Improved Sleeping Bandits with Stochastic Actions Sets and Adversarial Rewards

Aadirupa Saha [1] **Pierre Gaillard** [2] **Michal Valko** [3]

Abstract

In this paper, we consider the problem of sleeping bandits with stochastic action sets and adversarial rewards. In this setting, in contrast to most work in bandits, the actions may not be available at all times. For instance, some products might be out of stock in item recommendation. The best existing efficient (i.e., polynomial-time) algorithms for this problem only guarantee an $O(T^{2/3})$ upper-bound on the regret. Yet, inefficient algorithms based on EXP4 can achieve $O(\sqrt{T})$. In this paper, we provide a new computationally efficient algorithm inspired by EXP3 satisfying a regret of order $O(\sqrt{T})$ when the availabilities of each action $i \in \mathcal{A}$ are independent. We then study the most general version of the problem where at each round available sets are generated from some unknown arbitrary distribution (i.e., without the independence assumption) and propose an efficient algorithm with $O(\sqrt{2^K T})$ regret guarantee. Our theoretical results are corroborated with experimental evaluations.

1. Introduction

The problem of standard multiarmed bandit (MAB) is well studied in machine learning (Auer, 2000; Vermorel & Mohri, 2005) and used to model online decision-making problems under uncertainty. Due to their implicit exploration-vs-exploitation tradeoff, bandits are able to model clinical trials, movie recommendations, retail management job scheduling etc., where the goal is to keep pulling the ‘best-item’ in hindsight through sequentially querying one item at a time and subsequently observing a noisy reward feedback of the queried arm (Even-Dar et al., 2006; Auer et al., 2002a; Auer, 2002; Agrawal & Goyal, 2012; Bubeck et al., 2012). However, in various real world applications, the decision space (set of arms $\mathcal{A}$) often changes over time due to unavailability of some items etc. For instance, in retail stores some items might go out of stock, on a certain day some websites could be down, some restaurants might be closed etc. This setting is known as *sleeping bandits* in online learning (Kanade et al., 2009; Neu & Valko, 2014; Kanade & Steinke, 2014; Kale et al., 2016), where at any round the set of available actions could vary stochastically based on some unknown distributions over $\mathcal{A}$ (Neu & Valko, 2014; Cortes et al., 2019) or adversarially (Kale et al., 2016; Kleinberg et al., 2010; Kanade & Steinke, 2014). Besides the reward model, the set of available actions could also vary stochastically or adversarially (Kanade et al., 2009; Neu & Valko, 2014). The problem is known to be NP-hard when both rewards and availabilities are adversarial (Kleinberg et al., 2010; Kanade & Steinke, 2014; Kale et al., 2016). In case of stochastic rewards and adversarial availabilities the achievable regret lower bound is known to be $\Omega(\sqrt{KT})$, K being the number of actions in the decision space $\mathcal{A} = [K]$. The well studied EXP4 algorithm does achieve the above optimal regret bound, although it is computationally inefficient (Kleinberg et al., 2010; Kale et al., 2016). However, the best known efficient algorithm only guarantees an $\tilde{O}((TK)^{2/3})$ regret,[1] which is not matching the lower bound both in K and T (Neu & Valko, 2014).

In this paper we aim to give computationally efficient and optimal $O(\sqrt{T})$ algorithms for the problem of sleeping bandits with adversarial rewards and stochastic availabilities. Our specific contributions are as follows:

Contributions

- We identified a drawback in the (sleeping) loss estimates in the prior work for this setting and gave an insight and margin for improvement over the best know rate (Section 3).
- We first study the setting when the availabilities of each item $i \in \mathcal{A}$ are independent and propose an EXP3-based algorithm (Alg. 1) with an $O(K^2\sqrt{T})$ regret guarantee (Theorem 2, Sec. 3).
- We next study the problem when availabilities are not

[1] Indian Institute of Science, Bangalore, India. [2] Sierra Team, Inria, Paris, France. [3] DeepMind, Paris, France. Correspondence to: Aadirupa Saha <aadirupa@iisc.ac.in>.

Proceedings of the 37^{th} International Conference on Machine Learning, Vienna, Austria, PMLR 119, 2020. Copyright 2020 by the author(s).

[1] $\tilde{O}(\cdot)$ notation hides the logarithmic dependencies.

independent and give an algorithm with an $O(\sqrt{2^K T})$ regret guarantee (Sec. 4).

- We corroborated our theoretical results with empirical evidence (Sec. 5).

2. Problem Statement

Notation. We denote by $[n] := \{1, 2, \ldots, n\}$. $\mathbf{1}(\cdot)$ denotes the indicator random variable which takes value 1 if the predicate is true and 0 otherwise. $\tilde{O}(\cdot)$ notation is used to hide logarithmic dependencies.

2.1. Setup

Suppose the decision space (or set of actions) is $[K] := \{1, 2, \ldots, K\}$ with K distinct actions, and we consider a T round sequential game. At each time step $t \in [T]$, the learner is presented a set of available actions at round t, say $S_t \subseteq [K]$, upon which the learner's task is to play an action $i_t \in S_t$ and consequently suffer a loss $\ell_t(i_t) \in [0, 1]$, where $\boldsymbol{\ell}_t := [\ell_t(i)]_{i \in [K]} \in [0, 1]^K$ denotes the loss of the K actions chosen obliviously independent of the available actions S_t at time t. We consider the following two types of availabilities:

Independent Availabilities. In this case we assume that the availability of each item $i \in [K]$ is independent of the rest $[K] \setminus \{i\}$, such that at each round item $i \in [K]$ is drawn in set S_t with probability $a_i \in [0, 1]$, or in other words, for all item $i \in [K]$, $\mathbf{1}(i \in S_t) \sim Ber(a_i)$, where availability probabilities $\{a_i\}_{i \in [K]}$ are fixed over time intervals t, independent of each other, and unknown to the learner.

General Availabilities. In this case each S_ts is drawn iid from some unknown distribution $\mathcal{P}$ over subsets $\{S \subseteq [K], |S| \geq 1\}$ with no further assumption made on the properties of $\mathcal{P}$. We denote by $P(S)$ the probability of occurrence of set S.

Analyses with independent and general availabilities are provided respectively in Sec. 3 and 4.

2.2. Objective

We define by a policy $\pi : 2^{[K]} \mapsto [K]$ to be a mapping from a set of available actions/experts to an item.

