Adversarial Blocking Bandits

Nicholas Bishop
University of Southampton, UK
nb8g13@soton.ac.uk

Hau Chan
University of Nebraska-Lincoln, USA
hchan3@unl.edu

Debmalya Mandal
Columbia University, USA
dm3557@columbia.edu

Long Tran-Thanh
University of Warwick, UK
long.tran-thanh@warwick.ac.uk

Abstract

We consider a general adversarial multi-armed blocking bandit setting where each played arm can be *blocked* (unavailable) for some time periods and the reward per arm is given at each time period *adversarially* without obeying any distribution. The setting models scenarios of allocating scarce limited supplies (e.g., arms) where the supplies replenish and can be reused only after certain time periods. We first show that, in the optimization setting, when the blocking durations and rewards are known in advance, finding an optimal policy (e.g., determining which arm per round) that maximises the cumulative reward is strongly NP-hard, eliminating the possibility of a fully polynomial-time approximation scheme (FPTAS) for the problem unless P = NP. To complement our result, we show that a greedy algorithm that plays the best available arm at each round provides an approximation guarantee that depends on the blocking durations and the path variance of the rewards. In the bandit setting, when the blocking durations and rewards are not known, we design two algorithms, `RGA` and `RGA-META`, for the case of bounded duration an path variation. In particular, when the variation budget B_T is known in advance, `RGA` can achieve $\mathcal{O}(\sqrt{T(2\tilde{D}+K)B_T})$ dynamic approximate regret. On the other hand, when B_T is not known, we show that the dynamic approximate regret of `RGA-META` is at most $\mathcal{O}((K+\tilde{D})^{1/4}\tilde{B}^{1/2}T^{3/4})$ where $\tilde{B}$ is the maximal path variation budget within each batch of `RGA-META` (which is provably in order of $o(\sqrt{T})$. We also prove that if either the variation budget or the maximal blocking duration is unbounded, the approximate regret will be at least $\Theta(T)$. We also show that the regret upper bound of `RGA` is tight if the blocking durations are bounded above by an order of $\mathcal{O}(1)$.

1 Introduction

This paper investigates the blocking bandit model where pulling an arm results in having that arm blocked for a deterministic number of rounds. For example, consider the classical problem of online task allocation, in which new task requests arrive at each time step, waiting to be assigned to one of many servers [Karthik et al., 2017]. Once a server is allocated to a task, it starts working on it, and becomes unavailable for future tasks until that task is done. If there are no servers available or none is allocated to the task at its arrival, the request will not be served and leave the system forever. A more recent example comes from the domain of expert crowdsourcing (e.g., Upwork, Outsourcely, etc.). In this setting, a job requester can sequentially choose from a pool of workers and allocate a short-term job/project to the worker [Ho and Vaughan, 2012, Tran-Thanh et al., 2014]. The stochastic version of this problem, where the rewards are randomly drawn from a distribution in an

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

i.i.d. manner, with the constraint that the blocking durations are fixed per arm over time, has been studied in [Basu et al., 2019] and [Basu et al., 2020]. However, in many applications, the stochastic setting is too restrictive and not realistic. For example, in the online task allocation problem, the tasks can be heterogeneous, and both the value and the serving time of the tasks can vary over time in an arbitrary manner. Furthermore, in the expert crowdsourcing setting, the time and quality workers need to deliver the job are unknown in advance, can vary over time, and do not necessarily follow an i.i.d. stochastic process. These examples demonstrate that for many real-world situations, the stochastic blocking bandit model is not an appropriate choice.

To overcome this issue, in this paper we propose the adversarial blocking bandit setting, where both the sequence of rewards and blocking durations per arm can be arbitrary. While the literature of adversarial bandits is enormous, to the best of our knowledge, this is the first attempt to address the effect of blocking in adversarial models. In particular, we are interested in a setting where the rewards are neither sampled i.i.d., nor maliciously chosen in an arbitrary way. Instead, in many real-world systems, the change in the value of rewards is rather slow or smooth over time (e.g., in the online task allocation problem, similar tasks usually arrive in batch, or in the crowdsourcing system, workers may have periods when they perform consistently, and thus, their performance slowly varies over time). To capture this, we assume that there is a path variation budget which controls the change of the rewards over time [1].

1.1 Main Contributions

In this paper, apart from the adversarial blocking bandit setting, we also investigate two additional versions of the model: (i) The offline MAXREWARD problem, where all the rewards and blocking durations are known in advance; and (ii) the online version of MAXREWARD, in which we see the corresponding rewards and blocking durations of the arms at each time step *before* we choose an arm to pull. Our main findings can be summarised as follows:

1. We prove that the offline MAXREWARD problem is strongly NP-hard (Theorem 1). Note that this result is stronger than the computational hardness result in Basu et al. [2019], which depends on the correctness of the randomised exponential time hypothesis.

2. We devise a provable approximation ratio for a simple online greedy algorithm, `Greedy-BAA`, for the online MAXREWARD problem (Theorem 2). Our approximation ratio, when applied to the stochastic blocking bandit model with fixed blocking durations, is slightly weaker than that of Basu et al. [2019]. However, it is more generic, as it can be applied to any arbitrary sequence of rewards and blocking durations.

3. For the bandit setting, we consider the case when both the maximal blocking duration and the path variance are bounded, and propose two bandit algorithms:

- We design `RGA` for the case of known path variation budget B_T. In particular, we show that `RGA` can provably achieve $\mathcal{O}\left(\sqrt{T(2\tilde{D}+K)B_T}\right)$ regret, where T is the time horizon, K is the number of arms, $\tilde{D}$ is the maximum blocking duration, and the regret is computed against the performance of `Greedy-BAA` (Theorem 3).

- For the case of unknown path variation budget B_T, we propose `RGA-META` that uses Exp3 as a meta-bandit algorithm to learn an appropriate path variation budget and runs `RGA` with it. We prove that `RGA-META` achieves $\mathcal{O}((K+\tilde{D})^{1/4}\tilde{B}^{1/2}T^{3/4})$ regret bound where $\tilde{B}$ is the maximal path variance within a single batch of the algorithm, and is in order of $\mathcal{O}(\sqrt{T})$ in the worst case (Theorem 4).

4. Finally, we also discuss a number of regret lower bound results. In particular, we show that if either B_T or $\tilde{D}$ is in $\Theta(T)$ (or unbounded), then the regret is at least $\Theta(T)$ (Claims 1 and 2). We also discuss that if $\tilde{D} \in \mathcal{O}(1)$, then there is a matching lower bound for the regret of `RGA` (Section 5).

[1] We will show in Section 5 that bounded variation budgets are necessary to achieve sub-linear regrets.

1.2 Related Work

Stochastic Blocking Bandits. The most relevant work to our setting is the stochastic blocking bandit model. As mentioned before, Basu et al. [2019] introduce and study this model where the reward per each time period is generated from a stochastic distribution with mean μ_k reward for each arm k and the blocking duration is fixed across all time period for each arm k (e.g., $D_t^k = D^k$ for all t and k). In the optimization setting where the mean rewards and blocking durations are known, they consider a simpler version of the MAXREWARD problem for their setting and show that the problem is as hard as the PINWHEEL Scheduling on dense instances [Jacobs and Longo, 2014] and provide that a simple greedy algorithm (see Algorithm 1) achieves an approximation ratio of $(1 - 1/e - O(1/T))$ where T is total time period. In the bandit setting, they provide lower and upper regret bounds that depend on the number of arms, mean rewards, and $\log(T)$. A very recent work [Basu et al., 2020] extends the stochastic blocking bandit to a contextual setting where a context is sampled according to a distribution each time period and the reward per arm is drawn from a distribution with the mean depending on the pulled arm and the given context. Similar to the work of Basu et al. [2019], Basu et al. [2020] derive an online algorithm with an approximation ratio that depends on the maximum blocking durations and provide upper and lower α-regret bounds of $O(\log T)$ and $\Omega(\log T)$, respectively. However, the results from this models cannot be directly applied to the adversarial setting due to the differences between the stochastic and adversarial reward generation schemes.

Budgeted and Knapsack Bandits. Since the underlying offline optimisation problem of our setting, MAXREWARD, can also be casted as an instance of the multiple-choice multidimensional knapsack problem, it is also worth mentioning the line of work in the bandit literature that solve online knapsack problems with bandit feedback. In these models, the pull of an arm requires the consumption of resources in $d \geq 1$ dimensions. The resource per arm is given either stochastic or adversarially in each time period and a (non replenishable) total budget $B = (B_1, ..., B_d)$ is available at the initial time period. The one-dimensional stochastic version of this setting is first studied in Tran-Thanh et al. [2010, 2012], Ding et al. [2013] under the name budgeted bandits, and is later extended to multiple dimensions (a.k.a. bandits with knapsack) by Badanidiyuru et al. [2013], Agrawal and Devanur [2014], Badanidiyuru et al. [2014]. More recently, Rangi et al. [2019] and Immorlica et al. [2019] initiate the study of adversarial knapsack bandits. Rangi et al. [2019] consider the $d = 1$ setting with a regret benchmark that is measured based on the best fixed-arm's reward to cost ratio. Under such a regret benchmark, they show that sub-linear regret (with respect to B and k) is possible in both the stochastic and adversarial settings. Immorlica et al. [2019] consider the $d \geq 1$ setting with a regret benchmark that is defined to be the ratio of the expected reward of the best fixed distribution over arms and the policy's expected reward. show that the ratio is at least $\Omega(\log T)$. However, none of the techniques developed in these work can be applied to our setting, due to the following reason: The results in the knapsack bandit models typically assume that the pulling costs are bounded above by a constant, and the budget is significantly larger than this constant to allow sufficient exploration. In contrast, when MAXREWARD is conversed into a knapsack model, many of its dimensions will have a budget of 1, and the corresponding pulling cost for that dimension is also 1 (due to the blocking condition).

Other Settings with Arm Availability Constraints. Other bandit models with arm availability cosntrainsts include the mortal bandits [Chakrabarti et al., 2009], sleeping bandits [Kleinberg et al., 2010, Kale et al., 2016], bandits with stochastic action sets [Neu and Valko, 2014], and combinatorial semi-bandits [Neu and Bartók, 2016]. We refer readers to [Basu et al., 2019] for a discussion of these models, including the relevance of the blocking bandit setting to online Markov decision processes.

