Stochastic Online Greedy Learning with Semi-bandit Feedbacks

(Full Version Including Appendices)

Tian Lin
Tsinghua University
Beijing, China
lintian06@gmail.com

Jian Li
Tsinghua University
Beijing, China
lapordge@gmail.com

Wei Chen
Microsoft Research
Beijing, China
weic@microsoft.com

Abstract

The greedy algorithm is extensively studied in the field of combinatorial optimization for decades. In this paper, we address the online learning problem when the input to the greedy algorithm is stochastic with unknown parameters that have to be learned over time. We first propose the greedy regret and -quasi greedy regret as learning metrics comparing with the performance of offline greedy algorithm. We then propose two online greedy learning algorithms with semi-bandit feedbacks, which use multi-armed bandit and pure exploration bandit policies at each level of greedy learning, one for each of the regret metrics respectively. Both algorithms achieve $O(\log T)$ problem-dependent regret bound (T being the time horizon) for a general class of combinatorial structures and reward functions that allow greedy solutions. We further show that the bound is tight in T and other problem instance parameters.

1 Introduction

The greedy algorithm is simple and easy-to-implement, and can be applied to solve a wide range of complex optimization problems, either with exact solutions (e.g. minimum spanning tree [22, 29]) or approximate solutions (e.g. maximum coverage [12] or influence maximization [19]). Moreover, for many practical problems, the greedy algorithm often serves as the first heuristic of choice and performs well in practice even when it does not provide a theoretical guarantee.

The classical greedy algorithm assumes that a certain *reward function* is given, and it constructs the solution iteratively. In *each phase*, it searches for a local optimal element to maximize the marginal gain of reward, and add it to the solution. We refer to this case as the *offline greedy* algorithm with a given reward function, and the corresponding problem the *offline problems*. The phase-by-phase process of the greedy algorithm naturally forms a *decision sequence* to illustrate the decision flow in finding the solution, which is named as the *greedy sequence*. We characterize the decision class as an *accessible set system*, a general combinatorial structure encompassing many interesting problems.

In many real applications, however, the reward function is *stochastic* and is not known in advance, and the reward is only instantiated based on the unknown distribution after the greedy sequence is selected. For example, in the influence maximization problem [19], social influence are propagated in a social network from the selected seed nodes following a stochastic model with unknown parameters, and one wants to find the optimal seed set of size k that generates the largest influence spread, which is the expected number of nodes influenced in a cascade. In this case, the reward of seed selection is only instantiated after the seed selection, and is only one of the random outcomes. Therefore, when the stochastic reward function is unknown, we aim at maximizing the expected reward overtime while gradually learning the key parameters of the expected reward functions. This

falls in the domain of *online learning*, and we refer the *online algorithm* as the strategy of the *player*, who makes sequential decisions, interacts with the environment, obtains feedbacks, and accumulates her reward. For *online greedy algorithms* in particular, at each time step the player selects and plays a candidate decision sequence while the environment instantiates the reward function, and then the player collects the values of instantiated function at every phase of the decision sequence as the feedbacks (thus the name of *semi-bandit feedbacks* [2]), and takes the value of the final phase as the reward cumulated in this step.

The typical objective for an online algorithm is to make sequential decisions against the optimal solution in the offline problem where the reward function is known a priori. For online greedy algorithms, instead, we compare it with the solution of the offline greedy algorithm, and minimize their gap of the cumulative reward over time, termed as the *greedy regret*. Furthermore, in some problems such as influence maximization, the reward function is estimated with error even for the offline problem [19] and thus the greedily selected element at each phase may contain some error. We call such greedy sequence as -*quasi greedy sequence*. To accommodate these cases, we also define the metric of -*quasi greedy regret*, which compares the online solution against the minimum offline solution from all -quasi greedy sequences.

In this paper, we propose two online greedy algorithms targeted at two regret metrics respectively. The first algorithm OG-UCB uses the stochastic multi-armed bandit (MAB) [25, 8], in particular the well-known UCB policy [3] as the building block to minimize the greedy regret. We apply the UCB policy to every phase by associating the confidence bound to each arm, and then choose the arm having the highest upper confidence bound greedily in the process of decision. For the second scenario where we allow tolerating -error for each phase, we propose a *first-explore-then-exploit* algorithm OG-LUCB to minimize the -quasi greedy regret. For every phase in the greedy process, OG-LUCB applies the LUCB policy [18, 9] which depends on the upper and lower confidence bound to eliminate arms. It first explores each arm until the lower bound of one arm is higher than the upper bound of any other arm within an -error, then the stage of current phase is switched to exploit that best arm, and continues to the next phase. Both OG-UCB and OG-LUCB achieve the problem-dependent $O(\log T)$ bound in terms of the respective regret metrics, where the coefficients in front of T depends on direct elements along the greedy sequence (a.k.a., its *decision frontier*) corresponding to the instance of learning problem. The two algorithms have complementary advantages: when we really target at greedy regret (setting to 0 for OG-LUCB), OG-UCB has a slightly better regret guarantee and does not need an artificial switch between exploration and exploitation; when we are satisfied with -quasi greedy regret, OG-LUCB works but OG-UCB cannot be adapted for this case and may suffer a larger regret. We also show a problem instance in this paper, where the upper bound is tight to the lower bound in T and other problem parameters.

We further show our algorithms can be easily extended to the knapsack problem, and applied to the stochastic online maximization for consistent functions and submodular functions, etc., in the supplementary material.

To summarize, our contributions include the following: (a) To the best of our knowledge, we are the first to propose the framework using the greedy regret and -quasi greedy regret to characterize the online performance of the stochastic greedy algorithm for different scenarios, and it works for a wide class of accessible set systems and general reward functions; (b) We propose Algorithms OG-UCB and OG-LUCB that achieve the problem-dependent $O(\log T)$ regret bound; and (c) We also show that the upper bound matches with the lower bound (up to a constant factor).

Due to the space constraint, the analysis of algorithms, applications and empirical evaluation of the lower bound are moved to the supplementary material.

Related Work. The multi-armed bandit (MAB) problem for both stochastic and adversarial settings [25, 4, 6] has been widely studied for decades. Most work focus on minimizing the cumulative regret over time [3, 16], or identifying the optimal solution in terms of pure exploration bandits [1, 18, 7]. Among those work, there is one line of research that generalizes MAB to combinatorial learning problems [8, 15, 2, 11, 24, 26, 9]. Our paper belongs to this line considering stochastic learning with semi-bandit feedbacks, while we focus on the greedy algorithm, the structure and its performance measure, which have not been addressed.

The classical greedy algorithms in the offline setting are studied in many applications [22, 29, 12, 5], and there is a line of work [17, 20] focusing on characterizing the greedy structure for solutions. We

adopt their characterizations of accessible set systems to the online setting of the greedy learning. There is also a branch of work using the greedy algorithm to solve online learning problem, while they require the knowledge of the exact form of reward function, restricting to special functions such as linear [2, 23] and submodular rewards [31, 14]. Our work does not assume the exact form, and it covers a much larger class of combinatorial structures and reward functions.

2 Preliminaries

Online combinatorial learning problem can be formulated as a repeated game between *the environment* and *the player* under stochastic multi-armed bandit framework.

Let $E = \{e_1, e_2, \dots, e_n\}$ be a finite *ground set* of size n, and $\mathcal{F}$ be a collection of subsets of E. We consider the *accessible set system* $(E, \mathcal{F})$ satisfying the following two axioms: (1) $\emptyset \in \mathcal{F}$; (2) If $S \in \mathcal{F}$ and $S \neq \emptyset$, then there exists some e in E, s.t., $S \setminus \{e\} \in \mathcal{F}$. We define any set $S \subseteq E$ as a *feasible set* if $S \in \mathcal{F}$. For any $S \in \mathcal{F}$, its *accessible set* is defined as $N(S) := \{e \in E \setminus S : S \cup \{e\} \in \mathcal{F}\}$. We say feasible set S is *maximal* if $N(S) = \emptyset$. Define the *largest length* of any feasible set as $m := \max_{S \in \mathcal{F}} |S|$ ($m \leq n$), and the *largest width* of any feasible set as $W := \max_{S \in \mathcal{F}} |N(S)|$ ($W \leq n$). We say that such an *accessible set system* $(E, \mathcal{F})$ is the *decision class* of the player. In the class of combinatorial learning problems, the size of $\mathcal{F}$ is usually very large (e.g., exponential in m, W and n).

Beginning with an empty set, the *accessible set system* $(E, \mathcal{F})$ ensures that any feasible set S can be acquired by adding elements one by one in some order (cf. Lemma A.1 in the supplementary material for more details), which naturally forms the decision process of the player. For convenience, we say the player can choose a *decision sequence*, defined as an ordered feasible sets $\tau := \langle S_0, S_1, \dots, S_k \rangle \in \mathcal{F}^{k+1}$ satisfying that $\tau = S_0 \quad S_1 \quad \dots \quad S_k$ and for any $i = 1, 2, \dots, k$, $S_i = S_{i-1} \cup \{s_i\}$ where $s_i \in N(S_{i-1})$. Besides, define decision sequence τ as *maximal* if and only if S_k is maximal.

Let Ω be an arbitrary set. The environment draws i.i.d. samples from Ω as $\omega_1, \omega_2, \dots$, at *each time* $t = 1, 2, \dots$, by following a predetermined but unknown distribution. Consider reward function $f : \mathcal{F} \times \Omega \to \mathbb{R}$ that is bounded, and it is non-decreasing[1] in the first parameter, while the exact form of function is agnostic to the player. We use a shorthand $f_t(S) := f(S, \omega_t)$ to denote the reward for any given S at time t, and denote the expected reward as $\overline{f}(S) := \mathbb{E}_{\omega_1}[f_1(S)]$, where the expectation $\mathbb{E}_{\omega_t}$ is taken from the randomness of the environment at time t. For ease of presentation, we assume that the reward function for any time t is normalized with arbitrary alignment as follows: (1) $f_t(\emptyset) = L$ (for any constant $L \geq 0$); (2) for any $S \in \mathcal{F}, e \in N(S)$, $f_t(S \cup \{e\}) - f_t(S) \in [0, 1]$. Therefore, reward function $f(\cdot, \cdot)$ is implicitly bounded within $[L, L + m]$.

We extend the concept of arms in MAB, and introduce notation $a := e|S$ to define *an arm*, representing the selected element e based on the prefix S, where S is a feasible set and $e \in N(S)$; and define $A := \{e|S : \forall S \in \mathcal{F}, \forall e \in N(S)\}$ as the *arm space*. Then, we can define the *marginal reward* for function f_t as $f_t(e|S) := f_t(S \cup \{e\}) - f_t(S)$, and the *expected marginal reward* for $\overline{f}$ as $\overline{f}(e|S) := \overline{f}(S \cup \{e\}) - \overline{f}(S)$. Notice that the use of arms characterizes the marginal reward, and also indicates that it is related to the player's previous decision.

2.1 The Offline Problem and The Offline Greedy Algorithm

In the *offline problem*, we assume that $\overline{f}$ is provided as a value oracle. Therefore, the objective is to find the optimal solution $S^* = \arg\max_{S \in \mathcal{F}} \overline{f}(S)$, which only depends on the player's decision. When the optimal solution is computationally hard to obtain, usually we are interested in finding a feasible set $S^+ \in \mathcal{F}$ such that $\overline{f}(S^+) \geq \alpha \overline{f}(S^*)$ where $\alpha \in (0, 1]$, then S^+ is called an α-*approximation* solution. That is a typical case where the greedy algorithm comes into play.

The *offline greedy* algorithm is a local search algorithm that refines the solution *phase by phase*. It goes as follows: (a) Let $G_0 = \emptyset$; (b) For each phase $k = 0, 1, \dots$, find $g_{k+1} = \arg\max_{e \in \mathcal{N}(G_k)} \overline{f}(e|G_k)$, and let $G_{k+1} = G_k \cup \{g_{k+1}\}$; (c) The above process ends when $N(G_{k+1}) = \emptyset$ (G_{k+1} is maximal). We define the maximal decision sequence $\tau^G :=$

[1] Therefore, the optimal solution is a maximal decision sequence.

$\langle G_0, G_1, \ldots, G_{m^G} \rangle$ (m^G is its length) found by the offline greedy as the *greedy sequence*. For simplicity, we assume that it is unique.

One important feature is that the greedy algorithm uses a polynomial number of calls (poly(m, W, n)) to the offline oracle, even though the size of F or A may be exponentially large.

In some cases such as the offline influence maximization problem [19], the value of $\overline{f}(\cdot)$ can only be accessed with some error or estimated approximately. Sometimes, even though $\overline{f}(\cdot)$ can be computed exactly, we may only need an approximate maximizer in each greedy phase in favor of computational efficiency (e.g., efficient submodular maximization [27]). To capture such scenarios, we say a maximal decision sequence $= \langle S_0, S_1, \ldots, S_{m'} \rangle$ is an *-quasi greedy sequence* (0), if the greedy decision can tolerate error every phase, i.e., for each $k = 0, 1, \ldots, m' - 1$ and $S_{k+1} = S_k \cup \{s_{k+1}\}$, $\overline{f}(s_{k+1} \mid S_k) \geq \max_{s \in \mathcal{N}(S_k)} \overline{f}(s \mid S_k) -$. Notice that there could be many -quasi greedy sequences, and we denote $^Q := \langle Q_0, Q_1, \ldots, Q_{m^Q} \rangle$ (m^Q is its length) as the one with the minimum reward, that is $\overline{f}(Q_{m^Q})$ is minimized over all -quasi greedy sequences.

2.2 The Online Problem

In the online case, in constrast $\overline{f}$ is not provided. The player can only access one of functions $f_1, f_2, \ldots$, generated by the environment, for each time step during a repeated game.

For each time t, the game proceeds in the following three steps: (1) The environment draws i.i.d. sample ω_t from its predetermined distribution without revealing it; (2) the player may, based on her previous knowledge, select a decision sequence $^t = \langle S_0, S_1, \ldots, S_{m^t} \rangle$, which reflects the process of her decision phase by phase; (3) then, the player plays ^{t} and gains reward $f_t(S_{m^t})$, while observes intermediate feedbacks $f_t(S_0), f_t(S_1), \ldots, f_t(S_{m^t})$ to update her knowledge. We refer such feedbacks as *semi-bandit feedbacks* in the decision order.