Regret definition The performance of the learner, measured with respect to the best policy in hindsight, is defined as:

$$R_T = \max_{\pi : 2^{[K]} \mapsto [K]} \mathbf{E}\left[\sum_{t=1}^{T} \ell_t(i_t) - \sum_{t=1}^{T} \ell_t(\pi(S_t))\right], \quad (1)$$

where the expectation is taken w.r.t. the availabilities and the randomness of the player's strategy.

Remark 1. *One obvious regret lower bound of the above objective is $\Omega(\sqrt{KT})$, which follows from the bound of standard MAB with adversarial losses (Auer et al., 2002a) for the special case when all the items are available at all times (even for availability-independent case). Interestingly, for a* harder *Sleeping-Bandits setting with adversarial availabilities the lower bound is $\Omega(K\sqrt{T})$ (Kleinberg et al., 2010), even for which no computationally efficient algorithm is known till date (EXP4 is the only algorithm which achieves the regret but it is computationally inefficient). Thus the interesting question to answer here is if for our setup–that lies in the middle-ground of (Auer et al., 2002a) and (Kleinberg et al., 2010)–is it possible to attend the $O(K\sqrt{T})$ learning rate? Here lies the primary objective of this work. To the best of our knowledge, there is no existing algorithm which are known to achieve this optimal rate and the best known efficient algorithm is only guaranteed to yield an $\tilde{O}((TK)^{2/3})$ regret (Neu & Valko, 2014).*

3. Proposed algorithm: Independent Availabilities

In this section we propose our first algorithm for the problem (Sec. 2), which is based on a variant of thr EXP3 algorithm with a 'suitable' loss estimation technique. Thm. 2 proves the optimality of its regret performance.

Algorithm description. Similar to EXP3 algorithm, at every round $t \in [T]$ we maintain a probability distribution $\mathbf{p}_t$ over the arm set $[K]$ and also the empirical availability of each item $\widehat{a}_{ti} = \frac{1}{t}\sum_{\tau=1}^{t} \mathbf{1}(i \in S_\tau)$. Upon receiving the available set S_t, the algorithm redistributes $\mathbf{p}_t$ only on the set of available items S_t, say $\mathbf{q}_t$, and plays an item $i_t \sim \mathbf{q}_t$. Subsequently the environment reveals the loss $\ell_t(i_t)$, and we update the distribution $\mathbf{p}_{t+1}$ using exponential weights on the loss estimates for all $i \in [K]$

$$\widehat{\ell}_t(i) = \frac{\ell_t(i)\mathbf{1}(i = i_t)}{\bar{q}_t(i) + \lambda_t}, \quad (2)$$

where λ_t is a scale parameter and $\bar{q}_t(i)$ (see definition (3)) is an estimation of $Pr_{S_t, i_t}(i_t = i)$, the probability of playing arm i at time t under the joint uncertainty in availability of i (due to $S_t \sim \mathcal{P}$) and the randomness of EXP3 algorithm (due to $i_t \sim \mathbf{p}_t$).

New insight compared to existing algorithms. It is crucial to note that one of our main contributions lies in the loss estimation technique $\widehat{\boldsymbol{\ell}}_t$ in (2). The standard loss estimates used by EXP3 (see (Auer et al., 2002b)) are of the form $\widehat{\ell}_t(i) = \ell_t(i)\mathbf{1}(i = i_t)/p_t(i)$. Yet, because of the unavailable actions, the latter is biased. The solution proposed by (Neu & Valko, 2014) (see Sec. 4.3) consists of using unbiased loss estimates of the form $\widehat{\ell}_t(i) = \ell_t(i)\mathbf{1}(i = i_t)/(\widehat{p}_t(i)\widehat{a}_{ti})$ where $\widehat{a}_{ti}$ and $\widehat{p}_t(i)$ are estimates for the availability probability a_i and for the weight $p_t(i)$ respectively. The suboptimal $O(T^{2/3})$ of their regret bound resulted from this

Algorithm 1 *Sleeping-EXP3*

```
Input:
    Item set: $[K]$, learning rate $\eta$, scale parameter $\lambda_t$
    Confidence parameter: $\delta > 0$
Initialize:
    Initial probability distribution $\mathbf{p}_1(i) = \frac{1}{K}$, $\forall i \in [K]$
while $t = 1, 2, \ldots$ do
    Define $q_t^S(i) := \frac{p_t(i)\mathbf{1}(i \in S)}{\sum_{j \in S} p_t(j)}$, $\forall i \in [K], S \subseteq [K]$
    Receive $S_t \subseteq [K]$
    Sample $i_t \sim \mathbf{q}_t^{S_t}$
    Receive loss $\ell_t(i_t)$
    Compute: $\widehat{a}_{ti} = \frac{\sum_{\tau=1}^{t} \mathbf{1}(i \in S_\tau)}{t}$
                $P_{\widehat{\mathbf{a}}}(S) = \Pi_{i=1}^{K} \widehat{a}_{ti}^{\mathbf{1}(i \in S)} (1 - \widehat{a}_{ti})^{1-\mathbf{1}(i \in S)}$
                $\bar{q}_t(i) = \sum_{S \in 2^{[K]}} P_{\widehat{\mathbf{a}}}(S) q_t^S(i)$
    Estimate loss: $\widehat{\ell}_t(i) = \frac{\ell_t(i)\mathbf{1}(i = i_t)}{\bar{q}_t(i) + \lambda_t}$, $\forall i \in [K]$
    Update $p_{t+1}(i) = \frac{p_t(i)e^{-\eta \widehat{\ell}_t(i)}}{\sum_{j=1}^{K} p_t(i)e^{-\eta \widehat{\ell}_t(j)}}$, $\forall i \in [K]$
end while
```

separated estimation of $\widehat{p}_t(i)$ and $\widehat{a}_{ti}$, which leads to a high variance in the analysis because $\widehat{p}_t(i) = 0$ whenever $i \notin S_t$.

We circumvent this problem by estimating them jointly as

$$\bar{q}_t(i) := \sum_{S \in 2^{[K]}} P_{\widehat{\mathbf{a}}}(S) q_t^S(i) \,, \tag{3}$$

where $P_{\widehat{\mathbf{a}}_t}(S) = \Pi_{i=1}^{K} \widehat{a}_{ti}^{\mathbf{1}(i \in S)} (1 - \widehat{a}_{ti})^{1-\mathbf{1}(i \in S)}$ is the empirical probability of the availability of set S, and for all $i \in [K]$

$$q_t^S(i) := \frac{p_t(i)\mathbf{1}(i \in S)}{\sum_{j \in S} p_t(j)} \,, \tag{4}$$

is the redistributed mass of $\mathbf{p}_t$ on support set S. As shown in Lem. 1, $\bar{q}_t(i)$ is a good estimate for $q_t^*(i) = \mathbf{E}_{S \sim \mathbf{a}}\big[q_t^S(i)\big]$, which is the conditional probability of playing action $i_t = i$ at time t. It turns out that $\bar{q}_t(i)$ is much more stable than $\widehat{p}_t(i)\widehat{a}_{ti}$ and therefore implies better variance control in the regret analysis. This improvement finally leads to the optimal $O(\sqrt{T})$ regret guarantee (Thm. 2). The complete algorithm is given in Alg. 1.

