$O(\log \log rank)$ 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 forms an independent set. An algorithm is *Known-Cardinality* if, in addition to (i) and (ii), it also knows, from the start, the cardinality n of the ground set of the Matroid.

We present here a Known-Cardinality algorithm with a *competitive-ratio* of $O(\log \log \rho)$, where ρ is the rank of the Matroid. The algorithm is also Order-Oblivious as defined by Azar *et al.* (2013). The prior known 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).

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 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

---

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

likewise if it is not. The goal is to design an algorithm for this problem wit ha 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 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 Vondráck [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 Vondráck 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 $O(\epsilon^{-1}(\log \log \rho) \log^{1+\epsilon} n)$. We believe, but do not prove explicitly, that 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 $O(\log \log \rho)$.

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, on 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.

One of the probabilistic inequalities we use is the key ingredient for ensuring that the Gap Algorithm works. It is not clear whether it is possible to prove such an inequality using only the techniques in [8], because they rely strongly on symmetry.

The paper is organized as follows: Section 2 contains the preliminaries required for the paper; Section 3 is an overview of the main result and techniques; Section 4 formally presents the Simple Algorithm and the Gap Algorithm; Section 5 presents the proof of the new probabilistic inequality;

Section 6 gives the proof of the new structural theorem for matroids; Section 7 contains the proof of the main result; and the Appendix mainly contains proofs that were added for the sake of completeness.

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 $[\alpha]$ to denote $\{1, 2, \ldots, \lfloor \alpha \rfloor\}$ for any non-negative real α. We use $[\alpha, \beta]$ to denote $\{i \in \mathbb{Z} \mid \alpha \leq i \leq \beta\}$ and $(\alpha, \beta]$ to denote $\{i \in \mathbb{Z} \mid \alpha < i \leq \beta\}$, and so on. For every $j \in \mathbb{Z}$ and $I \subseteq \mathbb{Z}$, we define $j > I$ if and only if $j > i$ for every $i \in I$, and $j < I$ if and only if $j < i$ for every $i \in I$. For every $I, J \subseteq \mathbb{Z}$, we define $J > I$ if and only if $\min J > \max I$, and we say I and J are *comparable* if either $J > I$, or $I > J$ or $I = J$. We use $med(f)$ to denote the *median* of a function f from a finite set to the non-negative reals. If there are two possible values for $med(f)$ 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 $A = \{a_1, a_2, \ldots, a_n\}$ and $W = \beta(n, 1/2)$; let $\pi : [n] \longrightarrow [n]$ 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 $val(e)$.

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

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

- $Cl(S) = \{e \in U \mid rank(S \cup \{e\}) = rank(S)\}$.

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. *$rank\,(S_1) \leq |S_1|$, where equality holds if and only if S_1 is an independent set,*

2. *if $S_1 \subseteq S_2$ or $S_1 \subseteq Cl\,(S_2)$, then $S_1 \subseteq Cl\,(S_1) \subseteq Cl\,(S_2)$ and $rank\,(S_1) \leq rank\,(S_2)$,*

3. *if $e \notin Cl\,(S_1)$, then $rank\,(S_1 \cup \{e\}) = rank\,(S_1) + 1$,*

4. *$rank\,(S_1 \cup S_2) \leq rank\,(S_1) + rank\,(S_2)$, and*

5. *suppose that S_1 is minimal such that $e \notin S_1$ but $e \in Cl\,(S_1)$, that is, e is not in the Closure of any subset of S_1, then $e^* \in Cl\,((S_1 \setminus \{e^*\}) \cup \{e\})$, for every $e^* \in S_1$.*

Proof.   We now prove 5. Assume for the sake of contradiction that there exists $e^* \in S_1$ such that $e^* \notin Cl\,((S_1 \setminus \{e^*\}) \cup \{e\})$. By the minimality of S_1, $e \notin Cl\,(S_1 \setminus \{e^*\})$ and hence, $rank\,((S_1 \setminus \{e^*\}) \cup \{e\}) = rank\,(S_1)$. Thus, by the initial assumption and Item 3, $rank\,((S_1 \setminus \{e^*\}) \cup \{e\} \cup \{e^*\}) = rank\,(S_1) + 1$. Yet, since $e \in Cl\,(S_1)$, we also have $rank\,((S_1 \setminus \{e^*\}) \cup \{e\} \cup \{e^*\}) = rank\,(S_1)$ which is a contradiction to the preceding equality. ∎

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

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 $B_i = \{e \in U \mid val(e) = 2^i\}$. 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 $OPT\,(S) = \max \left\{ \sum_{e \in S'} val(e) \,\middle|\, S' \subseteq S \text{ and } S' \in \mathcal{I} \right\}$.*

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

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

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

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

1. *$LOPT\,(S) \geq OPT\,(S)$,*

2. *$LOPT\left(B_{J_1}^S\right) = \sum_{i \in J_1} 2^i \cdot rank\left(B_i^S\right)$ and*

3. *if $J_1 \cap J_2 = \emptyset$, then $LOPT\left(B_{J_1 \cup J_2}^S\right) = LOPT\left(B_{J_1}^S\right) + LOPT\left(B_{J_2}^S\right)$.*

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 $\mathcal{S}$ for the secretary problem (together with its corresponding analysis) is **order-oblivious** if, on a randomly ordered input vector $(v_{i_1}, \ldots, v_{i_n})$:*

1. *(algorithm) $\mathcal{S}$ sets a (possibly random) number k, observes without accepting the first k values $S = \{v_{i_1}, \ldots, v_{i_k}\}$, and uses information from S to choose elements from $V = \{v_{i_{k+1}}, \ldots, v_{i_n}\}$.*

2. *(analysis) $\mathcal{S}$ 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 Overview

3.1 Overview of Main Algorithm

We start this section with a high level description of the Main Algorithm. The input to the Main Algorithm is the number of indices n in a randomly ordered input vector $(e_1, \ldots, e_n)$, where $\{e_1, \ldots, e_n\}$ 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 $U \backslash F$ 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 3.3.

