# Forced Exploration in Bandit Problems

**Qi Han, Li Zhu *, Fei Guo**

School of Software Engineering, Xi'an Jiaotong University, 28 Xianning West Road, Xi'an, Shaanxi, 710049, China
{qihan19,co.fly}@stu.xjtu.edu.cn, zhuli@xjtu.edu.cn

## Abstract

The multi-armed bandit(MAB) is a classical sequential decision problem. Most work requires assumptions about the reward distribution (e.g., bounded), while practitioners may have difficulty obtaining information about these distributions to design models for their problems, especially in non-stationary MAB problems. This paper aims to design a multi-armed bandit algorithm that can be implemented without using information about the reward distribution while still achieving substantial regret upper bounds. To this end, we propose a novel algorithm alternating between greedy rule and forced exploration. Our method can be applied to Gaussian, Bernoulli and other subgaussian distributions, and its implementation does not require additional information. We employ a unified analysis method for different forced exploration strategies and provide problem-dependent regret upper bounds for stationary and piecewise-stationary settings. Furthermore, we compare our algorithm with popular bandit algorithms on different reward distributions.

## Introduction

The multi-armed bandit (MAB) is a classical reinforcement learning problem. It simulates the process of pulling different arms of a slot machine, where each arm has an unknown reward distribution and only the selected arm's reward is observed. The learner aims to find the optimal policy that maximizes the cumulative reward. To achieve better long-term rewards, the learner must balance exploration and exploitation.

MAB has been widely used in many sequential decision tasks, such as online recommendation systems (Li et al. 2011; Li, Karatzoglou, and Gentile 2016), online advertisement campaign (Schwartz, Bradlow, and Fader 2017) and diagnosis and treatment experiment (Vermorel and Mohri 2005). Most of the existing multi-armed bandit algorithms are based on Upper Confidence Bound (UCB) (Auer, Cesa-Bianchi, and Fischer 2002) and Thompson Sampling (TS) (Thompson 1933), which often rely on assumptions about the reward distribution. For example, some studies assume that the reward distribution follows a sub-Gaussian distribution, and the algorithm implementation requires knowledge of the variance parameter. Additionally, some works assume that the reward distribution is bounded, and the algorithm design necessitates specific upper-bound information.

In the standard MAB model, the reward distribution remains constant. However, in real-world scenarios, the distribution of rewards may vary over time. In the context of clinical trials, the target disease may undergo mutations, causing the initially optimal treatment to potentially become less effective compared to another candidate (Gorre et al. 2001). Similarly, in online recommendation systems, user preferences are prone to evolve (Wu, Iyer, and Wang 2018), rendering collected data progressively outdated. During the past ten years, several works have been conducted on non-stationary multi-armed bandit problems. These approaches can be broadly classified into two categories: one category involves detecting changes in the reward distribution using change-point detection algorithms (Liu, Lee, and Shroff 2018; Cao et al. 2019; Auer, Gajane, and Ortner 2019; Chen et al. 2019; Besson et al. 2022). In contrast, the other category focuses on mitigating the impact of past observations in a passive manner (Garivier and Moulines 2011; Raj and Kalyani 2017; Trovo et al. 2020). Among them, Auer, Gajane, and Ortner (2019); Chen et al. (2019); Besson et al. (2022) can derive regret bounds without knowing the number of changes. However, as in the stationary settings, these algorithms also require prior information on the reward distributions for their implementation.

Recently, the non-parametric multi-armed bandit algorithm has garnered considerable attention (Lattimore 2017; Kveton et al. 2019a,b; Riou and Honda 2020; Baudry, Russac, and Cappé 2021; Liu et al. 2022). The implementation of these algorithms does not require strong parametric assumptions on the reward distributions, that is, a single implementation can be applied to several different reward distributions. Although the implementation of some mentioned algorithms (Kveton et al. 2019b,b; Riou and Honda 2020) does not need to know the information of reward distribution in advance, they require that the reward distribution is bounded. Therefore, they cannot be applied to a wider unbounded distribution. Wang et al. (2020) propose a perturbation based exploration method called *ReBoot* which has been analyzed only for Gaussian distributions. Chan (2020) propose SSMC by comparing the subsample means of the leading arm with the sample means of its competitors for

*Li Zhu is the corresponding author.

one-parameter exponential families distributions. However, as with the existing subsampling algorithms (Baransi, Maillard, and Mannor 2014), they have to store the entire history of rewards for all the arms. Baudry, Russac, and Cappé (2021) propose LB-SDA and LB-SDA-LM with a deterministic subsampling rule for one-parameter exponential families distributions. LB-SDA-LM is introduced to address the issue of high storage space and computational complexity. By using a sliding window, LB-SDA can be applied to non-stationary settings. While it cannot be applied to Gaussian distributions with unknown variance and the arms must come from the same one-parameter exponential family. Lattimore (2017) propose an algorithm that can be applied to various common distributions but requires the learner to know the specific kurtosis. Liu et al. (2022) propose the extended robust UCB policies without knowing the knowledge of an upper bound on specific moments of reward distributions. However, their algorithm needs a specific moment control coefficient as input.

Table 1 lists the required assumptions and preliminary information for some mentioned algorithms. Our algorithm is applicable to sub-Gaussian distributions. This assumption covers common distributions such as Gaussian, bounded, and Bernoulli distributions. The implementation of our algorithm does not require knowledge of the variance parameter for sub-Gaussian distributions.

