

$O(\log \log \text{rank})$ competitive-ratio for the Matroid Secretary Problem

Oded Lachish*

Abstract

In the *Matroid Secretary Problem* (MSP), the elements of the ground set of a Matroid are revealed on-line one by one, each together with its value. An algorithm for the Matroid Secretary Problem is *Matroid-Unknown* if, at every stage of its execution: (i) it only knows the elements that have been revealed so far and their values, and (ii) it has access to an oracle for testing whether or not any subset of the elements that have been revealed so far is an independent set. An algorithm is *Known-Cardinality* if, in addition to (i) and (ii), it also initially knows the cardinality of the ground set of the Matroid.

We present here a Known-Cardinality and *Order-Oblivious* algorithm that, with constant probability, selects an independent set of elements, whose value is at least the optimal value divided by $O(\log \log \rho)$, where ρ is the rank of the Matroid; that is, the algorithm has a *competitive-ratio* of $O(\log \log \rho)$. The best previous results for a Known-Cardinality algorithm are a competitive-ratio of $O(\log \rho)$, by Babaioff *et al.* (2007), and a competitive-ratio of $O(\sqrt{\log \rho})$, by Chakraborty and Lachish (2012).

In many non-trivial cases the algorithm we present has a competitive-ratio that is better than the $O(\log \log \rho)$. The cases in which it fails to do so are characterized. Understanding these cases may lead to improved algorithms for the problem or, conversely, to non-trivial lower bounds.

1 Introduction

The *Matroid Secretary Problem* is a generalization of the *Classical Secretary Problem*, whose origins seem to still be a source of dispute. One of the first papers on the subject [12], by Dynkin, dates back to 1963. Lindley [21] and Dynkin [12] each presented an algorithm that achieves a *competitive-ratio* of e , which is the best possible. See [14] for more information about results preceding 1983.

In 2007, Babaioff *et al.* [4] established a connection between the Matroid Secretary Problem and *mechanism design*. This is probably the cause of an increase of interest in generalizations of the *Classical Secretary Problem* and specifically the Matroid Secretary Problem.

In the Matroid Secretary Problem, we are given a Matroid $\{U, \mathcal{I}\}$ and a value function assigning non-negative values to the Matroid elements. The elements of the Matroid are revealed in an on-line fashion according to an unknown order selected uniformly at random. The value of each element

*Birkbeck, University of London, London, UK. Email: oded@dcs.bbk.ac.uk

is unknown until it is revealed. Immediately after each element is revealed, if the element together with the elements already selected does not form an independent set, then that element cannot be selected; however, if it does, then an irrevocable decision must be made whether or not to select the element. That is, if the element is selected, it will stay selected until the end of the process and likewise if it is not. The goal is to design an algorithm for this problem with a small competitive-ratio, that is the ratio between the maximum sum of values of an independent set and the expected sum of values of the independent set returned by the algorithm.

An algorithm for the *Matroid Secretary Problem* (MSP) is called *Matroid-Unknown* if, at every stage of its execution, it only knows (i) the elements that have been revealed so far and their values and (ii) an oracle for testing whether or not a subset of the elements that have been revealed so far forms an independent set. An algorithm is called *Known-Cardinality* if it knows (i), (ii) and also knows from the start the cardinality n of the ground set of the Matroid. An algorithm is called *Matroid-Known*, if it knows, from the start, everything about the Matroid except for the values of the elements. These, as mentioned above, are revealed to the algorithm as each element is revealed.

Related Work Our work follows the path initiated by Babaioff *et al.* in [4]. There they formalized the Matroid Secretary Problem and presented a Known-Cardinality algorithm with a competitive-ratio of $\log \rho$. This line of work was continued in [8], where an algorithm with a competitive-ratio of $O(\sqrt{\log \rho})$ was presented. In Babaioff *et al.* [4] (2007), it was conjectured that a constant competitive-ratio is achievable. The best known result for a *Matroid-Unknown* algorithm, implied by the works of Gharan and Vondrák [15] and Chakraborty and Lachish [8] (2012): for every fixed $\epsilon > 0$, there exists a Matroid-Unknown algorithm with a competitive-ratio of $O(\epsilon^{-1}(\sqrt{\log \rho}) \log^{1+\epsilon} n)$. Gharan and Vondrák showed that a lower bound of $\Omega(\frac{\log n}{\log \log n})$ on the competitive-ratio holds in this case.

Another line of work towards resolving the Matroid Secretary Problem is the study of the Secretary Problem for specific families of Matroids. Most of the results of this type are for Matroid-Known algorithms and all achieve a constant competitive-ratio. Among the specific families of Matroids studied are *Graphic Matroids* [4], *Uniform/Partition Matroids* [3, 19], *Transversal Matroids* [9, 20], *Regular and Decomposable Matroids* [11] and *Laminar Matroids* [17]. For surveys that also include other variants of the Matroid Secretary Problem see [23, 18, 10].

There are also results for other generalizations of the Classical Secretary Problem, including the *Knapsack Secretary Problem* [3], *Secretary Problems with Convex Costs* [5], *Sub-modular Secretary Problems* [6, 16, 13] and *Secretary problems via linear programming* [7].

Main result We present here a Known-Cardinality algorithm with a competitive-ratio of $O(\log \log \rho)$. The algorithm is also Order-Oblivious as defined by Azar *et al.* [2]). Definition 13 is a citation of their definition of an Order-Oblivious algorithm for the Matroid Secretary Problem. According to [15], this implies that, for every fixed $\epsilon > 0$, there exists a Matroid-Unknown algorithm with a competitive-ratio of $O(\epsilon^{-1}(\log \log \rho) \log^{1+\epsilon} n)$. Our algorithm is also Order-Oblivious as in Definition 1 of [2], and hence, by Theorem 1 of [2], this would imply that there exists a Single Sample Prophet Inequality for Matroids with a competitive-ratio of $O(\log \log \rho)$.

In many non-trivial cases the algorithm we present has a competitive-ratio that is better than

the $O(\log \log \rho)$. The cases in which it fails to do so are characterized. Understanding these cases may lead to improved algorithms for the problem or, conversely, to non-trivial lower bounds.

High level description of result and its relation to previous work. As in [4] and [8], here we also partition the elements into sets which we call *buckets*. This is done by rounding down the value of each element to the largest possible power of two and then, for every power of two, defining a bucket to be the set of all elements with that value. Obviously, the only impact this has on the order of the competitive-ratio achieved is a constant factor of at most 2.

We call our algorithm the Main Algorithm. It has three consecutive stages: *Gathering stage*, *Preprocessing stage* and *Selection stage*. In the Gathering stage it waits, without selecting any elements, until about half of the elements of the matroid are revealed. The set F that consists of all the elements revealed during the Gathering stage is the input to the Preprocessing stage. In the Preprocessing stage, one out of the following three types of output is computed: (i) a non negative value, (ii) a set of bucket indices, or (iii) a critical tuple. Given the output of the Preprocessing stage, before any element is revealed the Main Algorithm chooses one of the following algorithms: the *Threshold Algorithm*, the *Simple Algorithm* or the *Gap Algorithm*. Then, after each one of the remaining elements is revealed, the decision whether to select the element is made by the chosen algorithm using the input received from the Preprocessing stage and the set of all the elements already revealed. Once all the elements have been revealed the set of selected elements is returned.

The Threshold Algorithm is chosen when the output to the Preprocessing stage is a non-negative value, which happens with probability half regardless of the contents of the set F . Given this input, the Threshold Algorithm, as in the algorithm for the Classical Secretary Problem, selects only the first element that has at least the given value. The Simple Algorithm is chosen when the output of Preprocessing stage is a set of bucket indices. The Simple Algorithm selects an element if it is in one of the buckets determined by the set of indices and if it is independent of all previously selected elements. This specific algorithm was also used in [8].

The Gap Algorithm is chosen when the output of Selection stage is a critical tuple, which we define further on. The Gap Algorithm works as follows: every element revealed is required to have one of a specific set of values and satisfy two conditions in order to be selected: it satisfies the first condition if it is in the closure of a specific subset of elements of F ; it satisfies the second condition if it is not in the closure of the union of the set of elements already selected and a specific subset of elements of F (which is different than the one used in the first condition).

The proof that the Main Algorithm achieves the claimed competitive-ratio consists of the following parts: a guarantee on the output of the Simple Algorithm as a function of the input and $U \setminus F$, where U is the ground set of the matroid; a guarantee on the output of the Gap Algorithm as a function of the input and $U \setminus F$; a combination of a new structural result for matroids and probabilistic inequalities that imply that if the matroid does not have an element with a large value, then it is possible to compute an input for either the Simple Algorithm or the Gap Algorithm that, with high probability, ensures that the output set has a high value. This guarantees the claimed competitive-ratio, since the case when the matroid has an element with a large value is dealt with by the Threshold Algorithm.

The paper is organized as follows: Section 2 contains the preliminaries; Section 3 presents Main Algorithm; Section 4 is devoted to the Simple Algorithm and the Gap Algorithm; Section 5 contains the required concentrations; the structural trade-off result is proved in Section 6; the main result appears in Section 7; and in Section 8 we characterize the cases in which the algorithm performs exactly as guaranteed and give non-trivial example in which the algorithm performs better than the guaranteed competitive-ratio.

2 Preliminaries

All logarithms are to the base 2. We use \mathbb{Z} to denote the set of all integers, \mathbb{N} to denote the non-negative integers and \mathbb{N}^+ to denote the positive integers. We use $\lceil \alpha \rceil$ to denote $\{1, 2, \dots, \lceil \alpha \rceil\}$ for any non-negative real α . We use $[\alpha, \beta]$ to denote $\{i \in \mathbb{Z} \mid \alpha \leq i \leq \beta\}$ and (α, β) to denote $\{i \in \mathbb{Z} \mid \alpha < i < \beta\}$, and so on. We use $\text{med}(f)$ to denote the *median* of a function f from a finite set to the non-negative reals. If there are two possible values for $\text{med}(f)$ the smaller one is chosen.

We define $\beta(n, 1/2)$ to be a random variable whose value is the number of successes in n independent probability $1/2$ Bernoulli trials.

Observation 1 *Let $A = \{a_1, a_2, \dots, a_n\}$ and $W = \beta(n, 1/2)$; let $\pi : [n] \rightarrow [n]$ be a permutation selected uniformly at random, and let $D = \{a_{\pi(i)} \mid i \in [W]\}$. For every $i \in [n]$, we have that $a_i \in D$ independently with probability $1/2$.*

Proof. To prove the proposition we only need to show that for every $C \subseteq A$, we have $D = C$ with probability 2^{-n} . Fix C . There are $\binom{n}{|C|}$ subsets of A of size $|C|$. D is equally likely to be one of these subsets. Hence, the probability that $|D| = |C|$ is $\binom{n}{|C|} \cdot 2^{-n}$ and therefore the probability that $D = C$ is $\binom{n}{|C|} \cdot 2^{-n} / \binom{n}{|C|} = 2^{-n}$. ■

2.1 Matroid definitions, notations and preliminary results

Definition 2 [Matroid] *A **matroid** is an ordered pair $M = (U, \mathcal{I})$, where U is a set of **elements**, called the **ground set**, and \mathcal{I} is a family of subsets of U that satisfies the following:*

- *If $I \in \mathcal{I}$ and $I' \subset I$, then $I' \in \mathcal{I}$*
- *If $I, I' \in \mathcal{I}$ and $|I'| < |I|$, then there exists $e \in I \setminus I'$ such that $I' \cup \{e\} \in \mathcal{I}$.*

*The sets in \mathcal{I} are called **independent sets** and a maximal independent set is called a **basis**.*

A *value function* over a Matroid $M = (U, \mathcal{I})$ is a mapping from the elements of U to the non-negative reals. Since we deal with a fixed Matroid and value function, we will always use $M = (U, \mathcal{I})$ for the Matroid. We set $n = |U|$ and, for every $e \in U$, we denote its value by $\text{val}(e)$.

Definition 3 [rank and Closure] *For every $S \subseteq U$, let*

- *$\text{rank}(S) = \max\{|S'| \mid S' \in \mathcal{I} \text{ and } S' \subseteq S\}$ and*
- *$\text{Cl}(S) = \{e \in U \mid \text{rank}(S \cup \{e\}) = \text{rank}(S)\}$.*

The following proposition captures a number of standard properties of Matroids; the proofs can be found in [22]. We shall only prove the last assertion.

Proposition 4 *Let S_1, S_2, S_3 be subsets of U and $e \in U$ then*

1. $\text{rank}(S_1) \leq |S_1|$, where equality holds if and only if S_1 is an independent set,
2. if $S_1 \subseteq S_2$ or $S_1 \subseteq \text{Cl}(S_2)$, then $S_1 \subseteq \text{Cl}(S_1) \subseteq \text{Cl}(S_2)$ and $\text{rank}(S_1) \leq \text{rank}(S_2)$,
3. if $e \notin \text{Cl}(S_1)$, then $\text{rank}(S_1 \cup \{e\}) = \text{rank}(S_1) + 1$,
4. $\text{rank}(S_1 \cup S_2) \leq \text{rank}(S_1) + \text{rank}(S_2)$,
5. $\text{rank}(S_1 \cup S_2) \leq \text{rank}(S_1) + \text{rank}(S_2 \setminus \text{Cl}(S_1))$, and
6. suppose that S_1 is minimal such that $e \in \text{Cl}(S_1 \cup S_2)$, but $e \notin \text{Cl}((S_1 \cup S_2) \setminus \{e^*\})$, for every $e^* \in S_1$, then $e^* \in \text{Cl}(\{e\} \cup ((S_1 \cup S_2) \setminus \{e^*\}))$, for every $e^* \in S_1$.