Connection to the scheduling literature. Notice that there is a strong connection between MAXREWARD and the interval scheduling problems. In particular, the MAXREWARD problem belongs to the class of fixed interval scheduling problems with arbitrary weight values, no preemption, and machine dependent processing time (see e.g., Kolen et al. [2007] for a comprehensive survey). This is one of the most general, and thus, hardest versions of the fixed interval scheduling literature (see, e.g., Kovalyov et al. [2007] for more details). In particular, MAXREWARD is a special case of this setting where for each task, the starting point of the feasible processing interval is equal to the arrival time. Note that to date, provable performance guarantees for fixed interval scheduling problems with arbitrary weight values only exist in offline, online but preemptive, or settings with some special uniformity assumptions (e.g., [Erlebach and Spieksma, 2000, Miyazawa and Erlebach, 2004, Bender et al., 2017, Yu and Jacobson, 2020]). Therefore, to our best knowledge, Theorem 2 in our paper is

the first result which provides provable approximation ratio for a deterministic algorithm in an online non-preemptive setting. Note that with some modifications, our proof can also be extended to the general online non-preemptive setting, i.e., online interval scheduling with arbitrary weight values, no preemption, and machine dependent processing time.

2 Preliminaries

Adversarial blocking bandit. In this paper we consider the following bandit setting. Let $\mathcal{K} = \{1, \dots, K\}$ be the set of K arms. Let $\mathcal{T} = \{1, \dots, T\}$ denote a sequence of T time steps, or decision points faced by a decision maker. At every time step $t \in \mathcal{T}$, the decision maker may pull one of the K arms. When pulling an arm $k \in \mathcal{K}$ at time step $t \in \mathcal{T}$, the reward $X_t^k \in [0, 1]$ is obtained. In addition, the pulled arm k is deterministically blocked and cannot be pulled for the next $(D_t^k - 1)$ time steps for some integer blocking duration $D_t^k \in \mathbb{Z}^+$. We also use the notation $\emptyset$ to denote the action of not pulling an arm. In which case, $X_t^\emptyset = 0$ and $D_t^\emptyset = 1$ for each time step t.

We denote by X^k the sequence of rewards over T time steps associated with an arm $k \in \mathcal{K}$ such that $X^k = \{X_t^k\}_{t=1}^T$. In addition, we denote by X the sequence of vectors of all K rewards such that $X = \{X^k\}_{k=1}^K$. Similarly, we denote by $D^k = \{D_t^k\}_{t=1}^T$ the sequence of blocking durations over T time steps associated with an arm k and denote by $D = \{D^k\}_{k=1}^K$ the sequence of vectors of all K blocking duration vectors.

In our model, the rewards and blocking durations of each arm can change an arbitrary number of times. We let $\tilde{D}$ ($\underset{\sim}{D}$) be the *maximal blocking duration* (*minimal blocking duration*) which is the upper bound (lower bound) of the largest (smallest) possible blocking duration. We denote by $\mathcal{D} = \{1, \dots, \tilde{D}\}^{K \times T}$ the set of all blocking duration vector sequences which are upper bounded by $\tilde{D}$. Note that D is defined with respect to minimal blocking duration $\underset{\sim}{D} = 1$. It is sometime be useful to define D for arbitrarily lower bound $\underset{\sim}{D}$.

Bounded path variation. Motivated by and adapted from a recent line of work in the bandit literature (e.g., [Besbes et al., 2014]), we assume that there is a *path variation budget* on the sequence of the rewards. In particular, the definition of path variation on the sequence of the rewards is defined to be

$$\sum_{t=1}^{T-1} \sum_{k=1}^{K} \left| X_{t+1}^k - X_t^k \right| .$$

We refer to B_T as the path variation budget over $\mathcal{T}$. We define the corresponding temporal uncertainty set as the set of reward vector sequences which satisfy the variation budget over the set of time steps $\{1, \dots, T\}$:

$$\mathcal{B} = \left\{ X \in [0, 1]^{K \times T} \ : \ \sum_{t=1}^{T-1} \sum_{k=1}^{K} \left| X_t^k - X_{t+1}^k \right| \leq B_T \right\}$$

Note that by setting $B_T = KT$ we can recover the standard unbounded version of our bandit model (as all the rewards are from $[0, 1]$). Note that our analysis also works for other variation budgets such as the maximum variation [Besbes et al., 2014] or the number of changes budgets [Auer et al., 2019]. See Section 5 for a more detailed discussion.

Arm pulling policy. Let U be a random variable defined over a probability space $(\mathbb{U}, \mathcal{U}, \mathbf{P}_u)$ Let $\pi_1 : \mathbb{U} \to \mathcal{K}$ and $\pi_t : [0, 1]^{t-1} \times \{1, \dots, \tilde{D}\}^{t-1} \times \mathbb{U} \to \mathcal{K}$ for $t = 2, 3, \dots$ be measurable functions With some abuse of notation we denote by $\pi_t \in \mathcal{K}$ the arm chosen at time t, that is given by

$$\pi_t = \begin{cases} \pi_1(U) & t = 1 \\ \pi_t(X_{t-1}^\pi, \dots, X_1^\pi, D_{t-1}^\pi, \dots, D_1^\pi, U) & t = 2, 3, \dots \end{cases}$$

Here X_t^π (resp. D_t^π) denotes the reward (resp. blocking duration) observed by the policy π at time t The mappings $\{\pi_t \ : \ t = 1, \dots, T\}$ together with the distribution $\mathbf{P}_u$ define the class of policies We define the class $\mathcal{P}$ of admissible policies to be those, at every time step, which choose an action which is not blocked. That is,

$$\mathcal{P} = \left\{ (\pi_1, \dots, \pi_T) \ : \ \pi_t \notin \{\pi_j : j + D_j^{\pi_j} - 1 \geq t, \ \forall j \leq t - 1\}, \ \forall t \in \{1, \dots, T\}, X \in \mathcal{B}, D \in \mathcal{D} \right\} .$$

In addition, let $A_t(\pi_1, \ldots, \pi_{t-1}) = \mathcal{K} \setminus \{\pi_j : j + D_j^{\pi_j} - 1 \geq t, \ \forall j \leq t-1\}$ denote the set of available arms at time step t (we will also use A_t for the sake of brevity).

Objective. The cumulative reward of a policy $\pi \in \mathcal{P}$ is defined to be $r(\pi) = \sum_{t=1}^{T} X_t^{\pi}$ where X_t^{π} is the reward obtained by policy π at time step t. Our objective is to find $\pi^* \in \mathcal{P}$ such that $\pi^* \in \arg\max_{\pi \in \mathcal{P}} \mathbb{E}^{\pi}[r(\pi)]$, where the expectation is over all possible randomisation coming from policy π.

Feedback. The difficulty of the optimisation problem depends on the information (or the feedback) we have about the rewards and blocking durations of the arms. In this paper, we consider three feedback models in increasing order of difficulty. In the simplest setting, we know the value of all X_t^k and D_t^k in advance. We refer to this setting as the (offline) MAXREWARD optimization problem. In the online version of MAXREWARD, we assume that X_t^k and D_t^k are not known in advance, but at each time step t, the value of X_t^k and D_t^k for all k at that particular time step t is revealed before we choose any arm to pull. Finally, in the (classical) bandit setting, we assume that only the reward and blocking duration of the chosen arms are revealed after that arm is pulled[2]. We will refer to third model as the *adversarial blocking bandit problem*.

3 The Offline and Online MAXREWARD Problems

We start with the analysis of the offline and online MAXREWARD problems. As a slight preview of the next subsections, computing an optimal solution of the offline MAXREWARD problem is strongly NP-hard even with bounded variation budget. Such result eliminates the possibility of a fully polynomial-time approximation scheme (FPTAS) for the problem unless P = NP. In addition, for the online MAXREWARD problem, we design an online greedy algorithm with provable approximation guarantee.

3.1 The Computational Complexity of the Offline MAXREWARD Problem

To show that the MAXREWARD problem is strongly NP-hard, we reduce from the Boolean satisfiability problem with three literals per clause (3-SAT), which is known to be strongly NP-complete [Garey and Johnson, 1979]. In a 3-SAT instance, we are given m variables and n clauses. Each clause consists of three literals, and each literal is either a variable or the negation of the variable. The problem is to determine if there is a boolean true/false assignment to each variable so that the given 3-SAT instance is true (i.e., each clause contains at least one true literal).

Theorem 1. *Computing an optimal solution for the MAXREWARD problem is strongly NP-hard. The hardness result holds even when the path variation is bounded.*

3.2 Online MAXREWARD Problem with Bounded Variation Budget

In this section, we consider the online version of MAXREWARD. We devise a simple online greedy algorithm, Greedy Best Available Arm (`Greedy-BAA`), in which, at each time step, the algorithm plays an available arm with the highest reward. Algorithm 1 provides a detail description of `Greedy-BAA`.

Below, we show that `Greedy-BAA` provides an approximation guarantee to the offline MAXREWARD problem that depends on the blocking durations and the variation budget.

Theorem 2. *Let $k^* = \arg\max_k \frac{D_{\max}^k}{D_{\min}^k}$ denote the arm with the highest max-min blocking duration ratio. Let π^+ denote the solution returned by `Greedy-BAA`, and π^* denote an optimal solution of the offline MAXREWARD problem, respectively. We state that:*

$$\left(1 + \frac{D_{\max}^{k^*}}{D_{\min}^{k^*}}\right) r(\pi^+) + \frac{D_{\max}^{k^*}}{D_{\min}^{k^*}} B_T \geq r(\pi^*),$$

That is, `Greedy-BAA` has an approximation ratio of $\left(1 + \frac{D_{\max}^{k^*}}{D_{\min}^{k^*}}\right)^{-1} \left(1 - \frac{D_{\max}^{k^*} B_T}{D_{\min}^{k^*} r(\pi^*)}\right)$.

[2]In this paper, due to space limits, we do not deal with the full information feedback model, in which the reward and blocking duration values of all the arms are revealed at each time step after the pull.