For any time $t = 1, 2, \ldots$, denote $^t = \langle S_0^t, S_1^t, \ldots, S_{m^t}^t \rangle$ and $S^t := S_{m^t}^t$. The player is to make sequential decisions, and the classical objective is to minimize the cumulative gap of rewards against the optimal solution [3] or the approximation solution [11]. For example, when the optimal solution $S^* = \arg\max_{S \in \mathcal{F}} \mathrm{E}[f_1(S)]$ can be solved in the offline problem, we minimize the expected *cumulative regret* $R(T) := T \cdot \mathrm{E}[f_1(S^*)] - \sum_{t=1}^{T} \mathrm{E}[f_t(S^t)]$ over the time horizon T, where the expectation is taken from the randomness of the environment and the possible random algorithm of the player. In this paper, we are interested in online algorithms that are comparable to the solution of the offline greedy algorithm, namely the greedy sequence $^G = \langle G_0, G_1, \ldots, G_{m^G} \rangle$. Thus, the objective is to minimize the *greedy regret* defined as

$$R^G(T) := T \cdot \mathrm{E}[f_1(G_{m^G})] - \sum_{t=1}^{\overline{X}} \mathrm{E}\left[f_t(S^t)\right] : \tag{1}$$

Given 0, we define the *-quasi greedy regret* as

$$R^Q(T) := T \cdot \mathrm{E}[f_1(Q_{m^Q})] - \sum_{t=1}^{\overline{X}} \mathrm{E}\left[f_t(S^t)\right] ; \tag{2}$$

where $^Q = \langle Q_0, Q_1, \ldots, Q_{m^Q} \rangle$ is the minimum -quasi greedy sequence.

We remark that if the offline greedy algorithm provides an -approximation solution (with $0 <$ 1), then the greedy regret (or -quasi greedy regret) also provides -approximation regret, which is the regret comparing to the fraction of the optimal solution, as defined in [11].

In the rest of the paper, our goal is to design the player's policy that is comparable to the offline greedy, in other words, $R^G(T) = T \cdot \overline{f}(G_{m^G}) - \frac{1}{T} \sum_{t=1}^{T} \mathrm{E}[f_t(S^t)] = o(1)$. Thus, to achieve sublinear greedy regret $R^G(T) = o(T)$ is our main focus.

3 The Online Greedy and Algorithm OG-UCB

In this section, we propose our Online Greedy (OG) algorithm with the UCB policy to minimize the greedy regret (defined in (1)).

Algorithm 1 OG

Require: MaxOracle

1: **for** $t = 1, 2, \ldots$ **do**
2: $S_0 \leftarrow \varnothing$; $k \leftarrow 0$; $h_0 \leftarrow$ true
3: **repeat** . online greedy procedure
4: $A \leftarrow \{e \mid S_k : \forall e \in N(S_k)\}$; $t' \leftarrow \sum_{a \in A} N(a) + 1$
5: $(s_{k+1} \mid S_k; h_k) \leftarrow$ MaxOracle$(A; \hat{X}(\cdot); N(\cdot); t')$. find the current maximal
6: $S_{k+1} \leftarrow S_k \cup \{s_{k+1}\}$; $k \leftarrow k + 1$
7: **until** $N(S_k) = \varnothing$. until a maximal sequence is found
8: Play sequence $\pi^t \leftarrow \langle S_0, \ldots, S_k \rangle$, observe $\{f_t(S_0), \ldots, f_t(S_k)\}$, and gain $f_t(S_k)$.
9: **for all** $i = 1, 2, \ldots, k$ **do** . update according to signals from MaxOracle
10: **if** $h_0, h_1, \ldots, h_{i-1}$ **are all** true **then**
11: Update $\hat{X}(s_i \mid S_{i-1})$ and $N(s_i \mid S_{i-1})$ according to (3).

Subroutine 2 UCB$(A; \hat{X}(\cdot); N(\cdot); t)$ to implement MaxOracle

Setup: confidence radius $\mathrm{rad}_t(a) := \sqrt{\frac{3 \ln t}{2N(a)}}$, for each $a \in A$

1: **if** $\exists a \in A$, $\hat{X}(a)$ is not initialized **then** . break ties arbitrarily
2: **return** $(a;$ true$)$. to initialize arms
3: **else** . apply UCB's rule
4: $I_t^+ \leftarrow \arg\max_{a \in A} \left\{ \hat{X}(a) + \mathrm{rad}_t(a) \right\}$, and **return** $(I_t^+;$ true$)$

For any arm $a = e \mid S \in A$, playing a at each time t yields the marginal reward as a random variable $X_t(a) = f_t(a)$, in which the random event $\omega_t \in \Omega$ is i.i.d., and we denote $\mu(a)$ as its true mean (i.e., $\mu(a) := \mathrm{E}[X_1(a)]$). Let $\hat{X}(a)$ be the empirical mean for the marginal reward of a, and $N(a)$ be the counter of the plays. More specifically, denote $\hat{X}_t(a)$ and $N_t(a)$ for particular $\hat{X}(a)$ and $N(a)$ at the beginning of the time step t, and they are evaluated as follows:

$$\hat{X}_t(a) = \frac{\sum_{i=1}^{t-1} f_i(a) \mathbb{I}_i(a)}{\sum_{i=1}^{t-1} \mathbb{I}_i(a)}; \quad N_t(a) = \sum_{i=1}^{t-1} \mathbb{I}_i(a); \tag{3}$$

where $\mathbb{I}_i(a) \in \{0, 1\}$ indicates whether a is updated at time i. In particular, assume that our algorithm is lazy-initialized so that each $\hat{X}(a)$ and $N(a)$ is 0 by default, until a is played.

The *Online Greedy* algorithm (OG) proposed in Algorithm 1 serves as a meta-algorithm allowing different implementations of Subroutine MaxOracle. For every time t, OG calls MaxOracle (Line 5, to be specified later) to find the local maximal phase by phase, until the decision sequence π^t is made. Then, it plays sequence π^t, observes feedbacks and gains the reward (Line 8). Meanwhile, OG collects the Boolean signals (h_k) from MaxOracle during the greedy process (Line 5), and update estimators $\hat{X}(\cdot)$ and $N(\cdot)$ according to those signals (Line 10). On the other hand, MaxOracle takes accessible arms A, estimators $\hat{X}(\cdot); N(\cdot)$, and counted time t', and returns an arm from A and signal $h_k \in \{$true$;$ false$\}$ to instruct OG whether to update estimators for the following phase.

The classical UCB [3] can be used to implement MaxOracle, which is described in Subroutine 2. We term our algorithm OG, in which MaxOracle is implemented by Subroutine 2 UCB, as Algorithm OG-UCB. A few remarks are in order: First, Algorithm OG-UCB chooses an arm with the highest upper confidence bound for each phase. Second, the signal h_k is *always* true, meaning that OG-UCB always update empirical means of arms along the decision sequence. Third, because we use lazy-initialized $\hat{X}(\cdot)$ and $N(\cdot)$, the memory is allocated only when it is needed.

3.1 Regret Bound of OG-UCB

For any feasible set S, define the *greedy element for* S as $g_S^* := \arg\max_{e \in \mathcal{N}(S)} \overline{f}(e \mid S)$, and we use $N_-(S) := N(S) \cap \{g_S^*\}$ for convenience. Denote $\mathcal{F}^\dagger := \{S \in \mathcal{F} : S$ is maximal$\}$ as the collection

of all maximal feasible sets in F. We use the following gaps to measure the performance of the algorithm.

Definition 3.1 (Gaps)**.** The *gap* between the maximal greedy feasible set G_{m^G} and any $S \in F$ is defined as $(S) := \overline{f}(G_{m^G}) - \overline{f}(S)$ if it is positive, and 0 otherwise. We define the *maximum gap* as $_{\max} = \overline{f}(G_{m^G}) - \min_{S \in \mathcal{F}^y} \overline{f}(S)$, which is the worst penalty for any maximal feasible set. For any arms $a = ejS \in A$, we define the *unit gap* of a (i.e., the gap for one phase) as

$$(a) = (ejS) := \begin{array}{ll} \overline{f}(g^*_S jS) - \overline{f}(ejS); & e \neq g^*_S \\ \overline{f}(g^*_S jS) - \max_{e' \in \mathcal{N}(S)} \overline{f}(e' jS); & e = g^*_S \end{array}. \tag{4}$$

For any arms $a = ejS \in A$, we define the *sunk-cost gap* (irreversible once selected) as

$$^*(a) = {}^*(ejS) := \max\left\{ \overline{f}(G_{m^G}) - \min_{V : V \in \mathcal{F}^y; S \cup \{e\} \prec V} \overline{f}(V); 0 \right\}; \tag{5}$$

where for two feasible sets A and B, $A \prec B$ means that A is a prefix of B in some decision sequence, that is, there exists a decision sequence $= hS_0 = ;; S_1; : : : ; S_k i$ such that $S_k = B$ and for some $j < k$, $S_j = A$. Thus, $^*(ejS)$ means the largest gap we may have after we have fixed our prefix selection to be $S \cup \{e\}$, and is upper bounded by $_{\max}$.

Definition 3.2 (Decision frontier)**.** For any decision sequence $= hS_0; S_1; : : : ; S_k i$, define *decision frontier* $( ) := \bigcup_{i=1}^{k} \{ejS_{i-1} : e \in N(S_{i-1})\} \setminus A$ as the arms need to be explored in the decision sequence , and $_-( ) := \bigcup_{i=1}^{k} \{ejS_{i-1} : \forall e \in N_-(S_{i-1})\}$ similarly.

Theorem 3.1 (Greedy regret bound)**.** *For any time T, Algorithm OG-UCB (Algorithm 1 with Subroutine 2) can achieve the greedy regret*

$$R^G(T) \leq \sum_{a \in \Gamma(^G)} \left(\frac{6\ ^*(a)\ \ln T}{(a)^2} + \frac{^2}{3} + 1 \right) ^*(a); \tag{6}$$

where ^{G} is the greedy decision sequence.

When $m = 1$, the above theorem immediately recovers the regret bound of the classical UCB [3] (with $^*(a) = (a)$). The greedy regret is bounded by $O\left(\frac{mW\Delta_{\max}\log T}{\Delta^2}\right)$ where is the minimum unit gap ($= \min_{a \in \mathcal{A}} (a)$), and the memory cost is at most proportional to the regret. For a special class of linear bandits, a simple extension where we treat arms ejS and ejS' as the same can make OG-UCB essentially the same as OMM in [23], while the regret is $O(\frac{n}{\Delta}\log T)$ and the memory cost is $O(n)$ (cf. Appendix F.1 of the supplementary material).

4 Relaxing the Greedy Sequence with ϵ-Error Tolerance

In this section, we propose an online algorithm called OG-LUCB, which learns an -quasi greedy sequence, with the goal of minimizing the -quasi greedy regret (in (2)). We learn -quasi-greedy sequences by a *first-explore-then-exploit policy*, which utilizes results from PAC learning with a fixed confidence setting. In Section 4.1, we implement MaxOracle via the LUCB policy, and derive its exploration time; we then assume the knowledge of time horizon T in Section 4.2, and analyze the -quasi greedy regret; and in Section 4.3, we show that the assumption of knowing T can be further removed.

4.1 OG with a first-explore-then-exploit policy

Given 0 and failure probability $2 (0; 1)$, we use Subroutine 3 LUCB $_{;}$ to implement the subroutine MaxOracle in Algorithm OG. We call the resulting algorithm OG-LUCB $_{;}$. Specifically, Subroutine 3 is adapted from CLUCB-PAC in [9], and specialized to explore the top-one element in the support of $[0; 1]$ (i.e., set $R = \frac{1}{2}$, width$(M) = 2$ and Oracle $=$ arg max in [9]). Assume that $I^{\text{exploit}}( )$ is lazy-initialized. For each greedy phase, the algorithm first explores each arm in A in the *exploration* stage, during which the return flag (the second return field) is always false; when the optimal one is found (initialize $I^{\text{exploit}}(A)$ with $\hat{f}_t$), it sticks to $I^{\text{exploit}}(A)$ in the *exploitation* stage

Subroutine 3 LUCB$_{;}$ $(A; \hat{X}(\,); N(\,); t)$ to implement MaxOracle

Setup: $\mathrm{rad}_t(a) := \sqrt{\frac{\ln(4Wt^3/\,)}{2N(a)}}$, for each $a \in A$; $I^{\text{exploit}}(\,)$ to cache arms for exploitation;

1: **if** $I^{\text{exploit}}(A)$ is initialized **then return** $(I^{\text{exploit}}(A); \text{true})$. in the *exploitation* stage
2: **if** $\exists a \in A$, $\hat{X}(a)$ is not initialized **then** . break ties arbitrarily
3: **return** $(a; \text{false})$. to initialize arms
4: **else**
5: $\hat{I}_t \leftarrow \arg\max_{a \in A} \hat{X}(a)$
6: $\forall a \in A$, $X'(a) \leftarrow \begin{cases} \hat{X}(a) + \mathrm{rad}_t(a); & a \neq \hat{I}_t \\ \hat{X}(a) - \mathrm{rad}_t(a); & a = \hat{I}_t \end{cases}$. perturb arms
7: $I'_t \leftarrow \arg\max_{a \in A} X'(a)$
8: **if** $X'(I'_t) - X'(\hat{I}_t) > \,$ **then** . not separated
9: $I''_t \leftarrow \arg\max_{i \in \{\hat{I}_t, I'_t\}} \mathrm{rad}_t(i)$, and **return** $(I''_t; \text{false})$. in the *exploration* stage
10: **else** . separated
11: $I^{\text{exploit}}(A) \leftarrow \hat{I}_t$. initialize $I^{\text{exploit}}(A)$ with $\hat{I}_t$
12: **return** $(I^{\text{exploit}}(A); \text{true})$. in the *exploitation* stage

for the subsequent time steps, and return flag for this phase becomes `true`. The main algorithm OG then uses these flags in such a way that it updates arm estimates for phase i if any only if all phases for $j < i$ are already in the exploitation stage. This avoids maintaining useless arm estimates and is a major memory saving comparing to OG-UCB.

In Algorithm OG-LUCB$_{;}$, we define the *total exploration time* $T^{\mathrm{E}} = T^{\mathrm{E}}(\,)$, such that for any time $t \geq T^{\mathrm{E}}$, OG-LUCB$_{;}$ is in the exploitation stage for all greedy phases encountered in the algorithm. This also means that after time T^{E}, in every step we play the same maximal decision sequence $= \langle S_0; S_1; \ldots; S_k \rangle \in \mathcal{F}^{k+1}$, which we call a *stable sequence*. Following a common practice, we define the *hardness coefficient* with prefix $S \in \mathcal{F}$ as

$$H_S := \sum_{e \in \mathcal{N}(S)} \frac{1}{\max\{\Delta(e|S)^2; \,^2\}}; \text{ where } \Delta(e|S) \text{ is defined in (4):} \tag{7}$$

Rewrite definitions with respect to the -quasi regret. Recall that $^{\mathrm{Q}} = \langle Q_0; Q_1; \ldots; Q_{m^{\mathrm{Q}}} \rangle$ is the minimum -quasi greedy sequence. In this section, we rewrite the gap $\Delta(S) := \max\{\overline{f}(Q_{m^{\mathrm{Q}}}) - \overline{f}(S); 0\}$ for any $S \in \mathcal{F}$, the maximum gap $_{\max} := \overline{f}(Q_{m^{\mathrm{Q}}}) - \min_{S \in \mathcal{F}^{\chi}} \overline{f}(S)$, and $^*(a) = \,^*(e|S) := \max\{\overline{f}(Q_{m^{\mathrm{Q}}}) - \min_{V: V \in \mathcal{F}^{\chi}; S \cup \{e\} \prec V} \overline{f}(V); 0\}$, for any arm $a = e|S \in A$.