Proof. We prove Item 6. The rest of the items are standard properties of Matroids.

Let $e^* \in S_1$. By Item 3, $\text{rank}(\{e\} \cup ((S_1 \cup S_2) \setminus \{e^*\}))$ is equal to $\text{rank}(S_1 \cup S_2)$ which is equal to $\text{rank}(\{e\} \cup S_1 \cup S_2)$ which in turn is equal to $\text{rank}(\{e\} \cup ((S_1 \cup S_2) \setminus \{e^*\}) \cup \{e^*\})$. Thus, again by Item 3, this implies that $e^* \in \text{Cl}(\{e\} \cup ((S_1 \cup S_2) \setminus \{e^*\}))$. ■

Assumption 5 $\text{val}(e) = 0$, for every $e \in U$ such that $\text{rank}(\{e\}) = 0$. For every $e \in U$ such that $\text{val}(e) > 0$, there exists $i \in \mathbb{Z}$ such that $\text{val}(e) = 2^i$.

In the worst case, the implication of this assumption is an increase in the competitive ratio by a multiplicative factor that does not exceed 2, compared with the competitive ratio we could achieve without this assumption.

Definition 6 [Buckets] For every $i \in \mathbb{Z}$, the i 'th bucket is $B_i = \{e \in U \mid \text{val}(e) = 2^i\}$. We also use the following notation for every $S \subseteq U$ and $J \subset \mathbb{Z}$:

- $B_i^S = B_i \cap S$,
- $B_J = \bigcup_{i \in J} B_i$ and
- $B_J^S = \bigcup_{i \in J} B_i^S$.

Definition 7 [OPT] For every $S \subseteq U$, let $\text{OPT}(S) = \max \left\{ \sum_{e \in S'} \text{val}(e) \mid S' \subseteq S \text{ and } S' \in \mathcal{I} \right\}$.

We note that if S is independent, then $\text{OPT}(S) = \sum_{e \in S} \text{val}(e)$.

Observation 8 For every independent $S \subseteq U$, $\text{OPT}(S) = \sum_{i \in \mathbb{Z}} 2^i \cdot \text{rank}(B_i^S)$.

Definition 9 [LOPT] For every $S \subseteq U$, we define $\text{LOPT}(S) = \sum_{i \in \mathbb{Z}} 2^i \cdot \text{rank}(B_i^S)$.

Observation 10 For every $S \subseteq U$ and $J_1, J_2 \subseteq \mathbb{Z}$,

1. $\text{LOPT}(S) \geq \text{OPT}(S)$,
2. $\text{LOPT}(B_{J_1}^S) = \sum_{i \in J_1} 2^i \cdot \text{rank}(B_i^S)$ and
3. if $J_1 \cap J_2 = \emptyset$, then $\text{LOPT}(B_{J_1 \cup J_2}^S) = \text{LOPT}(B_{J_1}^S) + \text{LOPT}(B_{J_2}^S)$.

2.2 Matroid Secretary Problem

Definition 11 [*competitive-ratio*] Given a Matroid $\mathcal{M} = (U, \mathcal{I})$, the competitive-ratio of an algorithm that selects an independent set $P \subseteq U$ is the ratio of $OPT(U)$ to the expected value of $OPT(P)$.

Problem 12 [*Known-Cardinality Matroid Secretary Problem*] The elements of the Matroid $M = (U, \mathcal{I})$ are revealed in random order in an on-line fashion. The cardinality of U is known in advance, but every element and its value are unknown until revealed. The only access to the structure of the Matroid is via an oracle that, upon receiving a query in the form of a subset of elements already revealed, answers whether the subset is independent or not. An element can be selected only after it is revealed and before the next element is revealed, and then only provided the set of selected elements remains independent at all times. Once an element is selected it remains selected. The goal is to design an algorithm that maximizes the expected value of $OPT(P)$, i.e., achieves as small a **competitive-ratio** as possible.

Definition 13 (*Definition 1 in [2]*). We say that an algorithm S for the secretary problem (together with its corresponding analysis) is **order-oblivious** if, on a randomly ordered input vector $(v_{i_1}, \dots, v_{i_n})$:

1. (algorithm) S sets a (possibly random) number k , observes without accepting the first k values $S = \{v_{i_1}, \dots, v_{i_k}\}$, and uses information from S to choose elements from $V = \{v_{i_{k+1}}, \dots, v_{i_n}\}$.
2. (analysis) S maintains its competitive ratio even if the elements from V are revealed in any (possibly adversarial) order. In other words, the analysis does not fully exploit the randomness in the arrival of elements, it just requires that the elements from S arrive before the elements of V , and that the elements of S are the first k items in a random permutation of values.

3 The Main Algorithm

The input to the Main Algorithm is the number of indices n in a randomly ordered input vector (e_1, \dots, e_n) , where $\{e_1, \dots, e_n\}$ are the elements of the ground set of the matroid. These are revealed to the Main Algorithm one by one in an on-line fashion in the increasing order of their indices. The Main Algorithm executes the following three stages:

1. **Gathering stage.** Let $W = \beta(n, 1/2)$. Wait until W elements are revealed without selecting any. Let F be the set of all these elements.
2. **Preprocessing stage.** Given only F , before any item of $U \setminus F$ is revealed, one of the following three types of output is computed: (i) a non-negative value, (ii) a set of bucket indices, or (iii) a critical tuple which is defined in Subsection 4.2.
3. **Selection stage.** One out of three algorithms is chosen and used in order to decide which elements from $U \setminus F$ to select, when they are revealed. If the output of Preprocessing stage is a non-negative value, then the *Threshold Algorithm* is chosen, if it is a set of bucket indices, then the *Simple Algorithm* is chosen and if it is a critical tuple, then the *Gap Algorithm* is chosen.

With probability $\frac{1}{2}$, regardless of F , the output of the Preprocessing stage is the largest value of the elements of F . The Threshold Algorithm, which is used in this case, selects the first revealed element of $U \setminus F$ that has a value at least as large as the output of the Preprocessing stage. This ensures that if $\max\{\text{val}(e) \mid e \in U\} \geq 2^{-19} \cdot \text{OPT}(U)$, then the claimed competitive-ratio is achieved. So for the rest of the paper we make the following assumption:

Assumption 14 $\max\{\text{val}(e) \mid e \in U\} < 2^{-19} \cdot \text{OPT}(U)$.

The paper proceeds as follows: in Subsection 4.1, we present the Simple Algorithm and formally prove a guarantee on its output; in Subsection 4.2, we define critical tuple, describe the Gap Algorithm and formally prove a guarantee on its output; in Section 5, prove the required concentrations; in Section 6, we prove our structural trade-off result; and in Section 7, we prove the main result.

4 The Simple Algorithm and the Gap Algorithm

In this section we present the pseudo-code for the Simple Algorithm and the Gap Algorithm, and prove the guarantees on the competitive-ratios they achieve. We start with the Simple Algorithm, which is also used in [8].

4.1 The Simple Algorithm

Algorithm 1 Simple Algorithm

Input: a set J of bucket indices

1. $P \leftarrow \emptyset$
2. immediately after each element $e \in U \setminus F$ is revealed, do
 - (a) if $\log \text{val}(e) \in J$ do
 - i. if $e \notin \text{Cl}(P)$ do $P \leftarrow P \cup \{e\}$

Output: P

We note that according to Steps 2a and 2(a)i, the output P of the Simple Algorithm always satisfies, $B_J^{U \setminus F} \subseteq \text{Cl}(P)$. Thus, since $P \subseteq B_J^{U \setminus F}$, the output P of the Simple Algorithm always satisfies, $\text{rank}(P) = \text{rank}(B_J^{U \setminus F})$. As a result, for every $j \in J$, we are guaranteed that P contains at least $\text{rank}(B_J^{U \setminus F}) - \text{rank}(B_{J \setminus \{j\}}^{U \setminus F})$ elements from $B_j^{U \setminus F}$. We capture this measure using the following definition:

Definition 15 [*uncov*] $\text{uncov}(R, S) = \text{rank}(R \cup S) - \text{rank}(R)$, for every $R, S \subseteq U$.

It is easy to show that

Observation 16 $\text{uncov}(R, S)$ is monotonic decreasing in R .

According to this definition, for every $j \in J$, we are guaranteed that P contains at least $\text{uncov}(B_{J \setminus \{j\}}^{U \setminus F}, B_j^{U \setminus F})$ elements from $B_j^{U \setminus F}$. We next prove this in a slightly more general setting that is required for the Gap Algorithm.

Lemma 17 *Suppose that the input to the Simple Algorithm is a set $J \subset \mathbb{Z}$ and, instead of the elements of $U \setminus F$, the elements of a set $S \subseteq U$ are revealed in an arbitrary order to the Simple Algorithm. Then the Simple Algorithm returns an independent set $P \subseteq S$ such that, for every $j \in J$, $\text{rank}(B_j^P) \geq \text{uncov}(B_{J \setminus \{j\}}^S, B_j^S)$.*

Proof. By the same reasoning as described in the beginning of this section, for every $j \in J$, we are guaranteed that P contains at least $\text{uncov}(B_{J \setminus \{j\}}^{U \setminus F}, B_j^{U \setminus F})$ elements from $B_j^{U \setminus F}$ and the result follows. ■

We next prove the following guarantee on the output of the Simple Algorithm, by using the preceding lemma.

Theorem 18 *Given a set $J \subset \mathbb{Z}$ as input, the Simple Algorithm returns an independent set $P \subseteq U \setminus F$ such that*

$$\text{OPT}(P) \geq \sum_{j \in J} 2^j \cdot \text{uncov}(B_{J \setminus \{j\}}^{U \setminus F}, B_j^{U \setminus F}).$$

Proof. By Observation 8, $\text{OPT}(P)$ is at least $\sum_{j \in J} 2^j \cdot \text{rank}(B_j^P)$, which is at least $\sum_{j \in J} 2^j \cdot \text{uncov}(B_{J \setminus \{j\}}^{U \setminus F}, B_j^{U \setminus F})$, by Lemma 17. The result follows. ■

We note that the above guarantee is not necessarily the best possible. However, it is sufficient for our needs because, as we show later on, with very high probability, for a specific family of sets J and every j in such J , we have that $\text{uncov}(B_{J \setminus \{j\}}^{U \setminus F}, B_j^{U \setminus F}) \approx \text{uncov}(B_{J \setminus \{j\}}^F, B_j^F)$. Thus, in relevant cases, we can approximate this guarantee using only the elements of F .

Corollary 19 *Given a set $J = \{k\}$ as input, the Simple Algorithm returns an independent set $P \subseteq U \setminus F$ such that $\text{OPT}(P) \geq 2^k \cdot \text{rank}(B_k^{U \setminus F})$.*

4.2 The Gap Algorithm

The subsection starts with a description of the input to the Gap Algorithm and how it works; afterwards it provides a formal definition of the Gap Algorithm and its input and then concludes with a formal proof of the guarantee on the Gap Algorithm's output.

Like the Simple Algorithm the elements of $U \setminus F$ are revealed to the Gap Algorithm one by one in an on-line manner. The input to the Gap Algorithm is a tuple (Block, Good, Bad), called a critical tuple. Block is a mapping from the integers \mathbb{Z} to the power set of the integers, such that if $\text{Block}(i)$ is not empty then $i \in \text{Block}(i)$. Block determines from which buckets the Gap Algorithm may select elements. Specifically, an element $e \in U \setminus F$ may be selected only if $\text{Block}(\log \text{val}(e))$ is not empty. Every pair of not empty sets $\text{Block}(i)$ and $\text{Block}(j)$, where $i \geq j$, are such that either $\text{Block}(i) = \text{Block}(j)$ or $\min \text{Block}(i) > \max \text{Block}(j)$ and the latter may occur only if $i > j$. We next formally define the critical tuple.

Definition 20 [*critical tuple, BLOCK*] (Block, Good, Bad), where Good, Bad and Block are mappings from \mathbb{Z} to $2^{\mathbb{Z}}$, is a **critical tuple** if the following hold for every $i, j \in \mathbb{Z}$ such that $i \geq j$ and $\text{Block}(i)$ and $\text{Block}(j)$ are not empty:

1. $i \in \text{Block}(i)$,

2. if $i > j$ either $\text{Block}(i) = \text{Block}(j)$ or $\min \text{Block}(i) > \max \text{Block}(j)$,
3. if $\text{Block}(i) = \text{Block}(j)$, then $\text{Good}(i) = \text{Good}(j)$ and $\text{Bad}(i) = \text{Bad}(j)$,
4. $\text{Block}(i) \cup \text{Bad}(i) \subseteq \text{Good}(i)$,
5. if $\min \text{Block}(i) > \max \text{Block}(j)$, then $\text{Bad}(i) \subseteq \text{Good}(i) \subseteq \text{Bad}(j) \subseteq \text{Good}(j)$,
6. $\max \text{Block}(i) < \min \text{Bad}(i)$.

