). In all experiments, for each value of S we show an averaged regret over 10,000 different runs. For each run, the lock-up periods in the experiments were randomly generated as follows. Until the total number of rounds reached T (i.e., $\sum_n L_n < T$), we appended a new period of size $\{1, ..., S\}$ with the same probability (Experiments 1) or the probability proportional to the inverse of size (Experiments 2). The last period was decreased to satisfy $\sum_n L_n = T$.

4. The setting of UCB-Tuned was the same as described in Section 4 of (Auer et al., 2002)
5. The parameters in ϵ_n-greedy$'$ were chosen to be empirically good (c.f. Section 4 in (Auer et al., 2002)). Therefore, it was no surprise ϵ_n-greedy$'$ performed better than UCB-E$'$.
6. S is the maximum period size to be possibly generated. L_{max}, the maximum period size to be actually generated, is smaller than or equal to S.

5.1.2. Settings of Experiments 3 and 4

In the second set of experiments (Experiments 3 and 4), we showed the regret as a function of rounds. The algorithms we used were UCB-E$'$ with parameter $a = 1/2 \log T$ (Experiment 3) and ϵ_n-greedy$'$ with parameters $(c, d) = (0.15, 0.1)$ (Experiment 4). In both experiments, the regrets were averaged over 10,000 different runs. In each run, the periods were generated as follows. For the first 2,000 rounds there were no lock-up periods (i.e., $L_1, ..., L_{2000} = 1$). From rounds 2,001 to 10,000, the periods were generated by a process similar to Experiments 2 and 4. Until the sum of the periods reached 10,000, we appended a new period of size $\{1, ..., 1000\}$ with the probability proportional to the inverse of size. We compared the base algorithm (UCB-E$'$ and ϵ_n-greedy$'$) before and after application of BaR. We also show the regret of the base algorithm run with no lock-up period (= standard stochastic bandit), which is much easier than lock-up bandit. As for the recommendation set, we used all periods larger than 400.

5.2. Experimental Results and Discussions

5.2.1. Results of Experiments 1 and 2

Figure 3 is the results of the experiments. In Experiments 1 and 2, we observed linear relation between the maximum period size and the regret for all algorithms. Note that, between Experiments 1 and 2, the number of large periods differed greatly. In Experiment 2, large periods had a small probability (inverse to its size) to be generated compared with Experiment 1; however, Experiments 1 and 2 look very much alike. This fact supports that the regret in lock-up bandit is dependent upon the size of the maximum periods.

5.2.2. Results of Experiments 3 and 4

Experiments 3 and 4 showed the effect of BaR. In both experiments, using BaR makes the regret significantly smaller. In Experiments 3, the results of [BaR, UCB-E$'$] were even better than those of the base algorithm in the standard bandit game. This is surprising because the bandit problem with lock-up periods is much more difficult than the standard bandit problem. This can be explained as follows. The regret of UCB-E$'$ is higher than that of ϵ_n-greedy$'$. This means that UCB-E$'$ does more exploration than it should and there is some room for exploitation.number of recommendation periods In the Experiments 4, the regret of [BaR, ϵ_n-greedy$'$] was higher than that of no lock-ups, which is natural. This results are not specific to UCB-E and ϵ_n-greedy. We also conducted experiments with many state-of-the-art algorithms (KL-UCB (Garivier and Cappé, 2011), Moss (Audibert and Bubeck, 2009) and UCB-Tuned (Auer et al., 2002)) and obtained similar results. They are not presented here due to space limitations.

5.2.3. Discussions

The use of the BaR meta-algorithm effectively reduces regret for the following reason. When T is large, the ratio of exploration to exploitation is small (i.e. $O(\log T/T) \to 0$). Therefore, if the forecaster does more exploration than it should do, restoring the optimal balance is virtually impossible. Conversely, if it does less exploration is smaller than it should, restoring the optimal balance is relatively easy. This is why BaR, which increases exploitation during the large lock-up periods, works well.