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 a non-negative value. The Threshold Algorithm, that 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 $\max\{val(e) \mid e \in U\} \geq 2^{-2^{34}} \cdot OPT(U)$, then the claimed competitive-ratio is achieved. Therefore, from here on, unless explicitly mentioned otherwise, we make the following assumption:

Assumption 14 $\max\{val(e) \mid e \in U\} < 2^{-2^{34}} \cdot OPT(U)$.

In this paper the constants have not been optimized.

The Simple Algorithm and the Gap Algorithm share a common scheme. They are both variations of the following Greedy Algorithm: start with an empty set P, then when an element e is revealed, add the element to P if $P \cup \{e\}$ is an independent set. Clearly, at the end of this process P is an independent set. In addition, by the definition of the *rank* function (Definition 3), it is easy to see that the rank of P is exactly the rank of the set of all the elements revealed. This implies that, in the trivial case where all the elements of the matroid have the same value, the Greedy Algorithm achieves a competitive-ratio of 1.

In the following subsection, we describe the Simple Algorithm, formally prove a guarantee on its output, and explain when the output of the Preprocessing stage is a set of bucket indices. In Subsection 3.3, we define critical tuple, describe the Gap Algorithm and explain when the output of the Preprocessing stage is a critical tuple. In Subsection 3.4, we explain the probabilistic inequality that validates the use of the Gap Algorithm. In Subsection 3.5, we explain the central structural results we use and, in Subsection 3.6, we explain how everything fits together.

3.2   The Simple Algorithm

Exactly the same algorithm is also used in [8]. We explain the algorithm and prove its correctness as a preparation towards the presentation of the Gap Algorithm.

Let J be a set of bucket indices that is the output of the Preprocessing stage. The Simple Algorithm, when executed in the Selection stage, receives J as an input and selects an independent subset P of $U \setminus F$ as follows: initially it sets $P = \emptyset$ and then, each time an element e is revealed, it is added to P if and only if (a) $\log val(e) \in J$ and (b) $P \cup \{e\}$ is an independent set.

Thus, the Simple Algorithm operates exactly like the Greedy Algorithm would if only the elements of $B_J^{U \setminus F}$ were revealed to it. Hence, the set P of elements selected by the Simple Algorithm satisfies $rank(P) = rank\left(B_J^{U \setminus F}\right)$. Now, for every $j \in J$, since the set $P \cap B_{J \setminus \{j\}}^{U \setminus F}$ has a rank of at most $rank\left(B_{J \setminus \{j\}}^{U \setminus F}\right)$, by Item 4 of Proposition 4, there are at least $rank\left(B_J^{U \setminus F}\right) - rank\left(B_{J \setminus \{j\}}^{U \setminus F}\right)$ elements of $B_j^{U \setminus F}$ in P. We use the following definition to succinctly express this fact.

Definition 15 [uncov] *For every $R, S \subseteq U$, let*

$$uncov\left(R, S\right) = rank\left(R \cup S\right) - rank\left(R \setminus S\right).$$

According to the preceding intuition, for every $j \in J$, $uncov\left(B_{J\setminus\{j\}}^{U\setminus F}, B_j^{U\setminus F}\right)$ is the minimum number of elements the Simple Algorithm will select from $B_j^{U\setminus F}$, given J. Thus, given a finite set $J \subset \mathbb{Z}$, the Simple Algorithm selects an independent set $P \subseteq U \setminus F$ such that $OPT\left(P\right)$ is at least $\sum_{j\in J} 2^j \cdot uncov\left(B_{J\setminus\{j\}}^{U\setminus F}, B_j^{U\setminus F}\right)$. This implies that, in order to achieve the claimed competitive-ratio, it is sufficient to find a set J such that $\sum_{j\in J} 2^j \cdot uncov\left(B_{J\setminus\{j\}}^{U\setminus F}, B_j^{U\setminus F}\right)$ is sufficiently large, with high probability. The problem is to find such a set. Searching for such a J is done by exhaustively checking whether a specific portion of the subsets of a specific set of bucket indices, called *Valuable*, contains such a set J. The proof that this is a good choice relies on the following precise definition of *Valuable*:

Definition 16 [Valuable] *We define* **Valuable** *to be the set of all $j \geq 4 + \log LOPT\left(F\right) - 2 \cdot \log rank\left(F\right)$ such that $rank\left(B_j^F\right) \geq \max\left\{1, \sqrt{\frac{LOPT(F)}{2^{j+8}}}\right\}$.*

It is easy to see that the size of *Valuable* is small, because $j \leq \log LOPT\left(F\right)$, for every j such that $B_j^F \neq \emptyset$ and hence the following observation holds:

Observation 17 $|Valuable| \leq 2 \cdot \log rank\left(F\right)$

In the formal part of the paper we use the preceding observation in a proof that, with very high probability, $rank\left(B_{J^*}^F\right) \approx rank\left(B_{J^*}^{U\setminus F}\right)$, for all $J^* \subseteq Valuable$. This ensures that if a set J as described is found, then with very high probability, for every $j \in J$, $\sum_{j\in J} 2^j \cdot uncov\left(B_{J\setminus\{j\}}^F, B_j^F\right) \approx \sum_{j\in J} 2^j \cdot uncov\left(B_{J\setminus\{j\}}^{U\setminus F}, B_j^{U\setminus F}\right)$. Consequently, if a set such as J exists, it will be found and this will ensure that the Simple Algorithm will achieve the claimed competitive-ratio.