The first crucial result we derive towards proving Thm. 2 is the following concentration guarantees on $\bar{q}_t$:

Lemma 1 (Concentration of $\bar{\mathbf{q}}_t$)**.** *Let $t \in [T]$ and $\delta \in (0, 1)$. Let $q_t^*(i) = \mathbf{E}_{S \sim \mathbf{a}}\big[q_t^S(i)\big]$ and $\bar{q}_t$ as defined in Equation (3). Then, with probability at least $1 - \delta$,*

$$|q_t^*(i) - \bar{q}_t(i)| \le 2K\sqrt{\frac{2\log(K/\delta)}{t}} + \frac{8K\log(K/\delta)}{3t} \,, \tag{5}$$

for all $i \in [K]$.

Using the result of Lem. 1, the following theorem analyses the regret guarantee of *Sleeping-EXP3* (Alg. 1).

Theorem 2 (*Sleeping-EXP3*: Regret Analysis)**.** *Let $T \ge 1$. The sleeping regret incurred by Sleeping-EXP3 (Alg. 1) can be bounded as:*

$$\begin{aligned} R_T &= \max_{\pi : 2^{[K]} \mapsto [K]} \mathbf{E}\Bigg[\sum_{t=1}^{T} \ell(i_t) - \sum_{t=1}^{T} \ell(\pi(S_t))\Bigg] \\ &\le 16K^2\sqrt{T \ln T} + 1 \,, \end{aligned}$$

for the parameter choices $\eta = \sqrt{(\log K)/(KT)}$, $\delta = K/T^2$, and

$$\lambda_t = \min\left\{2K\sqrt{\frac{2\log(K/\delta)}{t}} + \frac{8K\log(K/\delta)}{3t}, 1\right\} .$$

Proof. **(sketch)** Our proof is developed based on the standard regret guarantee of the EXP3 algorithm for the classical problem of multiarmed bandits with adversarial losses (Auer et al., 2002b; Auer, 2002). Precisely, consider any fixed set $S \subseteq [K]$, and suppose we run EXP3 algorithm on the set S, over any nonnegative sequence of losses $\widehat{\ell}_1, \widehat{\ell}_2, \ldots \widehat{\ell}_T$ over items of set S, and consequently with weight updates $\mathbf{q}_1^S, \mathbf{q}_2^S, \ldots \mathbf{q}_T^S$ as per the EXP3 algorithm with learning rate η. Then from the standard regret analysis of the EXP3 algorithm (Cesa-Bianchi & Lugosi, 2006), we get that for any $i \in S$:

$$\sum_{t=1}^{T} \big\langle \mathbf{q}_t^S, \widehat{\ell}_t \big\rangle - \sum_{t=1}^{T} \widehat{\ell}_t(i) \le \frac{\log K}{\eta} + \eta \sum_{t=1}^{T} \sum_{k \in S} q_t^S(k)\widehat{\ell}_t(k)^2 \,.$$

Let $\pi^* : S \mapsto [K]$ be any strategy. Then, applying the above regret bound to the choice $i = \pi^*(S)$ and taking the expectation over $S \sim P_{\mathbf{a}}$ and over the possible randomness of the estimated losses, we get

$$\begin{aligned} \sum_{t=1}^{T} \mathbf{E}\Big[\big\langle \mathbf{q}_t^S, \widehat{\ell}_t \big\rangle\Big] - \sum_{t=1}^{T} \mathbf{E}\Big[\widehat{\ell}_t(\pi^*(S))\Big] \le \\ \frac{\log K}{\eta} + \eta \sum_{t=1}^{T} \mathbf{E}\Bigg[\sum_{k \in S} q_t^S(k)\widehat{\ell}_t(k)^2\Bigg] \,. \end{aligned} \tag{6}$$

Now towards proving the actual regret bound of *Sleeping-EXP3* (recall the definition from Eqn. (1)), we first need to establish the following three main sub-results that relate the different expectations of Inequality (6) with quantities related to the actual regret (in Eqn. (1)).

Lemma 3. *Let $\delta \in (0, 1)$. Let $t \in [T]$. Define $\mathbf{q}_t^S$ as in (4) and $\widehat{\ell}_t$ as in (2). Assume that i_t is drawn according to $\mathbf{q}_t^{S_t}$ as defined in Alg. 1. Then,*

$$\mathbf{E}\big[\ell_t(i_t)\big] \le \mathbf{E}\big[\big\langle \mathbf{q}_t^S, \widehat{\ell}_t \big\rangle\big] + 2K\lambda_t + \frac{\delta}{\lambda_t} \,,$$

for $\lambda_t = 2K\sqrt{\frac{2\log(K/\delta)}{t}} + \frac{8K\log(K/\delta)}{3t}$.

Lemma 4. *Let $\delta \in (0,1)$. Let $t \in [T]$. Define $\widehat{\ell}_t$ as in (2) and assume that i_t is drawn according to $\mathbf{q}_t^{S_t}$ as defined in Alg. 1. Then for any $i \in [K]$,*

$$\mathbf{E}\big[\widehat{\ell}_t(i)\big] \leq \ell_t(i) + \frac{\delta}{\lambda_t}\,,$$

for $\lambda_t = 2K\sqrt{\frac{2\log(K/\delta)}{t}} + \frac{8K\log(K/\delta)}{3t}$.

Lemma 5. *Let $\delta \in (0,1)$. Let $t \in [T]$. Define $\mathbf{q}_t^S$ as in (4) and $\widehat{\ell}_t$ as in (2). Then,*

$$\mathbf{E}\left[\sum_{i\in S} q_t^S(i)\widehat{\ell}_t(i)^2\right] \leq K + \frac{\delta}{\lambda_t^2}.$$

for $\lambda_t = 2K\sqrt{\frac{2\log(K/\delta)}{t}} + \frac{8K\log(K/\delta)}{3t}$.