The following theorem shows that, with high probability, we can find a stable -quasi greedy sequence, and the total exploration time is bounded.

Theorem 4.1 (High probability exploration time)**.** *Given any* $\in (0; 1)$, *suppose after the total exploration time* $T^{\mathrm{E}} = T^{\mathrm{E}}(\,)$, *Algorithm OG-LUCB*$_{;}$ *(Algorithm 1 with Subroutine 3) sticks to a stable sequence* $= \langle S_0; S_1; \ldots; S_{m'} \rangle$ *where* m' *is its length. With probability at least* $1 - m$, *the following claims hold: (1)* $\in \mathcal{F}^{m'+1}$ *is an -quasi greedy sequence; (2) The total exploration time satisfies that* $T^{\mathrm{E}} \leq 127 \sum_{k=0}^{m'-1} H_S \ln(1996WH_S/\,)$;

4.2 Time Horizon T is Known

Knowing time horizon T, we may let $= \frac{1}{T}$ in OG-LUCB$_{;}$ to derive the -quasi regret as follows.

Theorem 4.2. *Given any* > 0. *When total time* T *is known, let Algorithm OG-LUCB*$_{;}$ *run with* $= \frac{1}{T}$. *Suppose* $= \langle S_0; S_1; \ldots; S_{m'} \rangle \in \Pi^{\mathrm{O}}$ *is the sequence selected at time* T. *Define function* $R^{\mathrm{Q};}(T) := \sum_{e|S \in \Gamma(\,)} \,^*(e|S) \min\{\frac{127}{\Delta(e|S)^2}; \frac{113}{^2}\} \ln(1996WH_ST) + \,_{\max} m$; *where* m *is the largest length of a feasible set and* H_S *is defined in* (7). *Then, the -quasi regret satisfies that* $R^{\mathrm{Q}}(T) \leq R^{\mathrm{Q};}(T) = O(\frac{WmH_{\max}}{\max\{\Delta^2; \,^2\}} \log T)$, *where* *is the minimum unit gap.*

In general, the two bounds (Theorem 3.1 and Theorem 4.2) are for different regret metrics, thus can not be directly compared. When $= 0$, OG-UCB is slightly better only in the constant before $\log T$.

Algorithm 4 OG-LUCB-R (i.e., OG-LUCB with Restart)

Require:

1: **for** epoch $\ell = 1, 2, \ldots$ **do**

2: Clean $\hat{X}(\cdot)$ and $N(\cdot)$ for all arms, and restart OG-LUCB$_{\gamma}$ with $\gamma = \frac{1}{\ell}$ (defined in (8)).

3: Run OG-LUCB$_{\gamma}$ for τ_ℓ time steps. (exit halfway, if the time is over.)

On other hand, when we are satisfied with γ-quasi greedy regret, OG-LUCB$_{\gamma}$ may work better for some large γ, for the bound takes the maximum (in the denominator) of the problem-dependent term $\Delta(e|S)$ and the fixed constant γ term, and the memory cost is only $O(mW)$.

4.3 Time Horizon T is not Known

When time horizon T is not known, we can apply the "squaring trick", and restart the algorithm for each epoch as follows. Define *the duration of epoch* ℓ as τ_ℓ, and its *accumulated time* as σ_ℓ, where

$$\tau_\ell := e^{2^\ell}; \qquad \sigma_\ell := \begin{cases} 0, & \ell = 0 \\ \sum_{s=1}^{\ell} \tau_s, & \ell \geq 1 \end{cases}. \tag{8}$$

For any time horizon T, define the *final epoch* $K = K(T)$ as the epoch where T lies in, that is $\sigma_{K-1} < T \leq \sigma_K$. Then, our algorithm OG-LUCB-R is proposed in Algorithm 4. The following theorem shows that the $O(\log T)$ γ-quasi regret still holds, with a slight blowup of the constant hidden in the big O notation (For completeness, the explicit constant before $\log T$ can be found in Theorem D.7 of the supplementary material).

Theorem 4.3. *Given any $\gamma \geq 0$. Use τ_ℓ and σ_ℓ defined in (8), and function $R^{Q,\gamma}(T)$ defined in Theorem 4.2. In Algorithm OG-LUCB-R, suppose $\Gamma^{(\ell)} = \langle S_0^{(\ell)}, S_1^{(\ell)}, \ldots, S_{m^{(\ell)}}^{(\ell)} \rangle$ is the sequence selected by the end of ℓ-th epoch of OG-LUCB$_{\gamma}$, where $m^{(\ell)}$ is its length. For any time T, denote final epoch as $K = K(T)$ such that $\sigma_{K-1} < T \leq \sigma_K$, and the γ-quasi regret satisfies that $R^Q(T) \leq \sum_{\ell=1}^{K} R^{Q,\gamma^{(\ell)}}(\tau_\ell) = O\left(\frac{Wm\Delta_{\max}}{\max\{\Delta^2, \gamma^2\}} \log T\right)$, where Δ is the minimum unit gap.*

5 Lower Bound on the Greedy Regret

Consider a problem of selecting one element each from m bandit instances, and the player sequentially collects prize at every phase. For simplicity, we call it the *prize-collecting problem*, which is defined as follows: For each bandit instance $i = 1, 2, \ldots, m$, denote set $E_i = \{e_{i,1}, e_{i,2}, \ldots, e_{i,W}\}$ of size W. The accessible set system is defined as (E, F), where $E = \bigcup_{i=1}^{m} E_i$, $F = \bigcup_{i=1}^{m} F_i \cup \{\emptyset\}$, and $F_i = \{S \subseteq E : |S| = i, \forall k : 1 \leq k \leq i, |S \setminus E_k| = 1\}$. The reward function $f : F \to [0, m]$ is non-decreasing in the first parameter, and the form of f is unknown to the player. Let minimum unit gap $\Delta := \min\{\overline{f}(g_S^* | S) - \overline{f}(e|S) : \forall S \in F, \forall e \in N_-(S)\} > 0$, where its value is also unknown to the player. The objective of the player is to minimize the greedy regret.

Denote the greedy sequence as $\Gamma^G = \langle G_0, G_1, \ldots, G_m \rangle$, and the greedy arms as $A^G = \{g^*_{G_{i-1}} | G_{i-1} : \forall i = 1, 2, \ldots, W\}$. We say an algorithm is *consistent*, if the sum of playing all arms $a \in A \setminus A^G$ is in $o(T^\alpha)$, for any $\alpha > 0$, i.e., $\mathrm{E}[\sum_{a \in A \setminus A^G} N_T(a)] = o(T^\alpha)$.

Theorem 5.1. *For any consistent algorithm, there exists a problem instance of the prize-collecting problem, as time T tends to ∞, for any minimum unit gap $\Delta \in (0, \frac{1}{4})$, such that $\Delta^2 \leq \frac{2}{3W^{\xi_m - 1}}$ for some constant $\xi \in (0, 1)$, the greedy regret satisfies that $R^G(T) = \Omega\left(\frac{mW \ln T}{\Delta^2}\right)$.*

We remark that the detailed problem instance and the greedy regret can be found in Theorem E.2 of the supplementary material. Furthermore, we may also restrict the maximum gap $\Delta_{\max}$ to $O(1)$, and the lower bound $R^G(T) = \Omega\left(\frac{mW\Delta_{\max} \ln T}{\Delta^2}\right)$, for any sufficiently large T. For the upper bound, OG-UCB (Theorem 3.1) gives that $R^G(T) = O(\frac{mW\Delta_{\max}}{\Delta^2} \log T)$, Thus, our upper bound of OG-UCB matches the lower bound within a constant factor.

Acknowledgments Jian Li was supported in part by the National Basic Research Program of China grants 2015CB358700, 2011CBA00300, 2011CBA00301, and the National NSFC grants 61202009, 61033001, 61361136003.

References

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

[2] J.-Y. Audibert, S. Bubeck, and G. Lugosi. Minimax policies for combinatorial prediction games. *arXiv preprint arXiv:1105.4871*, 2011.

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

[4] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire. The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing*, 32(1):48–77, 2002.

[5] A. Björner and G. M. Ziegler. Introduction to greedoids. *Matroid applications*, 40:284–357, 1992.

[6] S. Bubeck and N. Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *arXiv preprint arXiv:1204.5721*, 2012.

[7] S. Bubeck, R. Munos, and G. Stoltz. Pure exploration in finitely-armed and continuous-armed bandits. *Theoretical Computer Science*, 412(19):1832–1852, 2011.

[8] N. Cesa-Bianchi and G. Lugosi. Combinatorial bandits. *Journal of Computer and System Sciences*, 78 (5):1404–1422, 2012.

[9] S. Chen, T. Lin, I. King, M. R. Lyu, and W. Chen. Combinatorial pure exploration of multi-armed bandits. In *NIPS*, 2014.

[10] W. Chen, L. V. Lakshmanan, and C. Castillo. *Information and Influence Propagation in Social Networks*, volume 5. Morgan & Claypool, 2013.

[11] W. Chen, Y. Wang, and Y. Yuan. Combinatorial multi-armed bandit: General framework and applications. In *ICML*, 2013.

[12] V. Chvatal. A greedy heuristic for the set-covering problem. *Mathematics of operations research*, 4(3): 233–235, 1979.

[13] N. Du, Y. Liang, M. Balcan, and L. Song. Influence function learning in information diffusion networks. In *ICML*, 2014.

[14] V. Gabillon, B. Kveton, Z. Wen, B. Eriksson, and S. Muthukrishnan. Adaptive submodular maximization in bandit setting. In *NIPS*. 2013.

[15] Y. Gai, B. Krishnamachari, and R. Jain. Learning multiuser channel allocations in cognitive radio networks: A combinatorial multi-armed bandit formulation. In *DySPAN*. IEEE, 2010.

[16] A. Garivier and O. Cappé. The kl-ucb algorithm for bounded stochastic bandits and beyond. *arXiv preprint arXiv:1102.2490*, 2011.

[17] P. Helman, B. M. Moret, and H. D. Shapiro. An exact characterization of greedy structures. *SIAM Journal on Discrete Mathematics*, 6(2):274–283, 1993.

[18] S. Kalyanakrishnan, A. Tewari, P. Auer, and P. Stone. Pac subset selection in stochastic multi-armed bandits. In *ICML*, 2012.

[19] D. Kempe, J. Kleinberg, and É. Tardos. Maximizing the spread of influence through a social network. In *SIGKDD*, 2003.

[20] B. Korte and L. Lovász. Greedoids and linear objective functions. *SIAM Journal on Algebraic Discrete Methods*, 5(2):229–238, 1984.

[21] A. Krause and D. Golovin. Submodular function maximization. *Tractability: Practical Approaches to Hard Problems*, 3:19, 2012.

[22] J. B. Kruskal. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical society*, 7(1):48–50, 1956.

[23] B. Kveton, Z. Wen, A. Ashkan, H. Eydgahi, and B. Eriksson. Matroid bandits: Fast combinatorial optimization with learning. *arXiv preprint arXiv:1403.5045*, 2014.

[24] B. Kveton, Z. Wen, A. Ashkan, and C. Szepesvari. Tight regret bounds for stochastic combinatorial semi-bandits. *arXiv preprint arXiv:1410.0949*, 2014.

[25] T. L. Lai and H. Robbins. Asymptotically efficient adaptive allocation rules. *Advances in applied mathematics*, 6(1):4–22, 1985.

[26] T. Lin, B. Abrahao, R. Kleinberg, J. Lui, and W. Chen. Combinatorial partial monitoring game with linear feedback and its applications. In *ICML*, 2014.

[27] B. Mirzasoleiman, A. Badanidiyuru, A. Karbasi, J. Vondrak, and A. Krause. Lazier than lazy greedy. In *Proc. Conference on Artificial Intelligence (AAAI)*, 2015.

[28] G. L. Nemhauser and L. A. Wolsey. Best algorithms for approximating the maximum of a submodular set function. *Mathematics of operations research*, 3(3):177–188, 1978.

[29] R. C. Prim. Shortest connection networks and some generalizations. *Bell system technical journal*, 36(6): 1389–1401, 1957.

[30] M. G. Rodriguez and B. Schölkopf. Influence maximization in continuous time diffusion networks. *arXiv preprint arXiv:1205.1682*, 2012.

[31] M. Streeter and D. Golovin. An online algorithm for maximizing submodular functions. In *NIPS*, 2009.

[32] N. White. *Matroid applications*, volume 40. Cambridge University Press, 1992.

Symbols in the main text	*Definition*
$(E; F)$	the ground set, and the collection of all feasible sets in Section 2
$N(S)$	the accessible set from prefix S in Section 2
n	the size of ground set $\|E\|$ in Section 2
m	the maximal length of any feasible set in Section 2
W	the maximal width of any accessible set in Section 2
$a = e\|S$, A	one arm, and the arm space in Section 2
$= \langle S_0;$ $; S_k \rangle$	a decision sequence in Section 2
$f(\ ;\ )$, $f_t(\ )$	the reward function, and its shorthand in Section 2
$\overline{f}(\ )$	the expected reward function in Section 2
$^{G} = \langle G_0;$ $; G_{m^G} \rangle$	the greedy sequence where m^G is its length in Section 2.1
$^{Q} = \langle Q_0;$ $; Q_{m^Q} \rangle$	the minimum -quasi greedy sequence where m^Q is its length in Section 2.1
$(\ )$	the decision frontier of in Section 3.1
$_{-}(\ )$	the decision frontier of , excluding all greedy elements in Section 3.1
$R(T)$	the cumulative regret in Section 2.2
$R^G(T)$, $R^Q(T)$,	the greedy regret ((1)), the -quasi greedy regret ((2))
$R\ (T)$	the -approximation regret discussed in Section 2.2 (formally defined in Definition B.1)
$F^{\dagger}$	the collection of all maximal feasible sets in Section 3.1
$\hat{X}(a)$, $N(a)$, $X(a)$	the mean estimator of $\{f_t(a)\}_{t=1}^{\infty}$, the counter, and the true mean in Section 3
$\hat{X}_t(a)$, $N_t(a)$	the particular $\hat{X}(a)$ and $N(a)$ at the beginning of the time step t, in Section 3
g_S^*	the greedy element of prefix S in Section 3.1
$N_{-}(S)$	the accessible set from prefix S, excluding the greedy element, in Section 3.1
(S)	the gap between ^{G} (or ^{Q}) and S in Definition 3.1 (or rewritten in Section 4.1)
$^{\max}$	the maximum gap of (S) defined in Definition 3.1 (or rewritten in Section 4.1)
(a) (or $(e\|S)$)	the unit gap of arm $a = e\|S$ defined in (4)
$^{*}(a)$ (or $^{*}(e\|S)$)	the maximum gap of selecting $a = e\|S$ defined in (5) (rewritten in Section 4.1)
H_S	the hardness coefficient in Section 4.1
T^{E}	the total exploration time for OG-LUCB until a stable sequence is found
$N_t^{E}(a)$	the counter of playing arm a during the exploration before time t in Section 4.1
ℓ, ℓ	the duration of ℓ-th epoch, and its accumulated time defined in (8)
$K = K(T)$	the final epoch of time horizon T defined in Section 4.3
$R^{Q;}\ (T)$	an intermediate form of the -quasi regret for sequence defined in Theorem 4.2

Table 1: List of symbols in the main text.