3.3 Overview of the Gap Algorithm

This algorithm ensures that the competitive-ratio of this paper is exponentially better than that of [8]. We start the overview with a simple scenario and some wishful thinking assumptions that will eventually lead to the Gap Algorithm.

Suppose that the output of Preprocessing stage consists of two disjoint sets of bucket indices K_1 and K_2, and a minimal independent set $Fence$ such that $B_{K_1}^{U\setminus F} \subseteq Cl\left(Fence\right)$ that satisfy the following:

1. $LOPT\left(B_{K_1}^{U\setminus F}\right) = LOPT\left(B_{K_2}^{U\setminus F}\right)$;

2. for every $j \in K_2$, $rank\left(Fence\right)$ is significantly smaller than $rank\left(B_j^{U\setminus F}\right)$, say $rank\left(B_j^{U\setminus F}\right) \geq 32 \cdot rank\left(Fence\right)$ (this is the wishful thinking);

3. for $i = 1$ or 2, if the Simple Algorithm received K_i, then it returns a set of elements P that satisfies $OPT\left(P\right) = LOPT\left(B_{K_i}^{U\setminus F}\right)$;

8

4. If the Simple Algorithm received $K_1 \cup K_2$, then it returns a set of elements P that satisfies $OPT(P) \approx LOPT\left(B_{K_2}^{U \setminus F}\right)$.

The following is a simple example of such a matroid. Suppose that $K_1 = \{6\}$, $K_1 = \{1\}$, $|B_{K_1}^{U \setminus F}| = n_1$, $B_{K_1}^{U \setminus F}$ and $B_{K_2}^{U \setminus F}$ contains a very large number of sets S such that, for every $e_1 \in S$, there exists $e_2 \in B_{K_1}^{U \setminus F}$ that satisfies $e_2 \in Cl(\{e_1\})$. Suppose also that $B_{K_2}^{U \setminus F}$ contains a set of $31 \cdot n_1$ elements that together with $B_{K_1}^{U \setminus F}$ form an independent set. Clearly, in this case, if the Simple Algorithm received $K_1 \cup K_2$, then it would hardly select any elements from $B_{K_1}^{U \setminus F}$ and hence will return a set of elements P that satisfies $OPT(P) \approx LOPT\left(B_{K_2}^{U \setminus F}\right)$. The wishful thinking is that we also had a minimal independent set $Fence$ such that $B_{K_1}^{U \setminus F} \subseteq Cl(Fence)$.

We note that, for a more specific but slightly different example, one can assume the matroid is a vector matroid, take $Fence$ to be a basis for $B_{K_1}^{U \setminus F}$, and $B_{K_2}^{U \setminus F}$ that contain a very large number of sets S, each spanning the same vector space spanned by $B_{K_1}^{U \setminus F}$. It should be assumed that the number of these sets is so large that, with very high probability, all the vectors, in at least one of them, are revealed before any vector in $B_{K_1}^{U \setminus F}$.

The algorithm we describe now for this scenario selects an independent set of elements whose sum is almost $LOPT\left(B_{K_1 \cup K_2}^{U \setminus F}\right)$. This, by the last assumption above, is much better than what would happen if the Simple Algorithm was used with $K_1 \cup K_2$. The algorithm initially sets P to be the empty set and for each element $e \in U \setminus F$ that is revealed, e is added to P if either $\log val(e) \in K_2$ and $e \notin Cl(P \cup Fence)$, or $\log val(e) \in K_1$ and $e \notin Cl(P)$.

Intuitively, for every $j \in K_2$, the fact that we require every element $e \in B_j^{U \setminus F}$ to satisfy the condition $e \notin Cl(P \cup Fence)$, in order to be selected, results in $|P \cap B_j^{U \setminus F}| \geq rank\left(B_j^{U \setminus F}\right) - |Fence|$. Hence, by assumption 2 above, for every $j \in K_2$, we have that $|P \cap B_j^{U \setminus F}| \geq (1 - \frac{1}{32}) \cdot rank\left(B_j^{U \setminus F}\right)$.

The situation is even better for K_1. According to the selection condition of elements in $B_{K_2}^{U \setminus F}$, it can never be the case that an element from $Cl(Fence)$ is in $Cl\left(P \cap B_{K_2}^{U \setminus F}\right)$ and hence, in particular, $Cl\left(P \cap B_{K_2}^{U \setminus F}\right)$ and $B_{K_1}^{U \setminus F}$ are disjoint. This in turn implies that the elements of $P \cap B_{K_2}^{U \setminus F}$ cannot in any way prevent an element of $B_{K_1}^{U \setminus F}$ from being selected. Thus, by assumption 3 above, for every $j \in K_1$, we have that $|P \cap B_j^{U \setminus F}| = rank\left(B_j^{U \setminus F}\right)$. Therefore, the set of elements P selected by this algorithm satisfies $OPT(P) \geq LOPT\left(B_{K_1}^{U \setminus F}\right) + (1 - \frac{1}{32}) \cdot LOPT\left(B_{K_2}^{U \setminus F}\right)$ which is almost $LOPT\left(B_{K_1 \cup K_2}^{U \setminus F}\right)$.