**Contributions** In this paper, we propose a bandit algorithm that achieves respectable upper bounds on regret without using the parameters of distribution model for implementation. Our method is simple and easy-to-implement with the core idea of forcing exploration. Each time step forces an arm to be explored or uses greedy rule to select an arm. Specifically, our algorithm takes a non-decreasing sequence $\{f(r)\}$ as input. This input sequence controls how each arm is forced to be pulled.

In the stationary settings, we provide problem-dependent regret upper bounds. This regret upper bound is related to the input sequence, i.e., different input sequence will lead to different upper bounds. For example, if $\{f(r)\}$ is taken as an exponential sequence, our algorithm can guarantee a problem-dependent asymptotic upper bound $O((\log T)^2)$ on the expected number of pulls of the suboptimal arm.

In the piecewise-stationary settings, we use a sliding window $\tau$ along with a reset strategy to adapt to changes in the reward distribution. The reset strategy is realized by periodically resetting the sequence $\{f(r)\}$. This ensures that the algorithm maintains its exploration capability after $\tau$ time steps. Furthermore, we show that our algorithm has the ability to achieve a problem-dependent asymptotic regret bound of order $\tilde{O}(\sqrt{TB_T})$ if the number of breakpoints $B_T$ is a constant independent of $T$. This asymptotic regret bound matches the lower bounds in finite time horizon (Garivier and Moulines 2011), up to logarithmic factors.

## Problem Formulation

**Stationary environments** Let's consider a multi-armed bandit problem with finite time horizon $T$ and a set of arms $\mathcal{A} := \{1, ..., K\}$. At time step $t$, the learner must choose an arm $i_t \in \mathcal{A}$ and receive the corresponding reward $X_t(i_t)$. The reward comes from a different distribution unknown to the learner and the expectation of $X_t(i)$ is denoted as $\mu(i) = \mathbb{E}[X_t(i)]$. Let $i^*$ denote the expected reward of the optimal arm, i.e. ,$\mu(i^*) = \max_{i\in\{1,...,K\}} \mu(i)$. The gap between the expectation of the optimal arm and the suboptimal arm is denoted as $\Delta(i) = \mu(i^*) - \mu(i)$.

The history $h_t$ is defined as the sequence of actions and rewards from the previous $t-1$ time steps. A policy, denoted as $\pi$, is a function $\pi(h_t) = i_t$ that selects arm $i_t$ to play at time step $t$ based on the history $h_t$. The performance of a policy $\pi$ is measured in terms of cumulative expected regret:

$$R_T^{\pi} = \mathbb{E}\left[\sum_{t=1}^{T} \mu(i^*) - \mu(i_t)\right], \tag{1}$$

where $\mathbb{E}[\cdot]$ is the expectation with respect to randomness of $\pi$. Let $k_T(i) = \sum_{t=1}^{T} \mathbb{1}\{i_t = i, \mu(i) \neq \mu(i^*)\}$, the regret can be denoted as

$$R_T^{\pi} = \sum_{i=1}^{K} \Delta(i)\mathbb{E}[k_T(i)]. \tag{2}$$

In later section, we provide a regret upper bound for our method by analyzing the upper bound on $\mathbb{E}[k_T(i)]$.

**Piecewise-stationary environments** The piecewise-stationary bandit problem has been extensively studied. Piecewise-stationary bandits pose a more challenging problem as the learner needs to balance exploration and exploitation within each stationary phase and during the changes between different phases. In this setting, the reward distributions remain constant for a certain period and change at unknown time steps, called *breakpoints*. The number of breakpoints is denoted as $B_T = \sum_{t=1}^{T-1} \mathbb{1}\{\exists i \in \mathcal{A} : \mu_t(i) \neq \mu_{t+1}(i)\}$. The optimal arm may vary over time and is denoted by $\mu_t(i_t^*) = \max_{i\in\{1,..,K\}} \mu_t(i)$. The performance of a policy is measured by

$$R_T^{\pi} = \mathbb{E}\left[\sum_{t=1}^{T} \mu_t(i_t^*) - \mu_t(i_t)\right]. \tag{3}$$

As in the stationary environment, the regret can be analyzed by bound the expectation of the number of pulls of suboptimal arm $i$ up to the time step $T$.

## Stationary Environments

### Forced Exploration

Our algorithm pulls arms in two ways. The first is based on the greedy rule, which pulls the arm with the largest value of the estimator. The second one is a forced exploration step. Specifically, the algorithm takes a sequence $\{f(1), f(2), ...\}$as input. Let $p(i)$ denote the number of times that arm $i$ is not pulled. At round $r$, if $p(i) \geq f(r)$, arm $i$ will be forced to pull once and reset $p(i)$ to 0. If all arms have been pulled at least once at round $r$, let $r = r + 1$ and repeat the process in the next round.

Algorithm 1 shows the pseudocode of our method. We require the input sequence to be non-decreasing to prevent the

| Algorithms | Assumption | Known |
|---|---|---|
| UCB,TS | Support in $[a, b]$ or $\sigma$-subgaussian | $a, b$ or $\sigma$ |
| Robust-UCB | $\mathbb{E}[\lvert X-\mu\rvert^{1+\epsilon}] < u, 0 < \epsilon \leq 1$ | $\epsilon, u$ |
| ReBoot(Wang et al. 2020) | Gaussian | - |
| SSMC,LB-SDA | one-parameter exponential families | - |
| (Lattimore 2017) | Bounded kurtosis $\mathbb{E}[(\frac{X-\mu}{\sigma})^4] < \kappa$ | $\kappa$ |
| (Liu et al. 2022) | - | moment control coefficient |
| Our method | subgaussian | - |

Table 1: Assumptions and implementation parameters

algorithm from over-exploring. If $f(r)$ is less than a small constant $c$ after $t$ time steps,

$$R_T^\pi > (K-1)\frac{T-t}{c}. \tag{4}$$

This shows that the algorithm can only obtain linear regret. Step 3 is the greedy rule. In Step 5, the algorithm pull the arm that has not been pulled more than $f(r)$ times. To simplify the notation, we use the equivalent notation of $\arg\max_i p(i)$.