6. Conclusion and future works

We proposed and studied a bandit game with a lock-up period restriction, which is expected to model the practical scenarios that naturally arise when we apply stochastic bandit to real problems. We studied how the exploration and exploitation balance is perturbed by lock-up restrictions and proposed methods to recover the balance. For further understanding of related problems, better bounds for the simple regret is of great interest. Contrary to the cumulative regret, the simple regret is less known. In our theory, the simple regret is important and finer bound preferred.

References

J.-Y. Audibert and S. Bubeck. Minimax policies for adversarial and stochastic bandits. In *COLT*, 2009.

J.-Y. Audibert, S. Bubeck, and R. Munos. Best arm identification in multi-armed bandits. In *COLT*, 2010.

J.Y. Audibert, R. Munos, and Cs. Szepesvri. Exploration-exploitation trade-off using variance estimates in multi-armed bandits. *Theoretical Computer Science*, 2008.

Peter Auer, Nicoló Cesa-bianchi, and Paul Fischer. Finite-time Analysis of the Multiarmed Bandit Problem. *Machine Learning*, 47:235–256, 2002.

Donald A. Berry. Modified Two-Armed Bandit Strategies for Certain Clinical Trials. *Journal of The American Statistical Association*, 73:339–345, 1978. doi: 10.1080/01621459.1978.10481579.

Sébastien Bubeck, Rémi Munos, and Gilles Stoltz. Pure exploration in multi-armed bandits problems. In *ALT*, pages 23–37, 2009.

Loc Bui, Ramesh Johari, and Shie Mannor. Committing bandits. In *NIPS*, pages 1557–1565, 2011.

Varsha Dani, Thomas P. Hayes, and Sham M. Kakade. Stochastic linear optimization under bandit feedback. In *COLT*, pages 355–366, 2008.

Aurélien Garivier and Olivier Cappé. The kl-ucb algorithm for bounded stochastic bandits and beyond. *JMLR*, 19:359–376, 2011.

J. C. Gittins. Multi-armed Bandit Allocation Indices. 1989.

Sudipto Guha and Kamesh Munagala. Multi-armed bandits with metric switching costs. In *ICALP (2)*, pages 496–507, 2009.

T. Jun. A survey on the bandit problem with switching costs. *De Economist*, 152(4):513–541, 2004.

Levente Kocsis and Csaba Szepesvári. Bandit based monte-carlo planning. In *ECML*, pages 282–293, 2006.

John Langford and Tong Zhang. The epoch-greedy algorithm for multi-armed bandits with side information. In *NIPS*, 2007.

Lihong Li, Wei Chu, John Langford, and Robert E. Schapire. A contextual-bandit approach to personalized news article recommendation. In *WWW*, pages 661–670, 2010.

Aditya Mahajan and Demosthenis Teneketzis. *Multi-Armed Bandit Problems*. doi: 10.1007/978-0-387-49819-5_6.

Oded Maron and Andrew W. Moore. Hoeffding Races: Accelerating Model Selection Search for Classification and Function Approximation. In *NIPS*, pages 59–66, 1993.

Volodymyr Mnih, Cs. Szepesv'ari, and Jean yves Audibert. Empirical Bernstein stopping. In *International Conference on Machine Learning*, pages 672–679, 2008. doi: 10.1145/1390156.1390241.

William H Press. Bandit solutions provide unified ethical models for randomized clinical trials and comparative effectiveness research. *PNAS*, 106:22387–22392, 2009. doi: 10.1073/pnas.0912378106.

Steven L. Scott. A modern bayesian look at the multi-armed bandit. *Appl. Stoch. Model. Bus. Ind.*, 26(6):639–658, November 2010. ISSN 1524-1904. doi: 10.1002/asmb.874. URL http://dx.doi.org/10.1002/asmb.874.