It is not clear how to find a replacement for $Fence$ using F. So, instead we use something slightly weaker. We will be satisfied if $Fence$ is replaced by a set B' such that $\sum_{j \in K_1} 2^j \cdot rank\left(B_j^{U \setminus F} \setminus Cl(B')\right)$ is very small. We use the following definition to capture this idea.

Definition 18 *[loss] For every* $R, S \subseteq U$, *let*

$$loss(R, S) = rank(S \setminus Cl(R)).$$

According to this definition, we want to find a B' such that $\sum_{j \in K_1} 2^j \cdot loss\left(B', B_j^{U \setminus F}\right)$ is very small. So, suppose that instead of the set $Fence$ we can find a set B' such that assumption 2 holds for

Fence replaced by B' and $loss\left(B', B_j^{U\setminus F}\right) \leq \frac{1}{16} \cdot rank\left(B_j^{U\setminus F}\right)$, for every $j \in K_1$. In this case we use an algorithm that is the same as the preceding algorithm except for the last condition. The algorithm initially sets P to be the empty set and for each element $e \in U \setminus F$ that is revealed, e is added to P if either $\log val(e) \in K_2$ and $e \notin Cl\left(P \cup B'\right)$, or $\log val(e) \in K_1$, $e \notin Cl\left(P\right)$ **and** $e \in Cl\left(B'\right)$.

By reasoning similar to that above, for every $j \in K_2$, we have that $|P \cup B_j^{U\setminus F}| \geq (1 - \frac{1}{32}) \cdot rank\left(B_j^{U\setminus F}\right)$; and for every $j \in K_1$, we have that $|P \cap B_j^{U\setminus F}| \geq (1 - \frac{1}{16}) \cdot rank\left(B_j^{U\setminus F}\right)$, by Item 4 of Proposition 4 and the fact that the rank of all the elements in $B_j^{U\setminus F}$ that are not in $Cl\left(B'\right)$ is at most $\frac{1}{16} \cdot rank\left(B_j^{U\setminus F}\right)$. Consequently, $OPT\left(P\right)$ is almost $LOPT\left(B_{K_1 \cup K_2}^{U\setminus F}\right)$. Note that, without the extra condition the algorithm would not work.

Further on we show that a set $B_{K'}^F$ such that $K_1 \subset K'$ and $K' \cap K_2 = \emptyset$ can be used for the role of B'.

Now we describe what happens when indeed the preceding replacement occurs and assumption 3 above does not necessarily hold. When the algorithm is dealing with elements in $B_{K_2}^{U\setminus F}$, the elements of $B_{K'}^F$ can be viewed as if they were initially selected to be in P and hence, by the same reasoning as for the Simple Algorithm, for every $j \in K_2$, we have that $|P \cap B_j^{U\setminus F}| \geq uncov\left(B_{K'}^F \cup B_{K_2\setminus\{j\}}^{U\setminus F}, B_j^{U\setminus F}\right)$. When the algorithm is dealing with elements in $B_{K_1}^{U\setminus F}$, for every $j \in K_1$ the rank of the set of elements in $B_j^{U\setminus F}$ satisfying the new condition is at least $rank\left(B_j^{U\setminus F}\right) - loss\left(B_{K'}^F, B_j^{U\setminus F}\right)$ and out of these the rank of those not selected is at most $rank\left(B_j^{U\setminus F}\right) - uncov\left(B_{K_1\setminus\{j\}}^F, B_j^{U\setminus F}\right)$. Thus, for every $j \in K_1$, we have that $|P \cap B_j^{U\setminus F}| \geq uncov\left(B_{K_1\setminus\{j\}}^F, B_j^{U\setminus F}\right) - loss\left(B_{K'}^F, B_j^{U\setminus F}\right)$. For the Gap Algorithm, which we describe next, we need a nested variation of the preceding setting, which is captured by the following definition.

Definition 19 *[critical tuple]* $(Block, Good, Bad)$, *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 *and* j *in* $\mathbb{Z}$:

1. *if* $Block(j) \neq \emptyset$, *then* $j \in Block(j)$

2. *if* $i \in Block(j)$, *then* $Block(i) = Block(j)$, $Good(i) = Good(j)$ *and* $Bad(i) = Bad(j)$,

3. $Block(i) \cap Bad(i) = \emptyset$ *and* $Block(i) \subseteq Good(i)$, *and*

4. *there exists a minimal set* $\{j_1, j_2, \ldots, j_s\}$ *that contains a distinct element from* $Block(\ell)$, *for every* $\ell \in \mathbb{Z}$ *such that* $Block(\ell) \neq \emptyset$, *and*

$$Bad(j_1) \subset Good(j_1) \subseteq Bad(j_2) \subset Good(j_2) \subseteq \cdots \subseteq Bad(j_s) \subset Good(j_s).$$

The following observation, follow directly from the previous definition.