**Related to Explore-Then-Committee (ETC)** (Garivier, Lattimore, and Kaufmann 2016) study ETC policy for two-armed bandit problems with Gaussian rewards. ETC explores until a stopping time $s$. Let

$$f(r) = \begin{cases} 1, r \leq s \\ 0, r > s \end{cases},$$

our method is equivalent to ETC. The essential difference is that ETC is non-adaptive, i.e., for different problem instances, the set of all exploration rounds and the choice of arms therein is fixed before the first round. For some common input sequences, such as $f(r) = \sqrt{T}$ or $f(r) = r$, the exploration schedule of our method is related to the history of pulled arms, so our method is adaptive.

**Related to $\epsilon_t$-greedy** $\epsilon_t$-greedy strategy tosses a coin with a success probability $\epsilon_t$ at time step $t$ and randomly selects an arm if success, otherwise the arm with the largest average reward is selected. If the exploration probability satisfies $\epsilon_t \sim t^{-\frac{1}{3}}$, this greedy strategy achieves regret bound on the order of $O(T^{\frac{2}{3}})$. Similar to our method, this strategy needs to decide whether to explore at each time step. The difference is that $\epsilon_t$-greedy is non-adaptive since it does not adapt their exploration schedule to the history.

## Regret Analysis

In this section, we bound the expectation of the number of pulls of suboptimal arms. The detailed proofs of Theorem and Corollary are at https://arxiv.org/abs/2312.07285.

Let $h_t(i)$ denotes the number of times arm $i$ is forced to pull (Step 5) up to time $t$. We give the following theorem and proof sketch.

**Theorem 1.** *Assume that the reward distribution is $\sigma$-subgaussian. Let $\{f(r)\}$ be a non-decreasing sequence, for any suboptimal arm $i$,*

$$\mathbb{E}[k_T(i)] \leq h_T(i) + 2me^{\frac{1}{m}}\sum_{t=1}^{T} e^{\frac{-h_t(i)}{m}}, \tag{5}$$

Algorithm 1: FE

**Input**: non-decreasing sequence $\{f(r)\}$, $K$ arms, horizon $T$

**Initialization:** $t = 1, r = 0, f(0) = 0, \forall i \in \{1, ..., K\}, p(i) = 0, flag(i) = 0$

1: **while** $t < T$ **do**
2: **if** $\forall i \in \{1, ..., K\}, p(i) < f(r)$ **then**
3: Pull arm $i_t = \arg\max_i \hat{\mu}(i)$.
4: **else**
5: Pull arm $i_t = \arg\max_i p(i)$
6: **end if**
7: Update the estimate $\hat{\mu}(i_t)$,$p(i_t) = 0, flag(i_t) = 1$
8: $p(i) = p(i) + 1$ for all unpulled arm $i$
9: **if** $\forall i \in \{1, ..., K\}, flag(i) == 1$ **then**
10: $r = r + 1$
11: $\forall i \in \{1, ..., K\}, flag(i) = 0$
12: **end if**
13: $t = t + 1$
14: **end while**

*where $m = \frac{8\sigma^2}{\Delta(i)^2}$.*

**Proof sketch** We bound the number of suboptimal arm's pulls in Step 3 and Step 5 in Algorithm 1 respectively. The number of pulls according to the greedy rule can be estimated using the properties of subgaussian. Summing the regret caused by the greedy rule and the forced exploration (which can be denoted as $h_T(i)$) leads to this Theorem.

Next, we give more specific upper bounds for different input sequences.

**Corollary 1** (Constant sequence)**.** *Let $f(r) = \sqrt{T}$. We have $h_t(i) \in [\frac{t}{\sqrt{T}} - 1, \frac{t}{\sqrt{T}} + 1]$. Therefore,*

$$\mathbb{E}[k_T(i)] \leq \sqrt{T}(1 + 2m^2 e^{\frac{2}{m}}) + 1. \tag{6}$$

**Corollary 2** (Linear sequence)**.** *Let $f(r) = r$. We have,*

$$\mathbb{E}[k_T(i)] \leq \sqrt{2T} + K^2 + 6m^3 e^{\frac{3}{m}} \tag{7}$$

**Corollary 3** (Exponential sequence)**.** *Let $f(r) = a^r (a > 1)$. We have,*

$$\mathbb{E}[k_T(i)] \leq \frac{\log(T(a-1)+1)}{\log(a)} + (K+1)\frac{\log(K+1)}{\log(a)} + 2me^{\frac{1}{m}}\sum_{t=1}^{T}(1 + \frac{a-1}{K+1}t)^{-\frac{1}{m\log(a)}}. \tag{8}$$

**Lower Bounds** Many studies have demonstrated the lower bounds of regret for the stationary multi-armed bandit problem (Slivkins et al. 2019; Lai, Robbins et al. 1985; Bubeck, Cesa-Bianchi, and Lugosi 2013). A commonly used problem-dependent lower bound for bounded rewards or subgaussian rewards with variance parameter 1 is

$$\liminf_{T\to\infty} \frac{R_T^\pi}{\log(T)} \geq \sum_{i:\Delta(i)\neq 0} \frac{C_\mathcal{I}}{\Delta(i)}, \tag{9}$$

where $C_\mathcal{I}$ is a problem-dependent constant.