With the above claims in place, we are now proceed to prove the main theorem: Let us denote the best policy $\pi^* := \arg\min_{\pi:2^{[K]}\mapsto[K]} \sum_{t=1}^T \mathbf{E}_{S_t\sim P_{\mathbf{a}}}[\ell(\pi(S_t))]$. Now, recalling from Eqn. (1), the actual regret definition of our proposed algorithm, and combining the claims from Lem. 3, 4, we first get:

$$R_T(\textit{Sleeping-EXP3}) = \sum_{t=1}^T \mathbf{E}\Big[\ell_t(i_t) - \ell_t(\pi^*(S_t))\Big]$$
$$\leq 2K\sum_{t=1}^T \lambda_t + 2\sum_{t=1}^T \frac{\delta}{\lambda_t} + \sum_{t=1}^T \mathbf{E}\Big[\big\langle \mathbf{q}_t^S, \widehat{\ell}_t\big\rangle - \widehat{\ell}_t(\pi^*(S))\Big]\,.$$

Then, we can further upper-bound the last term on the right-hand-side using Inequality (6) and Lem. 5, which yields

$$R_T(\textit{Sleeping-EXP3})$$
$$\leq 2K\sum_{t=1}^T \lambda_t + 2\sum_{t=1}^T \frac{\delta}{\lambda_t} + \frac{\log K}{\eta} + \eta KT + \eta\sum_{t=1}^T \frac{\delta}{\lambda_t^2}$$
$$\leq \frac{\log K}{\eta} + \eta KT + 2K\sum_{t=1}^T \lambda_t + 3\sum_{t=1}^T \frac{\delta}{\lambda_t^2}\,, \tag{7}$$

where in the last inequality we used that $\eta \leq 1$ and $\lambda_t \leq 1$. Otherwise, we can always choose $\min\{1, \lambda_t\}$ instead of λ_t in the algorithm and Lem. 1 would still be satisfied.

The proof is concluded by replacing $\lambda_t = 2K\sqrt{\frac{2\log(K/\delta)}{t}} + \frac{8K\log(K/\delta)}{3t}$ and by bounding the two sums as follows:

$$\sum_{t=1}^T \lambda_t \leq 2K\sqrt{2\log\left(\frac{K}{\delta}\right)T} + \frac{8K}{3}\log\left(\frac{K}{\delta}\right)(1+\log T)$$

and using $\lambda_t \geq 2K\sqrt{2\log(K/\delta)/t}$, we have

$$\sum_{t=1}^T \frac{1}{\lambda_t^2} \leq \frac{1}{8K^2\log(K/\delta)}\sum_{t=1}^T t \leq \frac{T^2}{8K^2\log(K/\delta)} \leq \frac{T^2}{8K^2}.$$

Then, using $\delta := K/T^2$, $\log(K/\delta) = 2\log(T)$, we can further upper-bound: $\sum_{t=1}^T \lambda_t \leq 4K\sqrt{T\log T} + \frac{8K}{3}(1 + \log T)(\log T) \leq 7K\sqrt{T\log T}$, and $3\sum_{t=1}^T \frac{\delta}{\lambda_t^2} \leq \frac{3}{8K} \leq 1$. Thus, upper-bounding the two sums into (7), we get

$$R_T(\textit{Sleeping-EXP3}) \leq \frac{\log K}{\eta} + \eta KT + 14K^2\sqrt{T\log T} + 1\,.$$

Optimizing $\eta = \sqrt{(\log K)/KT}$ and upper-bounding $\sqrt{KT\log K} \leq K^2\sqrt{T}$, finally concludes the proof. □

The above regret bound is of order $\tilde{O}(K^2\sqrt{T})$, which is optimal in T, unlike any previous work which could only achieve $\tilde{O}((KT)^{2/3})$ regret guarantees (Neu & Valko, 2014) at best. Thus our regret guarantee is only suboptimal in terms of K, as the lower bound of this problem is known to be $\Omega(\sqrt{KT})$ (Kleinberg et al., 2010; Kanade et al., 2009). However, it should be noted that in our experiments (see Figure 4), the dependence of our regret on the number of arms behaves similarly to other algorithms although their theoretical guarantees expect better dependencies on K. The sub-optimality could thus be an artifact of our analysis, but despite our efforts, we have not been able to improve it. We think this may come from our proof of Lem. 1, in which we see two gross inequalities that may cost us this dependence on K. First, the proof upper-bounds $|q_t^{(S)}(i)| \leq 1$, while in average over i and S the latter is around $1/K$. Yet, dependence problems condemn us to use this worst-case upper-bound. Secondly, the proof uses uniform bounds of $|a_i - \widehat{a}_{ti}|$ over $i = 1, \ldots, K$ when the estimation errors could offset each other.

Note also that the regret bound in the theorem is worst-case. An interested direction for future work would be to study whether it is possible to derive an instance-dependent bound, based on the a_i instances. Typically, K could be replaced by the expected number of active experts. A first step in this direction would be to start from Inequality (15) in the proof of Lem. 1 and try to keep the dependence on the a_i distribution along the proof.

Finally, note that the algorithm only requires the beforehand knowledge of the horizon T to tune its hyper-parameter. However, the latter assumption can be removed by using standard calibration techniques such as the doubling trick (see (Cesa-Bianchi & Lugosi, 2006)).

3.1. Efficient *Sleeping-EXP3*: Improving Computational Complexity

Thm. 2 shows the optimality of *Sleeping-EXP3* (Alg. 1), but its one limitation lies in computing the probability estimates

$$\bar{q}_t(i) := \sum_{S\in 2^{[K]}} P_{\hat{\mathbf{a}}}(S)q_t^S(i), \forall i \in [K],$$

which requires $O(2^K)$ computational complexity per round.

In this section we show how to get around with this problem just by approximating $\bar{q}_t(i)$ by an empirical estimate

$$\tilde{q}_t(i) := \frac{1}{t}\sum_{\tau=1}^{t} q_t^{S_t^{(\tau)}}(i)\,, \tag{8}$$

where $S_t^{(1)}, S_t^{(2)}, \ldots S_t^{(t)}$ are t independent draws from the distribution $P_{\widehat{\mathbf{a}}}$, i.e. $P_{\widehat{\mathbf{a}}_t}(S) := \Pi_{i=1}^{K}\widehat{a}_{ti}^{\mathbf{1}(i\in S)}(1-\widehat{a}_{ti})^{1-\mathbf{1}(i\in S)}$ for any $S \subseteq [K]$ (recall the notation from Sec. 3). The above trick proves useful with the crucial observation that $q_t^{S_t^{(1)}}(i), q_t^{S_t^{(2)}}(i), \ldots q_t^{S_t^{(T)}}(i)$ are independent of each other (given the past) and that each $q_t^{S_t^{(\tau)}}(i)$ are unbiased estimated of $\bar{q}_t(i)$. That is, $\mathbf{E}_{\widehat{\mathbf{a}}_t}[q_t^{S_t^{(\tau)}}(i)] = \bar{q}_t(i)$, $\forall i \in [K], \tau \in [t]$. By classical concentration inequalities, this precisely leads to fast concentration of $\tilde{q}_t(i)$ to $\bar{q}_t(i)$ which in turn concentrates to $\mathbf{q}_t^*(i)$ (by Lem. 1). Combining these results, thus one can obtain the concentration of $\tilde{q}_t(i)$ to $\mathbf{q}_t^*(i)$ as shown in Lem. 6.