Appendix

The appendix is organized as follows.

Above all, for ease of reading, we list the symbols and their definitions in the main text in Table 1.

In Appendix A, we show that any feasible set in F is accessible by a decision sequence, and demonstrate the concrete decision classes in our model with a few examples.

In Appendix B, we formally define *-approximation regret*, and give some simple propositions to show that such a regret can be derived immediately from the greedy regret and the -quasi greedy regret if the offline solution is an -approximation solution.

In Appendix C, we first establish lemmas by the set decomposition for Algorithm OG-UCB, and prove that OG-UCB can achieve the problem-dependent $O(\log T)$ greedy regret.

In Appendix D, we relax the greedy sequence to tolerate -error. We prove the high-probability exploration time of for Algorithm OG-LUCB in the beginning. By letting $= \frac{1}{T}$ with the known time horizon T, we show that the first-explore-then-exploit policy of Algorithm OG-LUCB (associate $= 1/T$) achieves the problem-dependent $O(\log T)$ -quasi regret bound. Then, we show that Algorithm OG-LUCB-R utilizing OG-LUCB can remove the dependence on T, and the $O(\log T)$ bound holds with a slight compensation for its constant.

In Appendix E, we construct a problem instance of the prize-collecting problem, and show the upper bound is tight with the lower bound (up to a constant factor).

In Appendix F, we show a simple extension to recover $O(\frac{n}{\Delta} \log T)$ bound for a linear bandit with matroid constraints, and discuss an extension of our model to the Knapsack problem.

In Appendix G, we first apply our algorithms to the top-m selection problem with a consistent function and the online submodular problem, and show the -approximation regret when the offline greedy yields an -approximation solution. Then we discuss the application to the online version of the influence maximization problem and the probabilistic set cover problem, which may remove the assumption on one particular diffusion model and avoid the issue of model misspecification.

In Appendix H, we evaluate the lower bound for the prize-collecting problem, and compare the regret of OG-UCB and the lower bound numerically.

A The Accessible Set System (E, F)

An accessible set system $(E; F)$ (in Section 2) satisfies two axioms:

(A1: Triviality axiom). $; \in F$;

(A2: Accessibility axiom). If $S \in F$ and $S \neq ;$, then there exists some e in E, s.t., $S \setminus \{e\} \in F$.

We claim that any set $S \in F$ can be obtained by adding one element at a time from the empty set. Formally, we have the following fact.

Lemma A.1 (Accessibility of any feasible set)**.** *For any set $S \in F$, denote $k = |S|$. There exists a sequence $\langle S_0; S_1; \ldots; S_k \rangle$, such that: (1) $S_0; S_1; \ldots; S_k \in F$; (2) $; = S_0 \subset S_1 \subset \cdots \subset S_k = S$; (3) $|S_i| - |S_{i-1}| = 1$, for each $i = 1; 2; \ldots; k$.*

Proof. If $S = ;$, it is obvious that the sequence $\{S_0\}$ satisfies the above three properties, therefore we only focus on a non-empty S, that is $k \geq 1$.

For any set $S \in F$, let $S_k = S$. From Axiom A2, we know that there exists some $e \in E$, $S_k \setminus \{e\} \in F$, thus we can denote $S_{k-1} = S_k \setminus \{e\}$. We can carry on this procedure, and apply Axiom A2 iteratively. Then we can get S_i, for each $i = k - 1; k - 2; \ldots; 0$. Therefore, the sequence $\langle S_0; S_1; \ldots; S_k \rangle$ we found satisfies the three properties, which ends the proof. □

Our characterization of accessible set system $(E; F)$ encompasses greedoids and matroids as special cases, which is studied in the previous literature [5, 17]. More specifically, from [17], we know that:

Definition A.1 (Greedoid)**.** A greedoid is an accessible set system $(E; F)$ satisfying:

(A3: Augmentation axiom). For all $S; T \in F$ such that $|S| > |T|$, there is an $x \in S \setminus T$ such that $T \cup \{x\} \in F$.

Definition A.2 (Matroid)**.** A matroid is a greedoid $(E; F)$ satisfying:

(A4: Hereditary axiom). If $S \in F$ and $T \subset S$, then $T \in F$.

In addition, we list a few concrete examples that can fit into our model.

Example A.1 (Top-m selection)**.** In accessible set system $(E; F)$, F contains all sets $S \subset E$, where $|S| \leq m$. Therefore, a maximal feasible set is a subset of size m.

Example A.2 (Spanning Tree)**.** Given a graph $G = (V; E)$, a forest F in G is a subset of edges in E that does not contains a cycle. The corresponding accessible set system is $(E; F)$, where $F = \{F \subset E : F \text{ is a forest}\}$. The maximal set of a spanning tree constraint is a forest that cannot include more edges.

Example A.3 (Gaussian Elimination Greedoids [32])**.** Let $M = (m_{ij}) \in \mathbb{K}^{m \times n}$ be a $n \times m$ matrix over an arbitrary field $\mathbb{K}$. The accessible set system is $(E; F)$ where $E = \{1; 2; \ldots; n\}$ and $F = \{A \subset E : \text{the submatrix } M_{\{1;2;\cdots;|A|\};A} \text{ is non-singular}\}$. It corresponds to the procedure of performing Gaussian Elimination on rows, which give rise to the sequence of column indices. A maximal feasible set A makes a submatrix that with the same rank as matrix M.

Example A.4 (Knapsack)**.** Let the ground set be $E = \{e_1, e_2, \ldots, e_n\}$ and the cost function be $c : E \to \mathbb{R}_{>0}$. Given the budget B for the knapsack problem, the accessible set system is (E, F), where $F = \{S \subseteq E : \sum_{e \in S} c(e) \leq B\}$. (E, F) corresponding to the knapsack problem is neither a greedoid nor a matroid, but nevertheless still an accessible set system. A maximal feasible set S is the one that cannot include any more element in $E \setminus S$ without violating the budget constraint, and maximal feasible sets may have different lengths.

B Some Simple Propositions

We first formally define α*-approximation regret*, the regret comparing to the α fraction of the optimal solution (as defined in [11]), in the following.

Definition B.1 (α-approximation regret)**.** Let $S^* = \arg\max_{S \in \mathcal{F}} \mathbb{E}[f_1(S)]$, and denote $\pi^t = \langle S_0^t, S_1^t, \ldots, S_{m^t}^t \rangle$, where $S^t := S_{m^t}^t$, as the decision sequence selected for any time $t = 1, 2, \ldots, T$. We define the α-approximation regret ($0 < \alpha \leq 1$) as

$$R_\alpha(T) := T \cdot \alpha \cdot \mathbb{E}[f_1(S^*)] - \sum_{t=1}^{T} \mathbb{E}[f_t(S^t)]. \tag{9}$$

Then, the next two simple propositions show that the α-approximation regret can be derived from the greedy regret and α-quasi greedy regret if the offline greedy algorithm achieves α-approximation solution.

Proposition B.1. *For the greedy sequence π^G, if the maximal feasible set in π^G is an α-approximation for the offline problem, where $\alpha \in (0, 1]$, then $R_\alpha(T) \leq R^G(T)$.*

Proof. For the greedy sequence $\pi^G := \langle G_0, G_1, \ldots, G_{m^G} \rangle$. Since G_{m^G} is an α-approximation solution, thus $\mathbb{E}[f_1(S^G)] = \overline{f}(S^G) \geq \alpha \overline{f}(S^*) = \alpha \mathbb{E}[f_1(S^*)]$. Therefore, it follows that

$$R^G(T) = T \cdot \mathbb{E}[f_1(S^G)] - \sum_{t=1}^{T} \mathbb{E}[f_t(S^t)] \leq T \cdot \alpha \cdot \mathbb{E}[f_1(S^*)] - \sum_{t=1}^{T} \mathbb{E}[f_t(S^t)] = R_\alpha(T).$$

□

A similar proposition holds for α-quasi greedy sequences. Among all α-quasi greedy sequences, we use $\pi^Q := \langle Q_0, Q_1, \ldots, Q_{m^Q} \rangle$ to denote the one with the minimum reward, where m^Q is its size.

Proposition B.2. *For an α-quasi greedy sequence $\pi^Q := \langle Q_0, Q_1, \ldots, Q_{m^Q} \rangle$, if Q_{m^Q} is an α-approximation for the offline problem, where $\alpha \in (0, 1]$, then $R_\alpha(T) \leq R^Q(T)$.*

Proof. Since Q_{m^Q} is an α-approximation, $\mathbb{E}[f_1(Q_{m^Q})] = \overline{f}(Q_{m^Q}) \geq \alpha \overline{f}(S^*) = \alpha \mathbb{E}[f_1(S^*)]$. Hence, we can get that

$$R^Q(T) = T \cdot \alpha \cdot \mathbb{E}[f_1(Q_{m^Q})] - \sum_{t=1}^{T} \mathbb{E}[f_t(S^t)] \leq T \cdot \alpha \cdot \mathbb{E}[f_1(S^*)] - \sum_{t=1}^{T} \mathbb{E}[f_t(S^t)] = R_\alpha(T).$$

□

C Analysis for OG-UCB in Section 3

In this section, we analyze the greedy regret bound for Algorithm OG-UCB (Algorithm 1 + Subroutine 2). In Appendix C.1, we first derive a general form of the greedy regret bound. Then, in Appendix C.2, we analyze the expected number of playing each arm in the decision frontier, and then obtain the desired greedy regret bound for Algorithm OG-UCB.

C.1 Set Decomposition

We first define some useful notations. For any time t, suppose Algorithm OG-UCB plays a maximal decision sequence $S^t := \langle S_0^t, S_1^t, \dots, S_{m^t}^t \rangle$ where $S^t := S_{m^t}^t$, and $S_i^t = S_{i-1}^t \cup \{s_i^t\}$ for each i. Denote the greedy sequence as $G := \langle G_0, G_1, \dots, G_{m^G} \rangle$, where $G_i = G_{i-1} \cup \{g_i\}$ for each i. For any time t and any $i = 1, 2, \dots$, define random event $O_i^t := \{G_i = S_i^t\}$ (we define $O_0^t = \{\emptyset = \emptyset\}$ always holds.) which means that OG-UCB finds the same prefix S_i^t as the greedy sequence does. For ease of notation, let $O_{0:k}^t := \bigcap_{i=0}^{k} O_i^t$ for each $k \geq 0$.

Then, we can get the following lemma by De Morgan's laws:

Lemma C.1 (Set-decomposition)**.** *Fix any $k \geq 1$, then $\overline{\bigcap_{i=1}^{k} O_i^t} = \overline{O_1^t} \cup (O_1^t \setminus \overline{O_2^t}) \cup \cdots \cup (\bigcap_{i=1}^{k-1} O_i^t \setminus \overline{O_k^t}) = \bigcup_{j=1}^{k} O_{0:j-1}^t \setminus \overline{O_j^t}$, and each term of its right-hand side is mutually exclusive.*

Notice that $O_{1:j-1}^t \setminus \overline{O_j^t}$ means that the first $j-1$ prefixes coincide with the greedy sequence. i.e., $G_i = S_i^t$, for $i = 1, \dots, j-1$; however, at j-th phase, it picks $s_j^t \neq g_j$, that is $s_j^t \in N_-(G_{j-1})$. Now, we can write the greedy regret as follows.

Lemma C.2. *For any time T, the greedy regret of Algorithm OG-UCB satisfies that:*

$$R^G(T) \leq \sum_{k=1}^{m^G} \mathbb{E}\left[\sum_{t=1}^{T} \Delta(S^t) \cdot \mathbb{1}\left\{O_{1:k-1}^t \setminus \overline{O_k^t}\right\}\right].$$

Proof. By definition, $\overline{f}(S) = \mathbb{E}_{I_1}[f_1(S)]$ and $\Delta(S) = \overline{f}(G_{m^G}) - \overline{f}(S)$ for any $S \in \mathcal{F}$. Therefore, we have that

$$\begin{aligned} R^G(T) &= T \cdot \mathbb{E}[f_1(G_{m^G})] - \sum_{t=1}^{T} \mathbb{E}\left[f_t(S^t)\right] \\ &= T \cdot \overline{f}(G_{m^G}) - \sum_{t=1}^{T} \mathbb{E}\left[\overline{f}(S^t)\right] \\ &= \mathbb{E}\left[\sum_{t=1}^{T} \Delta(S^t) \cdot \mathbb{1}\{G_{m^G} \neq S^t\}\right]. \end{aligned}$$

From the definition, we know that if $\bigcap_{i=1}^{m^G} O_i^t$ occurs, then event $\{G_{m^G} = S^t\}$ is true. As its contrapositive, $\{G_{m^G} \neq S^t\}$ implies that $\setminus \bigcap_{i=1}^{m^G} O_i^t$ occurs. Therefore, we can get

$$\begin{aligned} \mathbb{E}\left[\sum_{t=1}^{T} \Delta(S^t) \cdot \mathbb{1}\{G_{m^G} \neq S^t\}\right] &\leq \mathbb{E}\left[\sum_{t=1}^{T} \Delta(S^t) \cdot \mathbb{1}\left\{\setminus \bigcap_{i=1}^{m^G} O_i^t\right\}\right] \\ &= \mathbb{E}\left[\sum_{t=1}^{T} \Delta(S^t) \cdot \mathbb{1}\left\{\bigcup_{k=1}^{m^G} O_{0:k-1}^t \setminus \overline{O_k^t}\right\}\right] \quad (10) \\ &\leq \mathbb{E}\left[\sum_{t=1}^{T} \Delta(S^t) \cdot \sum_{k=1}^{m^G} \mathbb{1}\left\{O_{0:k-1}^t \setminus \overline{O_k^t}\right\}\right] \quad (11) \\ &= \sum_{k=1}^{m^G} \mathbb{E}\left[\sum_{t=1}^{T} \Delta(S^t) \cdot \mathbb{1}\left\{O_{0:k-1}^t \setminus \overline{O_k^t}\right\}\right], \end{aligned}$$

where (10) is due to Lemma C.1, and (11) is by the union bound. □

C.2 Proof of Theorem 3.1

We use the following well-known probability inequality.