Define $f^{-1}(r)$ as $\min\{x : f(x) \geq r\}$. Let $t_0$ denote the number of all time steps satisfying $f(r) \leq K$, our method has a simple problem-dependent lower bound:

$$\max\{n : \sum_{r=f^{-1}(K+1)}^{n} f(r) \leq T - t_0\}. \tag{10}$$

Notethat, this lower bound only considers forced exploration and does not take into account the regret incurred by the greedy rule. The true regret lower bound is larger than the one computed by Equation 10.

If $f(r)$ is constant sequence ($f(r) = \sqrt{T}$) or linear sequence ($f(r) = r$), this problem-dependent lower bound is $\Omega(\sqrt{T})$. According to Corollary 1 and Corollary 2, the lower bound of constant sequence and linear sequence is on the same order of the upper bound.

If $f(r)$ is exponential sequence ($f(r) = a^r$), this lower bound is $\Omega(\frac{\log(T)}{\log(a)})$.

**Upper Bound of Exponential Sequence** From the lower bounds of the above three sequences, the regret upper bound of exponential sequence can be expected to reach the lower bound (Equation 9). However, it often fails to achieve this goal. Since there is lack of knowledge about $\sigma$ and $\Delta(i)$, we can't tune the parameter $a$ to obtain an optimal upper bound in Equation 8.

If there is no information about rewards distributions, there are two simple ways to set the exponential sequence:

- $a$ is a small constant independent of $T$. If the value of $m = \frac{8\sigma^2}{\Delta(i)^2}$ is appropriate such that
$$m\log(a) < 1, \tag{11}$$
we have
$$\mathbb{E}[k_T(i)] = O(me^{1/m}\log(T)).$$
This upper bound is optimal with respect to the order of $T$. However, in general, we cannot guarantee the parameter $a$ can satisfy Equation 11.
- $a$ is associated with $T$, such as $a = e^{\frac{1}{\log(T)}}$. Then, $f(r) = e^{\frac{r}{\log(T)}}$, we can get the asymptotic regret
$$\mathbb{E}[k_T(i)] = O(me^{\frac{1}{m}}(\log(T))^2). \tag{12}$$
Like the other two sequences, this problem-dependent asymptotic upper bound also matches the order of lower bound. Our upper bound is not optimal and there has been work to show that optimality is impossible. For example, recently Agrawal, Koolen, and Juneja (2021); Ashutosh et al. (2021) have proved the impossibility of a problem-dependent logarithmic regret for light-tailed distributions without further assumptions on the tail parameters.

**Remark 1.** We use an example to illustrate Corollary 3 and Equation 12. Assume that the reward distribution follows a Gaussian distribution with variance 1. If $T$ is sufficiently large, it holds that $m\log(a) = \frac{m}{\log(T)} < 1$. If $m > 1$, we get

$$\mathbb{E}[k_T(i)] \leq (K+2)(\log(T))^2 + e(K+1)\frac{16(\log(T))^2}{(\Delta(i))^2}. \tag{13}$$

If $m \leq 1$, the above equation holds obviously. We can derive the following regret upper bound:

$$\begin{aligned} R_T^\pi \leq\ & 8\sqrt{e}(K+1)\sqrt{T}\log(T) \\ & + (K+2)(\log(T))^2 \sum_{i=1}^{K} \Delta(i). \end{aligned} \tag{14}$$

The above regret bound matches the optimal upper bounds $\tilde{O}(\sqrt{T})$ in stationary multi-armed bandits. The extra terms $\log(T)$ and $K$ are due to the forced exploration. Note that, this regret upper bound also holds asymptotically.

## Non-Stationary Environments

In this section, we consider the piecewise-stationary settings. We employ a method often used in non-stationary bandits problems - sliding windows. One might think that all we need is to add sliding windows to the mean estimator $\hat{\mu}$. However, this does not work in non-stationary situations. Consider an input sequence $f(r)$ that can be incremented to $+\infty$. If $f(r) > T$ holds after some time step $t$, then between the time steps in $[t, T]$, the algorithm will not force to explore any arm but only pulls the arm based on the value of the mean estimator. If the reward distribution changes, the algorithm will suffer from very poor performance. To keep the exploration ability when the reward distribution changes, we propose to periodically reset the exploration sequence.

**Reset** $\{f(r)\}$

Define

$$\hat{\mu}_t(\tau, i) = \frac{1}{N_t(\tau, i)} \sum_{s=t-\tau+1}^{t} X_s(i)\mathbb{1}\{i_s = i\},$$

$$N_t(\tau, i) = \sum_{s=t-\tau+1}^{t} \mathbb{1}\{i_s = i\}.$$

$\hat{\mu}_t(\tau, i)$ denote the sliding window estimator, which using only the $\tau$ last pulls. We reset $r = 1$ every $\tau$ time steps to make the input sequence re-grow from $f(1)$. This simple reset strategy ensures that each arm is forced to be explored a certain number of times in each $\tau$ interval.