We define $\text{BLOCK} = \{i \mid \text{Block}(i) \neq \emptyset\}$.

For a depiction of the preceding structure see Figure 1.

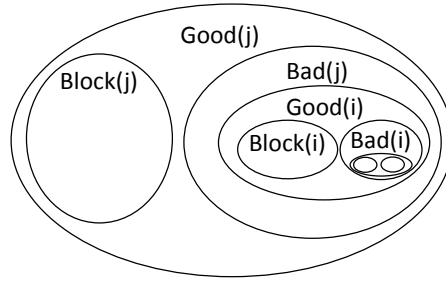


Figure 1: [critical tuple, where $\min \text{Block}(i) > \max \text{Block}(j)$]

The following observation, follow directly from the preceding definition.

Observation 21 *If $(\text{Block}, \text{Good}, \text{Bad})$ is a **critical tuple**, then*

1. the sets in $\{\text{Block}(j)\}_{j \in \mathbb{Z}}$ are pairwise-disjoint,
2. $\text{Block}(i) \cap \text{Bad}(i) = \emptyset$, for every $i \in \mathbb{Z}$, and
3. for every i and j in $\bigcup_{\ell \in \mathbb{Z}} \text{Block}(\ell)$, if $j \notin \text{Good}(i)$, then $i > j$ and $\text{Good}(i) \subseteq \text{Bad}(j)$.

The mappings Good and Bad are used in order to determine if an element can be selected as follows: an element $e \in U \setminus F$ such that $\log \text{val}(e) \in \text{Block}(\log \text{val}(e))$ is selected if it satisfies two conditions: (i) $e \in \text{Cl}(B_{\text{Good}(i)}^F)$; and (ii) e is in the closure of the union of $B_{\text{Bad}(i)}^F$ and all the previously selected elements. We next explain why this strategy works.

Clearly, the only elements in $B_i^{U \setminus F}$ that do not satisfy condition (i) are those in $B_j^{U \setminus F} \setminus \text{Cl}(B_{\text{Good}(i)}^F)$. An essential part of our result is an upper bound on the rank of the set $B_i^{U \setminus F} \setminus \text{Cl}(B_{\text{Good}(i)}^F)$ and hence we use the following definition to capture this quantity.

Definition 22 [*loss*] *For every $R, S \subseteq U$, let $\text{loss}(R, S) = \text{rank}(S \setminus \text{Cl}(R))$.*

According to this definition and the preceding explanation we are guaranteed that the rank of the set of elements in $B_i^{U \setminus F}$ that satisfy condition (i) is at least $\text{rank}(B_i^{U \setminus F}) - \text{loss}(B_{\text{Good}(i)}^F, B_i^{U \setminus F})$.

For every $j \in \text{Block}(i)$, let $S_j = B_j^{U \setminus F} \cap \text{Cl}(B_{\text{Good}(j)}^F)$, that is, the elements of S_j are the elements of $B_j^{U \setminus F}$ that satisfy condition (i). We will show that such an element satisfies condition (ii) if it is not in the closure of the union of $B_{\text{Bad}(i)}^F$ and **only** all the elements from $B_i^{U \setminus F}$ that were previously selected. The reason this happens is that, for every $j' > i$, such that $\min \text{Block}(j') > \max \text{Block}(i)$ all the element selected from $B_{j'}^{U \setminus F}$, satisfy condition (i) and hence are in $\text{Cl}(B_{\text{Bad}(i)}^F)$, and for every $j' < i$, such that $\max \text{Block}(j') < \min \text{Block}(i)$ the condition (ii) ensures, for every $j \in \text{Block}(j')$, that each element selected from $B_j^{U \setminus F}$ will not prevent the selection of any element from S_i because $S_i \subseteq \text{Cl}(B_{\text{Good}(i)}^F) \subset \text{Cl}(B_{\text{Bad}(j)}^F)$.

Thus, when restricted to the elements of $\bigcup_{j \in \text{Block}(i)} S_j$, the Gap Algorithm can be viewed as if it was executing the Simple Algorithm with input $J = \text{Block}(i) \cup \text{Bad}(i)$ and the elements revealed are those of $S = B_{\text{Bad}(i)}^F \cup \bigcup_{j \in \text{Block}(i)} S_j$, which are revealed in an arbitrary order, except that the elements of $B_{\text{Bad}(j)}^F$ are revealed first. Thus, using Lemma 17, it is straight forward to see that at least $\text{uncov}(B_{\text{Bad}(i)}^F \cup \bigcup_{j \in \text{Block}(i) \setminus \{i\}} S_j, S_{i'})$ are selected from $S_{i'}$, for every $i' \in \text{Block}(i)$. We shall show, that this term, is at least $\text{uncov}(B_{\text{Bad}(j)}^F \cup B_{\text{Block}(i) \setminus \{i\}}^{U \setminus F}, S_{i'})$, which in turn is at least $\text{uncov}(B_{\text{Bad}(j)}^F \cup B_{\text{Block}(i) \setminus \{i\}}^{U \setminus F}, B_{i'}^{U \setminus F}) - \text{loss}(B_{\text{Good}(i)}^F, B_i^{U \setminus F})$. In Section 5, we show that with high probability, by using only the elements of F , we can approximate $\text{uncov}(B_{\text{Bad}(j)}^F \cup B_{\text{Block}(i) \setminus \{i\}}^{U \setminus F}, B_{i'}^{U \setminus F})$ and upper bound $\text{loss}(B_{\text{Good}(i)}^F, B_i^{U \setminus F})$. In Section 6, we use the result of Section 5 to show that, if there is no element with a very high value, then either the Simple Algorithm or the Gap Algorithm will achieve the required competitive-ratio and we can choose the proper option using only the elements of F .

Algorithm 2 Gap Algorithm

Input: a critical tuple (**Block**, **Good**, **Bad**)

1. $P \leftarrow \emptyset$
2. immediately after each element $e \in U \setminus F$ is revealed do
 - (a) $\ell \leftarrow \log \text{val}(e)$
 - (b) if $\text{Block}(\ell) \neq \emptyset$ do
 - i. if $e \in \text{Cl}(B_{\text{Good}(\ell)}^F)$, do
 - A. if $e \notin \text{Cl}(P \cup B_{\text{Bad}(\ell)}^F)$, do $P \leftarrow P \cup \{e\}$

Output: P

Lemma 23 *Let i be such that $\text{Block}(i) \neq \emptyset$ and P as it was in any stage in an arbitrary execution of the Gap Algorithm. If $e \in B_{\text{Block}(i)}^{U \setminus F} \cap \text{Cl}(B_{\text{Good}(i)}^F)$, then $e \notin \text{Cl}(P \cup B_{\text{Bad}(i)}^F)$ if and only if $e \notin \text{Cl}((P \cap B_{\text{Block}(i)}^{U \setminus F}) \cup B_{\text{Bad}(i)}^F)$.*

Proof. Let $e \in B_{\text{Block}(i)}^{U \setminus F} \cap \text{Cl}(B_{\text{Good}(i)}^F)$. We note that the "only if" condition trivially holds and

hence we only prove the "if" condition. Let $P^* = P \cap B_{\text{Block}(i)}^{U \setminus F}$. Assume that $e \in \text{Cl}(P \cup B_{\text{Bad}(i)}^F)$. Let C be a minimal subset of $P \setminus (P^* \cup \text{Cl}(B_{\text{Bad}(i)}^F))$ such that $e \in \text{Cl}(C \cup B_{\text{Bad}(i)}^F \cup P^*)$. We shall show that $C = \emptyset$ and hence $e \in \text{Cl}((P \cap B_{\text{Block}(i)}^{U \setminus F}) \cup B_{\text{Bad}(i)}^F)$. Hence the result then follows.

Let e' be the latest element C added to P and let $j = \log \text{val}(e')$. According to construction, the elements of C were selected by the Gap Algorithm and hence $\text{Block}(j) \neq \emptyset$. Also, by construction, $\text{Block}(i) \neq \text{Block}(j)$, since otherwise $e' \in P^* = P \cap B_{\text{Block}(i)}^{U \setminus F}$.

Suppose that $\min \text{Block}(j) > \max \text{Block}(i)$. By Items 4 and 5 of Definition 20, this implies that $\text{Block}(j) \subseteq \text{Good}(j) \subseteq \text{Bad}(i)$. Since e' was selected by the Gap Algorithm, by Step 2(b)i, this implies that $e' \in \text{Cl}(B_{\text{Good}(j)}^F) \subseteq \text{Cl}(B_{\text{Bad}(i)}^F)$. This contradicts the choice of $e' \in C \subseteq P \setminus (P^* \cup \text{Cl}(B_{\text{Bad}(i)}^F))$.

Suppose on the other hand that $\max \text{Block}(j) < \min \text{Block}(i)$. By Items 4 and 5 of Definition 20, this implies that $\text{Block}(i) \cup \text{Bad}(i) \subseteq \text{Good}(i) \subseteq \text{Bad}(j)$ and hence $e \in B_{\text{Block}(i)}^{U \setminus F} \cap \text{Cl}(B_{\text{Good}(i)}^F) \subseteq \text{Cl}(B_{\text{Bad}(j)}^F)$ and, using Item 5 of the definition of a critical tuple, $B_{\text{Bad}(i)}^F \cup P^* \subseteq B_{\text{Bad}(i)}^F \cup \text{Cl}(B_{\text{Good}(i)}^F) \subseteq \text{Cl}(B_{\text{Bad}(j)}^F)$ since, by Step 2(b)i, every element in P^* is in $\text{Cl}(B_{\text{Good}(i)}^F)$.

Since e' was the latest element in C added to P , by Item 6 of Proposition 4, $e' \in \text{Cl}(\{e\} \cup (C \cup B_{\text{Bad}(i)}^F \cup P^*) \setminus \{e'\})$. Since $e \in \text{Cl}(B_{\text{Bad}(j)}^F)$ and $B_{\text{Bad}(i)}^F \cup P^* \subseteq \text{Cl}(B_{\text{Bad}(j)}^F)$, we see that $e' \in \text{Cl}((C \cup B_{\text{Bad}(j)}^F) \setminus \{e'\})$. Therefore, e' did not satisfy the condition in Step 2(b)iA. This contradicts the fact that e' was added to P . \blacksquare

Theorem 24 *Given a critical tuple $(\text{Block}, \text{Good}, \text{Bad})$ as input, Algorithm 2 returns an independent set of elements $P \subseteq U \setminus F$ such that*

$$\text{OPT}(P) \geq \sum_{j \in \text{BLOCK}} 2^j \cdot \left(\text{uncov}(B_{\text{Bad}(j)}^F \cup B_{\text{Block}(j) \setminus \{j\}}^{U \setminus F}, B_j^{U \setminus F}) - \text{loss}(B_{\text{Good}(j)}^F, B_j^{U \setminus F}) \right).$$

Proof. Step 2(b)iA implies that P is always an independent set. Let $j \in \text{BLOCK}$ and, for every $i \in \text{Block}(j)$, let $S_i = B_i^{U \setminus F} \cap \text{Cl}(B_{\text{Good}(i)}^F)$.

We note that, by definition, for every $i \in \text{Block}(j)$, every element in S_i satisfies the condition in Step 2(b)i. Consequently, by Lemma 23, the Gap Algorithm processes the elements in $\bigcup_{i \in \text{Block}(j)} S_i$, as if it was the Simple Algorithm in the following setting: the input is a set $J = \text{Block}(j) \cup \text{Bad}(j)$ and the elements revealed are those of $S = B_{\text{Bad}(j)}^F \cup \bigcup_{i \in \text{Block}(j)} S_i$, which are revealed in an arbitrary order, except that the elements of $B_{\text{Bad}(j)}^F$ are revealed first. Thus, by Lemma 17, $\text{rank}(B_j^P)$ is at least $\text{uncov}(B_{\text{Bad}(j)}^F \cup \bigcup_{i \in \text{Block}(j) \setminus \{j\}} S_i, S_j)$.