Fact C.3 (Chernoff-Hoeffding Bound). *$Z_1, Z_2, \cdots, Z_m$ are i.i.d. random variables supported on $[0,1]$ with mean μ. Define the mean estimator as $\hat{Z} = \frac{1}{m}\sum_{i=1}^{m} Z_m$. For any $a > 0$, it holds that*

$$\mathrm{P}\left[\hat{Z} > \mu + a\right] \le \exp\left(-2a^2 m\right), \quad \mathrm{P}\left[\hat{Z} < \mu - a\right] \le \exp\left(-2a^2 m\right).$$

Recall that the unit gap for any arm $a = e|S$ (defined in (4)) is

$$\Delta(a) = \Delta(e|S) = \begin{cases} \overline{f}(g_S^*|S) - \overline{f}(e|S), & e \neq g_S^* \\ \overline{f}(e|S) - \max_{e' \in \mathcal{N}(S)} \overline{f}(e'|S), & e = g_S^* \end{cases}.$$

Define the threshold function for any arm $a = e|S \in A$ as

$$\ell_t(a) = \frac{6 \ln t}{(\Delta(a))^2}; \tag{12}$$

and the random event

$$E^t(e|S) = \{N_t(e|S) \ge \ell_t(e|S)\} \tag{13}$$

meaning that the arm $e|S$ is *sufficient sampled* at time t.

Lemma C.4 and Lemma C.5 is a technique adapted from [3], which can bound the error probability of the UCB policy for any sufficiently sampled arm.

Lemma C.4. *Consider two arms a_1 and a_2 that generate random variables on the support $[0,1]$, denoted as X_1 and X_2, respectively, and denote $\mu_i = \mathrm{E}[X_i]$ for $i = 1, 2$. Let $\hat{X}_i$ be the average of all i.i.d. samples from X_i (i.e., the empirical mean of X_i), and $N(a_i)$ be the counter of samples. Denote $\hat{X}_{i,N(a_i)}$ be the average of the first $N(a_i)$ samples. W.L.O.G, assume $d = \mu_1 - \mu_2 > 0$. Define function $\mathrm{rad}_t(a) := \sqrt{\frac{3 \ln t}{2N(a)}}$. For $t = 1, 2, \cdots$, if $N(a_1), N(a_2) \ge 1$ and $N(a_2) > \frac{6 \ln t}{d^2}$, the following holds:*

$$\mathrm{P}\left[\hat{X}_{1,N(a_1)} + \mathrm{rad}_t(a_1) \le \hat{X}_{2,N(a_2)} + \mathrm{rad}_t(a_2)\right] \le \mathrm{P}\left[\hat{X}_{1,N(a_1)} \le \mu_1 - \mathrm{rad}_t(a_1)\right] + \mathrm{P}\left[\hat{X}_{2,N(a_2)} \ge \mu_2 + \mathrm{rad}_t(a_2)\right].$$

Proof. Since $N(a_1), N(a_2) \ge 1$ and $N(a_2) > \frac{6 \ln t}{d^2}$, we can get $\mathrm{rad}_t(a_2) < \frac{d}{2}$ and $\mathrm{rad}_t(a_1) > 0$. Thus, $\mu_1 > \mu_2 + 2\,\mathrm{rad}_t(a_2)$. It is easy to verify that event $\hat{X}_{1,N(a_1)} + \mathrm{rad}_t(a_1) \le \hat{X}_{2,N(a_2)} + \mathrm{rad}_t(a_2)$ implies at least one of the following events must hold:

$$\hat{X}_{1,N(a_1)} \le \mu_1 - \mathrm{rad}_t(a_1) \tag{14}$$

$$\hat{X}_{2,N(a_2)} \ge \mu_2 + \mathrm{rad}_t(a_2). \tag{15}$$

Otherwise, assume that both of (14) and (15) are false, then $\hat{X}_{1,N(a_1)} + \mathrm{rad}_t(a_1) > \mu_1 > \mu_2 + 2\,\mathrm{rad}_t(a_2) > \hat{X}_{2,N(a_2)} + \mathrm{rad}_t(a_2)$, which causes a contradiction. Therefore, we can get $\mathrm{P}\left[\hat{X}_{1,N(a_1)} + \mathrm{rad}_t(a_1) \le \hat{X}_{2,N(a_2)} + \mathrm{rad}_t(a_2)\right] = \mathrm{P}[(14) \text{ or } (15) \text{ is true}]$, and the lemma can be concluded by the union bound. □

For any arm $a \in F$, we denote the upper bound as

$$U_t(a) = \hat{X}(a) + \mathrm{rad}_t(N(a)). \tag{16}$$

As (3) defined in the main text, the counter $N_t(a) = \sum_{i=1}^{t-1} \mathbb{I}_i\{a\}$ is used to denoted the particular $N(a)$ at the beginning of the time step t, where $\mathbb{I}_i\{a\} \in \{0,1\}$ indicates whether arm a is updated at time i. (For OG-UCB, it is always updated once chosen.)

We need the following lemma, which bounds the expected value of the counter.

Lemma C.5. *For any time horizon T and any $k \ge 1$, for any $e \in N_-(G_{k-1})$, its counter satisfies*

$$\mathrm{E}[N_{T+1}(e|G_{k-1})] \le \ell_T(e|G_{k-1}) + \frac{2}{3} + 1. \tag{17}$$

Proof. Consider a fixed prefix G_{k-1} and an arm $e \in N_-(G_{k-1})$. Suppose each arm in $N(G_{k-1})$ is initialized (played once). Line 11 of Subroutine 2 (the UCB policy) indicates that, selecting element $e \in N_-(G_{k-1})$ for phase k implies that the random event $\{U_t(g_k|G_{k-1}) \geq U_t(e|G_{k-1})\}$ occurs. Then, we can bound the counter of playing $e|G_{k-1}$ by considering two disjoint random events, $E^t(e|G_{k-1})$ and $\overline{E^t(e|G_{k-1})}$ (sufficiently sampled and insufficiently sampled, respectively). That is

$$\begin{aligned}
&\mathbb{E}[N_{T+1}(e|G_{k-1})] \\
&= \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{I}\left\{O^t_{0:k-1} \wedge U_t(g_k|G_{k-1}) \geq U_t(e|G_{k-1})\right\}\right] \\
&= \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{I}\left\{O^t_{0:k-1} \wedge \overline{E^t(e|G_{k-1})} \wedge U_t(g_k|G_{k-1}) \geq U_t(e|G_{k-1})\right\}\right] \\
&\quad + \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{I}\left\{O^t_{0:k-1} \wedge E^t(e|G_{k-1}) \wedge U_t(g_k|G_{k-1}) \geq U_t(e|G_{k-1})\right\}\right] \\
&\leq d_T(e|G_{k-1})e + \sum_{t=1}^{T} \mathbb{P}\left\{O^t_{0:k-1} \wedge E^t(e|G_{k-1}) \wedge U_t(g_k|G_{k-1}) \geq U_t(e|G_{k-1})\right\}, \qquad (18)
\end{aligned}$$

where (18) follows from the definition of $E^t(e|G_{k-1})$. Now we use Lemma C.4, in which arms are $a_1 = g_k|G_{k-1}$ and $a_2 = e|G_{k-1}$; empirical means $\hat{X}_1$, $\hat{X}_2$ are associated with $\hat{X}(g_k|G_{k-1})$ and $\hat{X}(e|G_{k-1})$, respectively; μ_1, μ_2 are used to denote their means; $N(a_1)$ and $N(a_2)$ are their counters. Then, for any t, we have that

$$\begin{aligned}
&\mathbb{P}\left\{\bigwedge_{i=1}^{k-1} O^t_i \wedge E^t(e|G_{k-1}) \wedge U_t(g_k|G_{k-1}) \geq U_t(e|G_{k-1})\right\} \\
&\leq \mathbb{P}\left\{E^t(e|G_{k-1}) \wedge U_t(g_k|G_{k-1}) \geq U_t(e|G_{k-1})\right\} \\
&\leq \sum_{N(a_1)=1}^{t} \mathbb{P}\left\{\hat{X}_{1;N(a_1)} \leq \mu_1 + \mathrm{rad}_t(a_1)\right\} + \sum_{N(a_2)=\lceil \mu_t(a_2) \rceil}^{t} \mathbb{P}\left\{\hat{X}_{2;N(a_2)} \geq \mu_2 + \mathrm{rad}_t(a_2)\right\} \\
&\leq \sum_{N(a_1)=1}^{t} t^{-3} + \sum_{N(a_2)=1}^{t} t^{-3} = 2t^{-2}; \qquad (19)
\end{aligned}$$

where (19) holds because of Fact C.3. Summing over $t = 1, \ldots, T$ and using the convergence of Riemann zeta function (i.e., $\sum_{t=1}^{\infty} \frac{1}{t^2} = \frac{\pi^2}{6}$), the proof is completed. □

Theorem C.6 (Restatement of Theorem 3.1)**.** *For any time T, Algorithm OG-UCB (Algorithm 1 + Subroutine 2) can achieve the greedy regret*

$$R^{\mathcal{G}}(T) \leq \sum_{a \in \Gamma \setminus \Gamma(\mathcal{G})} \frac{6\,\Delta^*(a)\,\ln T}{(\Delta(a))^2} + \left(\frac{\pi^2}{3} + 1\right)\Delta^*(a);$$

where $\mathcal{G}$ is the greedy decision sequence.

Proof. Now fix any k, and consider each term of the right-hand side in Lemma C.2, that is $\mathbb{E}\left[\sum_{t=1}^{T} \mathbb{I}\{S^t) \mid O^t_{0:k-1} \setminus \overline{O^t_k}\}\right]$.

Suppose $O^t_{0:k-1} \setminus \overline{O^t_k}$ happens, that is the algorithm selects G_{k-1} at the first $k-1$ phases, and then picks $e \in N_-(G_{k-1})$. By definition of the sunk-cost gap in (5), we can see that the regret at time t is no more than $\Delta^*(e|G_{k-1})$. We we can get:

$$\mathbb{E}\left[\sum_{t=1}^{T} \mathbb{I}\{S^t) \mid O^t_{0:k-1} \setminus \overline{O^t_k}\}\right]$$

$$\mathbb{E}\left[\sum_{t=1}^{\infty} (S^t) \mid O^t_{0:k-1} \left(\sum_{e\in\mathcal{N}(G_{k-1})} \mathbb{I}_t\{e|G_{k-1}\}\right)\right] \tag{20}$$

$$\sum_{e\in\mathcal{N}(G_{k-1})} \mathbb{E}\left[\sum_{t=1}^{\infty} (S^t) \mid O^t_{0:k-1} \mathbb{I}_t\{e|G_{k-1}\}\right]$$

$$\sum_{e\in\mathcal{N}(G_{k-1})} {}^*(e|G_{k-1}) \mathbb{E}\left[\sum_{t=1}^{\infty} \mid O^t_{0:k-1} \mathbb{I}_t\{e|G_{k-1}\}\right]$$

$$\sum_{e\in\mathcal{N}(G_{k-1})} {}^*(e|G_{k-1}) \mathbb{E}[N_{T+1}(e|G_{k-1})] \tag{21}$$

where (20) is by the union bound, and (21) is by the definition of $N_{T+1}(e|G_{k-1})$ and $\mathbb{I}\{O^t_{0:k-1}\}$ 1. Thus, the greedy regret satisfies that

$$R^G(T) \le \sum_{k=1}^{m^G} \mathbb{E}\left[\sum_{t=1}^{\infty} (S^t) \mid O^t_{1:k-1} \setminus \overline{O^t_k}\right] \tag{22}$$

$$\le \sum_{k=1}^{m^G} \sum_{e\in\mathcal{N}(G_{k-1})} {}^*(e|G_k) \mathbb{E}[N_T(e|G_{k-1})] \tag{23}$$

$$= \sum_{a\in\Gamma(G)} {}^*(a) \mathbb{E}[N_T(a)] \tag{24}$$

$$\le \sum_{a\in\Gamma(G)} \frac{6\ {}^*(a) \ln T}{(a)^2} + \frac{2}{3} + 1 \quad {}^*(a) \quad ; \tag{25}$$

where (22) is derived from Lemma C.2; (23) is from (21); (24) is due to Definition 3.2; and (25) follows Lemma C.5. Therefore, the theorem is concluded. □

D Analysis for OG-LUCB $_{;}$ and OG-LUCB-R in Section 4

In this section, we provide the analysis of Algorithm OG-LUCB $_{;}$ (Algorithm 1 + Subroutine 3) and Algorithm OG-LUCB-R (Algorithm 4). In the following, we first derive the probability inequality of confidence events. Second, we analyze the exploration time in Appendix D.1. Assuming that the total time horizon T is known, we provide an $O(\log T)$ -quasi greedy regret in Appendix D.2. Finally, we show that we can remove the assumption of knowing T and still obtain an $O(\log T)$ regret bound (by Algorithm OG-LUCB-R).

Given any arm $a = e|S \in A$, define event $C^t(e|S) = \left\{\hat{X}_t(e|S) \quad X(e|S) < \mathrm{rad}_t(e|S)\right\}$ (confidence bound holds). For any prefix S, define $C^t(S) = \bigcap_{e\in\mathcal{N}(S)} C^t(e|S)$ for all the arms $e|S$.

Recall that for any S, $e \in N(S)$, by definition

$$(e|S_k) = \begin{cases} \overline{f}(g^*_S|S) \quad \overline{f}(e|S); & e \neq g^*_S \\ \overline{f}(e|S) \quad \max_{a\in\mathcal{N}(S)\setminus\{g_S\}} \overline{f}(a|S); & e = g^*_S \end{cases},$$

and the hardness coefficient with the prefix S as

$$H_S := \sum_{e\in\mathcal{N}(S)} \frac{1}{\max\{ \quad (e|S)^2; \ ^2\}}.$$

Lemma D.1. *Fix any* $\in (0;1)$. *Suppose* $\mathrm{rad}_t(s) = \sqrt{\frac{\ln(4Wt^3/\ )}{2N_t(s)}}$. *For any prefix* S, $\bigcap_{t=1}^{\infty} C_t(S)$ *holds with probability at least* 1 .