The size of this reset interval is set to $\tau$ for simplicity and ease of analysis. One can use intervals of other sizes. For

**Algorithm 2:** SW-FE

**Input**: non-decreasing sequence $\{f(r)\}$,sliding window $\tau$, $K$ arms, horizon $T$
**Initialization:** $t = 1, r = 0, f(0) = 0, \forall i \in \{1, ..., K\}, p(i) = 0, flag(i) = 0$

**while** $t < T$ **do**
  **if** $\forall i \in \{1, ..., K\}, p(i) < f(r)$ **then**
    Pull arm $i_t = \arg\max_i \hat{\mu}_t(\tau, i)$.
  **else**
    Pull arm $i_t = \arg\max_i p(i)$
  **end if**
  Update the estimate $\hat{\mu}_t(\tau, i_t)$,$p(i_t) = 0, flag(i_t) = 1$
  $p(i) = p(i) + 1$ for all unpulled arm $i$
  **if** $\forall i \in \{1, ..., K\}, flag(i) == 1$ **then**
    $r = r + 1$
    $\forall i \in \{1, ..., K\}, flag(i) = 0$
  **end if**
  **if** $t \bmod \tau == 0$ **then**
    $r = 1$
  **end if**
  $t = t + 1$
**end while**

our method, the selection of the reset interval should ensure that each arm can be forced to explore a certain number of times within the interval $[t - \tau + 1, t]$. Algorithm 2 shows the pseudocode of our method for non-stationary settings. Compared to Algorithm 1, only a sliding window is added to the estimator and $\{f(r)\}$ is reset periodically.

## Regret Analysis

In non-stationary setting, more regret is incurred compared to the stationary setting. This is the cost that must be paid to adapt to changes in the reward distributions. For our approach, the regret that arises more than the stationary environment comes from two aspects. The first is that due to the use of sliding window, the historical data used to estimate arm expectations are limited to at most $\tau$. The second, which is unique to our method, is that the number of times the suboptimal arm is pulled increases due to the periodically resetting of the exploration sequence.

Let $h_t(\tau, i)$ denote the number of forced pulls for arm $i$ in the $\tau$ last plays. Since we use the sliding window and a reset strategy, $h_t(\tau, i)$ will first increase and then change within a certain range. Let $\Delta_T(i) = \min\{\Delta_t(i) : i \neq i_t^*, t \leq T\}$, be the minimum difference between the expected reward of the best arm $i_t^*$ and the expected reward of arm $i$ in all times $T$ when arm $i$ is not the best arm.

**Theorem 2.** *Assume that the reward distribution is $\sigma$-subgaussian. Let $\{f(r)\}$ be an non-decreasing sequence,$\tau$ is the sliding window, for any suboptimal arm $i$,*

$$\mathbb{E}[k_T(i)] \leq \frac{T}{\tau}\left(h_\tau(\tau, i) + m_T e^{\frac{1}{m_T}} \sum_{t=1}^{\tau} e^{\frac{-h_t(\tau,i)}{m_T}}\right) + \frac{T}{\tau}(1 + 2m_T \log(\tau)) + B_T\tau, \tag{15}$$

*where $m_T = \frac{8\sigma^2}{\Delta_T(i)^2}$.*

**Proof sketch** The proof is adapted from the analysis of SW-UCB (Garivier and Moulines 2011). The regret comes from three aspects: greedy rules, forced exploration and the analysis methods of sliding windows. The forced exploration incurs regrets with upper bounds $\frac{T}{\tau} h_\tau(\tau, i)$. The analysis approach of the sliding window itself has a regret upper bound of $B_T\tau$. The regret caused by greedy rule can be estimated by regret decomposition. $\sum_{t=1}^{T} \mathbb{1}\{i_t = i \neq i_t^*, N_t(\tau, i) < A(\tau)\}$ can be bounded by $\lceil T/\tau \rceil A(\tau)$, $A(\tau) = 2m_T \log(\tau)$. We can decompose the regret in the following way:

$$\{i_t = i \neq i_t^*, N_t(\tau, i) > A(\tau)\} \subset$$
$$\{\mu_t(\tau, i) > \mu_t(i) + \frac{\Delta(i)}{2}, N_t(\tau, i) > A(\tau)\} \cup$$
$$\{\mu_t(\tau, i_t^*) \leq \mu_t(i_t^*) - \frac{\Delta(i)}{2}\}$$

The analysis methods of these two parts are similar to the stationary settings. Summing over all leads to this Theorem.

Similar to the stationary setting, we provide the specific bounds for some explore sequence. Our method requires that each arm can be forced to explore within $[t - \tau + 1, t]$. Constant sequences cannot be set to the same value as $\sqrt{T}$ in the stationary scenario. This could potentially result in a long time steps without exploring the optimal arm. Since the size of the sliding window is $\tau$, we set the constant sequence as $f(r) = \sqrt{\tau}$.

**Corollary 4.** *Let $f(r) = \sqrt{\tau}$. We have,*

$$\mathbb{E}[k_T(i)] \leq B_T\tau + \frac{T}{\tau}(1 + 2m_T \log(\tau) + \sqrt{\tau} m_T^2 e^{\frac{2}{m_T}}) + \frac{T}{\tau}\left(1 + \sqrt{\tau}\right) \tag{16}$$

**Corollary 5.** *Let $f(r) = r$. We have,*

$$\mathbb{E}[k_T(i)] \leq B_T\tau + \frac{T}{\tau}(1 + 2m_T \log(\tau) + 3m_T^3 e^{\frac{3}{m_T}}) + \frac{T}{\tau}\left(K^2 + \sqrt{2\tau}\right) \tag{17}$$