Lemma 6 (Concentration of $\tilde{q}_t(i)$)**.** *Let $t \in [T]$ and $\delta \in (0,1)$. Let $q_t^*(i) = \mathbf{E}_{S\sim\mathbf{a}}\big[q_t^S(i)\big]$ and $\tilde{q}_t$ as defined in Equation (8). Then, with probability at least $1-\delta$,*

$$|q_t^*(i) - \tilde{q}_t(i)| \le 4K\sqrt{\frac{\log(2K/\delta)}{t}} + \frac{8K\log(2K/\delta)}{3t}\,,$$

for all $i \in [K]$.

Remark 2. *Note that for estimating $\tilde{\mathbf{q}}_t$, we can not use the observed sets $S_1, S_2 \ldots, S_t$, instead of resampling $S_t^{(1)}, S_t^{(2)} \ldots, S_t^{(t)}$ again–this is because in that case the resulting numbers $q_t^{S_1}(i), q_t^{S_2}(i), \ldots q_t^{S_t}(i)$ would no longer be independent, and hence can not derive the concentration result of Lem. 6 (see proof in Appendix A.3 for details).*

Using the result of Lem. 6, we now derive the following theorem towards analyzing the regret guarantee of the computationally efficient version of *Sleeping-EXP3*.

Theorem 7 (*Sleeping-EXP3* (Computationally efficient version): Regret Analysis)**.** *Let $T \ge 1$. The sleeping regret incurred by the efficient approximation of Sleeping-EXP3 (Alg. 1) can be bounded as:*

$$R_T \quad\le\quad 20K^2\sqrt{T\log T} + 1\,,$$

for the parameter choices $\eta = \sqrt{\frac{\log K}{KT}}$, $\delta = 2K/T^2$ and $\lambda_t := 4K\sqrt{\log(2K/\delta)/t} + 8K\log(2K/\delta)/3t$.

Furthermore, the per-round time and space complexities of the algorithm are $O(tK)$ and $O(K)$ respectively.

Proof. **(sketch)** The regret bound can be proved using similar steps as described for Thm. 2, except now we replace the concentration result of Lem. 6 in place of Lem. 1.

Computational complexity: At any round $t \ge 1$, the algorithm requires only an $O(K)$ cost to update $\widehat{a}_{ti}$, $\widehat{\ell}_t(i)$ and $p_{t+1}(i)$, $\forall i \in [K]$. Resampling t subsets $S_t^{(\tau)}$ and computing $\{q_t^{S_t^{(\tau)}}(i)\}_{i\in[K]}$ requires another $O(tK)$ cost, resulting in the claimed computational complexity.

Spatial complexity: We only need to keep track of $\widehat{\mathbf{a}}_t \in [0,1]^K$ and $\mathbf{p}_t \in [0,1]^K$ making the total storage complexity just $O(K)$ (noting $\tilde{\mathbf{q}}_t$ can be computed sequentially). □

4. Proposed algorithm: General Availabilities

Setting. In this section we assume *general subset availabilities* (see Sec. 2).

Algorithm 2 *Sleeping-EXP3G*

```
Input:
    Learning rate $\eta > 0$, scale parameter $\lambda_t$
    Confidence parameter: $\delta > 0$
Initialize:
    Initial probability distribution $\mathbf{p}_1(i) = \frac{1}{K}$, $\forall i \in [K]$
while $t = 1, 2, \ldots$ do
    Receive $S_t$
    Compute $q_t(i) = \frac{p_t(i)\mathbf{1}(i\in S_t)}{\sum_{j\in S_t} p_t(j)}$, $\forall i \in [K]$
    Sample $i_t \sim \mathbf{q}_t$
    Receive loss $\ell_t(i_t)$
    Compute: $\bar{q}_t(i) := \frac{1}{t}\sum_{\tau=1}^{t} q_t^{S_\tau}(i)$
    Estimate loss bound $\widehat{\ell}_t(i) = \frac{\ell_t(i)\mathbf{1}(i=i_t)}{\bar{q}_t(i)+\lambda_t}$
    Update $p_{t+1}(i) = \frac{p_t(i)e^{-\eta\widehat{\ell}_t(i)}}{\sum_{j=1}^{K} p_t(i)e^{-\eta\widehat{\ell}_t(j)}}$, $\forall i \in [K]$
end while
```

4.1. Proposed Algorithm: *Sleeping-EXP3G*

Main idea. By and large, we use the same EXP3 based algorithm as proposed for the case of *independent availabilities*, the only difference lies in using a different empirical estimate

$$\bar{q}_t(i) := \frac{1}{t}\sum_{\tau=1}^{t} q_t^{S_\tau}(i)\,. \tag{9}$$

In hindsight, the above estimate $\bar{q}_t(i)$ is equal to the expectation $\mathbf{E}_{S\sim\widehat{P}_t}[q_t^S]$, i.e., $\bar{q}_t(i) = \sum_{S\in 2^{[K]}} \widehat{P}(S)q_t^S(i)$, where $\widehat{P}_t(S) := \frac{1}{t}\sum_{\tau=1}^{t}\mathbf{1}(S_\tau = S)$ is the empirical probability of set S at time t. The rest of the algorithm proceeds the same as Alg. 1, the complete description is given in Alg. 2.

4.2. Regret Analysis

We first analyze the concentration of $\bar{q}_t(i)$–the empirical probability of playing item i at any round t, and the result goes as follows:

Lemma 8 (Concentration of $\bar{q}_t(i)$)**.** *Let $t \in [T]$. Let $q_t^*(i) = \mathbf{E}_{S\sim\mathbf{a}}\big[q_t^S(i)\big]$, and define $\bar{q}_t(i)$ as in Equation (9).*

Then, with probability at least $(1-\delta)$,

$$|q_t^*(i) - \bar{q}_t(i)| \le \sqrt{\frac{2^{K+1}}{t} \ln \frac{2^K}{\delta}} + \frac{2^{K+1}}{3t} \ln \frac{2^K}{\delta}\,,$$

for all $i \in [K]$.