Algorithm 1: `Greedy-BAA`

Input : T, K, $\{X_t^k\}_{k\in\mathcal{K},t\in\mathcal{T}}$, $\{D_t^k\}_{k\in\mathcal{K},t\in\mathcal{T}}$ - An instance of the MAXREWARD Problem
Output : $\pi^+ = (\pi_1^+, \pi_2^+, ..., \pi_T^+) \in \mathcal{P}$ - A greedy solution to the MAXREWARD Problem

$\pi^+ = (\emptyset, ..., \emptyset)$;
for $j \leftarrow 1$ **to** T **do**
  Select $\pi_j^+ \in \arg\max_{k_j \in A_j(\pi_1^+,...,\pi_{j-1}^+)\cup\emptyset} X_j^{k_j}$
    # See the preliminary section for definitions
end
return π^+

Note that as $D_{\min}^{k^*} \geq \underset{\sim}{D}$ and $D_{\max}^{k^*} \leq \tilde{D}$, the approximation ratio above can be further bounded above by $\left(1 + \frac{\tilde{D}}{\underset{\sim}{D}}\right)^{-1}\left(1 - \frac{\tilde{D}B_T}{\underset{\sim}{D}r(\pi^*)}\right)$.

Comparison to the result of Basu et al. [2019]. We note that Basu et al. [2019] has studied the MAXREWARD problem with path variation budget $B_T = 0$ (i.e., the reward values are fixed over time) and homogeneous blocking durations per arm (i.e., when the blocking duration per arm do not change over time). In that case, our proof provides an approximation ratio of $1/2$ whereas Basu et al. [2019] provides an approximation ratio of $O(1 - 1/e - O(1/T))$. Their technique uses a much complicated LP-bounding technique/proof that does not directly generalize to the case of $B_T > 0$ with varying blocking durations. On the other hand, our approximation ratio result holds for the general case. For example, if B_T grows slower than $r(\pi^+)$ with T, our algorithm guarantees an approximation ratio of $(1 + 2\frac{\tilde{D}}{\underset{\sim}{D}})^{-1}$.

4 The Adversarial Blocking Bandit Problem

Given the investigation of the (offline and online) MAXREWARD problems in the previous section, we now turn to the main focus of our paper, namely the online MAXREWARD problem with bandit feedback, a.k.a the adversarial blocking bandit problem. While the regret analyses are typically done by benchmarking against the best fixed policy in hindsight, we can easily show that in our setting, this benchmark would perform arbitrarily poorly, compared to the offline optimal solution. Therefore, instead of following the standard regret analysis, we are interested in comparing the performance of the designed algorithms to that of the offline optimal solution. Therefore, we will use the following regret definition:

Dynamic approximate regret. We compare the performance of a policy with respect to the dynamic oracle algorithm that returns the offline optimal solution of MAXREWARD .We define the α-regret under a policy $\pi \in \mathcal{P}$ as the worst case difference between an (offline) α-optimal sequence of actions and the expected performance under policy π. More precisely, let π^* denote the arm pulling policy of that dynamic oracle algorithm. The α-regret of a policy $\pi \in \mathcal{P}$ against π^* is defined to be

$$\mathcal{R}_\pi^\alpha(B_T, \tilde{D}, T) = \alpha r(\pi^*) - \mathbb{E}[r(\pi)]$$

where the expectation is over all the possible randomisation of π. Note that this regret notion is stronger than the regret against the best fixed policy in hindsight, as it is easy to show that the best fixed policy can perform arbitrarily badly, compared to π^*.

4.1 Blocking Bandit with Known Path Variation Budget

We now turn to describe our new bandit algorithm, `RGA`, designed for the adversarial blocking bandit problem. This algorithm can be described as follows:

1. We split the time horizon $\mathcal{T}$ into batches $\mathcal{T}_1, \ldots, \mathcal{T}_m$ of size Δ_T each (except possibly the last batch):

$$\mathcal{T}_j = \{t \in \{1, \ldots, \Delta_T\} \ : \ (j-1)\Delta_T + t \leq \min\{j\Delta_T, T\}\}, \quad \text{for all } j = 1, \ldots, m$$

where $m = \left\lceil \frac{T}{\Delta_T} \right\rceil$ is the number of batches.

Algorithm 2: Repeating Greedy Algorithm (RGA)

Input: Δ_T.

while $1 \leq j \leq \left\lceil \frac{T}{\Delta_T} \right\rceil$ **do**
  Set $\tau = 1$
  while $\tau \leq \Delta_T$ **do**
    if *($1 \leq \tau \leq K$)* **then**
      Pull arm $k = \tau \mod K + 1$
      Receive reward and blocking duration (X_τ^k, D_τ^k)
      Set $\hat{X}_t^k = X_\tau^k$ for all $t \in [1, \Delta_T]$.
    if *($K + 1 \leq \tau \leq \tilde{D} + K$)* **then**
      Pull no arms
    if *($\tilde{D} + K + 1 \leq \tau \leq \Delta_T - \tilde{D}$)* **then**
      Pick arms according to
       GREEDY-BAA$(\Delta_T - 2\tilde{D} - K, K, \hat{X}^1, \ldots, \hat{X}^K, D^1, \ldots, D^K)$
    if *($\Delta_T - \tilde{D} + 1 \leq \tau \leq \Delta_T$)* **then**
      Pull no arms
    $\tau \leftarrow \tau + 1$
  $j \leftarrow j + 1$

2. Within each batch we spend the first K rounds pulling each arm. Without loss of generality, we shall assume that arm k is pulled on round k. After this we spend the next $\tilde{D}$ rounds pulling no arms. This ensures that all arms will be available when we next pull an arm.

3. Then, up until the final $\tilde{D}$ rounds we play Greedy-BAA using the rewards observed in the first K rounds as the fixed rewards for each arm.

4. In the final $\tilde{D}$ rounds of each batch, we again pull no arms. This ensures that all of the arms are available at the beginning of the next batch.

Theorem 3. *Suppose that the variation budget B_T is known in advance and maximal duration $\tilde{D} \geq 1$ such that $\tilde{D}B_T \in \mathrm{o}(T)$. The α-regret of RGA, where $\alpha = \frac{\tilde{D}}{\tilde{D}+\tilde{D}}$, is at most $\mathcal{O}\left(\sqrt{T(2\tilde{D} + K)B_T}\right)$ when the parameter when Δ_T is set to $\left\lceil \sqrt{\frac{(T+1)(2\tilde{D}+K)}{2B_T}} \right\rceil$.*

Note that this bound is sub-linear in T if $\tilde{D}B_T = \mathrm{o}(T)$ (e.g., $\tilde{D}$ is bounded above by a constant and $B_T \in \mathrm{o}(T)$). It is also worth noting that while $\alpha = \frac{1}{1+\tilde{D}}$ might imply that RGA can perform better than the worst-case performance of Greedy-BAA, with $B_T \in \mathrm{o}(T)$ it is not the case (see Section E in the appendix for more details).

4.2 Blocking Bandit with Unknown Path Variation Budget

Note that RGA requires knowledge of B_T in order to properly set Δ_T. To resolve this issue we propose META-RGA, a meta-bandit algorithm, where each arm corresponds to an instance of the RGA algorithm whose Δ_T parameter tuned for a different variation budget. The time horizon $\mathcal{T}$ is broken into meta-blocks of length H. At the start of each meta-block an arm (i.e., an instance of RGA with its corresponding budget) is selected according to the well known Exp3 algorithm [Auer et al., 2002]. The RGA is then played for the next H time steps with optimally tuned restarts (see Theorem 3 for more details). At the end of the meta-block, the Exp3 observes a reward corresponding to the total reward accumulated by the chosen RGA in this meta-block. The intuition of this idea is that the meta-bandit will learn which budget will be the best upper bound for RGA.

In what follows, we shall denote the set of arms available to the Exp3 algorithm by $\mathcal{J}$, and denote the corresponding set of variation budgets by $\mathcal{J}_B$. The META-RGA algorithm uses $\lceil \log_2(KT) \rceil + 1$ meta-arms with budgets $\mathcal{J}_B = \{2^0, 2^1, \ldots, 2^{\lceil \log_2(KT) \rceil}\}$. That is, the budget values are powers of 2

Algorithm 3: Meta Repeating Greedy Algorithm (`META-RGA`)

Input: $T, K, \gamma \in (0,1]$, batch length H.

Initialize: $|\mathcal{J}| = \lceil \log_2(KT) \rceil + 1$, $\mathcal{J}_B = \{2^0, 2^1, \dots, 2^{\lceil \log_2(KT) \rceil}\}$, $w_i(1) = 1$ for $i = 1, \dots, |\mathcal{J}|$.

for $\tau = 1, \dots, \lceil \frac{T}{H} \rceil$ **do**

  Set

$$p_i(\tau) = (1-\gamma)\frac{w_i(\tau)}{\sum_{j=1}^{|\mathcal{J}|} w_j(\tau)} + \frac{\gamma}{|\mathcal{J}|} \quad i = 1, \dots, |\mathcal{J}|$$

  Draw i_τ randomly according to the probabilities $p_1(\tau), \dots, p_{|\mathcal{J}|}(\tau)$

  Run `RGA` in batch τ with budget $\mathcal{J}_B[i_\tau] = 2^{i_\tau - 1}$ and optimally tuned restarts

  Receive reward $x_{i_t}(\tau) \in [0, H]$ at the end of the batch

  for $j = 1, \dots, |\mathcal{J}|$ **do**

$$\hat{x}_j(\tau) = \begin{cases} \frac{x_j(\tau)}{p_j(\tau)} & \text{if } j = i_\tau \\ 0 & \text{otherwise} \end{cases}$$

$$w_j(\tau+1) = w_j(\tau)\exp(\gamma \hat{x}_j(\tau)/(H|\mathcal{J}|))$$

up to the smallest 2-power, which is still larger than KT, which is the ultimate upper bound of the path variation budget (as $B_T \leq KT$). In addition, let B_i denote the total path variance within batch i, and $\tilde{B} = \max_i B_i$. We state the following:

Theorem 4. *Suppose that the variation budget B_T is unknown in advance to us. In addition, suppose that the maximal blocking duration $\tilde{D} \geq 1$ such that $\tilde{D}B_T \in \mathrm{o}(T)$. The α-regret of `RGA-META`, where $\alpha = \frac{1}{1+\tilde{D}}$, is at most*

$$\mathcal{O}\left(\tilde{B}^{1/2}T^{3/4}(2\tilde{D}+K)^{1/4}\ln(KT)^{1/4}\ln(\ln(KT))^{1/4}\right)$$

when the parameters of `RGA-META` are set as follows:

$$H = \sqrt{\frac{T(2\tilde{D}+K)}{\ln(KT)\ln(\ln(KT))}}, \quad \gamma = \min\left\{1, \sqrt{\frac{\ln(KT)\ln(\ln(KT))}{(e-1)T}}\right\}.$$

Note that since $\tilde{B} \leq HK$ by definition (the maximum path variance within a batch is at most HK), by setting $H = \sqrt{\frac{T(2\tilde{D}+K)}{\ln(KT)\ln(\ln(KT))}}$ we always get sub-linear regret in T if $\tilde{D} \in \mathcal{O}(1)$ (i.e., is bounded above by a constant). Otherwise we need to have $\tilde{B}^2\tilde{D} \in \mathrm{o}(T)$. Furthermore, when $\tilde{B}$ is small, our regret bound tends to $\mathcal{O}(T^{3/4})$. Thus, it is still an open question whether we can get a tighter upper bound (e.g., $\mathcal{O}(\sqrt{T})$) for this case (i.e., when the variation budget is unknown).