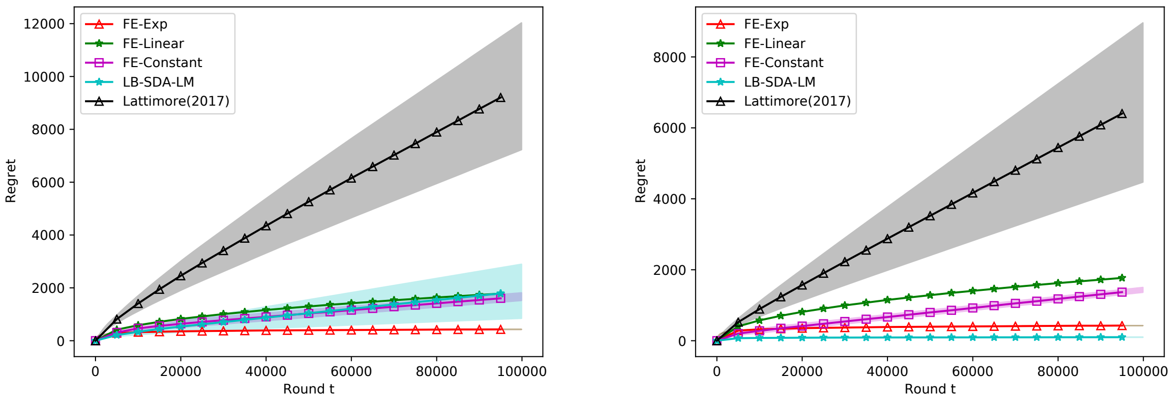
**Corollary 6.** *Let $f(r) = a^r (a > 1)$. We have,*

$$\mathbb{E}[k_T(i)] \leq B_T\tau + \frac{T}{\tau} m_T e^{\frac{1}{m_T}} \sum_{t=1}^{\tau} \left(1 + \frac{a-1}{K+1} t\right)^{-\frac{1}{m_T \log(a)}} + \frac{T}{\tau}\left(1 + 2m_T \log(\tau) + (K+2)\frac{\log(\tau+1)}{\log(a)}\right) \tag{18}$$

**Remark 2.** The sliding window methods (Garivier and Moulines 2011; Baudry, Russac, and Cappé 2021) generally suggest that the size of sliding window is $\tau = \sqrt{T \log(T)/B_T}$. For constant and linear sequence, we get

$$\mathbb{E}[k_T(i)] = O(T^{\frac{3}{4}} \sqrt{B_T \log(T)}).$$

For exponential sequence, we can take $a = e^{\frac{1}{\log(\tau)}}$ similar to stationary settings. It can be observed that setting the sliding




(a) (b)

Figure 1: Settings with $K = 10, T = 100000$. Gaussian rewards (a), Bernoulli rewards (b).

window to $\tau = \sqrt{T/B_T}\log(T)$ for exponential sequence will yield a smaller upper bound, effectively reducing it by $\sqrt{\log(T)}$. If the number of breakpoints is constant, we have the following asymptotic bound

$$\mathbb{E}[k_T(i)] = O(\sqrt{TB_T}\log(T)). \tag{19}$$

# Experiments

## Stationary Settings

In this section, we compare our method with other non-parametric bandit algorithms on Gaussian and Bernoulli distribution rewards[1]. Our method is instantiated by three different sequence: FE-Constant, FE-Linear, FE-Exp. They use constant ($f(r) = \sqrt{T}$), linear ($f(r) = r$), and exponential ($f(r) = e^{\frac{r}{\log(T)}}$) sequences, respectively. We compare the above three instances of our method with two representative non-parametric algorithms: LB-SDA-LM and Lattimore(2017). Due to the potentially high time complexity of LB-SDA algorithm, we turn to comparing LB-SDA-LM, an alternative algorithm that achieves the same theoretical results but with much lower complexity. The implementation of Lattimore(2017) seems challenging, we use a roughly equivalent and efficiently computable alternative (Lattimore 2017).

The means and variances of Gaussian distributions are randomly generated from uniform distribution:

$$\mu(i) \sim U(0, 1),$$
$$\sigma(i) \sim U(0, 1).$$

The means of Bernoulli distribution are also generated from $U(0, 1)$. The time horizon is set as $T = 100000$. We fix the number of arms as $K = 10$. We measure the performance of each algorithm with the cumulative expected regret defined in Equation 1. The expected regret is averaged on 100 independently runs. The 95% confidence interval is obtained by performing 100 independent runs and is shown as a semi-transparent region in the figure.

Figure 1 shows the results of Gaussian and Bernoulli rewards. Constant and linear sequences exhibit similar performance. The implementation of Lattimore(2017) using the approximation method may lead to significant variance in experimental results, and its performance could be inferior to other methods. LB-SDA-LM is applicable to single-parameter exponential family distributions. This method demonstrates optimal performance on Bernoulli distributions. However, its performance is notably weaker on Gaussian distributions with unknown variance in $(0, 1)$. Our approach, FE-EXP, although its theoretical upper bound is asymptotic and not optimal, achieves remarkable performance on Gaussian and Bernoulli rewards.

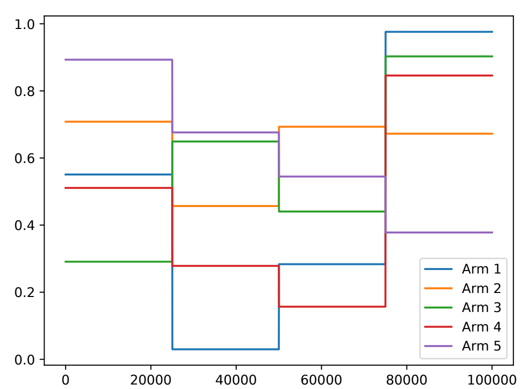
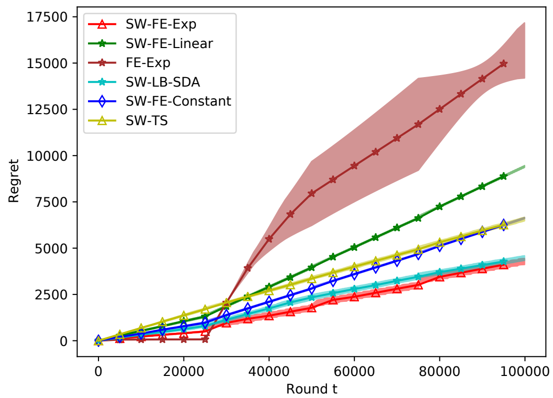



Figure 2: $K = 5$,$B_T = 5$ for Gaussian rewards.