Observation 20 *If* $(Block, Good, Bad)$ *is a **critical tuple**, then*

1. *the sets in* $\{Block(j)\}_{j \in \mathbb{Z}}$ *are pairwise-disjoint, and*

2. *for every i and j in $\bigcup_{\ell \in \mathbb{Z}} Block(\ell)$, if $j \notin Good(i)$, then $Good(i) \subseteq Bad(j)$.*

We note that the previous scenario can be captured by the above definition if $\{Block(j)\}_{j \in \mathbb{Z}} = \{K_1, K_2\}$, $Good(j) = K'$ and $Bad(j) = \emptyset$, for every $j \in K_1$, and $Good(j) = \emptyset$ and $Bad(j) = K'$, for every $j \in K_2$. Thus, in this more general definition, the role of K', is done by two possibly different sets.

The Gap Algorithm initially sets P to be the empty set and for each element $e \in U \setminus F$ that is revealed, e is added to P if $\log val(e) \in \cup_{j \in \mathbb{Z}} Block(j)$, and $e \in Cl\left(B_{Good(\log val(e))}^{F}\right)$ and $e \notin Cl\left(P \cup B_{Bad(\log val(e))}^{F}\right)$. The guarantee proved for Gap Algorithm, in Section 4, is that the set of elements P is independent and

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

Thus, the goal is to search for a critical tuple $(Block, Good, Bad)$, for which the previous expression is sufficiently large. As in the case of the Simple Algorithm, the search is restricted to the members of $Valuable$. The ability to find a good candidate for such a critical tuple, when the Simple Algorithm cannot be used, enables the Main Algorithm to achieve a competitive-ratio that is exponentially better than that of [8]. The reason this can be done, is that by using the probabilistic inequality we describe in Subsection 3.4, we prove that, with very high probability, for every $K \subseteq Valuable$ and every $j \in K$, $loss\left(B_K^{F}, B_j^{U \setminus F}\right)$ is less than $uncov\left(B_{K \setminus \{j\}}^{F}, B_j^{F}\right)$ plus a bit. Thus, together with what we know about $uncov$ from the Simple Algorithm, the indication that a critical tuple $(Block, Good, Bad)$ is good for our purposes is that $\sum_{j \in \bigcup_{i \in \mathbb{Z}} Block(i)} 2^j \cdot \left(uncov\left(B_{Bad(j)}^{F} \cup B_{Block(j) \setminus \{j\}}^{F}, B_j^{F}\right) - uncov\left(B_{Good(j) \setminus \{j\}}^{F}, B_j^{F}\right) \right)$ is sufficiently large.

In the following subsection, we explain how we prove the probabilistic inequality that is used for bounding $loss$ by $uncov$. In the subsection afterwards, we explain why in the Preprocessing stage a critical tuple as above can be found when a set as required for the Simple Algorithm can not be found. In the last subsection of this section, we give a short explanation how everything fits together to form the main result.

3.4 Overview of central probabilistic inequality

We give here an overview of the proof of Theorem 23, which is stated and proved in Section 5 and asserts that, for every finite subset K of $\mathbb{Z}$ and every $k \in K$, if $rank(B_k)$ is sufficiently large then, with very high probability, $loss\left(B_k^{U \setminus F}, Cl\left(B_K^{F}\right)\right)$ is at most slightly larger than $uncov\left(B_{K \setminus \{k\}}^{F}, B_k^{F}\right)$.

The proof is based on a process of exposing the elements of B_K one by one in an on-line manner according to a specific predetermined order, where an element in B_K is *exposed* when it is determined whether it is in F or in $U \setminus F$. During the process two sets are constructed H and $\tilde{H}$. They are both initially empty. The predetermined order is as follows:

1. All the elements in $B_{K \setminus \{k\}}$ are exposed according to an arbitrary predetermined order, and none of them are added to either H or $\tilde{H}$.

2. The elements of B_k are exposed according to a predetermined order, which is represented by a labeled tree and ensures that during the process elements $e \in B_k$ such that $e \notin Cl\left(B^F_{K \setminus \{k\}} \cup H\right)$ are exposed first. An element $e \in B_k$ such that $e \notin Cl\left(B^F_{K \setminus \{k\}} \cup H\right)$ is added to H if it is in F, and otherwise it is added to $\tilde{H}$.

It turns out, as we shall explain later, that $|H| = uncov\left(B^F_{K \setminus \{k\}}, B^F_k\right)$, $|\tilde{H}| \geq uncov\left(B^F_{K \setminus \{k\}}, B^F_k\right)$ and, with very high probability, $|\tilde{H}|$ is at most slightly larger than $|H|$. It is easy to see that this implies Theorem 23.

It remains to explain why the preceding three assertions hold. We start by explaining the first. According to Item 2, H is an independent set and $rank\left(B^F_{K \setminus \{k\}} \cup H\right) = rank\left(B^F_K\right)$, because every element in B^F_k must be also in $Cl\left(B^F_{K \setminus \{k\}} \cup H\right)$. The required equality follows, by the definition of $uncov$ (Definition 15). We next explain why $|\tilde{H}| \geq uncov\left(B^F_{K \setminus \{k\}}, B^F_k\right)$.