5 Discussions

In this section we will provide some intuitions why we set B_T and $\tilde{D}$ to be small in the previous sections. In particular, we show that if either the variation budget or the maximum blocking duration is large, the lower bound of the α-regret is $\Theta(T)$. We also discuss a potential lower bound for the α-regret of the adversarial blocking bandit problem in the case of $B_T \in \mathrm{o}(KT)$ and $\tilde{D} \in \mathcal{O}(1)$. Finally, we will also discuss how our results change if we use other types of variation budgets.

Large variation budget. Consider the case when $B_T \in \Theta(T)$. Theorem 3 indicates that the upper bound of the α-regret is $\Theta(T)$ where $\alpha = \frac{1}{1+\tilde{D}}$ as defined in Theorem 3. Indeed, we show that this is the best possible we can achieve:

Claim 1. *For any $T > 0$ and $B_T \in \Theta(KT)$, there exists a sequence of rewards and blocking durations X and D such that $\mathcal{R}_\pi^\alpha(B_T, \tilde{D}, T) = \Theta(T)$ for that particular (X, D).*

Large blocking durations. If $\tilde{D} \in \Theta(T)$ and α is the approximation ratio of `Greedy-BAA`:

Claim 2. *For any $T > 0$ and $\tilde{D} \in \Theta(T)$, there exists a sequence of rewards and blocking durations X and D such that $\mathcal{R}_{\pi}^{\alpha}(B_T, \tilde{D}, T) = \Theta(T)$ for that particular (X, D).*

Note that our regret bounds only make sense if $\tilde{D}B_T \in \mathrm{o}(T)$. Thus, it is still an open question whether we can achieve sub-linear α-regret bounds in T if both $B_T, \tilde{D} \in \mathrm{o}(T)$ but $\tilde{D}B_T \in \Omega(T)$.

Almost matching regret lower bound for `RGA`. Consider the case when $\tilde{D} = \mathcal{O}(1)$. This implies that the α-regret bound of `RGA` is reduced to $\mathcal{O}(\sqrt{KTB_T})$. This in fact matches the known lower bounds of the 1-regret for the case of $\tilde{D} = 1$ (i.e., no blocking) from the literature [Auer et al., 2019]. In particular, with $\tilde{D} = 1$, the `Greedy-BAA` algorithm becomes optimal (see, e.g., Section 4.3 of Basu et al. [2019] for the discussion of this), and thus, the α-regret notion becomes 1-regret. Therefore, if there exists an algorithm which could achive an α-regret better than $\mathcal{O}(\sqrt{KTB_T})$ in our setting, then it would be able to achieve $\mathcal{O}(\sqrt{KTB_T})$ 1-regret for the standard (i.e., non-blocking) adversarial bandit as well.

It is also worth noting that when $\tilde{D}$ is not bounded above by a constant, or the variation budget B_T is not known in advance, it is still not known what the regret lower bound would be.

Other variation budget definitions. There are a number of different variation budget definitions in the literature [Besbes et al., 2014, Wei and Luo, 2018, Auer et al., 2019]. It is worth noting that our analysis works in a similar way for the maximum variation budget $B_T^{\max}$ and number of changes budget L_T, which can be defined as follows:

$$B_T^{\max} = \sum_{t,t+1 \in \mathcal{T}} \max_{k \in \mathcal{K}} |X_{t+1}^k - X_t^k|, \quad L_T = \#\{t : 1 \leq t \leq T-1, \exists k : X_t^k \neq X_{t+1}^k\}$$

If we use these variation budgets instead, the regret in Theorem 3 will be modified to $\mathcal{O}\left(\sqrt{(2\tilde{D}+K)TB_T^{\max}}\right)$ and $\mathcal{O}\left(\sqrt{(2\tilde{D}+K)TL_T}\right)$, respectively. Furthermore, the approximation ratio of `Greedy-BAA` will also change. In particular, it becomes $\left[\left(1+\tilde{D}\right) + \tilde{D}KB_T^{\max}/r(\pi^+)\right]^{-1}$ and $\left[\left(1+\tilde{D}\right) + \tilde{D}KL_T/r(\pi^+)\right]^{-1}$. We refer the reader to Section C in the appendix for a more detailed discussion. It remains as future work to derive regret bounds for the other variation budgets.

Broader Impact

The paper examines a novel multi-armed bandit problem in which the decision-making agent aims to receive as many (cumulative) rewards as possible over a finite period subject to constraints. Our focus and results are largely theoretical. In particular, our contributions advance our understanding of multi-armed bandit models and its theoretical limitations and benefit the general (theoretical) machine learning community, specifically the multi-armed bandit and online learning communities. In addition, we do not expect that our theoretical findings can be directly used in more applied domains.

Acknowledgments and Disclosure of Funding

Nicholas Bishop was supported by the ECS PhD scholarship scheme from the University of Southampton, UK. Debmalya Mandal was supported by a Columbia Data Science Institute Post-Doctoral Fellowship.

References

Shipra Agrawal and Nikhil R Devanur. Bandits with concave rewards and convex knapsacks. In *Proceedings of the fifteenth ACM conference on Economics and computation*, pages 989–1006, 2014.

Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire. The nonstochastic multiarmed bandit problem. *SIAM journal on computing*, 32(1):48–77, 2002.

Peter Auer, Pratik Gajane, and Ronald Ortner. Adaptively tracking the best bandit arm with an unknown number of distribution changes. In *Conference on Learning Theory*, pages 138–158, 2019.

Ashwinkumar Badanidiyuru, Robert Kleinberg, and Aleksandrs Slivkins. Bandits with knapsacks. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pages 207–216. IEEE, 2013.

Ashwinkumar Badanidiyuru, John Langford, and Aleksandrs Slivkins. Resourceful contextual bandits. In *Conference on Learning Theory*, pages 1109–1134, 2014.

Soumya Basu, Rajat Sen, Sujay Sanghavi, and Sanjay Shakkottai. Blocking bandits. In *Advances in Neural Information Processing Systems 32*, pages 4784–4793, 2019.

Soumya Basu, Orestis Papadigenopoulos, Constantine Caramanis, and Sanjay Shakkottai. Contextual blocking bandits. *arXiv*, abs/2003.03426, 2020.

Marco Bender, Clemens Thielen, and Stephan Westphal. Online interval scheduling with a bounded number of failures. *Journal of Scheduling*, 20(5):443–457, 2017.

Omar Besbes, Yonatan Gur, and Assaf Zeevi. Stochastic multi-armed-bandit problem with non-stationary rewards. In *Proceedings of the 27th International Conference on Neural Information Processing Systems*, pages 199–207, 2014.

Sébastien Bubeck and Nicolo Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Machine Learning*, 5(1):1–122, 2012.

Deepayan Chakrabarti, Ravi Kumar, Filip Radlinski, and Eli Upfal. Mortal multi-armed bandits. In *Advances in neural information processing systems*, pages 273–280, 2009.

Wenkui Ding, Tao Qin, Xu-Dong Zhang, and Tie-Yan Liu. Multi-armed bandit with budget constraint and variable costs. In *Twenty-Seventh AAAI Conference on Artificial Intelligence*, 2013.

Thomas Erlebach and Frits CR Spieksma. Simple algorithms for a weighted interval selection problem. In *International Symposium on Algorithms and Computation*, pages 228–240. Springer, 2000.

Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., 1979.

Chien-Ju Ho and Jennifer Wortman Vaughan. Online task assignment in crowdsourcing markets. In *Twenty-sixth AAAI conference on artificial intelligence*, 2012.

N. Immorlica, K. A. Sankararaman, R. Schapire, and A. Slivkins. Adversarial bandits with knapsacks. In *2019 IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 202–219, 2019.

Tobias Jacobs and Salvatore Longo. A new perspective on the windows scheduling problem. *ArXiv*, abs/1410.7237, 2014.

Satyen Kale, Chansoo Lee, and David Pal. Hardness of online sleeping combinatorial optimization problems. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 2181–2189. Curran Associates, Inc., 2016.

A Karthik, Arpan Mukhopadhyay, and Ravi R Mazumdar. Choosing among heterogeneous server clouds. *Queueing Systems*, 85(1-2):1–29, 2017.

Robert Kleinberg, Alexandru Niculescu-Mizil, and Yogeshwer Sharma. Regret bounds for sleeping experts and bandits. *Mach. Learn.*, 80(2–3):245–272, 2010.

Antoon WJ Kolen, Jan Karel Lenstra, Christos H Papadimitriou, and Frits CR Spieksma. Interval scheduling: A survey. *Naval Research Logistics (NRL)*, 54(5):530–543, 2007.

Mikhail Y Kovalyov, CT Ng, and TC Edwin Cheng. Fixed interval scheduling: Models, applications, computational complexity and algorithms. *European journal of operational research*, 178(2): 331–342, 2007.

Hiroyuki Miyazawa and Thomas Erlebach. An improved randomized on-line algorithm for a weighted interval selection problem. *Journal of Scheduling*, 7(4):293–311, 2004.