Appendix: Proofs

In this section, we prove the theorems in this paper. The overall goal with the proofs is to show that the existing bounds in stochastic bandit also holds even in the existence of lock-up periods.

Array-UCB

The proofs of the cumulative regrets in UCB and UCB-E rely on the same bound. To avoid redundancy, we define Array-UCB, the generalization of UCB and UCB-E.

Definition 8 (Array-UCB) *Array-UCB is defined as the index-based policy with index*

$$B_{i,s,t} = \hat{X}_{i,s} + \sqrt{\frac{a(t)}{s}}, \tag{10}$$

where, $(a(1), a(2), ...)$ is an array of real numbers. For each round t, the forecaster selects the arm i with maximum $B_{i,T_i(t-1),t-1}$

When $a(t) = 2\log t$, Array-UCB is equal to UCB. When $a(t) = a$ (constant), Array-UCB is equal to UCB-E.

Proofs of Theorem 3 and Theorem 6

In this subsection, we prove Theorem 3 and Theorem 6, the uniform cumulative regret of UCB$'$ and UCB-E$'$. We convert Array-UCB to an algorithm for lock-up bandit with the procedure of Proposition 2. We call the converted algorithm Array-UCB$'$.

Theorem 9 *For Array-UCB$'$ with $a(t) \geq 2\log t$, the cumulative regret R is upper bounded as*

$$\mathbb{E}[\mathbf{R}(L_1, ..., L_N)] \leq \sum_{i \neq i^*} \left\{ \frac{4a_{max}}{\Delta_i} + \Delta_i L_{max} \left(1 + \frac{\pi}{3}\right) \right\}, \tag{11}$$

where $a_{max} = \max_{t \in \{1,...,T\}} a(t)$.

Theorem 3 and 6 are directly derived as the specialization of Theorem 9 with $a(t) = 2\log t$ and $a(t) = a \geq 2\log T$.

Proof [Theorem 9]

The proof is based on Proof 1 in (Auer et al., 2002). The base proof is applied to UCB in stochastic bandit. We extend this proof in two respects. First, the UCB index is generalized to the Array-UCB index (Equation (10)). Second, we take lock-up periods into consideration.

We upper bound $T_i(T)$, or the number of rounds suboptimal arm i is pulled in T rounds. Let $c_{t,s} = \sqrt{a(t)/s}$. Remember (s_n, f_n) tuple means the start and end round of the period n. We use both the period-wise notation with symbol n and round-wise notation with symbol t (c.f. Section 2.1).

$$T_i(L_1, ..., L_N) = \sum_{n=1}^{N} L_n \mathbb{I}\{I_n = i\} \tag{12}$$

$$= (l + L_{max} - 1) + \sum_{n=K+1}^{N} L_n \mathbb{I}\{I_n = i, T_i(s_n - 1) \geq l\}, \tag{13}$$

where the transformation at (13) comes from the fact that T_i is at most $l + L_{max} - 1$ at the first period after T_i exceeds or equals l. The condition i is selected at $n \geq 2$ is transformed as follows.

$$\mathbb{I}\{I_n = i\} \leq \mathbb{I}\{\hat{X}_{i^*,T_{i^*}(s_n-1)} + c_{s_n-1,T_{i^*}(s_n-1)} \leq \tag{14}$$

$$\hat{X}_{i,T_i(s_n-1)} + c_{s_n-1,T_i(s_n-1)}\} \tag{15}$$

$$\leq \mathbb{I}\{\min_{0<t_1<s_n} \hat{X}_{i^*,t_1} + c_{s_n-1,t_1} \leq \max_{l<t_2<s_n} \hat{X}_{i,t_2} + c_{s_n-1,t_2}\} \tag{16}$$

$$\leq \sum_{t_1=1}^{s_n-1} \sum_{t_2=1}^{s_n-1} \mathbb{I}\{\hat{X}_{i^*,t_1} + c_{s_n-1,t_1} \leq \hat{X}_{i,t_2} + c_{s_n-1,t_2}\}. \tag{17}$$