Proof. From Fact C.3, we know that for any $e \in N(S)$,

$$\mathbb{P}\left[C^t(e|S)\right] = \mathbb{P}\left[\hat{X}_t(e|S) \quad X(e|S) < \mathrm{rad}_t(e|S); N_t(e|S) = 1; \quad ; t\right]$$

$$1 \quad 2 \quad t \quad \frac{}{4Wt^3} = 1 \quad \frac{}{2Wt^2}:$$

Then, by summing all time steps t and all elements $e \in N(S)$ ($jN(S)j \quad W$), we can conclude that

$$P \left[\bigcap_{t=1}^{\infty} C^t(S) \right] \quad 1 \quad \sum_{t=1}^{\infty} \sum_{e \in \mathcal{N}(S)} \frac{}{2Wt^2} \quad 1 \quad \sum_{t=1}^{\infty} \frac{}{2t^2} \quad 1 \quad \frac{2}{12} \quad 1 \quad ; \tag{26}$$

where (26) is by Riemann zeta function. □

D.1 Exploration time for OG with a first-explore-then-exploit policy (OG-LUCB $_{;}$)

Since we use a specialized version of CLUCB-PAC in [9] as MaxOracle to explore the top-one element in the support of $[0; 1]$ for each phase, and henceforth our analysis starts from the result of sample complexity of CLUCB-PAC by setting $R = \frac{1}{2}$ and width$(M) = 2$.

In Algorithm OG-LUCB $_{;}$, for any arm $a = ejS \in A$, denote $N_t^{\mathsf{E}}(a)$ as the counter of playing arm a during the exploration stage at the beginning of time step t. The algorithm turns from the exploration to the exploitation some time and the counter will not change, therefore we may use $N_{\infty}^{\mathsf{E}}(a)$ to obtain its final value. For any prefix S, denote arms $A = \{ejS : 8e \in N(S)\}$. Denote $t_0(S) := \sum_{e|S \in A} N_{t_0}^{\mathsf{E}}(ejS)$ as the total exploration time of all elements in $N(S)$, such that for any time $t \quad t_0(S)$, $I^{\text{exploit}}(A)$ is initialized.

Notice that we use CLUCB-PAC for each phase, therefore the following lemma can be adapted from the intermediate step in proving Theorem 5 of [9].

Lemma D.2. *For any phase $k \quad 0$, fix prefix S_k. Suppose $\quad 0, \quad \in (0; 1)$, and $\bigcap_{t=1}^{\infty} C^t(S_k)$ holds. If it goes to the exploitation stage (Line 12 of Subroutine 3) with $I^{\text{exploit}}(A) = s^*$ at time $t_0 = t_0(S_k)$, then:*

1. $f(s^*jS_k) \quad f(g^*_{S_k}jS_k) \quad ;$
2. *In addition, for any $s \in N(S_k)$,*

$$N_{t_0}^{\mathsf{E}}(sjS_k) \quad \min \quad \frac{18}{(sjS_k)^2}; \frac{16}{2} \quad \ln \quad 4Wt_0^3 = \quad + 1: \tag{27}$$

Notice that event $\bigcap_{t=1}^{\infty} C^t(S_k)$ holds with probability at least $1 \quad$, which is guaranteed by Lemma D.1. From the above lemma, we may further derive the following bound for $t_0(S_k)$.

Lemma D.3. *With the same setting as Lemma D.2. For $t_0 = t_0(S_k)$, denote $t_c = t_c(S_k) := 499H_{S_k} \ln(4WH_{S_k} =) + 2W$ then we can get:*

1. $t_0 \quad t_c:$
2. $\ln \quad 4Wt_c^3 = \quad 7 \ln \quad 1996WH_{S_k} = \quad :$

Proof. **Property (1):** First of all, (1) holds trivially if $W \quad \frac{t_0}{2}$, thus we only need to show the case $W < \frac{t_0}{2}$.

Since $t_0 = \sum_{e \in \mathcal{N}(S_k)} N_{t_0}^{\mathsf{E}}(ejS_k)$, it can be implied from Lemma D.2 that, with probability at least $1 \quad$,

$$t_0 \quad 18H_{S_k} \ln(4Wt_0^3 =) + W: \tag{28}$$

We assume $t_0 = CH_{S_k} \ln(4WH_{S_k} =) + W$, for some constant $C > 0$. When $W < \frac{t_0}{2}$, $t_0 < 2CH_{S_k} \ln(4WH_{S_k} =)$. Then, rewrite (28) as

$$t_0 \quad W + 18H_{S_k} \ln(4W =) + 54H_{S_k} \ln(t_0)$$

$$
\begin{aligned}
&< W + 18H_{S_k}\ln(4W/\ ) + 54H_{S_k}\ln(2CH_{S_k}\ln(4WH_{S_k}/\ ))\\
&\quad W + 18H_{S_k}\ln(4W/\ ) + 54H_{S_k}\ \ln(2C) + \ln(H_{S_k})\ + 54H_{S_k}\ln(4WH_{S_k}/\ )\\
&\quad W + 72H_{S_k}\ln(4WH_{S_k}/\ ) + 54H_{S_k}\ \ln(2C)\ln(4WH_{S_k}/\ ) + \ln(4WH_{S_k}/\ )\\
&\quad W + (126 + 54\ln 2C)\,H_{S_k}\ln(4WH_{S_k}/\ ): \qquad (29)
\end{aligned}
$$

Solve $126 + 54\ln 2C < C$, and we can get the minimum integer solution $C = 499$. When $C \quad 499$, from (29), $t_0 < W + CH_{S_k}\ln(4WH_{S_k}/\ ) = t_0$, which cause a contradiction. Thus, we can conclude that $t_0 \quad 499H_{S_k}\ln(4WH_{S_k}/\ ) + 2W$.

Property (2): We can simplify $\ln\ 4Wt_c^3/\ $ in (27) as follows.

From Property 1, since $\ln(a+b) \quad \ln(a) + \ln(b)$ for all $a; b \quad e$, we have

$$
\begin{aligned}
\ln(t_c) \quad &\ln\ 499H_{S_k}\ln(4WH_{S_k}/\ )\ + \ln(4W) \qquad (30)\\
&\ln(499H_{S_k}) + \ln\ln(4WH_{S_k}/\ ) + \ln(4W)\\
&\ln(499H_{S_k}) + 2\ln(4W) + \ln\ H_{S_k}\ + \ln(1/\ );
\end{aligned}
$$

then

$$
\begin{aligned}
\ln\ 4Wt_c^3/\ \quad &\ln(4W) + \ln(1/\ ) + 3\ln(t_c)\\
&7\ln(4W) + 4\ln(1/\ ) + 6\ln\ 499H_{S_k}\\
&7\ln\ 1996WH_{S_k}/\ \ ;
\end{aligned}
$$

which ends the proof. □

Define random event iteratively, for any $k \quad 1$, given prefix S_{k-1},

$$
Q_k(S_{k-1}) = \ \overline{f}(s_k j S_{k-1})\ \ \overline{f}(g^*_{S_k\ _1} j S_{k-1})\ \ ; \qquad (31)
$$

where $s_k = I^{\text{exploit}}(A)$ and $A = \{e j S_{k-1} : 8e \ 2\ N(S_{k-1})\}$.

Theorem D.4 (Restatement of Theorem 4.1)**.** *Given any* $\ 0$ *and* $\ 2\ (0;1)$*, suppose after the total exploration time* $T^{\mathrm{E}} = T^{\mathrm{E}}(\ )$*, Algorithm OG-LUCB* $_{\ ;}$ *sticks to a stable sequence* $\ = \langle S_0; S_1;\ \ ; S_{m'}\rangle$*. With probability at least* $1\ \ m$ *(*m *is the largest length of a feasible set), the following three claims hold:*

1. *is an* $\ $*-quasi greedy sequence;*
2. *Let* $t_c(S) := 499H_S\ln(4WH_S/\ ) + 2W$*. For any arm* $e j S$ *in the decision frontier* $\ (\ )$*,*

$$
N^{\mathrm{E}}_{\infty}(e j S)\ \ \min\ \ \frac{18}{(e j S)^2}; \frac{16}{\ ^2}\ \ \ln\ 4Wt_c^3(S)/\ \ + 1: \qquad (32)
$$

3. *The total exploration time* T^{E} *satisfies that*

$$
T^{\mathrm{E}} \quad \sum_{e|S\in\Gamma(\ )} N^{\mathrm{E}}_{\infty}(e j S)\ \ 127\sum_{k=0}^{m'-1} H_S\ln(1996WH_S/\ ): \qquad (33)
$$

Proof. **Property (1):** Similar to the set decomposition in Lemma C.1, by De Morgan's laws, we know that

$$
\overline{\bigcap_{i=1}^{m'} Q^t_i(S_{i-1})} = \overline{Q^t_1(S_0)}\ \ [\ \ Q^t_1(S_0) \setminus \overline{Q^t_2(S_1)}\ \ [\ \ \ [\ \Big(\bigcap_{i=1}^{m'-1} Q^t_i(S_{i-1})\Big) \setminus \overline{Q^t_k(S_{m'-1})}\Big) :
$$

By Lemma D.2, we know that for $k = 1; 2;\ \ ; m'$, the k-th term above satisfies

$$
\mathrm{P}\Big[\Big(\bigcap_{i=1}^{k-1} Q^t_i(S_{i-1})\Big) \setminus \overline{Q^t_k(S_{k-1})}\Big]\ \ \mathrm{P}\Big[\overline{Q^t_k(S_{k-1})}\Big]\ \ : \qquad (34)
$$

Therefore, by union bound and $m' \quad m$, we have that

$$\mathrm{P}\left[\bigcap_{i=1}^{m'} Q_i^t(S_{k-1})\right] = 1 \quad \mathrm{P}\left[\bigcap_{i=1}^{m'} Q_i^t(S_{k-1})\right] \quad 1 \quad \sum_{k=1}^{m'} \quad 1 \quad m \ ;$$

and we prove that is an -quasi greedy sequence.

Property (2): Assume that $\bigcap_{i=1}^{m'} Q_i^t(S_{i-1})$ holds. Since we denote $t_c(S) = 499H_S \ln(4WH_S/\ ) + 2W$, applying Property 1 of Lemma D.3, we can get $t_0(S_{k-1}) \quad t_c(S_{k-1})$. Then, use Property 2 of Lemma D.2, and it follows immediately that

$$N_\infty^{\mathrm{E}}(ejS_{k-1}) \quad \min \quad \frac{18}{(ejS_{k-1})^2}; \frac{16}{2} \quad \ln \ 4Wt_c^3(S_{k-1})/ \quad + 1; \tag{35}$$

where ejS_{k-1} is any arm occurred in the decision frontier ( ).

Property (3): It is easy to see that the total exploration time $T^{\mathrm{E}} \quad \sum_{e|S\in\Gamma(\ )} N_\infty^{\mathrm{E}}(ejS)$. Sum up each phase $k = 0; 1; \quad ; m' \quad 1$ and each element $e \ 2 \ N(S_k)$, and use $\ln \ 4Wt_c^3(S_{k-1}) \ 7 \ln \ 1996WH_{S_k}/\ $ by Property 2 of Lemma D.3. Therefore we can get:

$$\begin{aligned} T^{\mathrm{E}} \quad & \sum_{e|S\in\Gamma(\ )} N_\infty^{\mathrm{E}}(ejS) \quad \sum_{k=0}^{m'-1} \sum_{e\in\mathcal{N}(S_k)} N_\infty^{\mathrm{E}}(ejS_k) \\ & \sum_{k=0}^{m'-1} 126H_{S_k} \ln \ 1996WH_{S_k}/\ \quad + jN(S_k)j \\ & 127 \sum_{k=0}^{m'-1} H_{S_k} \ln \ 1996WH_{S_k}/\ \quad : \end{aligned} \tag{36}$$

□

D.2 Time Horizon T is Known

We first assume that we know the total time horizon T. In this case, we may let $= \frac{1}{T}$ in OG-LUCB $_{\ ;\ }$, and we can obtain the following theorem.

Theorem D.5 (Restatement of Theorem 4.2)**.** *Given any 0. When total time T is known, in Algorithm OG-LUCB $_{\ ;\ }$, we associate $= \frac{1}{T}$. Suppose $= \{S_0; S_1; \quad ; S_{m'}\}$ is the sequence selected by Algorithm OG-LUCB $_{\ ;\ }$ at time T, and define function*

$$R^{\mathrm{Q};\ }(T) := \sum_{e|S\in\Gamma(\ )} {}^*(ejS) \min \quad \frac{127}{(ejS)^2}; \frac{113}{2} \quad \ln(1996WH_S T) + \quad {}_{\max} m; \tag{37}$$

where m is the largest length of a feasible set and H_S is defined in (7). Then, the -quasi regret satisfies

$$R^{\mathrm{Q}}(T) \quad R^{\mathrm{Q};\ }(T) = O \quad \frac{Wm \ {}_{\max}}{\max\{ \ ^2; \ ^2\}} \log T \quad ; \tag{38}$$

where is the minimum unit gap.

Proof. First, we claim that

$$R^{\mathrm{Q}}(T) \quad \sum_{e|S\in\Gamma(\ )} {}^*(ejS) \quad \min \quad \frac{18}{(ejS)^2}; \frac{16}{2} \quad \ln \ 4Wt_{c;T}^3(S) \quad T \quad + 1 \quad + \quad {}_{\max} m; \tag{39}$$

where $t_{c;T}(S) = 499H_S \ln(4WH_S \quad T) + 2W$ for any S.

From Property 1 of Theorem D.4 , we know that with probability 1 $\frac{m}{T}$, the sequence is a stable -greedy sequence, i.e., all confidence events $\bigcap_{i=1}^{m'} Q_i^t(S_{k-1})$ hold.

We may sum up exploration numbers of the arms ($N_\infty^{\mathsf{E}}(\ )$'s) in Property 2 of Theorem D.4, so that the first term of (39) can be easily derived. For the second term of (39), it is because OG-LUCB $_{;}$ pays at most regret $_{\max}T$ with probability $\frac{m}{T}$, when any confidence events fails, which contributes at most $_{\max}m$ to the regret.