Gergely Neu and Gábor Bartók. Importance weighting without importance weights: An efficient algorithm for combinatorial semi-bandits. *The Journal of Machine Learning Research*, 17(1): 5355–5375, 2016.

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

Anshuka Rangi, Massimo Franceschetti, and Long Tran-Thanh. Unifying the stochastic and the adversarial bandits with knapsack. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 3311–3317, 2019.

Long Tran-Thanh, Archie Chapman, Enrique Munoz de Cote, Alex Rogers, and Nicholas R Jennings. Epsilon–first policies for budget–limited multi-armed bandits. In *Twenty-Fourth AAAI Conference on Artificial Intelligence*, 2010.

Long Tran-Thanh, Archie Chapman, Alex Rogers, and Nicholas R Jennings. Knapsack based optimal policies for budget–limited multi–armed bandits. In *Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.

Long Tran-Thanh, Sebastian Stein, Alex Rogers, and Nicholas R Jennings. Efficient crowdsourcing of unknown experts using bounded multi-armed bandits. *Artificial Intelligence*, 214:89–111, 2014.

Chen-Yu Wei and Haipeng Luo. More adaptive algorithms for adversarial bandits. In *Conference On Learning Theory*, pages 1263–1291, 2018.

Ge Yu and Sheldon H Jacobson. Approximation algorithms for scheduling c-benevolent jobs on weighted machines. *IISE Transactions*, 52(4):432–443, 2020.

A Proofs for Theorems from Section 3

Proof of Theorem 1. Given a 3-SAT instance of m variables, $v_1, ..., v_m$, and n clauses, $C_1, ..., C_n$, we construct an instance of the MAXREWARD (decision) problem as follows.

- For each variable v_j, we create two arms k_j and $\bar{k}_j$.
- For each variable v_j, we set $X_j^{k_j} = X_j^{\bar{k}_j} = 1$ and $D_j^{i_j} = D_j^{\bar{i}_j} = T_0$ for some $T_0 > 0$ (all other rewards and blocking durations are set to zero and one, respectively, by default at this point).
- For each clause C_l and each literal $l(v)$ in C_l, we set $X_{m+l}^{l(v)} = 1$. Note that all other rewards and blocking durations are also zero and one, respectively, except those from the above.
- We let $T_0 = m + n$, $V = n + m$ and $B_{T_0} = O(T_0 m)$.

Clearly, the constructed instance's parameters are polynomial bounded. Given the MAXREWARD (decision) problem instance, our goal is to find $\pi^* = (\pi_1^*, ..., \pi_T^*) \in \mathcal{P}$ such that $r(\pi^*) \geq V$.

Claim 3. *There a solution to the MAXREWARD problem if only if there is a solution to 3-SAT.*

Proof of Claim 3. 3-*SAT solution* $\implies$ *MAXREWARD solution.* Suppose that we have a solution for the 3-SAT problem. It follows that there is an assignment to each variable v_i such that each clause is true. To construct a solution to the MAXREWARD problem, we perform the following. For each variable v_j that is set to true (false) and for each clause C_l containing v_j ($\bar{v}_j$), we play arm k_j ($\bar{k}_j$) at time $m + l$ to obtain a reward of 1, which corresponds to setting $\pi_{m+l}^* = k_j$ ($\pi_{m+l}^* = \bar{k}_j$). Since we can only play an arm each time period, ties can be broken arbitrarily. From this partial solution, we obtain a cumulative reward of n since all of the n clauses are satisfied by at least one literal. Observe that for the remaining arm $\bar{k}_j$ (k_j) for the variable v_j that is set to be true (false), it is valid to play $\bar{k}_j$ (k_j) at time j to obtain a reward of 1, which corresponds to setting $\pi_j^* = \bar{k}_j$ ($\pi_j^* = k_j$). From this partial solution, we obtain a value of m since there are m variables. It is not hard to see that the constructed π^* is a valid solution with $r(\pi^*) \geq n + m = V$.

MAXREWARD solution $\implies$ 3-*SAT solution.* Suppose now that we have a solution for the MAXREWARD problem. It follows that there a solution i_T^* such that $r(\pi^*) \geq V = n + m$. Note that for any feasible solution π, $r(\pi) \leq n + m$ as from time periods 1 to m and $m + 1$ to $m + n$ we can obtain at most a reward of m and n, respectively. Thus, $r(\pi^*) = n + m$. To obtain a reward of m from time period $j = 1, ..., m$, we must play either k_j or $\bar{k}_j$. To obtain a reward of n from time period $j = m + 1, ..., m + n$, we must play exactly one of the arms corresponds to the clause C_{j-m}. Thus, to construct an assignment for the 3-SAT instance, we let v_j to be false (true) if k_j ($\bar{k}_j$) is played at time j. Such assignment ensures that, for each clause C_l, at least one of the literals is true, which corresponds to having one of the (literal) arms played that isn't blocked. □

Together with the above claim, we have completed the reduction and showed that the MAXREWARD problem is strongly NP-hard with bounded path variation. Note that in this proof, we relied on a special set of problem instances where both T_0 and B_{T_0} are in the range of max delay $D = n + m$. This might cause some issues for the other analysis in the paper (e.g., the performance guarantee in Theorem 2 is designed for some more well behaved cases). Therefore, we still need to modify the proof above to cover more generic cases.

Now, let $k_1, k_2 \geq 2$ arbitrary integers. For each of the construction above of T_0 time steps, we pad it with another $(k_1 - 1)T_0$ time steps where each arm has blocking value 1 and reward value 0. Together with the first T_0 time steps, these form an interval of $k_1 T_0$ time steps, called large blocks. We then set $T = k_2 k_1 T_0$ and concatenate k_2 copies of these large blocks together. It is easy to see that the proof of reduction above is still valid, but now we have time horizon $T = k_1 k_2 D$, and variation budget $B = k_2 D$. By varying k_1 and k_2, we can set the relationship of T, B, and D to be arbitrary. This concludes the proof. □

Proof of Theorem 2. Let $\pi^* = (\pi_1^*, ..., \pi_T^*) \in \arg\max_{\pi \in \mathcal{P}} r(\pi)$ be an optimal solution. Let $\pi^+ = (\pi_1^+, ..., \pi_T^+) \in \mathcal{P}$ be a solution returned by `Greedy-BAA`. Consider a time period $t \in \mathcal{T}$ where

$\pi_t^* \neq \pi_t^+$. There are two cases in which π_t^* is not selected by `Greedy-BAA`. The first case is where $X_t^{i_t^*} \leq X_t^{i_t^+}$. The second case is where `Greedy-BAA` played the arm π_t^* at (the most recent) time $1 \leq t' < t$ and it is blocked for $D_{t'}^{\pi_{t'}^+}$ time steps. Since $\pi_{t'}^+ = \pi_t^*$, we let $\pi_{t'}^+ = j \in \mathcal{K}$. The difference of the reward is given by

$$|X_{t'}^j - X_t^j| = |X_{t'}^j - X_{t'+1}^j + X_{t'+1}^j - X_t^j| \leq |X_{t'}^j - X_{t'+1}^j| + |X_{t'+1}^j - X_t^j| \leq \sum_{\bar{t}=t'}^{t-1} |X_{\bar{t}}^j - X_{\bar{t}+1}^j|,$$

where the inequalities resulted from adding and subtracting the corresponding terms and applying the triangle inequalities for the absolute value function repeatedly. Thus,

$$X_{t'}^j + \sum_{\bar{t}=t'}^{t-1} |X_{\bar{t}}^j - X_{\bar{t}+1}^j| \geq X_t^j$$

Let $\mathrm{Blk}(t', j)$ be the set of time periods in which playing arm j is optimal in $(\pi_1^*, ..., \pi_T^*)$, but arm j is blocked by playing it at time t' via `Greedy-BAA`. Note that $|\mathrm{Blk}(t', j)| \leq \frac{D_{\max}^j}{D_{\min}^j}$ where where $D_{\max}^j$ and $D_{\min}^j$ are the maximum and minimum blocking duration of arm j across all the time periods, respectively. This is because in the time intervals from $t'+1$ to $t' + D_{\max}^j$ arm j can be played at most $\frac{D_{\max}^j}{D_{\min}^j}$ times by any algorithm. This gives us

$$\sum_{t \in \mathrm{Blk}(t',j)} X_t^j \leq \sum_{t \in \mathrm{Blk}(t',j)} \left(X_{t'}^j + \sum_{\bar{t}=t'}^{t-1} |X_{\bar{t}}^j - X_{\bar{t}+1}^j| \right) \leq \frac{D_{\max}^j}{D_{\min}^j} \left(X_{t'}^j + \sum_{\bar{t}=t'}^{\max(\mathrm{Blk}(t',j))-1} |X_{\bar{t}}^j - X_{\bar{t}+1}^j| \right).$$

Note that, for any $\bar{t} \neq t'$ such that $\pi_{\bar{t}}^G = \pi_{t'}^G = j$, $\mathrm{Blk}(t', j) \cap \mathrm{Blk}(\bar{t}, j) = \emptyset$.