Let $B_j^{U \setminus F}$, $R_1 = B_{\text{Bad}(j)}^F \cup \bigcup_{i \in \text{Block}(j) \setminus \{j\}} S_i$ and $R_2 = B_{\text{Bad}(j)}^F \cup B_{\text{Block}(j) \setminus \{j\}}^{U \setminus F}$. Then, $\text{rank}(B_j^P) \geq \text{uncov}(R_1, S_j)$. Since $R_1 \subseteq R_2$, by Observation 16, we have that $\text{uncov}(R_1, S_j) \geq \text{uncov}(R_2, S_j)$. By definition, $\text{uncov}(R_2, S_j) = \text{uncov}(R_2, B_j^{U \setminus F}) - (\text{rank}(R_2 \cup B_j^{U \setminus F}) - \text{rank}(R_2 \cup S_j))$. Finally, $S_j = B_j^{U \setminus F} \cap \text{Cl}(B_{\text{Good}(i)}^F)$, we see that $\text{rank}(R_2 \cup B_j^{U \setminus F}) - \text{rank}(R_2 \cup S_j)$ does not exceed $\text{rank}(B_j^{U \setminus F} \setminus \text{Cl}(B_{\text{Good}(i)}^F)) = \text{loss}(B_{\text{Good}(i)}^F, B_j^{U \setminus F})$. Therefore, $\text{uncov}(R_2, S_j) \geq \text{uncov}(R_2, B_j^{U \setminus F}) - \text{loss}(B_{\text{Good}(i)}^F, B_j^{U \setminus F})$. Thus, by Observation 8, the result follows. \blacksquare

5 Prediction

In this section, we prove that for a specific subset of the integers, which we denote by $Super$ and later, with constant probability, for every $K, K' \subseteq Super$, where $\max K' < \min K$ and $\min\{\text{rank}(B_i) \mid i \in K\} \geq \text{rank}(B_{\min K})^{\frac{2}{3}}$, and for every $j \in K$ we have that (i) $\text{uncov}(B_{K'}^F \cup B_{K' \setminus \{j\}}^{U \setminus F}, B_j^{U \setminus F})$ is approximately $\text{uncov}(B_{K' \cup K \setminus \{j\}}^F, B_j^F)$; and (ii) $\text{loss}(B_K^F, B_j^{U \setminus F})$ is bounded above by approximately $\text{uncov}(B_{K \setminus \{j\}}^F, B_j^F)$. This result enables us at Preprocessing stage of the Main Algorithm to chose whether to select elements using the Simple Algorithm or the Gap Algorithm, and to compute the input to the chosen algorithm.

In Subsection 5.2, we use the Talagrand inequality for the unquantified version of (i), in Subsection 5.1, we use Martingales and Azuma's inequality fur the unquantified version of (ii) and in Subsection 5.3, we define the set $Super$ and use the Union Bound together with the results in the previous sections to prove the main result of this section.

5.1 Upper Bounding loss

Theorem 25 *Let $K \subset \mathbb{Z}$, be finite and non-empty and $k \in K$ then,*

$$\text{prob}\left(\text{loss}(B_K^F, B_k^{U \setminus F}) \leq \text{uncov}(B_{K \setminus \{k\}}^F, B_k^F) + 4 \cdot \text{rank}(B_k)^{\frac{3}{4}}\right) > 1 - e^{-\text{rank}(B_k)^{\frac{1}{2}}}.$$

Proof. We fix $S = B_{K \setminus \{k\}}^F$ and let $m = \text{rank}(B_k)$. We initially let both H^F and $H^{U \setminus F}$ be empty sets. Then, we repeat the following $4m$ times: if there exists an element in $B_k \setminus (H^F \cup H^{U \setminus F})$ that is not in $\text{Cl}(S \cup H^F)$, then we pick such an element arbitrarily, if it is in F , then we add it to H^F and otherwise we add it to $H^{U \setminus F}$.

We observe that every time an element is added to H^F it is independent of $\text{Cl}(S \cup H^F)$ and hence it increases by one the quantity $\text{uncov}(S, H^F) = \text{rank}(S \cup H^F) - \text{rank}(S)$. Thus, if after $4m$ repetitions there are no elements in $B_k \setminus \text{Cl}(S \cup H^F)$, then the preceding quantity cannot be increased further by adding elements from $B_k \setminus (H^F \cup H^{U \setminus F})$ to H^F and therefore $\text{uncov}(S, H^F) = |H^F|$. Since, in this case every element in $B_k^F \setminus H^F$ is in $\text{Cl}(S \cup H^F)$ and $H^F \subseteq B_k^F$, we see that $\text{Cl}(S \cup B_k^F) = \text{Cl}(S \cup H^F)$. Therefore, $\text{rank}(S \cup B_k^F) = \text{rank}(S \cup H^F)$. Hence, by the definition of uncov , $\text{uncov}(S, B_k^F) = \text{uncov}(S, H^F) = |H^F|$.

We also observe that every time an element is added to $H^{U \setminus F}$ it may increase by one the quantity $\text{loss}(S \cup H^F, H^{U \setminus F}) = \text{rank}(H^{U \setminus F} \setminus \text{Cl}(S \cup H^F))$. We note that if after $4m$ repetitions there are no elements in $B_k \setminus \text{Cl}(S \cup H^F)$, then the preceding quantity cannot be increased further by adding elements from $B_k \setminus (H^F \cup H^{U \setminus F})$ to $H^{U \setminus F}$ and therefore $\text{loss}(S \cup H^F, H^{U \setminus F}) \leq |H^{U \setminus F}|$. Since, in this case every element in $B_k^{U \setminus F} \setminus H^{U \setminus F}$ is in $\text{Cl}(S \cup H^F)$ and $H^{U \setminus F} \subseteq B_k^{U \setminus F}$, we see that $\text{loss}(S \cup H^F, H^{U \setminus F}) = \text{loss}(S \cup H^F, B_k^{U \setminus F})$. We note that $S \cup B_k^F = B_K^F$ and we already proved $\text{Cl}(S \cup B_k^F) = \text{Cl}(S \cup H^F)$. Thus, $\text{loss}(B_K^F, B_k^{U \setminus F}) = \text{loss}(S \cup H^F, H^{U \setminus F}) \leq |H^{U \setminus F}|$.

Next we show that, $\||H^F| - |H^{U \setminus F}|\| \leq 4 \cdot m^{\frac{3}{4}}$, with probability at least $1 - e^{-2m^{\frac{1}{2}}}$, and afterwards

we show that, with probability at least $1 - e^{-\frac{m}{2}}$, after $4m$ repetitions, there are no elements in $B_k \setminus \text{Cl}(S \cup H^F)$. By the union bound, this implies the theorem.

We define the variables Z_i so that $Z_0 = 0$ and (i) $Z_i = Z_{i-1} - 1$ if in the i 'th repetition an element was added to H^F ; (ii) $Z_i = Z_{i-1} + 1$ if in the i 'th repetition an element was added to $H^{U \setminus F}$; and (iii) $Z_i = Z_{i-1}$ if nothing happened in the i 'th repetition.

We note that, for every $i > 0$, either $Z_i = Z_{i-1}$ or Z_i is distributed uniformly over $\{Z_{i-1} - 1, Z_{i-1} + 1\}$ and hence $E(Z_i | Z_{i-1}) = Z_{i-1}$, where $E()$ denotes the expected value. Consequently, we have a martingale. Thus, by Azuma's inequality, $Z_{4m} > 4 \cdot m^{\frac{3}{4}}$ with probability less than $e^{-2 \cdot m^{\frac{1}{2}}}$. Since, $Z_{4m} = |H^{U \setminus F}| - |H^F|$ we have proved the first inequality. We now proceed to the second.

We define the variables X_i so that $X_i = 1$ if in the i th repetition the element processed was in F and otherwise $X_i = 0$. By definition, for every $i \in [4m]$, if $Z_i = Z_{i-1} - 1$, then $X_i = 1$. If $|H^F| = m$ after $4m$ repetitions, then $\text{rank}(S \cup H^F) = \text{rank}(S) + \text{rank}(B_k)$, which can only happen if $B_k \subseteq \text{Cl}(S \cup H^F)$. This implies that $B_k \setminus \text{Cl}(S \cup H^F)$ is empty. So $B_k \setminus \text{Cl}(S \cup H^F)$ is not empty after $4m$ only if $\sum_{i=1}^{4m} X_i < m$. By Observation 1, for every $i \in [4m]$, X_i is independently distributed uniformly over $\{0, 1\}$. By the Chernoff inequality, with probability at least $1 - e^{-\frac{m}{2}}$, $\sum_{i=1}^{4m} X_i \geq m$. ■

5.2 Talagrand based concentrations

This subsection is very similar to one that appears in [8], we include it for the sake of completeness. The following definition is an adaptation of the Lipschitz condition to our setting.

Definition 26 [*Lipschitz*] Let $f : U \rightarrow \mathbb{N}$. If $|f(S_1) - f(S_2)| \leq 1$ for every $S_1, S_2 \subseteq U$ such that $|(S_1 \setminus S_2) \cup (S_2 \setminus S_1)| = 1$, then f is **Lipschitz**.

Definition 27 [*Definition 3, Section 7.7 of [1]*] Let $f : \mathbb{N} \rightarrow \mathbb{N}$. h is f -certifiable if whenever $h(x) \geq s$ there exists $I \subseteq \{1, \dots, n\}$ with $|I| \leq f(s)$ so that all $y \in \Omega$ that agree with x on the coordinates I have $h(y) \geq s$.

Observation 28 For every finite $K \subset \mathbb{Z}$, the rank function over subsets of B_K is Lipschitz and f -certifiable with $f(s) = \text{rank}(B_K)$, for all s .

Proof. The rank function is Lipschitz, by the definition of the rank function (Definition 3). By Item 2 of Proposition 4, for every $S \subseteq R \subseteq B_K$, we have that $\text{rank}(S) \leq \text{rank}(R) \leq \text{rank}(B_K)$. Thus, the rank function over subsets of B_K is f -certifiable with $f(s) = \text{rank}(B_K)$. ■

The succeeding theorem is a direct result of Theorem 7.7.1 from [1].

Theorem 29 If h is Lipschitz and f certifiable, then for x selected uniformly from Ω and all b, t , $\Pr[h(x) \leq b - t\sqrt{f(b)}] \cdot \Pr[h(x) \geq b] \leq e^{-t^2/4}$.

Lemma 30 Let $t \geq 2$, $j \in \mathbb{Z}$, $K, K' \subseteq \mathbb{Z}$, where $\min K' < \max K$ and $k \in K$ then, $\text{prob}\left(\left|\text{rank}(B_j^F) - \text{rank}(B_j^{U \setminus F})\right| \geq 2t\sqrt{\text{rank}(B_j)}\right) \leq e^{1.4 - \frac{t^2}{4}}$ and

$$\text{prob}\left(\left|\text{uncov}(B_{K'}^F \cup B_{K \setminus \{k\}}^{U \setminus F}, B_k^{U \setminus F}) - \text{uncov}(B_{K' \cup K \setminus \{k\}}^F, B_k^F)\right| \geq 4t\sqrt{\text{rank}(B)}\right) \leq e^{2.1 - \frac{t^2}{4}}.$$

Proof. Let $S \in \{B_{K' \cup K}^F, B_{K' \cup K \setminus \{k\}}^F, B_{K'}^F \cup B_K^{U \setminus F}, B_{K'}^F \cup B_{K \setminus \{k\}}^{U \setminus F}\}$. By Observation 28, the *rank* function is Lipschitz and *rank*-certifiable.

Clearly, since F and $U \setminus F$ are both distributed uniformly, with probability at least $\frac{1}{2}$, we have that $\text{rank}(S) \geq \text{med}(\text{rank}(S))$. Hence, taking $b = \text{med}(\text{rank}(S)) + t\sqrt{\text{rank}(B_{K' \cup K})}$, by Theorem 29, we get that $\text{rank}(S) - \text{med}(\text{rank}(S)) \geq t\sqrt{\text{rank}(B_{K' \cup K})}$, with probability at most $2e^{-\frac{t^2}{4}}$. In a similar manner, by taking $b = \text{med}(\text{rank}(S))$, we get that $\text{med}(\text{rank}(S)) - \text{rank}(S) \geq t\sqrt{\text{rank}(B_{K' \cup K})}$, with probability at most $2e^{-\frac{t^2}{4}}$. Thus, by the union bound, $|\text{rank}(S) - \text{med}(\text{rank}(S))| \geq t\sqrt{\text{rank}(B_{K' \cup K})}$, with probability at most $4e^{-\frac{t^2}{4}}$.

We note that, since F and $U \setminus F$ are identically distributed and $K \cap K^* = \emptyset$, we have that $\text{med}(\text{rank}(B_{K' \cup K}^F)) = \text{med}(\text{rank}(B_{K'}^F \cup B_K^{U \setminus F}))$ and $\text{med}(\text{rank}(B_{K' \cup K \setminus \{k\}}^F)) = \text{med}(\text{rank}(B_{K'}^F \cup B_{K \setminus \{k\}}^{U \setminus F}))$. Consequently, the second part of the result follows, by the union bound and the definition of *uncov* (Definition 15). The first part follows in a similar manner the preceding analysis. ■

5.3 Union bound

Definition 31 [*Super*] We define *Super* = $\left\{i \mid \text{rank}(B_i) > (2^{-i-4} \cdot \text{LOPT}(U))^{\frac{3}{4}}\right\}$.

When the following theorem is used later, the notations K' and K are replaced once with $\text{Bad}(i)$ and $\text{Block}(i)$, respectively, another time with the empty set and the input to the Simple Algorithm, respectively.