According to Item 2, $Cl\left(B^F_{K \setminus \{k\}} \cup H\right) = Cl\left(B^F_K\right)$ and every element in $B^{U \setminus F}_k \setminus \tilde{H}$ is in $Cl\left(B^F_{K \setminus \{k\}} \cup H\right)$. Thus, $\tilde{H}$ contains all the elements in $B^{U \setminus F}_k \setminus Cl\left(B^F_K\right)$ and the required inequality follows.

We now explain the probabilistic part. We note that every element in $H \cup \tilde{H}$ was selected to the specific set with probability $\frac{1}{2}$ independently, so any event in which $|\tilde{H}|$ is significantly larger than $|H|$ has a very low probability and therefore, the required probabilistic inequality follows. The actual proof relies on the predetermined order in order to use probabilities that are conditioned on the number of elements in $H \cup \tilde{H}$.

3.5 Overview of structural results used

In this section we explain how we prove that if the set of bucket indices $Valuable$ does not contain a *heavy* set then it has a *heavy* critical tuple $(Block, Good, Bad)$, where by heavy set and heavy critical tuple we mean the following: a set $J \subseteq Valuable$ is heavy if $\sum_{j \in J} 2^j \cdot uncov\left(B^F_{J \setminus \{j\}}, B^F_j\right)$ is sufficiently large; and a critical tuple $(Block, Good, Bad)$ is heavy if

$$\sum_{j \in \bigcup_{i \in \mathbb{Z}} Block(i)} 2^j \cdot \left(uncov\left(B^F_{Bad(j)} \cup B^F_{Block(j) \setminus \{j\}}, B^F_j\right) - uncov\left(B^F_{Good(j) \setminus \{j\}}, B^F_j\right)\right)$$

is sufficiently large. We note that both these structures can be found using exhaustive search and the knowledge about the matroid that consists of F and whatever can be gained by access to the matroid oracle. We also note that given a heavy set the Simple Algorithm is guaranteed to achieve the claimed competitive-ratio and given a heavy critical tuple $(Block, Good, Bad)$ the Gap Algorithm is guaranteed to achieve the claimed competitive-ratio.

We now describe a technique we use for constructing a critical tuple from a family of comparable subsets of $Valuable$. Afterwards we explain its role in finding a heavy critical tuple. Given a partition $\{H_1, H_2, \ldots, H_s\}$ of a set H and sets $H^*_1, H^*_2, \ldots, H^*_s$ such that $H_1 > H_2 > \cdots > H_s$ and, for every $i \in [s]$, $H^*_i \subset H_i$, we construct the tuple $(Block, Good, Bad)$ as follows: for every $i \in \bigcup_{j \in [s]} H^*_j$, we let $Block(i) = H^*_i$, $Bad(i) = \{t \in H \mid t > H^*_i\}$ and $Good(i) = H_i \cup Bad(i)$, and for every other i, we let $Block(i) = Bad(i) = Good(i) = \emptyset$. It is easy to see that this construction satisfies the definition of a critical tuple (Definition 19).

12

Recall that one of the features a heavy critical tuple, implicitly mentioned in Subsection 3.3, is that, for every $i \in \sum_{j \in \mathbb{Z}} Block(j)$, we have that $rank\left(B_i^F\right)$ is significantly larger than $rank\left(B_{Bad(i)}^F\right)$. This is the reason why we search for a heavy critical tuple not in $Valuable$, but instead in a subset of it that consists of the members of a *strong* sequence H consisting of the integers $h_1, h_2, \ldots, h_k$, which is a sequence that has the following properties: it is strictly monotonically increasing; $LOPT\left(B_H^F\right)$ is close to $LOPT\left(B_{Valuable}^F\right)$; and for every $j \in [k-1]$, $0 < rank\left(B_{h_j}^F\right) \le \frac{1}{32} \cdot rank\left(B_{h_{j+1}}^F\right)$.

The next step towards finding a heavy critical tuple is to partition H into a family of comparable sets, that is denoted by $Partition(H)$, such that, for every set H_i in the family, $rank\left(B_{\max H_i}^F\right)$ is significantly larger than $rank\left(B_{H_i}^F\right)^c$, where $c < 1$ is defined below. This property is required in order to ensure that the extra terms, that are added because of the use of the probabilistic inequalities, are negligible.

In order to find the sets for the construction of the heavy critical tuple, we use an iterative process that applies one of three operation on a set depending on which one of the following types it is:

1. a set $K \subseteq H$ is *negligible* if $LOPT\left(B_K^F\right)$ is significantly small,

2. a set $K \subseteq H$ is *useful* if it has a subset K^* such that $\sum_{j \in K^*} 2^j \cdot \left(uncov\left(B_{M(K^*) \cup K^* \setminus \{j\}}^F, B_j^F\right) - uncov\left(B_{M(K) \cup K \setminus \{j\}}^F, B_j^F\right)\right)$ is sufficiently large, and

3. a set $K \subseteq H$ is *splittable* it has a partition into two comparable sets, each with a $LOPT$ measure that is not too small.

The iterative process starts with $Partition(H)$. On each iteration it goes over all the sets, removing every set that is negligible and then replacing every set that is splittable and not useful or negligible with the two sets of a partition that makes it splittable. The iterations ends when all the remaining sets are useful.

We note that splittable sets are replaced with sets that are strictly smaller and that sets of size 1 are not splittable. We also note that, as we prove later on, every one of the sets in the process is either negligible or useful or splittable or some combination of these. Thus, after a finite number of iterations, all the remaining sets are useful and it easy to see that all these sets are also pairwise comparable. The aforementioned process, for constructing a critical tuple is applied to the remaining sets.