The condition $\hat{X}_{i^*,t_1} + c_{s_n-1,t_1} \leq \hat{X}_{i,t_2} + c_{s_n-1,t_2}$ in (17) implies that at least one of the following three conditions must hold.

$$\hat{X}_{i^*,t_1} \leq \mu^* - \sqrt{\frac{a(s_n-1)}{t_1}}, \tag{18}$$

$$\hat{X}_{i,t_2} \geq \mu_i + \sqrt{\frac{a(s_n-1)}{t_2}}, \tag{19}$$

$$\mu^* < \mu_i + 2\sqrt{\frac{a(s_n-1)}{t_2}}. \tag{20}$$

Now, we bound the probabilities of inequalities (18), (19), and (20). First, (20) never occurs when $l \geq \lceil \frac{4a_{max}}{\Delta_i^2} \rceil$. The probability of (18) is upper bounded as

$$\begin{aligned}
\mathbb{P}[(18)\,\text{is true}] &= \mathbb{P}\left[\hat{X}_{i^*,t_1} \leq \mu^* - \sqrt{\frac{a(s_n-1)}{t_1}}\right] && (21)\\
&\leq \mathbb{P}\left[\hat{X}_{i^*,t_1} \leq \mu^* - \sqrt{\frac{2\log(s_n-1)}{t_1}}\right] && (22)\\
&\leq \exp(-4\log(s_n-1)) \leq (s_n-1)^{-4}, && (23)
\end{aligned}$$

where we use the assumption $a(t) > 2\log t$ at (22) and the Hoeffding inequality at (23). By using the same arguments, we obtain the same bound for (19). By using inequalities (13), (17) and (23), we obtain

$$\begin{aligned}
\mathbb{E}[T_i(N)] &\leq \left(\left\lceil \frac{4a_{max}}{\Delta_i^2} \right\rceil + L_{max} - 1\right) && (24)\\
&\quad + \sum_{n=K+1}^{N} \sum_{t_1=1}^{s_n-1} \sum_{t_2=1}^{s_n-1} L_s \Big\{ \mathbb{P}[((18)\,\text{is true})] + \mathbb{P}[((19)\,\text{is true})] \Big\} && (25)\\
&\leq \left(\frac{4a_{max}}{\Delta_i^2} + L_{max}\right) + L_{max} \sum_{n=K+1}^{N} \sum_{t_1=1}^{s_n-1} \sum_{t_2=1}^{s_n-1} (2(s_n-1)^{-4}) && (26)\\
&\leq \left(\frac{4a_{max}}{\Delta_i^2} + L_{max}\right) + L_{max} \sum_{t=1}^{\infty} (2t^{-2}) && (27)\\
&\leq \left(\frac{4a_{max}}{\Delta_i^2} + L_{max}\right) + L_{max} \cdot \frac{\pi^2}{3} = \frac{4a_{max}}{\Delta_i^2} + L_{max}\left(1 + \frac{\pi^2}{3}\right). && (28)
\end{aligned}$$

■

Proof of Theorem 7

Proof [Theorem 7]

We extend Theorem 1 in (Audibert et al., 2010) to lock-up bandit. Consider an event

$$\xi = \left\{ \forall i \in \{1,...,K\}, t \in \{1,...,T\}, |\hat{X}_{i,t} - \mu_i| < \frac{1}{5}\sqrt{\frac{a}{t}} \right\}. \quad (29)$$