Using Lem. 8 we now analyze the regret bound of Alg. 2.

Theorem 9 (*Sleeping-EXP3G*: Regret Analysis)**.** *Let* $T \ge 1$. *Suppose we set* $\eta = \sqrt{(\log K)/(KT)}$, $\delta = 2^K/T^2$, *and* $\lambda_t = \sqrt{(2^{K+1}/t)\ln(2^K/\delta)} + 2^{K+1}\ln(2^K/\delta)/(3t)$. *Then, the regret incurred by Sleeping-EXP3G (Alg. 2) can be bounded as:*

$$R_T \le K\sqrt{2^{K+4}T\log T} + K2^{K+3}(\log T)^2\,.$$

Furthermore, the per-round space and time complexities of the algorithm are $O(tK)$.

Proof. **(sketch)** The proof proceeds almost similar to the proof of Thm. 2 except now the corresponding version of the main lemmas, aka. Lem. 3,4, and 5 are satisfied but for $\lambda_t = \sqrt{\frac{2^{K+1}}{t}\ln\frac{2^K}{\delta}} + \frac{2^{K+1}}{3t}\ln\frac{2^K}{\delta}$, since here we need to use the concentration Lem. 8 instead of Lem. 1.

Similar to the proof of Thm. 2 and following the same notation, we first combine claims from Lem. 3, 4 to get:

$$\begin{aligned} R_T(\textit{Sleeping-EXP3G}) &= \sum_{t=1}^T \mathbf{E}\Big[\ell_t(i_t) - \ell_t(\pi^*(S_t))\Big] \\ &\le \sum_{t=1}^T \mathbf{E}\big[\langle \mathbf{q}_t^S, \widehat{\ell}_t\rangle + 2K\lambda_t + \frac{2\delta}{\lambda_t}\big] - \mathbf{E}[\widehat{\ell}_t(\pi^*(S_t))] \\ &\overset{(a)}{\le} \frac{\log K}{\eta} + \eta\sum_{t=1}^T \mathbf{E}\Big[\sum_{k\in S} q_t^S(k)\widehat{\ell}_t(k)^2\Big] \\ &\qquad + 2\sum_{t=1}^T \big(K\lambda_t + \frac{\delta}{\lambda_t}\big) \\ &\overset{(b)}{\le} \frac{\log K}{\eta} + \eta KT + \sum_{t=1}^T \big(2K\lambda_t + \frac{2\delta}{\lambda_t} + \frac{\eta\delta}{\lambda_t^2}\big) \\ &\le \frac{\log K}{\eta} + \eta KT + \sum_{t=1}^T \big(2K\lambda_t + \frac{3\delta}{\lambda_t^2}\big), \end{aligned}$$

where Inequality (a) and (b) respectively follow from (6) and Lem. 5. The last inequality holds because $\eta \le 1$ and $\lambda_t \le 1$. To conclude the proof, it now only remains to compute the sums and to choose the parameters $\delta = 2^K/T^2$ and $\eta = \sqrt{(\log K)/KT}$. Using $\lambda_t \ge \sqrt{2^{K+1}/t}$, we have $\sum_{t=1}^T \frac{\delta}{\lambda_t^2} \le \frac{\delta T^2}{2^{K+1}} \le 1$, and since $\log(2^K/\delta) = 2\log T$, we further have

$$\lambda_t = \sqrt{\frac{2^{K+1}}{t}\ln T} + \frac{2^{K+1}}{3t}\ln T$$

which entails:

$$\begin{aligned} \sum_{t=1}^T \lambda_t &\le \sqrt{2^{K+1}T\log T} + \frac{2^{K+1}}{3}(\log T)(1+\log T) \\ &\le \sqrt{2^{K+1}T\log T} + 2^{K+2}(\log T)^2. \end{aligned}$$

Finally, substituting η and the above bounds in the regret upper-bound yields the desired result.

Complexity analysis. The only difference with Alg. 1 lies in computing $\bar{q}_t(i)$. Following a similar argument given for proving the computational complexity of Thm. 7, this can also be performed with a computational cost of $O(tK)$. Yet, now the algorithm specifically needs to keep in memory the empirical distribution of $S_1, \ldots, S_t$ and thus a space complexity of $O(K + \min\{tK, 2^K\})$ is required. □

Our regret bound in Thm. 9 has the optimal $\sqrt{T}$ dependency—to the best of our knowledge, *Sleeping-EXP3G* (Alg 2) is the first *computationally efficient* algorithm to achieve $O(\sqrt{T})$ guarantee for the problem of *Sleeping-Bandits*. Of course the EXP4 algorithm is known to attain the optimal $O(\sqrt{KT})$ regret bound, however it is computationally infeasible (Kleinberg et al., 2010) due to the overhead of maintaining a combinatorial policy class.

Yet, on the downside, it is worth pointing out that the regret bound of Thm. 9 only provides a sublinear regret in the regime $2^K \le O(T)$, in which case algorithms such as EXP4 can be efficiently implemented. However, we still believe our algorithm to be an interesting contribution because it completes another side of the computational-performance trade-off. It is possible to move the exponential dependence on the number of experts from the computational complexity to the regret bound.

Another argument in favor of this algorithm is that it provides an efficient alternative algorithm to EXP4 with regret guarantees in the regime $2^K \le O(T)$. In the other regime, though we could not prove any meaningful regret guarantee, Alg. 4.1 performs very well in practice as shown by our experiments. We believe the 2^K constant in the regret to be an artifact of our analysis. However, removing it seems to be highly challenging due to dependencies between q_t^S and $S_1, \ldots, S_t$. An analysis of the concentration of $\bar{q}_t$ (defined in (9)) to q_t^* without exponential dependence on K proved to be particularly complicated. We leave this question for future research.