## Non-stationary Settings

In this section, we compare our method with other non-stationary bandit algorithms. Specifically, our method em-

[1]https://github.com/qh1874/Force_Explor



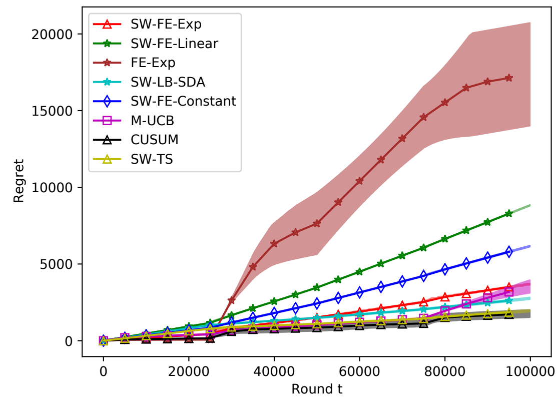




Figure 3: Settings with $K = 5, B_T = 5, T = 100000$. Gaussian rewards (a), Bernoulli rewards (b).

ploys four instances: constant sequence with sliding window (SW-FE-Constant), linear sequence with sliding window (SW-FE-Linear), exponential sequence with sliding window (SW-FE-EXP), and the exponential sequence without sliding window (FE-EXP). We use FE-EXP to evaluate the improvement obtained thanks to the employment of the sliding window. We also compare our method with another non-parametric algorithm named SW-LB-SDA (Baudry, Russac, and Cappé 2021). Furthermore, we compare with some novel and efficient algorithms such as CUSUM (Liu, Lee, and Shroff 2018), M-UCB (Cao et al. 2019) only in Bernoulli distribution rewards. Moreover, we compare with SW-TS (Trovo et al. 2020). This method requires information about the Gaussian rewards to be known in advance. There is no theoretical proof yet for SW-TS except for Bernoulli rewards.

**Tune parameters** Following Remark 2, we set $\tau = \sqrt{T/B_T}\log(T)$ for SW-FE-Exp, $\tau = \sqrt{T\log(T)/B_T}$ for SW-FE-Constant and SW-FE-Linear. We set $\tau = \sqrt{T\log(T)/B_T}$ for LB-SDA and SW-TS. For change-point detection algorithm M-UCB, we set $w = 800, b = \sqrt{w/2\log(2KT^2)}$ suggested by (Cao et al. 2019). But set the amount of exploration $\gamma = \sqrt{KB_T\log(T)/T}$. In practice, it has been found that using this value instead of the one guaranteed in (Cao et al. 2019) will improve empirical performance (Baudry, Russac, and Cappé 2021). For CUSUM, following from (Liu, Lee, and Shroff 2018), we set $\alpha = \sqrt{B_T/T\log(T/B_T)}$ and $h = \log(T/B_T)$. For our experiment settings, we choose $M = 50, \epsilon = 0.05$.

The time horizon is set as $T = 100000$. We split the time horizon into 5 phases of equal length and fix the number of arms to $K = 5$. Each stationary phase, the reward distributions will be regenerated in the same way as stationary settings. Figure 2 depicts the expected rewards for Gaussian arms with $K = 5$ and $B_T = 5$. Gaussian and Bernoulli distributions are generated in the same way as in the stationary setting. The expected regret is averaged on 10 independently runs. The 95% confidence interval is obtained by performing 10 independent runs and is shown as a semi-transparent region in the figure.

Figure 3 shows the results of Gaussian and Bernoulli rewards for piecewise-stationary settings. M-UCB and CUMSUM require that the rewards are bounded, which is not applicable to Gaussian rewards. We only conducted experiments on Bernoulli rewards. FE-EXP is an algorithm for stationary MAB problems, so it oscillates a lot at the breakpoint. SW-FE-Constant and SW-FE-Linear have similar performance, with SW-FE-Constant even performing better. This could be attributed to the significant impact of problem-dependent term $m_T = \frac{8\sigma^2}{\Delta_T(i)^2}$ on the performance. The regret upper bound of SW-FE-Constant is controlled by $m_T^2$, while that of SW-FE-Linear is controlled by $m_T^3$. Our algorithm, SW-FE-EXP, exhibits competitive results on both Gaussian and Bernoulli rewards.

## Conclusion

In this paper, we have developed a forced exploration algorithm for both stationary and non-stationary multi-armed bandit problems. This algorithm has broad applicability to various reward distributions, and its implementation does not require the use of reward distribution information. We employ a unified analytical approach for different input sequences and provide regret upper bounds. Experimental results demonstrate that despite the asymptotic nature of our regret upper bounds, our approach achieves comparable performance to current popular algorithms.

## References

Agrawal, S.; Koolen, W. M.; and Juneja, S. 2021. Optimal best-arm identification methods for tail-risk measures. *Advances in Neural Information Processing Systems*, 34: 25578–25590.