Theorem 32 If $\text{rank}(U) > 2^{19}$ then, with probability at least $\frac{1}{4}$, the following event holds: for every $i \in \text{Super}$, $K, K' \subseteq \{j \in \text{Super} \mid j \leq \log \text{LOPT}(U) - 2^7 \cdot \log \log \text{rank}(U)\}$, where $\min K' > \max K$ or $K' = \emptyset$, $\min\{\text{rank}(B_j) \mid j \in K\} \geq (2^{-5} \cdot \text{rank}(B_{\min K}))^{\frac{8}{9}}$, and every $k \in K$, the following hold:

1. $4 \cdot \text{rank}(F) > \text{rank}(U)$,
2. $|\text{rank}(B_i^F) - \text{rank}(B_i^{U \setminus F})| < 4 \cdot \text{rank}(B_i)^{\frac{2}{3}}$,
3. $|\text{uncov}(B_{K'}^F \cup B_{K \setminus \{k\}}^{U \setminus F}, B_k^{U \setminus F}) - \text{uncov}(B_{K' \cup K \setminus \{k\}}^F, B_k^F)| < 8 \cdot \text{rank}(B_{K' \cup K})^{\frac{3}{4}}$,
4. $\text{loss}(B_K^F, B_k^{U \setminus F}) \leq \text{uncov}(B_{K \setminus \{k\}}^F, B_k^F) + 8 \cdot \text{rank}(B_k)^{\frac{3}{4}}$.

Proof. Let C be a maximal independent set in U . By Observation 1 and the Chernoff bound, $\text{prob}\left(|F \cap C| \leq \frac{1}{4} \cdot \text{rank}(U)\right) \leq e^{-2^{-3} \cdot \text{rank}(U)} < \frac{1}{8}$, where the last inequality follows from $\text{rank}(U) > 2^{19}$. By the definition of *rank*, $|F \cap C| > \frac{1}{4} \cdot \text{rank}(U)$ implies Item 1.

Let $c = \log \text{LOPT}(U) - 2^7 \cdot \log \log \text{rank}(U)$, $K, K' \subseteq \text{Super}$, where $\min K' > \max K$ and $\min\{\text{rank}(B_i) \mid i \in K\} \geq \text{rank}(B_{\min K})^{\frac{8}{9}}$, $k \in K$ and $t = 2 \cdot \text{rank}(B_{K' \cup K})^{\frac{1}{4}} \geq 2$.

Consequently, by the union bound, Theorem 25 and Lemma 30, at least one of Items 3 and 4 does not hold for K', K and k , with probability at most $e^{2.1 - \frac{t^2}{4}} + e^{-\text{rank}(B_k)^{\frac{1}{2}}}$, which does not exceed $e^{2.1 - \text{rank}(B_{K' \cup K})^{\frac{1}{2}}} + e^{-\text{rank}(B_k)^{\frac{1}{2}}}$ which, in turn, is less than $e^{2.75 - (2^{-5} \cdot \text{rank}(B_{\min K}))^{\frac{4}{9}}}$, because

$k \in K$ and $2^{-5} \cdot \min\{\text{rank}(B_i) \mid i \in K\} \geq \text{rank}(B_{\min K})^{\frac{8}{9}}$. By the definition of *Super*, we see that $e^{2.75 - (2^{-5} \cdot \text{rank}(B_{\min K}))^{\frac{4}{9}}} \leq e^{2.75 - (2^{\log \text{LOPT}(U) - \min K - 12})^{\frac{1}{3}}}$.

Let $z \in \text{Super}$. Since $z \in K \subseteq \text{Super}$, and according to the definition of K and K' , for every possible value of z there are at most $|K| \leq c - z$ possible choices of k and $(c - z)2^{c-z}$ choices of K and K' . Consequently, by the union bound, the probability that at least one of Items 3 and 4 does not hold for some K', K and k , is at most $\sum_{z \in \text{Super}} (c - z)2^{2c-z} e^{2.75 - (2^{\log \text{LOPT}(U) - z - 12})^{\frac{1}{3}}}$. Taking $y = c - z$, the previous value is bounded above by $\sum_{y \in \mathbb{N}} y^2 2^y e^{2.75 - (2^{\log \text{LOPT}(U) - c + y - 12})^{\frac{1}{3}}} < \frac{1}{8}$, since $c = \log \text{LOPT}(U) - 2^7 \cdot \log \log \text{rank}(U)$ and $\text{rank}(U) > 2^{19}$.

Let $k' \in \text{Super}$, and $t' = 2 \cdot \text{rank}(B_{k'})^{\frac{1}{6}} \geq 2$. By Lemma 30, Item 2 does not hold, with probability at most $e^{1.4 - \text{rank}(B_{k'})^{\frac{1}{3}}}$. Since $\text{rank}(B_{k'}) \geq (2^{-k'-4} \cdot \text{LOPT}(U))^{\frac{3}{4}}$, by the definition of *Super*, we see that $e^{1.4 - \text{rank}(B_{k'})^{\frac{1}{3}}} \leq e^{1.4 - (2^{\log \text{LOPT}(U) - k' - 4})^{\frac{1}{4}}}$. Therefore, by the union bound, the probability that Item 2 does not hold for any $k' \in \text{Super}$, is at most $\sum_{k' \in \text{Super}} e^{1.4 - (2^{\log \text{LOPT}(U) - k' - 4})^{\frac{1}{4}}}$. Taking $y' = \max \text{Super} - k'$, the previous value is bounded above by $\sum_{y' \in \mathbb{N}} e^{1.4 - (2^{\log \text{LOPT}(U) - \max \text{Super} + y' - 4})^{\frac{1}{4}}} < \frac{1}{2}$, where the last inequality follows from Assumption 14. Consequently, by the union bound the result follows. \blacksquare

6 Structural Theorem

In this section we assume that all the elements of F have been revealed and hence F is treated as fixed.

Definition 33 [M_H] For every $K \subseteq H \subset \mathbb{Z}$ be let $M_H(K) = \{i \in H \mid i > \max K\}$. We omit the subscript when clear from context.

Definition 34 [*manageable set*] A set of integers K is *manageable* if, for every $j \in K$, we have that $\text{rank}(B_j^F) \geq (\frac{1}{2} \cdot \sum_{i \in K} \text{rank}(B_i^F))^{\frac{8}{9}}$.

Definition 35 [*Critical family*] Let $L \subset \mathbb{Z}$, let \mathcal{H} be a family of subsets of L , and let $H = \bigcup_{H' \in \mathcal{H}} H'$, then \mathcal{H} is a *critical family* for L if the following hold:

1. $\text{LOPT}(B_H^F) \geq \frac{1}{18} \cdot \text{LOPT}(B_L^F)$,
2. for every pair H_1 and H_2 of distinct sets in \mathcal{H} , either $\max H_1 < \min H_2$ or $\max H_2 < \min H_1$,
3. for every $H' \in \mathcal{H}$ and $j \in H'$, $\text{uncov}(B_{M_H(H')}^F, B_j^F) \geq \frac{15}{16} \cdot \text{rank}(B_j^F)$,
4. every set in \mathcal{H} is manageable and
5. for every $i \in H$, $2 \cdot \text{rank}(B_i^F) \geq \text{rank}(B_{M_H(\{i\}) \cup \{i\}}^F)$.

Lemma 36 Let $L \subset \mathbb{Z}$. If $\text{rank}(F) > 2^{16}$, then there exists a critical family \mathcal{H} for L of cardinality at most $8 \cdot \log \log \text{rank}(F)$.

Proof. Define, $w : L \rightarrow \mathbb{N}$ as follows: for every $i \in L$, $w(i) = \text{rank}(B_i^F)$. Let $m = \sum_{j \in L} w(j) \cdot 2^j = \text{LOPT}(B_L^F)$. Let s_1 be maximum so that $w(s_1) > 0$, and inductively define, s_{i+1} to be the maximum integer such that $w(s_{i+1}) \geq 2 \cdot w(s_i)$. Let k be the maximum integer such that s_k is defined. It follows that $s_1 > s_2 > \dots > s_k$. For every $i \in [k]$, the sum of $w(j) \cdot 2^j$ over all $j \in L$, where $s_{i+1} < j < s_i$ when $i < k$, is at most $2 \cdot w(s_i) \cdot 2^{s_i}$. Thus, $m \leq 3 \cdot \sum_{i=1}^k w(s_i) \cdot 2^{s_i}$ and so $\sum_{i=1}^k w(s_i) \cdot 2^{s_i} \geq \frac{m}{3}$.

Let $R_j = \{s_i \mid i = j \pmod{5}\}$. By the pigeon hole principle, there exists $q \in [6]$ such that the sum of $\sum_{i \in R_q} 2^i \cdot w(s_i) \geq \frac{m}{18}$. Let $\ell_1 = \max R_q$ and r_1 be the minimum member of R_q such that $w(\ell_1) \geq w(r_1)^{\frac{8}{9}}$ and set $H_1 = [\ell_1, r_1] \cap R_q$. Now, inductively, for every $i > 1$, let ℓ_i be the maximum member of R_q that is smaller than r_{i-1} , and r_i be the minimum member of R_q such that $w(\ell_i) \geq w(r_i)^{\frac{8}{9}}$ and $H_i = [\ell_i, r_i] \cap R_q$. Let g be the maximum integer for which H_g is defined, $\mathcal{H} = \{H_i\}_{i \in [g]}$ and $H = \bigcup_{i=1}^g H_i$. We note that, by construction, $r_g = \min R_q$ and $H = R_q$.

We next bound above g . If $g \leq 4$, then $g \leq 8 \cdot \log \log \text{rank}(F)$, since $\text{rank}(F) \geq 2^{16}$. Suppose that $g > 4$. By construction, $w(\ell_i) < w(\ell_{i+1})^{\frac{8}{9}}$, for every $i \in [g-1]$, and hence $\text{rank}(F) \geq w(\ell_g) > w(\ell_2)^{\left(\frac{9}{8}\right)^{g-2}}$. Since, by construction, $w(\ell_2) \geq 32 \cdot w(\ell_1) \geq 32$, the preceding inequality and the fact that $\text{rank}(F) > 2^{16}$ imply that $g \leq 8 \cdot \log \log \text{rank}(F)$.

By construction, **Items 2 and 5 of Definition 35** holds. Also $\sum_{p \in H} 2^p \cdot w(s_p) = \sum_{p \in R_q} 2^p \cdot w(s_p) \geq \frac{m}{18}$. so **Item 1 of Definition 35** holds. Let $H' \in \mathcal{H}$ and $j \in H'$. By construction, $\text{rank}(B_j^F) = w(j) \geq w(\min H')^{\frac{8}{9}} \geq \left(\frac{1}{2} \cdot \sum_{p \in H'} w(p)\right)^{\frac{8}{9}} = \left(\frac{1}{2} \cdot \sum_{p \in H'} B_p^F\right)^{\frac{8}{9}}$. Thus, **Item 4 of Definition 35** holds.

By the definition of uncov (Definition 15), $\text{uncov}\left(\text{rank}\left(B_{M_H(H')}^F\right), \text{rank}\left(B_j^F\right)\right) = \text{rank}\left(B_{M_H(H') \cup \{j\}}^F\right) - \text{rank}\left(B_{M_H(H')}^F\right) \geq \text{rank}\left(B_j^F\right) - \text{rank}\left(B_{M_H(H')}^F\right)$. This is bounded below by $\frac{15}{16} \cdot \text{rank}\left(B_j^F\right)$ because, by construction, for every $j \in H'$ we have $\text{rank}\left(B_j^F\right) \geq 16 \cdot \sum_{i \in M_H(H')} \text{rank}\left(B_i^F\right) \geq 16 \cdot \text{rank}\left(B_{M_H(H')}^F\right)$. Consequently, **Item 3 of Definition 35** holds. ■

Definition 37 [useful] Let $K^* \subseteq K \subseteq H \subset \mathbb{Z}$. If the following hold:

1. $\text{LOPT}\left(B_{K^*}^F\right) > \frac{1}{32} \cdot \text{LOPT}\left(B_K^F\right)$ and
2. $\sum_{j \in K^*} 2^j \cdot \left(\text{uncov}\left(B_{M_H(K^*) \cup K^* \setminus \{j\}}^F, B_j^F\right) - \text{uncov}\left(B_{M_H(K) \cup K \setminus \{j\}}^F, B_j^F\right)\right) \geq \frac{\text{LOPT}\left(B_K^F\right)}{2^{11} \cdot \log \log \text{rank}(F)}$.

then K^* is **useful** for K in H and K is **useful** in H .

Definition 38 [splittable] $K \subset \mathbb{Z}$ is **splittable** if it has a bipartition $\{K_1, K_2\}$ such that

1. $\min K_1 > \max K_2$ and
2. $\text{LOPT}\left(B_{K_i}^F\right) > \frac{1}{32} \cdot \text{LOPT}\left(B_K^F\right)$, for every $i = 1, 2$.

Definition 39 [negligible] A subset K of a set $H \subset \mathbb{Z}$ is **negligible** for H if $\text{LOPT}\left(B_K^F\right) < \frac{\text{LOPT}\left(B_H^F\right)}{8 \cdot \log \text{rank}(F)}$. When H is clear from context, we just say K is negligible.