5. Experiments

In this section we present the empirical evaluation of our proposed algorithms (Sec. 3 and 4) comparing their performances with the two existing *sleeping bandit* algorithms that apply to our problem setting, i.e. for adversarial losses

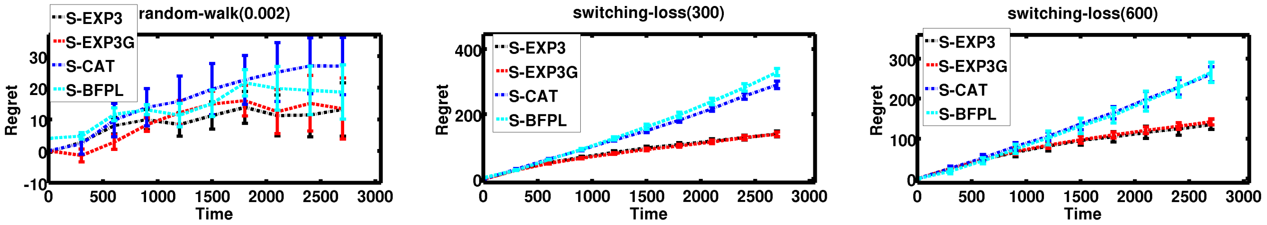

Figure 1. Regret vs Time: Independent availabilities

and stochastic availabilities. Thus we report the comparative performances of the following algorithms:

1. *Sleeping-EXP3*: Our proposed Alg. 1 (the efficient version as described in Sec. 3.1).
2. *Sleeping-EXP3G*: Our proposed Alg. 2.
3. *Sleeping-Cat*: The algorithm proposed by (Neu & Valko, 2014) (precisely their Algorithm for semi-bandit feedback in Sec. 4.3).
4. *Bandit-SFPL*: The algorithm proposed by (Kanade et al., 2009) (see Fig. 3, BSFPL algorithm, Sec. 2).

Performance Measures. In all cases, we report the cumulative regret of the algorithms for $T = 5000$ time steps, each averaged over 50 runs. In the following subsections, we analyze our experimental evaluations for both independent and general (non-independent) availabilities.

5.1. Independent Availabilities

In this case the item availabilities are assumed to be independent at each round (description in Sec. 2).

Environments. We consider $K = 20$ and generate the probabilities of item availabilities $\{a_i\}_{i\in[K]}$ independently and uniformly at random from the interval $[0.3, 0.9]$. We use the following loss generation techniques: (1) *Switching loss or SL(τ)*. We generate the loss sequence such that the best performing expert changes after every τ length epochs. (2) *Markov loss or ML(p)*. Similar to the setting used in (Neu & Valko, 2014), losses for each arm are constructed as random walks with Gaussian increments of standard deviation p, initialized uniformly on $[0, 1]$ such that losses outside $[0, 1]$ are truncated. The explicit values used for τ and p are specified in the corresponding figures. The algorithm parameters η, λ_t, δ are set as defined in Thm7.

Remarks. From Fig. 1 it clearly shows that regret bounds of our proposed algorithm *Sleeping-EXP3* and *Sleeping-EXP3G* outperform the other two due to their orderwise optimal $O(\sqrt{T})$ regret performance (see Thm. 2 and 9). In particular, *Bandit-SFPL* performs the worst due to its initial $O(T^{4/5})$ exploration rounds and uniform exploration phases thereafter. *Sleeping-Cat* gives a much competitive regret bound compared to *Bandit-SFPL* however, still lags behind due to the $O(T^{2/3})$ regret guarantee (see Sec. 3.1 for a detailed explanation).

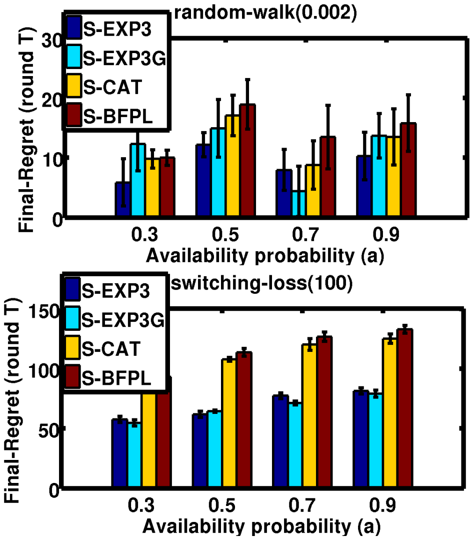

Figure 2. Final regret (at round T) vs availability probabilities((a))

5.2. Regret vs Varying Availabilities.

We next conduct a set of experiments to compare the regret performances of the algorithms with varying availability probabilities: For this we assign same availability $a_i = a \in [0.1]$ to every item $i \in [K]$ for $a = 0.3, 0.5, 0.7, 0.9$ and plot the final cumulative regret of each algorithm.

Remarks From Fig. 2, we again note our algorithms outperform the other two by a large margin for almost every p. The performance of BSFPL is worse, it steadily decreases with increasing availability probability due to the explicit $O(T^{4/5})$ exploration rounds in the initial phase of BSFPL, and even thereafter it keeps on suffering the loss of the uniform policy scaled by the exploration probability.

5.3. Correlated (General) Availabilities

We now assess the performances when the availabilities of items are dependent (description in Sec. 2).

Environments. To enforce dependencies of item availabilities we generate each set S_t by drawing a random sample

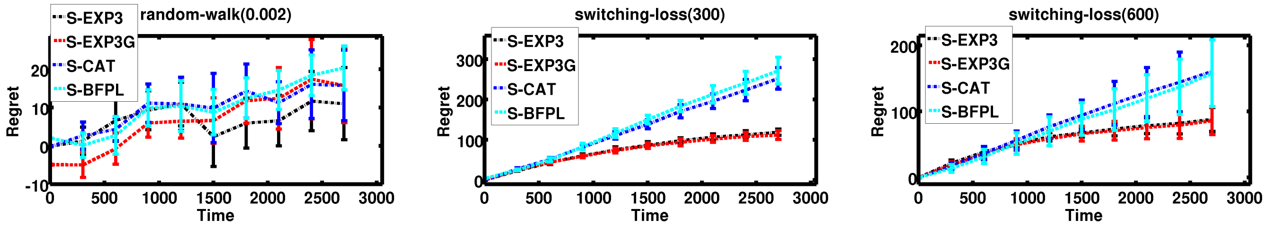

Figure 3. Regret vs time: General availabilities

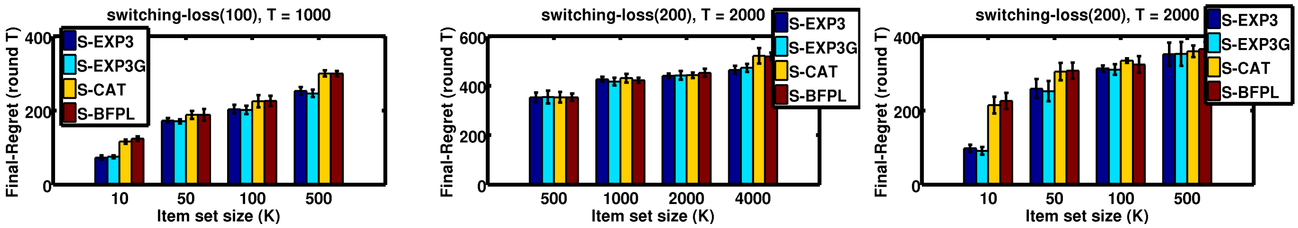

Figure 4. Final regret (at round T) vs item size (K)