As a result, each arm π_t^+ can be used to cover some part of the optimal solution under case 1 and/or case 2 for each time period $t \in \mathcal{T}$. It follows that

$$\begin{aligned}
r(\pi^*) = \sum_{t=1}^T X_t^{\pi_t^*} &\leq \sum_{t=1}^T X_t^{\pi_t^+} + \sum_{t=1}^T \frac{D_{\max}^{\pi_t^+}}{D_{\min}^{\pi_t^+}} \left(X_t^{\pi_t^+} + \sum_{\bar{t}=t}^{\max(\mathrm{Blk}(t,\pi_t^+))-1} |X_{\bar{t}}^{\pi_t^+} - X_{\bar{t}+1}^{\pi_t^+}| \right) \\
&\leq \sum_{t=1}^T X_t^{\pi_t^+} + \frac{D_{\max}^{j^*}}{D_{\min}^{j^*}} \sum_{t=1}^T \left(X_t^{\pi_t^+} + \sum_{\bar{t}=t}^{\max(\mathrm{Blk}(t,\pi_t^+))-1} |X_{\bar{t}}^{\pi_t^+} - X_{\bar{t}+1}^{\pi_t^+}| \right) \\
&= \sum_{t=1}^T X_t^{\pi_t^+} + \frac{D_{\max}^{j^*}}{D_{\min}^{j^*}} \left(\sum_{t=1}^T X_t^{\pi_t^+} + \sum_{t=1}^T \sum_{\bar{t}=t}^{\max(\mathrm{Blk}(t,\pi_t^+))-1} |X_{\bar{t}}^{\pi_t^+} - X_{\bar{t}+1}^{\pi_t^+}| \right) \\
&\leq \sum_{t=1}^T X_t^{\pi_t^+} + \frac{D_{\max}^{j^*}}{D_{\min}^{j^*}} \left(\sum_{t=1}^T X_t^{\pi_t^+} + \sum_{i \in \mathcal{K}} \sum_{t=1}^{T-1} |X_t^i - X_{t+1}^i| \right) \\
&\leq \sum_{t=1}^T X_t^{\pi_t^+} + \frac{D_{\max}^{j^*}}{D_{\min}^{j^*}} \left(\sum_{t=1}^T X_t^{\pi_t^+} + B_T \right) \leq \left(1 + \frac{D_{\max}^{j^*}}{D_{\min}^{j^*}} \right) r(\pi^+) + \frac{D_{\max}^{j^*}}{D_{\min}^{j^*}} B_T,
\end{aligned}$$

where the first inequality is from applying case 1 and case 2, the second inequality is from replacing the ratio by $\frac{D_{\max}^{j^*}}{D_{\min}^{j^*}}$, the arm j^* with the highest max-min blocking duration ratio, the third equality by distributing the summations, the fourth inequality by first grouping the time periods that each arm i is played and then applying the sum (which is bounded by T), and the fifth inequality is by definition. Rearranging the terms, we obtain our claimed result. □

B Proofs for Theorems from Section 4

Proof of Theorem 3. Recall that we want to compute the α-regret of `RGA` against the optimal offline solution of MAXREWARD. Let π denote the policy generated by `RGA`, and recall that π^* is the

policy of the optimal solution. For the sake of simplicity, we will refer to π^* as $*$ in the indices. Let $\alpha = \frac{\tilde{D}}{\tilde{D}+\tilde{D}}$. The α-regret of RGA incurred in a batch $\mathcal{T}_j$ is given by:

$$\sum_{t \in \mathcal{T}_j} (\alpha X_t^* - X_t^\pi) \tag{1}$$

In the first K rounds a loss of at most K can be accumulated. Similarly for the next $\tilde{D}$ time steps and the last $\tilde{D}$ time steps, a loss of at most $\tilde{D}$ is accumulated. Let $\mathcal{T}_j'$ denote the time steps in batch $\mathcal{T}_j$, excluding the first $\tilde{D} + K$ and the last $\tilde{D}$ rounds.

$$\sum_{t \in \mathcal{T}_j} (\alpha X_t^* - X_t^\pi) \leq (2\tilde{D} + K) + \sum_{t \in \mathcal{T}_j'} (\alpha X_t^* - X_t^\pi) \tag{2}$$

Let $\hat{X}_t^\pi$ denote the reward of the arm played by policy π at time step t which was observed in the first K rounds of batch $\mathcal{T}_j$ and let B_j denote the path variance within this batch (i.e., batch $\mathcal{T}_j$):

$$B_j = \sum_{t \in \mathcal{T}_j} \sum_{k \in \mathcal{K}} \left| X_{t+1}^k - X_t^k \right|$$

Then we have

$$\begin{aligned}
\sum_{t \in \mathcal{T}_j'} (\alpha X_t^* - X_t^\pi) &= \sum_{t \in \mathcal{T}_j'} (\alpha \hat{X}_t^* - \hat{X}_t^\pi) + \sum_{t \in \mathcal{T}_j'} (\alpha X_t^* - \alpha \hat{X}_t^*) + \sum_{t \in \mathcal{T}_j'} (\hat{X}_t^\pi - X_t^\pi) \\
&\leq \sum_{t \in \mathcal{T}_j'} |X_t^* - \hat{X}_t^*| + \sum_{t \in \mathcal{T}_j'} |X_t^\pi - \hat{X}_t^\pi| \\
&\leq \sum_{t \in \mathcal{T}_j'} 2B_j \\
&\leq 2\Delta_T B_j
\end{aligned} \tag{3}$$

The first inequality comes from the fact that $\alpha = \frac{1}{1+\tilde{D}} \leq 1$ and RGA runs Greedy-BAA with fixed estimates $\hat{X}_t$, which is α-optimal for an instance of MAXREWARD with fixed $\hat{X}^k$ values (we apply Theorem 2 with variation budget 0). Thus, $\sum_{t \in \mathcal{T}_j'}^T \hat{X}_t^\pi \geq \alpha \sum_{t \in \mathcal{T}_j'}^T \hat{X}_t^*$. The second inequality comes from the following observation: For each $\in \mathcal{T}_j'$ and $k \in \mathcal{K}$, we have

$$|X_t^k - \hat{X}_t^k| \leq \sum_{t, t+1 \in \mathcal{T}_j}^T |X_{t+1}^k - \hat{X}_t^k| \leq B_j \tag{4}$$

Recall that $\hat{X}_t^k$ is the value of first pull of arm k in the batch. Therefore, the difference between that first observed value and X_t^k can be bounded above by the sum of reward changes from round to round of arm k, which is further bounded above by the path variation budget B_j of that batch. The third inequality in Eq. (3) comes from the fact that the length of the batch is at most Δ_T. Replacing Eq. (3) into Eq. (2) we get:

$$\sum_{t \in \mathcal{T}_j}^T (X_t^* - X_t^\pi) \leq 2\Delta_T B_j + (2\tilde{D} + K)$$

Summing over all batches we have the following bound on regret:

$$\begin{aligned}
\mathcal{R}_\pi^\alpha(B_T, \tilde{D}, T) &\leq \sum_{j=1}^m 2\Delta_T B_j + \left\lceil \frac{T}{\Delta_T} \right\rceil (2\tilde{D} + K) \\
&\leq 2B_T \Delta_T + \left\lceil \frac{T}{\Delta_T} \right\rceil (2\tilde{D} + K) \\
&\leq 2B_T \Delta_T + \frac{T+1}{\Delta_T} (2\tilde{D} + K)
\end{aligned}$$

Since $B_T \leq TK$ by definition and both $\tilde{D}, K \geq 1$, we have $\sqrt{\frac{(T+1)(2\tilde{D}+K)}{2B_T}} \geq 1$, and thus, $\left\lceil \sqrt{\frac{(T+1)(2\tilde{D}+K)}{2B_T}} \right\rceil \leq \sqrt{\frac{(T+1)(2\tilde{D}+K)}{2B_T}} + 1 \leq 2\sqrt{\frac{(T+1)(2\tilde{D}+K)}{2B_T}}$. By setting $\Delta_T = \left\lceil \sqrt{\frac{(T+1)(2\tilde{D}+K)}{2B_T}} \right\rceil \leq 2\sqrt{\frac{(T+1)(2\tilde{D}+K)}{2B_T}}$ we obtain the desired result.

□

Proof of Theorem 4. Let π denote `META-RGA`. The α-regret of `META-RGA` can be expressed as follows:

$$\sum_{i=1}^{\lceil \frac{T}{H} \rceil} \sum_{t=(i-1)H+1}^{\max(T,iH)} (\alpha X_t^* - X_t^\pi)$$

Let B_i denote the total path variance within batch i. Of all the `RGA` instances (i.e., meta-arms) available to there must be an instance who is a associated with a candidate budget $\tilde{B}$ such that:

$$\max_i B_i \leq \tilde{B} \leq 2 \max_i B_i \tag{5}$$

Le $\tilde{\pi}$ denote the policy of this `RGA` instance. Using $\tilde{\pi}$ we can decompose the regret of `META-RGA` as follows:

$$\left[\sum_{i=1}^{\lceil \frac{T}{H} \rceil} \sum_{t=(i-1)H+1}^{\max(T,iH)} (\alpha X_t^* - X^{\tilde{\pi}}) \right] + \left[\sum_{i=1}^{\lceil \frac{T}{H} \rceil} \left(\sum_{t=(i-1)H+1}^{\max(T,iH)} X_t^{\tilde{\pi}} \right) - \left(\sum_{t=(i-1)H+1}^{\max(T,iH)} X^\pi \right) \right] \tag{6}$$

The second term of Eq (6) can be further bounded as follows. Note that the `RGA` instance with policy $\tilde{\pi}$ might not be the best fixed meta-arm in hindsight, whose policy is denoted by π^+. Thus, we have:

$$\left[\sum_{i=1}^{\lceil \frac{T}{H} \rceil} \left(\sum_{t=(i-1)H+1}^{\max(T,iH)} X_t^{\tilde{\pi}} \right) - \left(\sum_{t=(i-1)H+1}^{\max(T,iH)} X^\pi \right) \right] \leq \left[\sum_{i=1}^{\lceil \frac{T}{H} \rceil} \left(\sum_{t=(i-1)H+1}^{\max(T,iH)} X_t^{\pi^+} \right) - \left(\sum_{t=(i-1)H+1}^{\max(T,iH)} X^\pi \right) \right]$$

The RHS of this is simply the difference between the rewards observed and accumulated by the Exp3 meta-algorithm and the best available `RGA` meta-arm in hindsight. Thus we can bound the second term with standard Exp3 regret bounds. Note that there are $\log_2(KT)$ arms available to the Exp3 algorithm, T/H is number of batches, and the maximum reward a meta-arm can receive within a batch is H (i.e., the length of each batch). Thus the second term can be bounded above by $\mathcal{O}\left(H\sqrt{T/H \ln(KT) \ln(\ln(KT))}\right) = \mathcal{O}\left(\sqrt{HT \ln(KT) \ln(\ln(KT))}\right)$.