Definition 40 [*burnt*] A subset K of a set $H \subset \mathbb{Z}$ is **burnt** for H if

$$\sum_{j \in K} 2^j \cdot \text{uncov} \left(B_{M_H(K) \cup K \setminus \{j\}}^F, B_j^F \right) > \frac{3}{4} \cdot \text{LOPT} \left(B_K^F \right).$$

Definition 41 [*critical-tree*] Let \mathcal{H} be a family of subsets of \mathbb{Z} and $H = \bigcup_{H' \in \mathcal{H}} H'$, a *critical-tree* for \mathcal{H} is a rooted tree whose vertices are subsets of H and that satisfies the following:

1. the root of the tree is H ,
2. the children of the root are the sets of \mathcal{H} ,
3. every leaf is either negligible, useful or burnt, and
4. every internal vertex K , except possibly the root, is splittable and neither useful, negligible nor burnt; moreover it has two children that form a bipartition of K as described in the definition of splittable.

Lemma 42 Suppose that \mathcal{T} is a critical-tree for a critical family \mathcal{H} . If $\text{rank}(F) > 2^{16}$, then the depth of \mathcal{T} does not exceed $2^7 \cdot \log \log \text{rank}(F)$.

Proof. Let K be a parent of a leaf in \mathcal{T} and d be the depth of K . We assume that K is not the root or one of its children, since otherwise the result follows immediately. By the definition of a critical-tree, each ancestor K^* of K , except for the root, is splittable and hence, by the definition of splittable, $\text{LOPT} \left(B_{K^*}^F \right) \leq \left(\frac{31}{32} \right)^{d-2} \cdot \text{LOPT} \left(B_H^F \right)$. Since K is not negligible, by definition, $\frac{\text{LOPT}(B_H^F)}{8 \cdot \log \text{rank}(F)} \leq \text{LOPT} \left(B_K^F \right)$. Therefore, $\frac{\text{LOPT}(B_H^F)}{8 \cdot \log \text{rank}(F)} \leq \left(\frac{31}{32} \right)^{d-2} \cdot \text{LOPT} \left(B_H^F \right)$. Consequently, since $\text{rank}(F) > 2^{16}$, we have that $d \leq 2^7 \cdot \log \log \text{rank}(F)$, which in turn implies the result. \blacksquare

Lemma 43 Let $L \subset \mathbb{Z}$ and \mathcal{H} be a critical family for L . Then, there exists a critical-tree \mathcal{T} for \mathcal{H} .

Proof. We construct \mathcal{T} as follows: we let $H = \bigcup_{H' \in \mathcal{H}} H'$ and set the root to be H and its children to be \mathcal{H} . Then, as long as there is a leaf K in the tree that is splittable but neither useful, negligible, nor burnt, we add two children K_1 and K_2 to K , where $\{K_1, K_2\}$ form a bipartition of K , as in the definition of splittable. If there are no such leaves, we stop.

By construction, every vertex K in \mathcal{T} is a subset of H and \mathcal{T} satisfies Items 1, 2 and 4 of the definition of a critical-tree. Suppose that every $K \subseteq H$, except for possibly the root, is at least one of the following: useful, negligible, burnt or splittable. This implies that \mathcal{T} also satisfies Item 3 of the definition of a critical-tree. Thus, \mathcal{T} is a critical-tree for \mathcal{H} .

We prove next that indeed, every $K \subseteq H$ is at least one of: useful, negligible, burnt or splittable. Fix $K \subseteq H$ and assume that K is neither burnt, negligible nor splittable. By the definition of splittable, there exists $\gamma \in K$ such that, $\text{LOPT} \left(B_\gamma^F \right) \geq \frac{15}{16} \cdot \text{LOPT} \left(B_K^F \right)$. Hence, the set $K^* = \{\gamma\}$ satisfies Item 1 of the definition of useful. We show next that K^* also satisfies Item 2 of the definition of useful, and therefore is useful.

Since K is not burnt, $2^\gamma \cdot \text{uncov} \left(B_{M(K) \cup K \setminus \{\gamma\}}^F, B_\gamma^F \right) \leq \frac{3}{4} \cdot \text{LOPT} \left(B_K^F \right)$, hence by Item 2 of the definition of useful, it is sufficient to show that $2^\gamma \cdot \text{uncov} \left(B_{M_H(\{\gamma\})}^F, B_\gamma^F \right) \geq \frac{7}{8} \cdot \text{LOPT} \left(B_K^F \right)$. Since

\mathcal{H} is a critical family, by Item 3 of Definition 35 and Observation 16, $\text{uncov}(B_{M_H(\{\gamma\})}^F, B_j^F) \geq \frac{15}{16} \cdot \text{rank}(B_\gamma^F)$. Consequently, because $\text{LOPT}(B_\gamma^F) \geq \frac{15}{16} \cdot \text{LOPT}(B_K^F)$, we have that $2^\gamma \cdot \text{uncov}(B_{M_H(\{\gamma\})}^F, B_\gamma^F) \geq \frac{7}{8} \cdot \text{LOPT}(B_K^F)$. \blacksquare

Theorem 44 *Let L be a set of at most $\frac{4}{3} \cdot \log \text{rank}(F) + 3$ integers. If $\text{rank}(F) > 2^{16}$ and $\sum_{j \in K} 2^j \cdot \text{uncov}(B_{K \setminus \{j\}}^F, B_j^F) \leq \frac{\text{LOPT}(B_L^F)}{2^{11 \cdot \log \log \text{rank}(F)}}$, for every manageable $K \subseteq L$, then there exists a critical tuple (Block, Good, Bad) such that, for every $i \in \mathbb{Z}$,*

1. $\text{Good}(i)$, $\text{Bad}(i)$ and $\text{Block}(i)$ are subsets of L ,
2. $\text{rank}(B_j^F) > \left(\frac{1}{8} \cdot \sum_{\ell \in \text{Bad}(i) \cup \text{Block}(i)} \text{rank}(B_\ell^F)\right)^{\frac{8}{9}}$, for every $j \in \text{Block}(i)$, and
3. $\sum_{j \in \text{BLOCK}} 2^j \left(\text{uncov}(B_{\text{Bad}(j) \cup \text{Block}(j) \setminus \{j\}}^F, B_j^F) - \text{uncov}(B_{\text{Good}(j) \setminus \{j\}}^F, B_j^F) \right) \geq \frac{\text{LOPT}(B_L^F)}{2^{10 \cdot \log \log \text{rank}(F)}}$.

Proof. By Lemma 36, there exists a critical family \mathcal{H} for L . Let $H = \bigcup_{H' \in \mathcal{H}} H'$. By Lemma 43, there exists a critical-tree \mathcal{T} for \mathcal{H} . Let \mathcal{Q} be the family containing all the leaves in \mathcal{T} . Let $\mathcal{Q}_{\text{useful}}$ be the family of all the sets in \mathcal{Q} that are useful in H . Define $\mathcal{Q}_{\text{burnt}}$ and $\mathcal{Q}_{\text{negligible}}$ in the same manner.

We construct a tuple (Block, Good, Bad) as follows: for each $K \in \mathcal{Q}_{\text{useful}}$, we pick a subset $K^* \subset K$, that is useful for K , arbitrarily; then, for each $i \in K^*$, we let $\text{Block}(i) = K^*$, $\text{Bad}(i) = M_H(K)$ and $\text{Good}(i) = K \cup M_H(K)$. Finally, for every i such that $\text{Block}(i)$ was not defined previously, we let $\text{Block}(i) = \text{Good}(i) = \text{Bad}(i) = \emptyset$.

By construction, (Block, Good, Bad) satisfies **Items 1 and 2** of the theorem and Items 1, 3, 4 and 6 of the definition of a critical tuple (Definition 20). By the definition of a splittable set and the definition of a critical-tree, for every pair of leaves K, K' of \mathcal{T} , either $\max K < \min K'$ or $\min K > \max K'$ and hence also $K \cap K' = \emptyset$. Thus, by construction, (Block, Good, Bad) also satisfies Items 2 and 5 of the definition of a critical tuple. Consequently, (Block, Good, Bad) is a critical tuple.

By the construction of (Block, Good, Bad) and the definition of useful, to prove Item 3 it is sufficient to show that $\text{LOPT}\left(\bigcup_{K \in \mathcal{Q}_{\text{useful}}} B_K^F\right) \geq \frac{1}{2} \cdot \text{LOPT}(B_H^F)$ since, by Item 1 of Definition 35, this implies that $\text{LOPT}\left(\bigcup_{K \in \mathcal{Q}_{\text{useful}}} B_K^F\right) \geq \frac{1}{36} \cdot \text{LOPT}(B_L^F)$. By Item 3 of Observation 10, $\text{LOPT}\left(\bigcup_{K \in \mathcal{Q}_{\text{useful}}} B_K^F\right)$ is at least $\text{LOPT}\left(\bigcup_{K \in \mathcal{Q}} B_K^F\right) - \text{LOPT}\left(\bigcup_{K \in \mathcal{Q}_{\text{burnt}}} B_K^F\right) - \text{LOPT}\left(\bigcup_{K \in \mathcal{Q}_{\text{negligible}}} B_K^F\right)$. To complete the proof, we bound each term in the preceding expression.

By the definition of a critical-tree, $\bigcup_{K \in \mathcal{Q}} K = H$. Hence, $\text{LOPT}\left(\bigcup_{K \in \mathcal{Q}} B_K^F\right) = \text{LOPT}(B_H^F)$. Since the sets in $\mathcal{Q}_{\text{negligible}}$ are subsets of H , pairwise disjoint and not-empty, we see that $|\mathcal{Q}| \leq |H|$. Thus, by the definition of negligible, $\text{LOPT}\left(\bigcup_{K \in \mathcal{Q}_{\text{negligible}}} B_K^F\right) \leq |H| \cdot \frac{\text{LOPT}(B_H^F)}{8 \cdot \log \text{rank}(F)}$. As $H \subseteq L$, $|H| \leq |L| \leq \frac{4}{3} \cdot \log \text{rank}(F) + 3$. Consequently, $\text{LOPT}\left(\bigcup_{K \in \mathcal{Q}_{\text{negligible}}} B_K^F\right) \leq \frac{1}{4} \cdot \text{LOPT}(B_H^F)$. We next bound $\text{LOPT}\left(\bigcup_{K \in \mathcal{Q}_{\text{burnt}}} B_K^F\right)$.

By the definition of burnt, $\text{LOPT}\left(\bigcup_{K \in \mathcal{Q}_{\text{burnt}}} B_K^F\right) \leq \frac{4}{3} \cdot \sum_{K \in \mathcal{Q}} \sum_{j \in K} 2^j \cdot \text{uncov}(B_{M(K) \cup K \setminus \{j\}}^F, B_j^F)$. This in turn is bounded above by the sum of:

- (a) $\sum_{H' \in \mathcal{H}} \sum_{j \in H'} 2^j \cdot \text{uncov} \left(B_{M(H') \cup H' \setminus \{j\}}^F, B_j^F \right)$ and
- (b) sum over every internal non-root vertex K , with children K_1 and K_2 , of $\sum_{\ell=1}^2 \sum_{j \in K_\ell} 2^j \cdot \text{uncov} \left(B_{M(K_\ell) \cup K_\ell \setminus \{j\}}^F, B_j^F \right) - \sum_{j \in K} 2^j \cdot \text{uncov} \left(B_{M(K) \cup K \setminus \{j\}}^F, B_j^F \right)$.

We note that sum (b) is the additional uncov measure because of the difference between the uncov of the children and their parent.

By construction, every $H' \in \mathcal{H}$ is manageable and therefore, by assumption, $\sum_{j \in H'} 2^j \cdot \text{uncov} \left(B_{M(H') \cup H' \setminus \{j\}}^F, B_j^F \right) \leq \sum_{j \in H'} 2^j \cdot \text{uncov} \left(B_{H' \setminus \{j\}}^F, B_j^F \right) \leq \frac{\text{LOPT}(B_L^F)}{2^{11} \cdot \log \log \text{rank}(F)}$, where the first inequality follows from Observation 16. Thus, the value of (a) is bounded above by $\frac{|\mathcal{H}| \cdot \text{LOPT}(B_L^F)}{2^{11} \cdot \log \log \text{rank}(F)}$. Because $|\mathcal{H}| \leq 8 \cdot \log \log \text{rank}(F)$, by Lemma 36, we see that $\frac{|\mathcal{H}| \cdot \text{LOPT}(B_L^F)}{2^{11} \cdot \log \log \text{rank}(F)} \leq 2^{-8} \cdot \text{LOPT}(B_L^F)$. This implies that the value of (a) is bounded above by $2^{-3} \cdot \text{LOPT}(B_H^F)$, by Item 1 of Definition 35.

By construction, every internal non-root vertex K of \mathcal{T} is splittable and not useful. Therefore its children do not satisfy Item 2 of the definition of useful. Hence, the value of (b) is bounded above by the sum of $\frac{2 \cdot \text{LOPT}(B_K^F)}{2^{11} \cdot \log \log \text{rank}(F)}$ over every internal non-root vertex K of \mathcal{T} . The sum of $\text{LOPT}(B_K^F)$ over all such vertices at any given depth is at most $\text{LOPT}(B_H^F)$. So, by Lemma 42, the value of (b) does not exceed $\frac{2 \cdot 2^7 \cdot \log \log \text{rank}(F) \cdot \text{LOPT}(B_H^F)}{2^{11} \cdot \log \log \text{rank}(F)}$. Consequently, (b) is bounded above by $\frac{1}{8} \cdot \text{LOPT}(B_H^F)$ and the result follows. \blacksquare

7 Main Result

The main result in this section is Theorem 48, which states that the Main Algorithm indeed has the claimed competitive-ratio. The proof of the theorem provides the details of how the Main Algorithm works and utilizes Theorems 18, 24, 32 and 44.