from a Gaussian$(\boldsymbol{\mu}, \Sigma)$ such that $\boldsymbol{\mu}_i = 0, \forall i \in [K]$, and Σ is some random $K \times K$ positive definite matrix, e.g. block diagonal matrix with strong correlations among certain groups of items. More precisely, at each round t, we first sample a random K-vector, say v_t, from Gaussian$(\boldsymbol{\mu}, \Sigma)$ and we set $S_t = i \in [K] | v_t(i) > 0$, i.e. S_t includes all those items whose corresponding coordinates are non-negative–this thus enforces item dependencies in the resulted S_t if Σ is block diagonal (or any correlation matrix). To generate the loss sequences, we use similar techniques described in Sec. 5.1. The algorithm parameters are set as defined in Thm9.

Remarks. From Fig. 3 one can again verify the superior performance of our algorithms over *Sleeping-Cat* and *Bandit-SFPL*, however the effect is only visible for large T as for smaller time steps t, the $O(2^K)$ terms dominates the regret performance, but as t shoots higher our optimal $O(\sqrt{T})$ rate outperforms the suboptimal $O(T^{2/3})$ and $O(T^{4/5})$ rates of *Sleeping-Cat* and *Bandit-SFPL* respectively.

5.4. Regret vs Varying Item-size (K).

Finally we also conduct a set of experiments changing the item set size K over a wide range ($K = 10$ to 4000). We report the final cumulative regret of all algorithms vs. K for different switching loss sequence for both independent and general availabilities, as specified in Fig. 4.

Remark. Fig. 4 shows that the regret of each algorithm increases with K, as expected. As before the other two baselines perform suboptimally in comparison to our algorithms, however the interesting thing to note is the relative performance of *Sleeping-EXP3* and *Sleeping-EXP3G*—as per Thm. 2 and 9, *Sleeping-EXP3* must outperform *Sleeping-EXP3G* with increasing K, however the effect does not seem to be so drastic experimentally, possibly revealing the scope of improving Thm. 9 in terms of a better dependency in K.

6. Conclusion and Future Work

We have presented a new approach that brought an improved rate for the setting of sleeping bandits with adversarial losses and stochastic availabilities including both minimax and instance-dependence guarantees. While our bounds guarantee a regret of $\tilde{O}(\sqrt{T})$, there are several open questions before the studied setting can be considered as closed. Firstly, for the case of independent availabilities, we provide a regret guarantee of $\tilde{O}(K^2\sqrt{T})$, leaving open whether $\tilde{O}(\sqrt{KT})$ is possible as in the standard non-sleeping setting. Secondly, while we provided computationally efficient (i.e., with per-round complexity of order $O(tK)$) *Sleeping-EXP3*, for the case of general availabilities and provided instance dependent regret guarantees for it, the worst case regret guarantee still amounts to $\tilde{O}(\sqrt{2^K T})$. Therefore, it is still unknown if for the general availabilities we can get an algorithm that would be both computationally efficient and have $\tilde{O}(\text{poly}(K)\sqrt{T})$ regret guarantee in the worst case. We would like to point out that the new techniques could be potentially used to provide new algorithms and guarantees in settings with similar challenges as in sleeping bandits, such as rotting or dying bandits. Finally, having algorithms for sleeping bandits with $\tilde{O}(\sqrt{T})$ regret guarantees, opens a way to deal with sleeping constraints in more challenging structured bandits with large or infinite number of arms and having the regret guarantee depend not on number of arms but rather some effective dimension of the arms' space.

Acknowledgements

We thank the anonymous reviewers for their valuable suggestions. The research presented was supported French National Research Agency project BOLD (ANR-19-CE23-0026-04) and by European CHIST-ERA project DELTA. We also wish to thank Antoine Chambaz and Marie-Hélène Gbaguidi for supporting Aadirupa's internship at Inria, and Rianne de Heide for the thorough proofreading.

References

Agrawal, S. and Goyal, N. Analysis of thompson sampling for the multi-armed bandit problem. In *Conference on Learning Theory*, pp. 39–1, 2012.

Auer, P. Using upper confidence bounds for online learning. In *Foundations of Computer Science, 2000. Proceedings. 41st Annual Symposium on*, pp. 270–279. IEEE, 2000.

Auer, P. Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research*, 3(Nov):397–422, 2002.

Auer, P., Cesa-Bianchi, N., and Fischer, P. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2-3):235–256, 2002a.

Auer, P., Cesa-Bianchi, N., Freund, Y., and Schapire, R. E. The nonstochastic multiarmed bandit problem. *SIAM journal on computing*, 32(1):48–77, 2002b.

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

Cesa-Bianchi, N. and Lugosi, G. *Prediction, learning, and games*. Cambridge university press, 2006.

Cortes, C., Desalvo, G., Gentile, C., Mohri, M., and Yang, S. Online learning with sleeping experts and feedback graphs. In *International Conference on Machine Learning*, pp. 1370–1378, 2019.

Even-Dar, E., Mannor, S., and Mansour, Y. Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems. *Journal of machine learning research*, 7(Jun):1079–1105, 2006.

Kale, S., Lee, C., and Pál, D. Hardness of online sleeping combinatorial optimization problems. In *Advances in Neural Information Processing Systems*, pp. 2181–2189, 2016.

Kanade, V. and Steinke, T. Learning hurdles for sleeping experts. *ACM Transactions on Computation Theory (TOCT)*, 6(3):11, 2014.

Kanade, V., McMahan, H. B., and Bryan, B. Sleeping experts and bandits with stochastic action availability and adversarial rewards. 2009.

Kleinberg, R., Niculescu-Mizil, A., and Sharma, Y. Regret bounds for sleeping experts and bandits. *Machine learning*, 80(2-3):245–272, 2010.

Neu, G. and Valko, M. Online combinatorial optimization with stochastic decision sets and adversarial losses. In *Advances in Neural Information Processing Systems*, pp. 2780–2788, 2014.

Vermorel, J. and Mohri, M. Multi-armed bandit algorithms and empirical evaluation. In *European conference on machine learning*, pp. 437–448. Springer, 2005.