To prove that the resulting critical tuple is heavy, we show that the $LOPT$ measure over all the remaining sets is not much smaller the $LOPT$ measure over the set of $Partition(H)$, which we show is sufficiently large. We prove this by showing that the $LOPT$ of the sets we removed is small. In the proof we also use the fact that $Partition(H)$ has only $O(\log \log rank(F))$ sets and hence, because there are no heavy sets, for every set H_i in the partition either $LOPT\left(B_H^F\right)$ is small or $\sum_{j \in H_i} 2^j \cdot uncov\left(B_{H_i \setminus \{j\}}^F, B_j^F\right)$ is small. Because the sum of $LOPT$ over the sets of $Partition(H)$ is large, this implies that the preceding sum is small. We also show that the same holds for the remaining sets despite the replacement of the splittable sets.

3.6 Main result

We give here a short sketch of the proof of the main result appearing in Section 7. The proof starts by showing that if Assumption 14 does not hold, then the claimed competitive-ratio is achieved, because with probability $\frac{1}{2}$ the output of the Preprocessing stage is a non-negative value. Afterwards it is shown that, if Assumption 14 does hold, then by using the structural and probabilistic techniques we described, we show that the output of the Preprocessing stage is such that the relevant algorithm achieves the claimed competitive-ratio.

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-ratio they achieves.

4.1 The Simple Algorithm

Algorithm 1 Simple Algorithm

Input: a set J of bucket indices

1. $P \longleftarrow \emptyset$

2. immediately after each element $e \in U \setminus F$ is revealed, do

   (a) if $\log val(e) \in J$, do
       i. if $e \notin Cl(P)$, do $P \longleftarrow P \cup \{e\}$

Output: P

Theorem 21 *Given a finite $J \subset \mathbb{Z}$, the Simple Algorithm selects an independent set $P \subseteq U \setminus F$ such that*

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

Proof.   According to the manner in which the Simple Algorithm selects elements, P is independent and

$$rank(P) = rank\left(B_J^P\right) = rank\left(B_J^{U \setminus F}\right).$$

Consequently, for every $j \in J$, since $B_{J \setminus \{j\}}^P \subseteq B_{J \setminus \{j\}}^{U \setminus F}$, by Items 2 and 4 of Proposition 4,

$$rank\left(B_j^P\right) \geq rank\left(B_J^P\right) - rank\left(B_{J \setminus \{j\}}^P\right) \geq rank\left(B_J^{U \setminus F}\right) - rank\left(B_{J \setminus \{j\}}^{U \setminus F}\right).$$

Thus, by the definition of *uncov* (Definition 15),

$$OPT(P) \geq \sum_{j \in J} rank\left(B_j^P\right) \geq \sum_{j \in J} uncov\left(B_{J \setminus \{j\}}^{U \setminus F}, B_j^{U \setminus F}\right).$$

By Observation 8, this implies the theorem. ∎

4.2 The Gap Algorithm

Algorithm 2 Gap Algorithm

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

1. $P \longleftarrow \emptyset$

2. immediately after each element $e \in U \setminus F$ is revealed, do

   (a) $i \longleftarrow \log val(e)$

   (b) if $i \in \bigcup_{j \in \mathbb{Z}} Block(j)$, do

      i. if $e \in Cl\left(B_{Good(i)}^{F}\right)$, do

         A. if $e \notin Cl\left(P \cup B_{Bad(i)}^{F}\right)$, do $P \longleftarrow P \cup \{e\}$

Output: P

Theorem 22 *Given a critical tuple $(Block, Good, Bad)$ as input, Algorithm 2 returns an independent set of elements $P \subseteq U \setminus F$ such that*

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

Proof. We note that Step 2(b)iA implies that P is always an independent set. We show that, after all the elements of $U \setminus F$ have been revealed, for every $j \in \bigcup_{i \in \mathbb{Z}} Block(i)$,

$$rank\left(B_j^P\right) \geq uncov\left(B_{Bad(j)}^{F} \cup B_{Block(j)\setminus\{j\}}^{U\setminus F}, B_j^{U\setminus F}\right) - loss\left(B_{Good(j)}^{F}, B_j^{U\setminus F}\right). \tag{1}$$

By Observation 8, this implies the theorem. Let $j \in \bigcup_{i \in \mathbb{Z}} Block(i)$ and

$$E_j = B_j^{U\setminus F} \setminus Cl\left(B_{Good(j)}^{F}\right). \tag{2}$$

Suppose first that

$$rank\left(B_{Bad(j)}^{F} \cup B_{Block(j)\setminus\{j\}}^{U\setminus F} \cup E_j \cup B_j^{P}\right) = rank\left(B_{Bad(j)}^{F} \cup B_{Block(j)}^{U\setminus F}\right) \tag{3}$$

We next show that (3) implies that the theorem holds and afterwards prove that (3) indeed holds. By Item 4 of Proposition 4,

$$rank\left(E_j \cup B_j^{P}\right) \geq rank\left(B_{Bad(j)}^{F} \cup B_{Block(j)\setminus\{j\}}^{U\setminus F} \cup E_j \cup B_j^{P}\right) - rank\left(B_{Bad(j)}^{F} \cup B_{Block(j)\setminus\{j\}}^{U\setminus F}\right).$$

Therefore, by (3),

$$rank\left(E_j \cup B_j^{P}\right) \geq rank\left(B_{Bad(j)}^{F} \cup B_{Block(j)}^{U\setminus F}\right) - rank\left(B_{Bad(j)}^{F} \cup B_{Block(j)\setminus\{j\}}^{U\setminus F}\right).$$

Hence, by Item 4 of Proposition 4 and the definition of *uncov* (Definition 15),

15

$$rank\left(B_j^P\right) \ge uncov\left(B_{Bad(j)}^F \cup B_{Block(j)\setminus\{j\}}^{U\setminus F}, B_j^{U\setminus F}\right) - rank\left(E_j\right).$$

Thus, inequality (1) follows immediately using (2) and the definition of *loss* (Definition 18).