One of the crucial details of the proof is that the Main Algorithm only involves a subset of the buckets. Specifically, those that belong to the set *Valuable*, defined as follows:

Definition 45 [*Valuable*, L , L'] We define $\text{Valuable} = \left\{ j \mid \text{rank}(B_j^F) > (2^{-j-1} \cdot \text{LOPT}(F))^{\frac{3}{4}} \right\}$, $L = \{i \in \text{Valuable} \mid i < \log \text{LOPT}(F) - 2^9 \cdot \log \log \text{rank}(F)\}$ and $L' = \text{Valuable} \setminus L$.

The importance of *Valuable*, L and L' , as implied by Lemma 46 below, is that Item 2 of Theorem 32 applies to every bucket in L' , and both Theorem 32 and Theorem 44 apply to the relevant subsets of L . The next result, Lemma 47, is required in order to bound the influence of the deviation in Theorem 32.

Lemma 46 *If the event of Theorem 32 holds, then*

1. $|L| \leq \frac{4}{3} \cdot \log \text{rank}(F) + 3$,
2. for every $j \in \text{Super}$, $3 \cdot \text{rank}(B_j^{U \setminus F}) \geq \text{rank}(B_j^F) \geq \frac{1}{4} \cdot \text{rank}(B_j)$,
3. if $\text{LOPT}(B_{L'}^F) < \frac{1}{12} \cdot \text{LOPT}(U)$, then $\text{LOPT}(B_L^F) \geq \frac{1}{12} \cdot \text{LOPT}(U)$ and

4. Valuable \subseteq Super.

Proof. We first prove Item 1. Using Definition 9, $\max L \leq \log \text{LOPT}(F)$. By Definition 45, $(2^{-\min L-1} \cdot \text{LOPT}(F))^{\frac{3}{4}} \leq \text{rank}(B_{\min L}^F) \leq \text{rank}(F)$ and hence $\min L \geq \log \text{LOPT}(F) - \frac{4}{3} \cdot \log \text{rank}(F) - 1$. Since L contains only integers, **Item 1** follows.

Let $j \in \text{Super}$. By the definition of *Super* (Definition 31), $\text{rank}(B_j) \geq (2^{-j-4} \cdot \text{LOPT}(U))^{\frac{3}{4}}$. Thus, as $j < \log \text{LOPT}(U) - 19$, we see that $\text{rank}(B_j) > 2^9$. This implies that $4 \cdot \text{rank}(B_j)^{\frac{2}{3}} \leq \frac{1}{2} \cdot \text{rank}(B_j)$. By Item 2 of Theorem 32, $|\text{rank}(B_j^{U \setminus F}) - \text{rank}(B_j^F)| \leq 4 \cdot \text{rank}(B_j)^{\frac{2}{3}}$. By Item 4 of Proposition 4, we also have $\text{rank}(B_j^{U \setminus F}) + \text{rank}(B_j^F) \geq \text{rank}(B_j)$. The preceding three inequalities imply that **Item 2** holds.

Suppose first that $\text{LOPT}(B_{\text{Valuable}}^F) \geq \frac{1}{6} \cdot \text{LOPT}(U)$. Then, **Item 3** holds and $\text{Valuable} \neq \emptyset$. Since $\text{LOPT}(F) \geq \text{LOPT}(B_{\text{Valuable}}^F)$, using Definition 45, for every $i \in \text{Valuable}$, $\text{rank}(B_i) \geq \text{rank}(B_i^F) > (2^{-i-4} \cdot \text{LOPT}(U))^{\frac{3}{4}}$. Hence, by Definition 31, **Item 4** follows.

Finally we prove that $\text{LOPT}(B_{\text{Valuable}}^F) \geq \frac{1}{4} \cdot \text{LOPT}(U)$. We do so by first proving that $\text{LOPT}(F) \geq \frac{2}{9} \cdot \text{LOPT}(U)$ and then that $\text{LOPT}(B_{\text{Valuable}}^F) \geq \frac{3}{4} \cdot \text{LOPT}(F)$.

Let J^* be the set of all integers i such that $1 \leq \text{rank}(B_i) \leq (2^{-i-4} \cdot \text{LOPT}(U))^{\frac{3}{4}}$. Thus, $\text{LOPT}(B_{J^*}) \leq \sum_{i \in J^*} 2^i \cdot (2^{-i-4} \cdot \text{LOPT}(U))^{\frac{3}{4}} = \sum_{i \in J^*} (2^{\frac{i}{3}-4} \cdot \text{LOPT}(U))^{\frac{3}{4}}$. This is less than $\sum_{\ell \geq 0} \left(2^{\frac{\max J^* - \ell}{3} - 4} \cdot \text{LOPT}(U)\right)^{\frac{3}{4}} \leq \text{LOPT}(U) \cdot \sum_{\ell > 0} 2^{-\frac{\ell}{4} - 4.75 - 2.75} < \frac{1}{9} \cdot \text{LOPT}(U)$, because $\max J^* < \log \text{LOPT}(U) - 19$, by Assumption 14. Now, since $\text{Super} \cup J^*$ contains the indices of all non-empty buckets, $\text{LOPT}(B_{\text{Super}}) \geq \text{LOPT}(U) - \text{LOPT}(B_{J^*}) > \frac{8}{9} \cdot \text{LOPT}(U)$. Thus, by Item 2 and Definition 9, $\text{LOPT}(F) \geq \text{LOPT}(B_{\text{Super}}^F) \geq \frac{1}{4} \cdot \text{LOPT}(B_{\text{Super}}) \geq \frac{2}{9} \cdot \text{LOPT}(U)$.

Let J be the set of all integers i such that $1 \leq \text{rank}(B_i^F) \leq (2^{-i-1} \cdot \text{LOPT}(F))^{\frac{3}{4}}$. Thus, $\text{LOPT}(B_J^F) \leq \sum_{i \in J} 2^i \cdot (2^{-i-1} \cdot \text{LOPT}(F))^{\frac{3}{4}} = \sum_{i \in J} (2^{\frac{i}{3}-1} \cdot \text{LOPT}(F))^{\frac{3}{4}}$. This is less than $\sum_{\ell \in J} \left(2^{\frac{\max J - \ell}{3} - 1} \cdot \text{LOPT}(F)\right)^{\frac{3}{4}} = \text{LOPT}(F) \cdot \sum_{\ell > 0} 2^{-\frac{\ell}{4} - 4 - 0.75} < \frac{1}{4} \cdot \text{LOPT}(F)$, because $\max J < \log \text{LOPT}(F) - 19$, by Assumption 14. Consequently, since $\text{Valuable} \cup J$ contains all the indices of non-empty buckets, $\text{LOPT}(B_{\text{Valuable}}^F) \geq \text{LOPT}(F) - \text{LOPT}(B_J^F) > \frac{3}{4} \cdot \text{LOPT}(F)$. ■

Lemma 47 *Suppose that $\text{rank}(F) > 2^{16}$. Let $K \subseteq L$ and $(\text{Block}, \text{Good}, \text{Bad})$ be a critical tuple. If Item 2 of Lemma 46 holds, K is manageable and $(\text{Block}, \text{Good}, \text{Bad})$ satisfies Items 1 and 2 of Theorem 44 then,*

1. $8 \cdot \text{rank}(B_K)^{\frac{3}{4}} \cdot \sum_{i \in K} 2^i < \frac{\text{LOPT}(F)}{2^{30} \cdot \log \log \text{rank}(F)}$.
2. $2^4 \cdot \sum_{i \in \text{BLOCK}} 2^i \cdot \text{rank}(B_{\text{Bad}(i) \cup \text{Block}(i)})^{\frac{3}{4}} < \frac{\text{LOPT}(F)}{2^{30} \cdot \log \log \text{rank}(F)}$.

Proof. Let $c^{-1} = \min\{\text{rank}(B_i^F) \mid i \in K \cup L\}^{\frac{1}{8}}$. By the properties of Matroids, Definition 34

and Item 2 of Lemma 46,

$$\begin{aligned}
8 \cdot \text{rank}(B_K)^{\frac{3}{4}} \cdot \sum_{i \in K} 2^i &\leq 8 \cdot \sum_{i \in K} 2^i \cdot \left(\sum_{j \in K} \text{rank}(B_j) \right)^{\frac{3}{4}} \\
&\leq 8 \cdot \sum_{i \in K} 2^i \cdot \left(4 \cdot \sum_{j \in K} \text{rank}(B_j^F) \right)^{\frac{3}{4}} \\
&< 2^9 \cdot \sum_{i \in K} 2^i \cdot \text{rank}(B_i^F)^{\frac{7}{8}} \\
&\leq 2^9 \cdot \min\{\text{rank}(B_i^F) \mid i \in K\}^{-\frac{1}{8}} \cdot \sum_{i \in K} 2^i \cdot \text{rank}(B_i^F) \\
&\leq 2^9 \cdot c \cdot \text{LOPT}(F)
\end{aligned} \tag{1}$$

By the properties of Matroids, Item 2 of Lemma 46, and Items 1 and 2 of Theorem 44,

$$\begin{aligned}
2^4 \cdot \sum_{i \in \text{BLOCK}} 2^i \cdot \text{rank}(B_{\text{Bad}(i) \cup \text{Block}(i)})^{\frac{3}{4}} &\leq 2^4 \cdot \sum_{i \in \text{BLOCK}} 2^i \cdot \left(4 \cdot \sum_{j \in \text{Bad}(i) \cup \text{Block}(i)} \text{rank}(B_j^F) \right)^{\frac{3}{4}} \\
&< 2^9 \cdot \sum_{i \in \text{BLOCK}} 2^i \cdot \text{rank}(B_i^F)^{\frac{7}{8}} \\
&\leq 2^9 \cdot c \cdot \text{LOPT}(F).
\end{aligned} \tag{2}$$

By Definition 45,

$$c^{-1} \geq \left(2^{-\log \text{LOPT}(F) + 2^9 \cdot \log \log \text{rank}(F) - 1} \cdot \text{LOPT}(F) \right)^{\frac{3}{32}} > \frac{1}{2} \cdot \log \text{rank}(F)^{30} \cdot \log \log \text{rank}(F),$$

so, $c^{-1} > 2^{36} \cdot \log \log \text{rank}(F)$, because $\text{rank}(F) > 2^{16}$. Thus, by (1) and (2), the result follows. ■

The following theorem is the main result of this paper.

Theorem 48 *The Main Algorithm is Order-Oblivious, Known-Cardinality and has returns and, with constant probability, returns an independent set of elements of value $\Omega\left(\frac{\text{OPT}(U)}{\log \log \text{rank}(U)}\right)$.*

Proof. We note that the Main Algorithm is Known-Cardinality, since the computation in Gathering stage is independent of the matroid elements and the computation in the Preprocessing and the Selection stages uses only elements of the matroid that have already been revealed. We also note that the Main Algorithm is Order-Oblivious, because by construction, and following Definition 13, the analysis depends on the elements in the sets F and $U \setminus F$ but not on their order.

By Assumption 14, the properties of Matroids and Observation 8, it follows that $\text{rank}(U) > 2^{19}$. Thus, the event in Theorem 32 holds with probability at least $\frac{1}{4}$. So, by the definition of competitive-ratio, it is sufficient to prove the result assuming the event in Theorem 32 holds. We proceed on this assumption. We note that this means that the conditions needed for Lemma 46 hold. By Item 1 of Theorem 32, we also have $\text{rank}(F) > 2^{16}$. Thus, the conditions needed for Lemma 47 also hold.

To conclude the proof we require the use of Items 3 and 4 of Theorem 32. We now, prove that they hold for the sets relevant to the proof. By Item 1 of Theorem 32 and Definition 45, we see that $\max L \leq \log \text{LOPT}(U) - 2^7 \cdot \log \log \text{rank}(U)$. Also, by Item 4 of Lemma 46 and Definition 45, $L \subseteq \text{Valuable} \subseteq \text{Super}$. Hence, Items 3 and 4 of Theorem 32 hold, for every $K, K' \subseteq L$, where $\min K' > \max K$ or $K' = \emptyset$ and $\min\{\text{rank}(B_j) \mid j \in K\} \geq (2^{-5} \cdot \text{rank}(B_{\min K}))^{\frac{8}{9}}$, and for every $k \in K$.

Case 1: Suppose that $\text{LOPT}(B_{L'}^F) \geq \frac{1}{12} \cdot \text{LOPT}(U)$. We observe that $|L'| \leq 2^9 \cdot \log \log \text{rank}(F)$ since, by Definition 9, $\max \text{Valuable} \leq \log \text{LOPT}(F)$. Therefore, by the Pigeon Hole Principle and Definition 9, there exists $k \in L'$ such that $\text{LOPT}(B_k^F) \geq \frac{\text{LOPT}(U)}{2^{13} \cdot \log \log \text{rank}(F)}$. By Items 2 and 4 of Lemma 46 and Corollary 19, on input $J = \{k\}$, the Simple Algorithm will return an independent set of elements with an optimal value of at least $\frac{\text{LOPT}(U)}{2^{15} \cdot \log \log \text{rank}(F)}$. Since $\text{rank}(F) < \text{rank}(U)$, it follows that this is $\Omega(\frac{\text{OPT}(U)}{\log \log \text{rank}(U)})$.