Then, apply Property 2 of Lemma D.3 and $= \frac{1}{T}$, i.e., ln $4WT_{c;T}^3(S)$ T $7\ln\ 1996WH_{S_k}=$, and henceforth it is obvious to see that

$$R^{\mathsf{Q}}(T) \quad R^{\mathsf{Q};\ }(T) = \sum_{e|S\in\Gamma(\ )} {}^*(ejS)\min\ \frac{127}{(ejS)^2};\frac{113}{2}\ \ln(1996WH_ST) + \ _{\max}m:$$

Furthermore, if we replace ${}^*(ejS)$ with the maximum gap $_{\max}$, and (ejS) with the minimum unit gap , it is easily derived that $R^{\mathsf{Q}}(T)$ $R^{\mathsf{Q};\ }(T) = O(\frac{Wm\Delta_{\max}}{\max\{\Delta^2;\ ^2\}}\log T)$, where W is the largest width and m is the largest length. □

Notice that, from the above theorem, the -quasi regret is bounded within $O(\log T)$ for any time horizon T, even though T may be less than the total exploration time $T^{\mathsf{E}} = T^{\mathsf{E}}\ \frac{1}{T}$. It is because $N_\infty^{\mathsf{E}}(\ )$ is the upper bound approaching the infinite time, and the regret is a non-decreasing function with the time horizon T. Therefore, the above regret is satisfied for $T < T^{\mathsf{E}}\ \frac{1}{T}$ as well.

D.3 Time Horizon T is not Known

When the time horizon T is not known, we use Algorithm OG-LUCB-R (Algorithm 4) which restarts the internal OG-LUCB $_{;}$ for different epochs. The following lemma is useful in proving Theorem D.7.

Lemma D.6. *For any $i = 1;2;\ \ ;k$, for any $c_i;b_i > 0$ and $_i = e^{2^i}$,*

$$\sum_{i=1}^{k}(c_i\ \ln(\ _i)+b_i)\quad 4\quad \max_{i=1;\cdots;k}c_i\quad \ln(\ _{k-1})+k\quad \max_{i=1;\cdots;k}b_i\ :$$

Proof. Since $_i = e^{2^i}$, then $\ln(\ _i) = 2^i$, an we can get

$$\sum_{i=1}^{k}\ln(\ _i) = 2^1+2^2+\ \ +2^k < 4\quad 2^{k-1} = 4\ln(\ _{k-1}):$$

Therefore, it follows immediately that

$$\sum_{i=1}^{k}(c_i\ \ln(\ _i)+b_i)\quad \max_{i=1;\cdots;k}c_i\sum_{i=1}^{k}\ln(\ _i)+k\quad \max_{i=1;\cdots;k}b_i$$

$$4\quad \max_{i=1;\cdots;k}c_i\quad \ln(\ _{k-1})+k\quad \max_{i=1;\cdots;k}b_i\ :$$

□

Theorem D.7 (Restatement of Theorem 4.3)**.** *Given any 0. Use $\ $ and $\ $ defined in (8), and function $R^{\mathsf{Q};\ }(T)$ defined Theorem 4.2. In Algorithm OG-LUCB-R, suppose $^{(\ )} = hS_0^{(\ )};S_1^{(\ )};\ \ ;S_{m^{(\ell)}}^{(\ )}i$ is the sequence selected by the end of $\ $-th epoch of OG-LUCB $_{;}$, where $m^{(\ )}$ is its length. For any time T, denote the final epoch as $K = K(T)$ such that $_{K-1} < T\ \ _K$, and the -quasi greedy regret satisfies that*

$$R^{\mathsf{Q}}(T)\quad \sum_{\ =1}^{K}R^{\mathsf{Q};\ ^{(\ell)}}(\ \ ): \tag{40}$$

Furthermore, we can get

$$R^{\mathrm{Q}}(T) \quad 4 \quad \max_{\ell=1,\cdots,K} c_\ell \quad \ln(T) + \quad \max_{\ell=1,\cdots,K} b_\ell \quad \log_2(2\ln T) \tag{41}$$

$$= O \quad \frac{Wm \quad \max}{\max f \quad ^2; \quad ^2 g} \log T \quad ; \tag{42}$$

where

$$c_\ell := \sum_{e|S\in\Gamma(\quad ^{(\ell)})} \quad ^*(e|S) \min \left(\frac{127}{(e|S)^2}; \frac{113}{2} \right) ;$$

$$b_\ell := \sum_{e|S\in\Gamma(\quad ^{(\ell)})} \quad ^*(e|S) \min \left(\frac{127}{(e|S)^2}; \frac{113}{2} \right) \ln(1996 W H_S) + \quad _{\max} m;$$

and *is the minimum unit gap.*

Proof. Algorithm OG-LUCB-R restarts the internal OG-LUCB $_{\ }$ for each epoch. Since -quasi regret for epoch ℓ during time $(\quad _{\ell-1}; \quad _\ell]$ is no more than $R^{\mathrm{Q};\ ^{(\ell)}}(\quad _\ell)$, then the -quasi greedy regret in (40) follows naturally by accumulating each interval.

For any ℓ, $\quad _{\ell-1} \quad _{\ell-1}; \quad _\ell = \sum_{i=1}^{\quad _\ell} e^{2^i} < 2e^{2^\ell} = 2 \quad _\ell$: Since $\quad _{K-1} < T \quad _K$, we have $\quad _{K-1} < T < 2 \quad _K$ and $K < \log_2(2\ln T)$.

Furthermore, we can get

$$\sum_{\ell=1}^{K} R^{\mathrm{Q};\ ^{(\ell)}}(\quad _\ell) \quad \sum_{\ell=1}^{K} (c_\ell \ln(\quad _i) + b_\ell) \tag{43}$$

$$4 \quad \max_{\ell=1,\cdots,K} c_\ell \quad \ln(\quad _{K-1}) + \quad \max_{\ell=1,\cdots,K} b_\ell \quad K \tag{44}$$

$$4 \quad \max_{\ell=1,\cdots,K} c_\ell \quad \ln(T) + \quad \max_{\ell=1,\cdots,K} b_\ell \quad \log_2(2\ln T); \tag{45}$$

where (43) is by definition of c_i and b_i, and the form of $R^{\mathrm{Q};\ ^{(\ell)}}$ defined (37); (44) is from Lemma D.6; and (45) is because $\quad _{K-1} < T$ and $K < \log_2(2\ln T)$. Use the similar technique in the proof of Theorem D.5, and we know that (45) is also in $O \quad \frac{Wm\Delta_{\max}}{\max\{\Delta^2; \quad ^2\}} \log T$. Therefore, the theorem is concluded. □

E Proof of Lower Bound in Section 5

In this section, we construct an instance of the prize-collecting problem, and provide its theoretical analysis. We first recall the *prize-collecting problem* defined in the main text as follows.

Problem. Consider m bandits, each of which has W elements. For each bandit $i = 1; 2; \cdots; m$, denote set $E_i = \{e_{i;1}; e_{i;2}; \cdots; e_{i;W}\}$. In this problem, the player needs to select one element from each bandit in order. The accessible set system is defined as $(E; F)$, where $E = \bigcup_{i=1}^{m} E_i$, $F = [\bigcup_{i=1}^{m} F_i \cup \{\emptyset\}$, and $F_i = \{S \quad E : |S| = i; \forall k : 1 \quad k \quad i; |S \setminus E_k| = 1\}$. The reward function $f : F \quad ! \ [0; m]$ is non-decreasing in the first parameter, where the form of f is unknown to the player. Let the minimum unit gap be

$$:= \min \ \{\overline{f}(g_S^* | S) \quad \overline{f}(e|S) : \forall S \in F; \forall e \in N_-(S) \quad > 0; \tag{46}$$

where the value of is also unknown to the player. The objective of the player is to minimize the greedy regret.

Instance P. We construct a problem instance P as follows.

We arbitrarily pick the greedy decision sequence $\pi^{\mathsf{G}} = \langle G_0, G_1, \ldots, G_m \rangle \in \mathcal{F}^{m+1}$ where $G_0 = \emptyset$ and $G_i = G_{i-1} \cup \{e_i^{\mathsf{G}}\}$ for each i.

Assume that $0 < \Delta_1 < \Delta_2 < \Delta_3 < 1$, and $\Delta := \Delta_2 - \Delta_1 > 0$. Consider that the environment's randomness comes from Bernoulli random variables $\omega_t = \langle \omega_{t,1}^1, \omega_{t,2}^1, \ldots, \omega_{t,1}^m, \omega_{t,2}^m \rangle \in \{0,1\}^{2m} = \Omega$, with $\mathbb{E}[\omega_{t,1}^i] = \Delta_1$ (low prize) for any $i = 1, 2, \ldots, m$; $\mathbb{E}[\omega_{t,2}^i] = \Delta_2$ (medium prize) for $i = 1, 2, \ldots, m-1$, and $\mathbb{E}[\omega_{t,2}^m] = \Delta_3$ (high prize). For convenience, given a feasible set S, we define indicator $\mathsf{I}_i^{\mathsf{G}} := \mathbb{I}\{e_i^{\mathsf{G}} \in S\}$ for each i. The exact form of reward function is $f(S, \omega_t) := \sum_{i=1}^m f^{(i)}(S, \omega_t)$, where for each $i = 1, 2, \ldots, m$,

$$f^{(i)}(S, \omega_t) = \begin{cases} \omega_{t,1}^i \overline{\mathsf{I}_i^{\mathsf{G}}} + \omega_{t,2}^i \mathsf{I}_i^{\mathsf{G}}, & G_{i-1} \subseteq S \\ \omega_{t,1}^i, & \text{otherwise} \end{cases}. \tag{47}$$

It is only accessed as a value oracle $f_t(S) := f(S, \omega_t)$ with a given feasible set S, and *the player does not know the form of the reward function*. For example, at time t, provided that we have already selected the greedy prefix G_{i-1} for the first $i-1$ phases (that is $\bigcup_{j=1}^{i-1} \{e_j^{\mathsf{G}}\} \subseteq S$), if we choose the greedy element e_i^{G} at phase i (that is $\mathsf{I}_i^{\mathsf{G}} = 1$), then the marginal reward $f^{(i)}(S, \omega_t) = \omega_{t,2}^i$, and the feedback for arm $e_i^{\mathsf{G}} | G_{i-1}$ is $f_t(e_i^{\mathsf{G}} | G_{i-1}) = f^{(i)}(S, \omega_t) = \omega_{t,2}^i$; otherwise, we will get $\omega_{t,1}^i$ for choosing the sub-optimal element. Moreover, only if all elements along the greedy sequence are chosen, that is $S = G$, can we get the marginal reward $f^{(m)}(S, \omega_t) = \omega_{t,2}^m$ for the last phase, where $\mathbb{E}[\omega_{t,2}^m] = \Delta_3$.

Collecting prizes means that the player should find the greedy sequence to gain medium prizes ($\mathbb{E}[\omega_{t,2}^i] = \Delta_2 > \Delta_1 = \mathbb{E}[\omega_{t,1}^i]$) for the first $m-1$ phases and achieve the high prize ($\mathbb{E}[\omega_{t,2}^m] = \Delta_3$) for the last phase. Since the maximal decision sequence is m phases, it is easy to infer that, the minimum reward and maximum reward in expectation are $m\Delta_1$ and $(m-1)\Delta_2 + \Delta_3$ respectively, and henceforth the maximum gap is $\Delta_{\max} = m - (\Delta_3 - \Delta_2)$. When the player mistakenly chooses a wrong element for some phase, denote the minimum gap as $\Delta_{\min} = \Delta + (\Delta_3 - \Delta_2)$, and $\Delta_{\min}$ is the minimum penalty incurred.

Denote greedy arms as $A^{\mathsf{G}} = \{e_i^{\mathsf{G}} | G_{i-1} : \forall i = 1, 2, \ldots, W\}$. We say an algorithm is *consistent*, if the sum of playing all arms $a \in A \setminus A^{\mathsf{G}}$ is in $o(T^\alpha)$, for any $\alpha > 0$, i.e., $\mathbb{E}[\sum_{a \in \mathcal{A} \setminus \mathcal{A}^{\mathsf{G}}} N_T(a)] = o(T^\alpha)$.

E.1 Lower Bound $\frac{mW\Delta_{\max}}{\Delta^2} \log T$

In this subsection, we set $\Delta_3 - \Delta_2$ to be a constant, and derive the $\frac{mW\Delta_{\min}}{\Delta^2} \log T$ greedy regret lower bound.

Fix any $\ell = 1, 2, \ldots, m-1$. As is illustrated in Figure 1, there exists some feasible sequence σ that coincides with the greedy sequence π^{G} for the first ℓ arms, and deviates from it for the next $d := m - \ell$ arms, that is $\sigma = \langle G_0, G_1, \ldots, G_\ell, S_{\ell+1}, \ldots, S_m \rangle$, and $S_{\ell+1} \nsubseteq G_\ell \cup \{e_i^{\mathsf{G}}\}$. For convenience, we refer those d arms as $a_1, a_2, \ldots, a_d$, and for any fixed time $T \geq T_0$ (T_0 is sufficiently large), we use $\vec{N} = (N(a_1), N(a_2), \ldots, N(a_d))$ to denote counters of $a_1, \ldots, a_d$, respectively, where each $N(a_k) := N_T(a_k)$. Obviously, $N(a_1) \geq N(a_2) \geq \cdots \geq N(a_d)$. Among all those sequences, since we have W options for each arm a_k after a_{k-1}, by Pigeonhole principle, there exists one sequence satisfying that

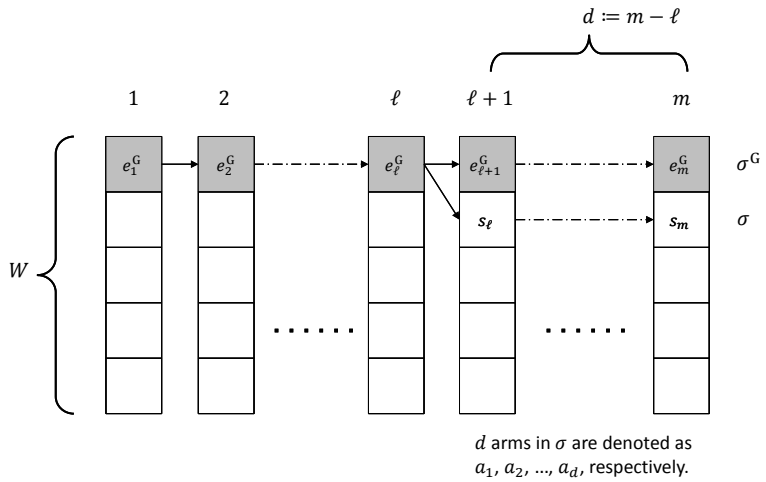
$$N(a_2) \leq \frac{1}{W} N(a_1), \; N(a_3) \leq \frac{1}{W} N(a_2), \; \ldots, \text{ and } N(a_d) \leq \frac{1}{W} N(a_{d-1}). \tag{48}$$

W.L.O.G., assume that $\vec{N}$ satisfies condition (48).