Ashutosh, K.; Nair, J.; Kagrecha, A.; and Jagannathan, K. 2021. Bandit algorithms: Letting go of logarithmic regret for statistical robustness. In *International Conference on Artificial Intelligence and Statistics*, 622–630. PMLR.

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

Auer, P.; Gajane, P.; and Ortner, R. 2019. Adaptively tracking the best bandit arm with an unknown number of distribution changes. In *Conference on Learning Theory*, 138–158. PMLR.

Baransi, A.; Maillard, O.-A.; and Mannor, S. 2014. Subsampling for multi-armed bandits. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2014, Nancy, France, September 15-19, 2014. Proceedings, Part I 14*, 115–131. Springer.

Baudry, D.; Russac, Y.; and Cappé, O. 2021. On limited-memory subsampling strategies for bandits. In *International Conference on Machine Learning*, 727–737. PMLR.

Besson, L.; Kaufmann, E.; Maillard, O.-A.; and Seznec, J. 2022. Efficient change-point detection for tackling piecewise-stationary bandits. *The Journal of Machine Learning Research*, 23(1): 3337–3376.

Bubeck, S.; Cesa-Bianchi, N.; and Lugosi, G. 2013. Bandits with heavy tail. *IEEE Transactions on Information Theory*, 59(11): 7711–7717.

Cao, Y.; Wen, Z.; Kveton, B.; and Xie, Y. 2019. Nearly optimal adaptive procedure with change detection for piecewise-stationary bandit. In *The 22nd International Conference on Artificial Intelligence and Statistics*, 418–427. PMLR.

Chan, H. P. 2020. The multi-armed bandit problem: An efficient nonparametric solution.

Chen, Y.; Lee, C.-W.; Luo, H.; and Wei, C.-Y. 2019. A new algorithm for non-stationary contextual bandits: Efficient, optimal and parameter-free. In *Conference on Learning Theory*, 696–726. PMLR.

Garivier, A.; Lattimore, T.; and Kaufmann, E. 2016. On explore-then-commit strategies. *Advances in Neural Information Processing Systems*, 29.

Garivier, A.; and Moulines, E. 2011. On upper-confidence bound policies for switching bandit problems. In *International Conference on Algorithmic Learning Theory*, 174–188. Springer.

Gorre, M. E.; Mohammed, M.; Ellwood, K.; Hsu, N.; Paquette, R.; Rao, P. N.; and Sawyers, C. L. 2001. Clinical resistance to STI-571 cancer therapy caused by BCR-ABL gene mutation or amplification. *Science*, 293(5531): 876–880.

Kveton, B.; Szepesvari, C.; Ghavamzadeh, M.; and Boutilier, C. 2019a. Perturbed-history exploration in stochastic multi-armed bandits. *arXiv preprint arXiv:1902.10089*.

Kveton, B.; Szepesvari, C.; Vaswani, S.; Wen, Z.; Lattimore, T.; and Ghavamzadeh, M. 2019b. Garbage in, reward out: Bootstrapping exploration in multi-armed bandits. In *International Conference on Machine Learning*, 3601–3610. PMLR.

Lai, T. L.; Robbins, H.; et al. 1985. Asymptotically efficient adaptive allocation rules. *Advances in applied mathematics*, 6(1): 4–22.

Lattimore, T. 2017. A scale free algorithm for stochastic bandits with bounded kurtosis. *Advances in Neural Information Processing Systems*, 30.

Li, L.; Chu, W.; Langford, J.; and Wang, X. 2011. Unbiased offline evaluation of contextual-bandit-based news article recommendation algorithms. In *Proceedings of the fourth ACM international conference on Web search and data mining*, 297–306.

Li, S.; Karatzoglou, A.; and Gentile, C. 2016. Collaborative filtering bandits. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, 539–548.

Liu, F.; Lee, J.; and Shroff, N. 2018. A change-detection based framework for piecewise-stationary multi-armed bandit problem. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Liu, K.; Chen, H.; Deng, W.; and Wu, T. 2022. The Extended UCB Policies for Frequentist Multi-armed Bandit Problems. arXiv:1112.1768.

Raj, V.; and Kalyani, S. 2017. Taming non-stationary bandits: A Bayesian approach. *arXiv preprint arXiv:1707.09727*.

Riou, C.; and Honda, J. 2020. Bandit algorithms based on thompson sampling for bounded reward distributions. In *Algorithmic Learning Theory*, 777–826. PMLR.

Schwartz, E. M.; Bradlow, E. T.; and Fader, P. S. 2017. Customer acquisition via display advertising using multi-armed bandit experiments. *Marketing Science*, 36(4): 500–522.

Slivkins, A.; et al. 2019. Introduction to multi-armed bandits. *Foundations and Trends® in Machine Learning*, 12(1-2): 1–286.

Thompson, W. R. 1933. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3-4): 285–294.

Trovo, F.; Paladino, S.; Restelli, M.; and Gatti, N. 2020. Sliding-window thompson sampling for non-stationary settings. *Journal of Artificial Intelligence Research*, 68: 311–364.

Vermorel, J.; and Mohri, M. 2005. Multi-armed bandit algorithms and empirical evaluation. In *Machine Learning: ECML 2005: 16th European Conference on Machine Learning, Porto, Portugal, October 3-7, 2005. Proceedings 16*, 437–448. Springer.

Wang, C.-H.; Yu, Y.; Hao, B.; and Cheng, G. 2020. Residual bootstrap exploration for bandit algorithms. *arXiv preprint arXiv:2002.08436*.

Wu, Q.; Iyer, N.; and Wang, H. 2018. Learning contextual bandits in a non-stationary environment. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, 495–504.