Now we turn to bound the first term of Eq (6). Each inner sum of the first term correspond to the α-regret of policy $\tilde{\pi}$ over a block of length H. Our idea is to use Theorem 3 to bound the regret of $\tilde{\pi}$ in each of batches i. In order to do so, we must check whether running `RGA` with budget $\tilde{B}$ in the batches (with time horizon H) will result in a valid Δ_H, that is $\Delta_H \geq 1$. From Eq (5) we know that $\tilde{B} \leq 2 \max_i B_i \leq 2HK$ (the second inequality comes from the definition of the total variance budget, which is at most HK for time horizon H). Therefore, from Theorem 3 we know that $\Delta_H \geq \sqrt{\frac{(H+1)(2\tilde{D}+K)}{2\tilde{B}}} > \sqrt{\frac{H(2\tilde{D}+K)}{4HK}} \geq 1$ if $\tilde{D} \geq \frac{3K}{2}$. Now, since $\tilde{D}$ is an upper bound of the maximal blocking duration, we can set it to be at least $\frac{3K}{2}$ to make $\Delta_H \geq 1$. Therefore, we can apply Theorem 3 to each of the batches. In particular, the α-regret of $\tilde{\pi}$ over batch i of length at most H and with optimally tuned restarts can be bounded as follows:

$$\begin{aligned} \sum_{t=(i-1)H+1}^{\max(T,iH)} (\alpha X_t^* - X^{\tilde{\pi}}) &\leq \sqrt{2\tilde{B}(H+1)(2\tilde{D}+K)} \\ &\leq 2\sqrt{\tilde{B}H(2\tilde{D}+K)} \end{aligned}$$

Summing over all blocks we have:

$$\sum_{i=1}^{\lceil \frac{T}{H} \rceil} \sum_{t=(i-1)H+1}^{\max(T,iH)} (\alpha X_t^* - X^{\tilde{\pi}}) \leq \left(\frac{T}{H}+1\right) 2\sqrt{\tilde{B}H(2\tilde{D}+K)} \\ \leq 4\frac{T}{\sqrt{H}}\sqrt{\tilde{B}(2\tilde{D}+K)} \tag{7}$$

Combining Eq (7) with the regret bound of the Exp3 meta-bandit algorithm, we get that the α-regret of `META-RGA` is at most

$$\mathcal{O}\left(\frac{T}{\sqrt{H}}\sqrt{\tilde{B}(2\tilde{D}+K)}\right) + \mathcal{O}\left(\sqrt{HT\ln(KT)\ln(\ln(KT))}\right) \tag{8}$$

By setting $H = \sqrt{\frac{T(2\tilde{D}+K)}{\ln(KT)\ln(\ln(KT))}}$ we get the desired regret bound. □

C Regret Analysis with Other Variation Budgets

In this section we show how our regret analysis can be adopted to the maximum variation budget $B_T^{\max}$ and number of changes budget L_T. For the sake of convenience, we repeat the definition of these budgets below:

$$B_T^{\max} = \sum_{t,t+1\in\mathcal{T}} \max_{k\in\mathcal{K}} |X_{t+1}^k - X_t^k|, \quad L_T = \#\{t : 1 \leq t \leq T-1, \exists k : X_t^k \neq X_{t+1}^k\}$$

It is easy to show that $B_T \leq KB_T^{\max} \leq KL_T$. Thus, by just replacing B_T with $KB_T^{\max}$ and KL_T we can already get regret bounds with the other two variation budgets.

However, we show that we can further improve these bounds by order of $\sqrt{K}$ as follows: We only need to modify the way we estimate the regret in Eq (4). In particular, recall that for each $\in \mathcal{T}_j'$ and $k \in \mathcal{K}$, we have

$$|X_t^k - \hat{X}_t^k| \leq \sum_{t,t+1\in\mathcal{T}_j}^{T} |X_{t+1}^k - \hat{X}_t^k| \leq \sum_{t,t+1\in\mathcal{T}_j}^{T} \max_{l\in\mathcal{K}} |X_{t+1}^l - \hat{X}_t^l| \leq B_j^{\max} \tag{9}$$

where $B_j^{\max}$ is the maximum variation budget of batch j. Replacing this back to Eq. (3) we get:

$$\begin{aligned} \sum_{t\in\mathcal{T}_j'} (\alpha X_t^* - X_t^\pi) &= \sum_{t\in\mathcal{T}_j'} (\alpha \hat{X}_t^* - \hat{X}_t^\pi) + \sum_{t\in\mathcal{T}_j'} (\alpha X_t^* - \alpha \hat{X}_t^*) + \sum_{t\in\mathcal{T}_j'} (\hat{X}_t^\pi - X_t^\pi) \\ &\leq \sum_{t\in\mathcal{T}_j'} |X_t^* - \hat{X}_t^*| + \sum_{t\in\mathcal{T}_j'} |X_t^\pi - \hat{X}_t^\pi| \\ &\leq \sum_{t\in\mathcal{T}_j'} 2B_j^{\max} \\ &\leq 2\Delta_T B_j^{\max} \end{aligned} \tag{10}$$

By following the same steps in the proof of Theorem 3, we get that the regret bound for `RGA` is $\mathcal{O}\left(\sqrt{(2\tilde{D}+K)TB_T^{\max}}\right)$ if Δ_T is optimally tuned to be $\left\lceil \sqrt{\frac{(T+1)(2\tilde{D}+K)}{2B_T^{\max}}} \right\rceil$.

Similarly, for number of changes L_T, we can rewrite Eq. (9) as follows:

$$\begin{aligned} |X_t^k - \hat{X}_t^k| &\leq \sum_{t,t+1\in\mathcal{T}_j}^{T} |X_{t+1}^k - \hat{X}_t^k| \leq \sum_{t,t+1\in\mathcal{T}_j}^{T} \mathcal{I}(X_{t+1}^k \neq \hat{X}_t^k) \\ &\leq \sum_{t,t+1\in\mathcal{T}_j}^{T} \mathcal{I}(\exists l \in \mathcal{K} : X_{t+1}^l \neq \hat{X}_t^l) \\ &\leq L_j \end{aligned} \tag{11}$$

where $\mathcal{I}(\cdot)$ is the indicator function, and L_j is the total number of changes in batch j. The rest is similar to the discussion above, and we can get $\mathcal{O}\left(\sqrt{(2\tilde{D}+K)TL_T}\right)$ regret bound for our algorithm (with $\Delta_T = \left\lceil \sqrt{\frac{(T+1)(2\tilde{D}+K)}{2L_T}} \right\rceil$).

Regarding the new values for the approximation ratio of `Greedy-BAA`, recall that the approximation bound for `Greedy-BAA` can be calculated as follows:

$$\begin{aligned} r(\pi^*) = \sum_{t=1}^{T} X_t^{\pi_t^*} &\leq \sum_{t=1}^{T} X_t^{\pi_t^+} + \sum_{t=1}^{T} \frac{D_{\max}^{\pi_t^+}}{D_{\min}^{\pi_t^+}} \left(X_t^{\pi_t^+} + \sum_{\bar{t}=t}^{\max(\mathrm{Blk}(t,\pi_t^+))-1} |X_{\bar{t}}^{\pi_t^+} - X_{\bar{t}+1}^{\pi_t^+}| \right) \\ &\leq \sum_{t=1}^{T} X_t^{\pi_t^+} + \frac{D_{\max}^{j^*}}{D_{\min}^{j^*}} \left(\sum_{t=1}^{T} X_t^{\pi_t^+} + \sum_{i\in\mathcal{K}} \sum_{t=1}^{T-1} |X_t^i - X_{t+1}^i| \right) \end{aligned} \tag{12}$$

Note that the term $\sum_{i\in\mathcal{K}} \sum_{t=1}^{T-1} |X_t^i - X_{t+1}^i|$ from Eq (12) can be bounded above by $KB_T^{\max}$ and KL_T, respectively. This implies that:

$$\begin{aligned} r(\pi^*) = \sum_{t=1}^{T} X_t^{\pi_t^*} &\leq \sum_{t=1}^{T} X_t^{\pi_t^+} + \frac{D_{\max}^{j^*}}{D_{\min}^{j^*}} \left(\sum_{t=1}^{T} X_t^{\pi_t^+} + \sum_{i\in\mathcal{K}} \sum_{t=1}^{T-1} |X_t^i - X_{t+1}^i| \right) \\ &\leq \sum_{t=1}^{T} X_t^{\pi_t^+} + \frac{D_{\max}^{j^*}}{D_{\min}^{j^*}} \left(\sum_{t=1}^{T} X_t^{\pi_t^+} + KB_T^{\max} \right) \\ &\leq \left(1 + \frac{D_{\max}^{j^*}}{D_{\min}^{j^*}} \right) r(\pi^+) + \frac{D_{\max}^{j^*}}{D_{\min}^{j^*}} KB_T^{\max} \end{aligned} \tag{13}$$

Similarly, we have:

$$\begin{aligned} r(\pi^*) = \sum_{t=1}^{T} X_t^{\pi_t^*} &\leq \sum_{t=1}^{T} X_t^{\pi_t^+} + \frac{D_{\max}^{j^*}}{D_{\min}^{j^*}} \left(\sum_{t=1}^{T} X_t^{\pi_t^+} + \sum_{i\in\mathcal{K}} \sum_{t=1}^{T-1} |X_t^i - X_{t+1}^i| \right) \\ &\leq \sum_{t=1}^{T} X_t^{\pi_t^+} + \frac{D_{\max}^{j^*}}{D_{\min}^{j^*}} \left(\sum_{t=1}^{T} X_t^{\pi_t^+} + KL_T \right) \\ &\leq \left(1 + \frac{D_{\max}^{j^*}}{D_{\min}^{j^*}} \right) r(\pi^+) + \frac{D_{\max}^{j^*}}{D_{\min}^{j^*}} KL_T \end{aligned} \tag{14}$$

Thus, we have the approximation ratio of $\left[\left(1+\tilde{D}\right) + \tilde{D}KB_T^{\max}/r(\pi^+) \right]^{-1}$ for the usage of maximum variation budget $B_T^{\max}$, and $\left[\left(1+\tilde{D}\right) + \tilde{D}KL_T/r(\pi^+) \right]^{-1}$ if we use the number of changes budget L_T.