Case 2: Suppose that $\text{LOPT}(B_L^F) < \frac{1}{12} \cdot \text{LOPT}(U)$ and there exists a manageable set $J \subseteq L$ such that $\sum_{j \in J} 2^j \cdot \text{uncov}(B_{J \setminus \{j\}}^F, B_j^F) \geq \frac{\text{LOPT}(B_L^F)}{2^{11} \cdot \log \log \text{rank}(F)}$. By Item 3 of Lemma 46, $\text{LOPT}(B_L^F) \geq \frac{1}{12} \cdot \text{LOPT}(U)$. By Definition 34 and Items 2 and 4 of Lemma 46, for every $j \in J$,

$$\text{rank}(B_j) \geq \text{rank}(B_j^F) \geq \left(\frac{1}{2} \cdot \sum_{i \in J} \text{rank}(B_i^F) \right)^{\frac{8}{9}} \geq \left(2^{-3} \cdot \text{rank}(B_{\min J}) \right)^{\frac{8}{9}}.$$

Thus, it follows that Item 3 of Theorem 32 holds with $K = J$ and $K' = \emptyset$. So, by Theorem 18, on input J , the Simple Algorithm will return an independent set of elements whose optimal value is at least $\frac{\text{LOPT}(B_L^F)}{2^{11} \cdot \log \log \text{rank}(F)} - 8 \cdot \sum_{j \in J} 2^j \cdot \text{rank}(B_j)^{\frac{3}{4}}$. Since $\text{rank}(F) < \text{rank}(U)$ and $\text{LOPT}(B_L^F) \geq \frac{1}{12} \cdot \text{LOPT}(U)$ and J is manageable, using Item 1 of Lemma 47, the preceding value is $\Omega(\frac{\text{OPT}(U)}{\log \log \text{rank}(U)})$.

Case 3: Suppose that $\text{LOPT}(B_{L'}^F) < \frac{1}{12} \cdot \text{LOPT}(U)$ and that the assumption that the manageable set J exists does not hold. By Item 1 of Lemma 46, Theorem 44 holds. Hence, there exists a critical tuple (Block, Good, Bad) as described in Theorem 44, which specifically satisfies:

$$\sum_{j \in \text{BLOCK}} 2^j \cdot \left(\text{uncov}(B_{\text{Bad}(j) \cup \text{Block}(j) \setminus \{j\}}^F, B_j^F) - \text{uncov}(B_{\text{Good}(j) \setminus \{j\}}^F, B_j^F) \right) \geq \frac{\text{LOPT}(B_L^F)}{2^{10} \cdot \log \log \text{rank}(F)}. \quad (3)$$

By Items 1 and 2 of Theorem 44 and Item 2 of Lemma 46, for every $i \in \text{BLOCK}$ and $j \in \text{Block}(i)$, we have that $\text{Good}(i)$, $\text{Bad}(i)$ and $\text{Block}(i)$ are subsets of L and

$$\text{rank}(B_j) \geq \text{rank}(B_j^F) > \left(\frac{1}{8} \cdot \sum_{\ell \in \text{Bad}(i) \cup \text{Block}(i)} \text{rank}(B_\ell^F) \right)^{\frac{8}{9}} \geq \left(2^{-5} \cdot \text{rank}(B_{\min \text{Block}(i)}) \right)^{\frac{8}{9}}.$$

Hence, using Definition 20, Theorem 24 and Items 3 and 4 of Theorem 32, we get that

$$\left| \text{uncov}(B_{\text{Bad}(i)}^F \cup B_{\text{Block}(i) \setminus \{i\}}^{U \setminus F}, B_i^{U \setminus F}) - \text{uncov}(B_{\text{Bad}(i) \cup \text{Block}(i) \setminus \{i\}}^F, B_i^F) \right| \leq 8 \text{rank}(B_{\text{Bad}(i) \cup \text{Block}(i)})^{\frac{3}{4}},$$

and

$$\text{loss}(B_{\text{Good}(i)}^F, B_i^{U \setminus F}) \leq \text{uncov}(B_{\text{Good}(i) \setminus \{i\}}^F, B_i^F) + 8 \cdot \text{rank}(B_i)^{\frac{3}{4}}.$$

So, by Theorem 24, using (3) it is straightforward to show that given (Block, Good, Bad) as input, the Gap Algorithm returns an independent set of elements whose optimal value is at least

$$\frac{\text{LOPT}(B_L^F)}{2^{10} \cdot \log \log \text{rank}(F)} - \sum_{j \in \text{BLOCK}} 2^j \cdot \left(8 \cdot \text{rank}(B_j)^{\frac{3}{4}} + 8 \cdot \text{rank}(B_{\text{Bad}(j) \cup \text{Block}(j)})^{\frac{3}{4}} \right).$$

This, in turn, is bounded below by $\frac{\text{LOPT}(B_L^F)}{2^{10} \cdot \log \log \text{rank}(F)} - 2^4 \cdot \sum_{j \in \text{BLOCK}} 2^j \cdot \text{rank}(B_{\text{Bad}(j) \cup \text{Block}(j)})^{\frac{3}{4}}$. Since $\text{rank}(F) < \text{rank}(U)$ and $\text{LOPT}(B_L^F) \geq \frac{1}{12} \cdot \text{LOPT}(U)$, using Item 2 of Lemma 47, the preceding value is $\Omega\left(\frac{\text{OPT}(U)}{\log \log \text{rank}(U)}\right)$.

Recall that, if Assumption 14 does not hold, then the Threshold Algorithm ensures that with constant probability an independent set of one element of value $\Omega\left(\frac{\text{OPT}(U)}{\log \log \text{rank}(U)}\right)$ is returned. Hence, we may assume that Case 1, Case 2 or Case 3 holds. So, in Preprocessing stage, we can check, using only the knowledge obtained about the elements of F via the oracle, which one of the cases hold as follows: First compute $\text{rank}(F)$. Then, use $\text{rank}(F)$ to determine the sets *Valuable*, L and L' . Now find every manageable subset of L and every critical tuple that satisfies the items of Theorem 44. Using this information check if there exists a bucket as guaranteed if Case 1 holds, a manageable set as guaranteed if Case 2 holds, or a critical tuple as guaranteed if Case 3 holds. The analysis of the cases ensures that at least one of the preceding exists. Pick arbitrarily if there exists more than one option. Finally, in the case of a single bucket or a manageable set proceed to Selection stage and use the Simple Algorithm, otherwise proceed to Selection stage and use the Gap Algorithm. ■

8 Discussion

The Main Algorithm achieves only the claimed competitive-ratio, when the following hold: the maximum value of an element of the Matroid is $O\left(\frac{\text{OPT}(U)}{\log \log \text{rank}(U)}\right)$ and, with probability at least $1 - O(\log \log \text{rank}(U)^{-1})$,

1. $2^j \cdot \text{rank}(B_j^F) = O\left(\frac{\text{LOPT}(U)}{\log \log \text{rank}(U)}\right)$, for every $j \in \text{Valuable}$,
2. $\sum_{j \in J} 2^j \cdot \text{uncov}(B_{J \setminus \{j\}}^F, B_j^F) = O\left(\frac{\text{LOPT}(U)}{\log \log \text{rank}(U)}\right)$, for every manageable subset J of the set L used in Theorem 48, and
3. for every critical tuple (Block, Good, Bad) in L , that satisfies the items of Theorem 44 $\sum_{j \in \mathbb{Z}} 2^j \cdot \left(\text{uncov}(B_{\text{Bad}(j) \cup \text{Block}(j) \setminus \{j\}}^F, B_j^F) - \text{uncov}(B_{\text{Good}(j) \setminus \{j\}}^F, B_j^F) \right) = O\left(\frac{\text{LOPT}(U)}{\log \log \text{rank}(U)}\right)$.

Clearly, the Gap Algorithm will perform better than claimed when the maximum value of an element of the Matroid is significantly larger than $\frac{\text{OPT}(U)}{\log \log \text{rank}(U)}$. The above also implies that the Gap Algorithm will perform better in many other cases, for example, if one of the following occurs with constant probability:

1. There exists $j \in \text{Valuable}$ such that $2^j \cdot \text{rank}(B_j^F) \gg \frac{\text{LOPT}(U)}{\log \log \text{rank}(U)}$, which implies that Item 1 above does not hold.

2. There exists a manageable $J \subseteq L$, such that $\sum_{j \in J} 2^j \cdot \text{rank}(B_j^F) \gg \frac{\text{LOPT}(U)}{\log \log \text{rank}(U)}$ and B_J is an independent set, which implies that Item 2 above does not hold.
3. The sum of $2^j \cdot \text{rank}(B_j^F)$ over every even $j \in L$, is significantly larger than $\frac{\text{LOPT}(U)}{\log \log \text{rank}(U)}$, and for every even j in the set L , that is used in Theorem 48, $\text{rank}(B_{i+1}^F) < 4 \cdot \text{rank}(B_i^F)$ and $\text{uncov}(B_{i+1}^F, B_i^F) = 0$. This implies that Item 3 above does not hold.

References

- [1] Noga Alon and Joel H. Spencer. *The Probabilistic Method*. Wiley, 2000.
- [2] Pablo D Azar, Robert Kleinberg, and S Matthew Weinberg. Prophet inequalities with limited information. In *Proceedings of the 45th symposium on Theory of Computing*, pages 123–136. ACM, 2013.
- [3] Moshe Babaioff, Nicole Immorlica, David Kempe, and Robert Kleinberg. A knapsack secretary problem with applications. In *APPROX/RANDOM*, pages 16–28, 2007.
- [4] Moshe Babaioff, Nicole Immorlica, and Robert Kleinberg. Matroids, secretary problems, and online mechanisms. In *SODA*, pages 434–443, 2007.
- [5] Siddharth Barman, Seeun Umboh, Shuchi Chawla, and David L. Malec. Secretary problems with convex costs. In *ICALP (1)*, pages 75–87, 2012.
- [6] MohammadHossein Bateni, MohammadTaghi Hajiaghayi, and Morteza Zadimoghaddam. Sub-modular secretary problem and extensions. *ACM Transactions on Algorithms (TALG)*, 9(4):32, 2013.
- [7] Niv Buchbinder, Kamal Jain, and Mohit Singh. Secretary problems via linear programming. *Mathematics of Operations Research*, 2013.
- [8] Sourav Chakraborty and Oded Lachish. Improved competitive ratio for the matroid secretary problem. In *SODA*, pages 1702–1712, 2012.
- [9] Nediialko B Dimitrov and C Greg Plaxton. Competitive weighted matching in transversal matroids. *Algorithmica*, 62(1-2):333–348, 2012.
- [10] Michael Dinitz. Recent advances on the matroid secretary problem. *ACM SIGACT News*, 44(2):126–142, 2013.
- [11] Michael Dinitz and Guy Kortsarz. Matroid secretary for regular and decomposable matroids. In *SODA*, pages 108–117. SIAM, 2013.
- [12] E. B. Dynkin. The optimum choice of the instant for stopping a markov process. *Sov. Math. Dokl.*, 4, 1963.

- [13] M. Feldman, J. Naor, and R. Schwartz. Improved competitive ratios for submodular secretary problems. *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 218–229, 2011.
- [14] P. R. Freeman. The secretary problem and its extensions: a review. *Internat. Statist. Rev.*, 51(2):189–206, 1983.
- [15] Shayan Oveis Gharan and Jan Vondrák. On variants of the matroid secretary problem. *Algorithmica*, 67(4):472–497, 2013.
- [16] Anupam Gupta, Aaron Roth, Grant Schoenebeck, and Kunal Talwar. Constrained non-monotone submodular maximization: offline and secretary algorithms. In *WINE*, pages 246–257, 2010.
- [17] Sungjin Im and Yajun Wang. Secretary problems: Laminar matroid and interval scheduling. In *SODA*, pages 1265–1274, 2005.
- [18] Patrick Jaillet, José A. Soto, and Rico Zenklusen. Advances on matroid secretary problems: Free order model and laminar case. *CoRR*, abs/1207.1333, 2012.
- [19] Robert Kleinberg. A multiple-choice secretary algorithm with applications to online auctions. In *SODA*, pages 630–631, 2005.
- [20] Nitish Korula and Martin Pál. Algorithms for secretary problems on graphs and hypergraphs. In *ICALP*, pages 508–520, 2009.
- [21] D. V. Lindley. Dynamic programming and decision theory. *Applied Statistics*, 10:39–51, 1961.
- [22] James G Oxley. *Matroid theory*, volume 3. Oxford university press, 2006.
- [23] José A Soto. Matroid secretary problem in the random-assignment model. *SIAM Journal on Computing*, 42(1):178–211, 2013.