By using the Hoeffding inequality for each event and union bound, we have $\mathbb{P}(\xi) \geq 1 - 2TK\exp(-\frac{2a}{25})$. Indeed, the event is the sufficient condition for that the empirically best arm corresponds to the truly best arm. Since we assume event ξ, it is enough to prove that

$$\frac{1}{5}\sqrt{\frac{a}{T_i(T)}} \leq \frac{\Delta_i}{2}, \forall i \in \{1,...,K\}, \quad (30)$$

or equivalently

$$T_i(T) \geq \frac{4}{25}\frac{a}{\Delta_i^2}. \tag{31}$$

First, we prove the upper bound of the number of the suboptimal arms pulled, namely

$$T_i(t) \leq \frac{36}{25}\frac{a}{\Delta_i^2} + L_{max}, \forall i \neq i^*. \tag{32}$$

Since the algorithm can select an arm only at the start of each lock-up period, we use induction based on each period. Namely, we show (32) is true at the end of any periods. Remember, we denote the start and end of the lock-up period n as (s_n, f_n). We also denote the UCB-E index as $B_{i,s} = \hat{X}_{i,s} + \sqrt{a/s}$. Obviously the inequality holds when $n = 1$. We now assume the inequality is true at time $n-1$. If $I_n \neq i$, $T_i(f_n) = T_i(f_{n-1})$ and the inequality still holds. On the other hand, if $I_n = i$ then it means $B_{i,T_i(s_n-1)} \geq B_{i^*,T_{i^*}(s_n-1)}$. Since event ξ holds, we have $B_{i^*,T_{i^*}(s_n-1)} \geq \mu^*$ and $B_{i,T_i(s_n-1)} \leq \mu_i + \frac{6}{5}\sqrt{\frac{a}{T_i(s_n-1)}}$. Summing up these conditions, we obtain $\frac{6}{5}\sqrt{\frac{a}{T_i(s_n-1)}} \geq \Delta_i$. The arm i is chosen during the lock-up period n. Since $f_n - (s_n - 1) = L_n \leq L_{max}$, (32) still holds.

Next, we prove the lower bound of suboptimal arms selected

$$T_i(t) \geq \frac{4}{25}\min\left(\frac{a}{\Delta_i^2}, \frac{25}{36}(T_{i^*}(t) - L_{max})\right), \forall i \neq i^*. \tag{33}$$

We also use induction based on each lock-up period. We assume (33) holds at the end of $n-1$. Then, at period n , if $B_{i,T_i(s_n-1)} > B_{i^*,T_{i^*}(s_n-1)}$, then T_{i^*} does not increase, so it still holds. On the other hand, in the case of $B_{i,T_i(s_n-1)} \leq B_{i^*,T_{i^*}(s_n-1)}$, T_{i^*} might increase. Since we are on ξ,

$$\mu^* + \frac{6}{5}\sqrt{\frac{a}{T_{i^*}(s_n-1)}} \geq B_{i^*,T_{i^*}(s_n-1)} \geq B_{i,T_i(s_n-1)} \geq \mu_i + \frac{4}{5}\sqrt{\frac{a}{T_i(s_n-1)}}, \tag{34}$$

which gives

$$T_i(s_n-1) \geq \frac{16}{25}\frac{a}{\left(\Delta_i + \frac{6}{5}\sqrt{\frac{a}{T_{i^*}(s_n-1)}}\right)^2}. \tag{35}$$

By using $u + v \geq 2\max(u, v)$, $T_i(f_n) = T_i(s_n - 1)$, and $T_{i^*}(f_n) \geq T_{i^*}(s_n - 1) + L_{max}$, (33) holds. From (33), we only have to show that, for all $i \neq i^*$

$$\frac{25}{36}(T_{i^*}(T) - L_{max}) \geq \frac{a}{\Delta_i^2}. \tag{36}$$

By using (32), we obtain

$$T_{i^*}(T) - L_{max} = T - L_{max} - \sum_{i \neq i^*} T_i(T) \geq T - KL_{max} - \frac{36}{25}a\sum_{i \neq i^*}\Delta_i^{-2} \geq \frac{36}{25}a\Delta^{-2}, \tag{37}$$

where, we use the assumption of theorem, or $\frac{36}{25}H_1 a \geq T - KL_{max}$ in the last inequality. ■