D Proof of Claims 1 and 2

Proof of Claim 1. Consider the case of $B_T = KT$. This implies that the rewards can change in an arbitrary way. Now consider the case when $D_t^k = 1$ for all $k \in \mathcal{K}$ and $t \in \mathcal{T}$ (i.e., there is no blocking at all). In this case, we have $\alpha = 1/2$. The main idea of the proof is to randomly generate the sequences of X_t^k in some way and prove that in expectation (over this randomisation), the α-regret is large. In particular, for any arm pulling policy π we have that:

$$\begin{aligned} \mathbb{E}\Big[\alpha \sum_t X_t^* - \sum_t X_t^\pi\Big] &= \mathbb{E}\Big[\alpha \sum_t \max_{k\in\mathcal{K}} X_t^k - \sum_t X_t^\pi\Big] \\ &\geq \mathbb{E}\Big[\alpha \sum_t \max_{k\in\mathcal{K}} X_t^k\Big] - \max_{k\in\mathcal{K}} \mathbb{E}\Big[\sum_t X_t^k\Big] + \max_{k\in\mathcal{K}} \mathbb{E}\Big[\sum_t X_t^k\Big] - \mathbb{E}\Big[\sum_t X_t^\pi\Big] \\ &\geq \alpha \sum_t \mathbb{E}\Big[\max_{k\in\mathcal{K}} X_t^k\Big] - \max_{k\in\mathcal{K}} \mathbb{E}\Big[\sum_t X_t^k\Big] + \tilde{R}_T^\pi \end{aligned} \tag{15}$$

where $\tilde{R}_T^\pi$ is the pseudo regret of π against the best fixed policy in hindsight. Now we use the standard stochastic setup to prove the lower bound of the pseudo regret: e.g., the arms are drawn from Bernoulli distributions with one arm to have reward mean of $\varepsilon + \sqrt{\frac{K}{T}}$, while the other arms have reward mean of ε (see, e.g., [Bubeck and Cesa-Bianchi, 2012] for the technical details). By doing so, we can prove that $\tilde{R}_T^\pi \geq \frac{1}{8}\sqrt{KT}$. In addition, we have that:

$$\begin{aligned}
&\alpha \sum_t \mathbb{E}\Big[\max_{k\in\mathcal{K}} X_t^k\Big] - \max_{k\in\mathcal{K}} \mathbb{E}\Big[\sum_t X_t^k\Big] \\
&\qquad = \alpha T\Big(1-(1-\varepsilon)^K(1-\sqrt{K/T}-\varepsilon)\Big) - T(\sqrt{K/T}+\varepsilon) \\
&\qquad = T\Big(\alpha\Big(1-(1-\varepsilon)^K(1-\sqrt{K/T}-\varepsilon)\Big) - \Big(\sqrt{K/T}+\varepsilon\Big)\Big)
\end{aligned} \tag{16}$$

Substituting $\alpha = 1/2$ and $\beta = (1-\varepsilon)^K$ we further have:

$$\begin{aligned}
&T\Big(\alpha\Big(1-(1-\varepsilon)^K(1-\sqrt{K/T}-\varepsilon)\Big) - \Big(\sqrt{K/T}+\varepsilon\Big)\Big) \\
&\qquad = T\Big((1-\beta)/2 - (\sqrt{K/T}+\varepsilon)(1-\beta/2)\Big) \\
&\qquad \geq T\Big((1-\beta)/2 - \varepsilon\Big) - \sqrt{KT}
\end{aligned} \tag{17}$$

Putting all these together we get:

$$\mathbb{E}\Big[\alpha \sum_t X_t^* - \sum_t X_t^\pi\Big] \geq T\Big((1-\beta)/2-\varepsilon\Big) - \frac{7}{8}\sqrt{KT} \tag{18}$$

It is easy to show that for any $K \geq 2$, with a sufficiently small ε, there exists a constant $c > 0$ such that $(1-\beta)/2-\varepsilon) > c$. This implies that $\mathbb{E}\Big[\alpha \sum_t X_t^* - \sum_t X_t^\pi\Big] \in \Theta(T)$, which concludes the proof. □

Note that the same proof works for any constant $\alpha > 0$.

Proof of Claim 2. Note that in this claim, the value of α is defined by the approximation ratio of `Greedy-BAA`, and not the value from Theorem 3. The reason for this modification is that $\alpha = \frac{1}{1+\bar{D}}$ in this case becomes $\Theta(1/T)$, which is not very meaningful. In particular, $\alpha = 1/T$ implies that, as the optimal solution is bounded above by T, a policy with sub-linear $\frac{1}{T}$-regret would only need to achieve $\Theta(1)$ performance, which is not difficult to achieve.

Now, consider the following two instances of a 2-arm bandit model: In problem instance P1, we have a bandit model with arms 1 and 2. For $t = 1$, we have $X_1^1 = 1$ with $D_1^1 = 1$, and $X_1^2 = 0$ with $D_1^2 = T$, respectively. For $t \geq 2$ we set $X_t^1 = 0$ with $D_t^1 = 1$, and $X_t^2 = 1$ with $D_t^2 = 1$ as well. It is clear that in this instance `Greedy-BAA` will also be the optimal solution, with pulling Arm 1 at $t = 1$ and repeatedly pulling Arm 2 afterwards (thus, the optimal performance is T). If any policy starts with pulling Arm 2 first, the total reward it can collect is 0 (as after pulling Arm 2 at $t = 1$, from $t = 2$, the only feasible arm is Arm 1 with reward 0).

We also design problem instance P2 by swapping the rewards and blocking durations of the 2 arms in P1 with each other. In this instance, the optimal solution is to pull Arm 2 first and then repeatedly pull Arm 1. For both P1 and P2, the path variation budget is $B = 2$.

Now, consider an arbitrary policy π. Suppose that π pulls Arm 1 with probability $p \in [0, 1]$. For now, assume that $p \leq 1/2$. In this case, if π is applied to P1, its expected reward will be $pT + (1-p)0 \leq T/2$, implying that the difference between the performance of π and that of `Greedy-BAA` is at least $T/2$. Similarly, if $p > 1/2$, we will consider P2. Putting these together we can see that for any arbitrary policy π, there exists a problem instance on which the approximate regret is at least $T/2$. □

E Performance Comparison between Greedy-BAA and RGA

From Theorem 2 we have that:

$$r(\pi^*) \leq \left(1 + \frac{D_{\max}^{k^*}}{D_{\min}^{k^*}}\right) r(\pi^+) + \frac{D_{\max}^{k^*}}{D_{\min}^{k^*}} B_T \leq \left(1 + \frac{\tilde{D}}{\underset{\sim}{D}}\right) r(\pi^+) + \frac{\tilde{D}}{\underset{\sim}{D}} B_T$$

where $r(\pi^*)$ is the (offline) optimal solution, and $r(\pi^+)$ is the performance of `Greedy-BAA`. This can be rewritten as:

$$\frac{\underset{\sim}{D}}{\underset{\sim}{D} + \tilde{D}} r(\pi^*) - \frac{\tilde{D}}{\underset{\sim}{D} + \tilde{D}} B_T \leq r(\pi^+). \tag{19}$$

For `RGA`, we know from Theorem 3 that

$$\frac{\underset{\sim}{D}}{\underset{\sim}{D} + \tilde{D}} r(\pi^*) - \mathcal{O}\left(\sqrt{T(2\tilde{D} + K)B_T}\right) \leq r(\text{RGA}). \tag{20}$$

If $B_T = \mathrm{o}(T)$, we have that $\sqrt{T(2\tilde{D} + K)B_T} > B_T$, thus the LHS of Eq (19) is larger than the LHS of Eq (20), which implies that the approximation ratio of `Greedy-BAA` is still a better performance guarantee than that of `RGA` (i.e., the the α-regret bound).

We further demonstrate this by running a small numerical experiment as follows: In this experiment we set $T = 10000$, $K = 10$, and the initial maximal path variation $B_T = 3$ (our results show a similar broad view for other parameter settings as well). We compare the performance of Greedy-BAA, `RGA`, and a random algorithm (which uniformly and randomly pull a feasible arm at each time step).

We consider reward vectors that have a reward of 1 for one arm and 0 for the others. We then divide the time horizon into switching blocks of fixed length. In each switching block the reward vector switches to another reward vector uniformly at random. For each fixed switching block size, We run the experiment 50 times, and plot the average performance in Figure 1 (the error bars with confidence value of 0.95 are too small that they were removed from the plots for the sake of visualisation). In particular, we plot the average collected reward value against the switching block size.

Note that for both the uniform random and `Greedy-BAA` algorithms the average reward is constant regardless of the size of the block chosen. This makes intuitive sense as `Greedy-BAA` sees the reward vector ahead of time so knows when to switch arms, whilst the random algorithm is just pulling arms at random and makes no attempt to track the best arm. Note that making the switching blocks large corresponds to reducing the variation budget. This is interesting as it seems, that as switching block length increases, `RGA` begins to approach the performance of `Greedy-BAA`. When the switching block size becomes too small the performance of `RGA` deteriorates and becomes equivalent to the performance of the uniform random policy. But in all the cases, the average performance of `RGA` is still below that of `Greedy-BAA`.

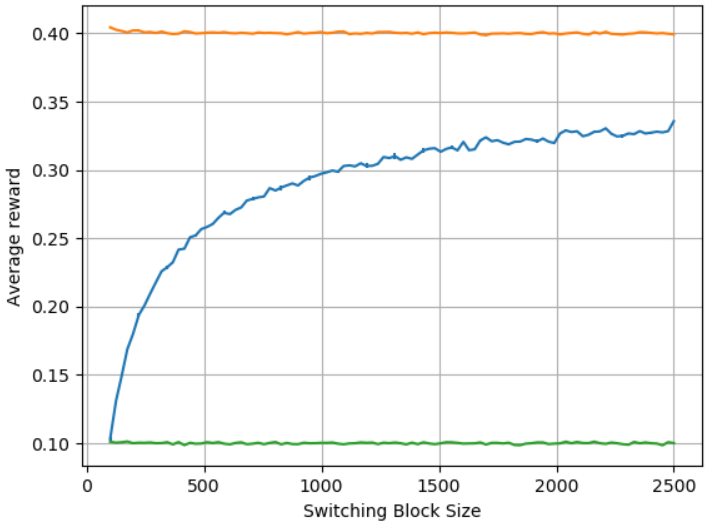

Figure 1: A performance comparison between `Greedy-BAA` (red), `RGA` (blue), and uniform random (green).