In Lemma E.1, we will show that when d is large enough, the number of playing a_1 is $\frac{1}{(\Delta_2 - \Delta_1)^2} \log T$.

Lemma E.1. *Assume that $\frac{1}{4} < \Delta_1 < \Delta_2 = \frac{1}{2} < \Delta_3 = \frac{3}{4}$. Suppose $W \geq 2$, and integer d satisifies that $d \geq \log_W \frac{2W}{3(\Delta_2 - \Delta_1)^2}$, (i.e., $(\Delta_2 - \Delta_1)^2 \geq \frac{2}{3W^{d-1}}$). For any consistent algorithm, the number*

Figure 1: Illustration of the sequence .



of playing arm a_1 *in the sequence satisfies that*

$$\lim_{t\to+\infty} \frac{\mathrm{E}[N_t(a_1)]}{\ln t} \quad \frac{0:39}{\mathrm{KL}(\ _1 jj \ _2)};$$

where $\mathrm{KL}(p||q) = p \ln \frac{p}{q} + (1 \quad p) \ln \frac{1-p}{1-q}$ *is Kullback-Leibler divergence of Bernoulli random variables.*

Proof. We keep all arms except $a_1; a_2; : : : ; a_d$ by default, and perturb those arms to construct the following two hypotheses:

$$\text{Null hypothesis } H^0 : \quad \overline{f}(a_i) = \ _1; i \ 2 \ f1; \quad ; d \quad 1g; \quad \overline{f}(a_d) = \ _1; \tag{49}$$

$$\text{Alternative hypothesis } H' : \quad \overline{f}(a_i) = \ ; i \ 2 \ f1; \quad ; d \quad 1g; \quad \overline{f}(a_d) = \ _3; \tag{50}$$

where $2 \ (\ _2; 1]$ is some constant to be determined later (in (51)). Note that $^{\mathrm{G}}$ is its greedy sequence in H^0, while is the greedy sequence in H'. For simplicity, we use P[ ] and P′[ ], E[ ] and E′[ ] to denote the probability and expectation under H^0 and H', respectively.

Fix any $2 \ (\frac{2}{3}; 1)$, and let be some constant such that

$$\ _2 < \quad 1; \text{ and } j\mathrm{KL}(\ _1 jj \ ) \quad \mathrm{KL}(\ _1 jj \ _2)j \quad \mathrm{KL}(\ _1 jj \ _2): \tag{51}$$

Define event $N := \left\{ N_{a_1} < \frac{1-}{\mathrm{KL}(\ _1 || \ )} \ln T \right\}$.

Denote $Z_{a;i} \ 2 \ f0; 1g$ as the i-th realization of playing arm a, and $(x; \ ) = \ ^x(1 \quad)^{1-x}$ for $x \ 2 \ f0; 1g$ and $2 \ [0; 1]$ as the probability for Bernoulli random variables. Then we can define function

$$L(N) := \ln \left(\prod_{i=1}^{N_{a_1}} \frac{(Z_{a_1;i}; \ _1)}{(Z_{a_1;i}; \ )} \prod_{i=1}^{N_{a_{d-1}}} \frac{(Z_{a_{d-1};i}; \ _1)}{(Z_{a_{d-1};i}; \ )} \prod_{i=1}^{N_{a_d}} \frac{(Z_{a_d;i}; \ _1)}{(Z_{a_d;i}; \ _3)} \right). \tag{52}$$

Fix any $2 \ (0; 3 \quad 2)$, and define event $L := \left\{ L(N) \quad (1 \quad) \ln T \right\}$. In the following, we will show that

$$\mathrm{P}[N] = \mathrm{P}\left[N(a_1) < \frac{1}{\mathrm{KL}(\ _1 jj \ )} \ln T \right] \ ! \ 0; \text{ as } T \ ! \ 1; \tag{53}$$

by proving that both $\mathrm{P}[N \setminus L]$ and $\mathrm{P}[N \setminus \overline{L}]$ tend to 0.

Step 1: for $\mathrm{P}[N \setminus L]$**.** For H' (is the greedy sequence), the sequence we play is the greedy sequence only if $a_1; \quad ; a_d$ are chosen simultaneously, thus the total number of playing is $N(a_d)$. Furthermore, for any consistent algorithm, for any 2 $(0; \ )$, we have $\mathrm{E}'[T \quad N(a_d)] = o(T \ )$ under H'. Since in our instance $N(a_1) \quad N(a_d)$ holds, thus $\mathrm{E}'[T \quad N(a_1)] \quad \mathrm{E}'[T \quad N(a_d)] = o(T \ )$.

$$\begin{aligned}
\mathrm{P}'[N] &= \mathrm{P}' \quad N(a_1) < \frac{1}{\mathrm{KL}(\ _1 jj \ )} \ln T \\
&= \mathrm{P}' \quad T \quad N(a_1) > T \quad \frac{1}{\mathrm{KL}(\ _1 jj \ )} \ln T \\
&\quad \frac{\mathrm{E}'[T \quad N(a_1)]}{T \quad \frac{1-}{\mathrm{KL}(\ _1 || \ )} \ln T} \qquad \{\text{Markov's inequality}\} \\
&= o(T^{\ -1}): \qquad (54)
\end{aligned}$$

Let $_x$ be the probability measure for H^0, then through the change of probability measure, we can get that

$$\begin{aligned}
\mathrm{P}'[N \setminus L] &= \int_{x \in \mathcal{N} \cap \mathcal{L}} \prod_{i=1}^{N(a_1)} \frac{(Z_{a_1,i}; \ _1)}{(Z_{a_1,i}; \ _1)} \prod_{i=1}^{N(a_{d\ 1})} \frac{(Z_{a_{d\ 1},i}; \ _1)}{(Z_{a_{d\ 1},i}; \ _1)} \prod_{i=1}^{N(a_d)} \frac{(Z_{a_d,i}; \ _2 + \frac{1}{4})}{(Z_{a_d,i}; \ _1)} \mathrm{d}\ _x \\
&= \int_{x \in \mathcal{N} \cap \mathcal{L}} \exp(\ L(N)) \mathrm{d}\ _x \qquad (55) \\
&\quad T^{\ -1} \int_{x \in \mathcal{N} \cap \mathcal{L}} \mathrm{d}\ _x = T^{\ -1} \mathrm{P}[N \setminus L]; \qquad (56)
\end{aligned}$$

where (55) is from (52), and (56) is because L holds. Thus, we can imply from (56) and (54) that

$$\mathrm{P}[N \setminus L] \quad T^{1-} \quad \mathrm{P}'[N \setminus L] \quad T^{1-} \quad \mathrm{P}'[N] = o(1): \qquad (57)$$

Step 2: for $\mathrm{P} \quad N \setminus \overline{L}$ **.** From Equation (52), we know that

$$\begin{aligned}
L(N) &= \sum_{i=1}^{N(a_1)} \ln \frac{(Z_{a_1,i}; \ _1)}{(Z_{a_1,i}; \ )} + \quad + \sum_{i=1}^{N(a_{d\ 1})} \ln \frac{(Z_{a_{d\ 1},i}; \ _1)}{(Z_{a_{d\ 1},i}; \ )} + \sum_{i=1}^{N(a_d)} \ln \frac{(Z_{a_d,i}; \ _2)}{(Z_{a_d,i}; \ _3)} \qquad (58) \\
&! \ (N(a_1) + N(a_2) + \quad + N(a_{d-1})) \mathrm{KL}(\ _1 jj \ ) + N(a_d) \quad \mathrm{KL}(\ _1 jj \ _3) =: \overline{L}(N); \qquad (59)
\end{aligned}$$

as $N(a_1); \quad ; N(a_d)$ tend to be sufficiently large. Due to condition (48) and the definition of N, we have

$$\begin{aligned}
\overline{L}(N) \quad N(1) \quad 1 + \frac{1}{W} + \quad + \frac{1}{W^{d-2}} \quad \mathrm{KL}(\ _1 jj \ ) + N(1) \frac{1}{W^{d-1}} \quad \mathrm{KL}(\ _1 jj \ _3) \\
N(1) \frac{1}{1 \quad \frac{1}{W}} \mathrm{KL}(\ _1 jj \ ) + N(1) \frac{1}{W^{d-1}} \quad \mathrm{KL}(\ _1 jj \ _3) \\
\frac{1}{1 \quad \frac{1}{W}} + \frac{1}{W^{d-1}} \frac{\mathrm{KL}(\ _1 jj \ _3)}{\mathrm{KL}(\ _1 jj \ )} \quad (1 \quad) \ln T: \qquad (60)
\end{aligned}$$

Since $2(x \quad y)^2 \quad \mathrm{KL}(x jj y) \quad \frac{(x-y)^2}{y(1-y)}$ for $x; y \ 2 \ [0; 1)$, $\frac{1}{4} < \ _1 < \ _2 < \ _3 = \frac{3}{4}$, and $\ _2 < \quad 1$, it can be implied that

$$\frac{\mathrm{KL}(\ _1 jj \ _3)}{\mathrm{KL}(\ _1 jj \ )} \quad \frac{(\ _3 \quad \ _1)^2}{\ _3(1 \quad \ _3) \ 2 \ (\quad \ _1)^2} \quad \frac{3}{2(\ _2 \quad \ _1)^2}: \qquad (61)$$

Assume that $W \quad 2$ and $W^{d-1} \quad \frac{2}{3(\ _2 - \ _1)^2}$, then from (60) and (61) we have $\overline{L}(N) \quad 3(1 \quad) \ln T$.

Now x any $\in (0; 3 \quad 2)$, and it is easy to know that $\bar{L}(N) \quad 3(1 \quad) \ln T < (1 \quad) \ln T$. Recall that $L(N) \; ! \; \bar{L}(N)$ from (59) and $N = f N(a_1) < \frac{1}{KL(\; _1 jj \;)} \ln T g \; \$ \; \sum_{j=0}^{\frac{1}{KL(\; _1 jj \;)} \ln T} f N(a_1) = j g$. Then, by the strong law of large numbers, we can get

$$P\{N \setminus \bar{L}\} = P\left\{N(a_1) < \frac{1}{KL(\; _1 jj \;)} \ln T \wedge L(N) > (1 \quad) \ln T\right\} \; ! \; 0; \text{ as } T \; ! \; 1 \quad : \tag{62}$$

Step 3: combine two parts. Because $d \quad \log_W \frac{2W}{3(\; _2 \quad _1)^2}$, we have that $W^d \; ^1 \; \frac{3}{2(\; _2 \quad _1)^2}$. For any $\in (\frac{2}{3}; 1)$, it can be derived from (57) and (62) that $\lim_{T \; ! \; 1} P[N] = \lim_{T \; ! \; 1} P\left\{N_T(a_1) < \frac{1}{KL(\; _1 jj \;)} \ln T \right\} = 0$. Then, from (51), we can get

$$\lim_{T \; ! \; 1} P\left\{N(a_1) < \frac{1}{(1 + \quad) KL(\; _1 jj \; _2)} \ln T \right\} = 0: \tag{63}$$

Therefore,

$$\lim_{t \; ! \; + 1} \frac{E[N_t(a_1)]}{\ln t} \quad \frac{0:39}{KL(\; _1 jj \; _2)}: \tag{64}$$

□

We will show that the greedy regret can be as large as $\Omega(\frac{mW \log T}{2})$, when (1) the unit gap $(\; _2 \; _1)$ is not exponentially small or (2) m is large enough, which will be specified quantitatively in Theorem E.2.

Theorem E.2. For instance P of the prize-collecting problem, for any $_1; \; _2; \; _3$ satisfying that $\frac{1}{4} < \; _1 < \; _2 = \frac{1}{2} < \; _3 = \frac{3}{4}$ and $W \quad 2$, assume that there exists some constant $2(0; 1)$ such that $m \quad \log_W \frac{2W}{3(\; _2 \quad _1)^2}$ (i.e., $(\; _2 \quad _1)^2 \quad \frac{2}{3W^{\; m \; 1}}$). Denote $= \; _2 \quad _1 > 0$. For any consistent algorithm, as time T tends to $+1$, the greedy regret satisfies that

$$R^G(T) \quad \frac{0:39 \quad (1 \quad) m(W \quad 1) \; _{min}}{KL(\; _1 jj \; _2)} \ln T; \tag{65}$$

where $KL(pjjq) = p \ln \frac{p}{q} + (1 \quad p) \ln \frac{1 \; p}{1 \; q}$ is Kullback-Leibler divergence for Bernoulli random variables, and $_{min} = + \; \frac{1}{4}$.

Proof. From the assumption, we know that $m \quad \log_W \frac{2W}{3(\; _2 \quad _1)^2}$. Fix any $` = 1; 2; \; ; (1 \quad) m$, then it is easy to see that the deviated $m \quad `$ arms (denoted as $a_1; a_2; \; ; a_d$) satisfying $d \quad \log_W \frac{2W}{3(\; _2 \quad _1)^2}$.

Denote the true greedy sequence as $G = hG_0; G_1; \; ; G_m i$. For any sequence coinciding with ^{G} for the prefix $hG_0; G_1; \; ; G_{`} i$, for phase $` + 1$, we can choose $W \quad 1$ candidates from $N(G_{`}) n f g_{G_{`}} g$. According to Lemma E.1, for each candidate, the consistent algorithm needs to play arm $_1$ for at least $\frac{0:39}{KL(\; _1 jj \; _2)} \ln T$ times. Since the penalty is at least $_{min} = + \; \frac{1}{4}$, therefore we can conclude that

$$R^G(T) \quad \sum_{`=1}^{(1 \quad) m} (W \quad 1) \; \frac{0:39 \quad _{min}}{KL(\; _1 jj \; _2)} \ln T = \frac{0:39 \quad (1 \quad) m(W \quad 1) \; _{min}}{KL(\; _1 jj \; _2)} \ln T: \tag{66}$$

□

From Theorem E.2 and $KL(\; _1 jj \; _2) \quad \frac{(\; _2 \quad _1)^2}{2(1 \quad _2)} = 4 \; ^2$, it is easy to check that the greedy regret is $\frac{mW}{2 \; _{min}} \log T$. Since $_{max} = m + \; \frac{1}{4}$ and $_{min} = + \; \frac{1}{4}$ in the setting of Theorem E.2, suppose we further assume that $= \; _2 \quad _1 = o(\frac{1}{m})$ (e.g., $= \frac{0:01}{m^{1:1}}$), then it indicates that $_{max} = \; _{min} + o(1)$. Thus, we can get that the lower bound $R^G(T) = \frac{mW}{2 \; _{max}} \log T$, which matches with the upper bound up to a constant factor.