Assume for the sake of contradiction that (3) does not hold. Then, since $B_{Bad(j)}^F \cup B_{Block(j)\setminus\{j\}}^{U\setminus F} \subseteq B_{Bad(j)}^F \cup B_{Block(j)}^{U\setminus F}$, by the definition of the *rank* function (Definition 3), there exists $e \in B_j^{U\setminus F}$ such that

$$e \notin Cl\left(B_{Bad(j)}^F \cup B_{Block(j)\setminus\{j\}}^{U\setminus F} \cup E_j \cup B_j^P\right). \tag{4}$$

By (2), since $e \in B_j^{U\setminus F}$, we get that $e \notin Cl\left(B_{Good(j)}^F\right)$ implies that $e \in E_j$. This contradicts (4). So it must be the case that

$$e \in Cl\left(B_{Good(j)}^F\right). \tag{5}$$

Consequently, e satisfies the conditions in Steps 2b and 2(b)i. We next prove that e also satisfies the condition in Step 2(b)iA. This completes the proof because it implies that $e \in B_j^P$, which contradicts (4).

Assume for the sake of contradiction that e does not satisfy the conditions in Step 2(b)iA. Hence, $e \in Cl\left(P \cup B_{Bad(j)}^F\right)$. Let C be a minimal subset of $P \cup B_{Bad(j)}^F$ such that $e \in Cl\left(C\right)$.

Suppose first that $C \subseteq Cl\left(B_{Good(j)}^F\right)$. Then, by the definition of a critical tuple (Definition 19), for every $e' \in C$, either $e' \in B_{Bad(j)}^F$ or $e' \in E_j \cup B_{Block(j)}^P$. Thus, $C \subseteq B_{Bad(j)}^F \cup E_j \cup B_{Block(j)}^P$. Consequently, because

$$B_{Bad(j)}^F \cup E_j \cup B_{Block(j)}^P \subseteq B_{Bad(j)}^F \cup B_{Block(j)\setminus\{j\}}^{U\setminus F} \cup E_j \cup B_j^P,$$

by Item 2 of Proposition 4,

$$e \in Cl\left(C\right) \subseteq Cl\left(B_{Bad(j)}^F \cup E_j \cup B_{Block(j)}^P\right) \subseteq Cl\left(B_{Bad(j)}^F \cup B_{Block(j)\setminus\{j\}}^{U\setminus F} \cup E_j \cup B_j^P\right).$$

This contradicts (4). Hence, we only need to deal with the case that $C \setminus Cl\left(B_{Good(j)}^F\right) \ne \emptyset$.

Suppose that $C \setminus Cl\left(B_{Good(j)}^F\right) \ne \emptyset$. By the definition of a critical tuple (Definition 19), $B_{Bad(j)}^F \subset B_{Good(j)}^F$ and therefore, $C \setminus Cl\left(B_{Good(j)}^F\right) \subseteq P$, because $C \subseteq P \cup B_{Bad(j)}^F$. Let e^* be the element from $C \setminus Cl\left(B_{Good(j)}^F\right)$ that was revealed last. For future reference, we note,

$$e^* \in P \tag{6}$$

Recall that, by construction, $e \in Cl\left(C\right)$, and C is minimal such that $e \in Cl\left(C\right)$, and $e \notin C$ because $e \notin P \cup B_{Bad(j)}^F$ and $C \subseteq P \cup B_{Bad(j)}^F$. Therefore, by Item 5 of Proposition 4,

$$e^* \in Cl\left(\left(C \setminus \{e^*\}\right) \cup \{e\}\right).$$

As a result, by (5) and Item 2 of Proposition 4,

$$e^* \in Cl\left(\left(C \setminus \{e^*\}\right) \cup \{e\} \cup B_{Good(j)}^F\right) = Cl\left(\left(\left(C \setminus B_{Good(j)}^F\right) \setminus \{e^*\}\right) \cup B_{Good(j)}^F\right).$$

16

Since $e^* \in P$, by Step 2b, we also have $\log val(e^*) \in \bigcup_{i \in \mathbb{Z}} Block(i)$, and hence by Observation 20, the fact that $\log val(e^*) \notin Good(j)$ implies that $B^F_{Good(j)} \subseteq B^F_{Bad(\log val(e'))}$. Thus, by Proposition 4.2,

$$e^* \in Cl\left(\left((C \setminus B^F_{Good(j)}) \setminus \{e^*\}\right) \cup B^F_{Good(j)}\right) \subseteq Cl\left(\left((C \setminus B^F_{Good(j)}) \setminus e^*\right) \cup B^F_{Bad(\log val(e^*))}\right).$$

Thus, because all the items in $C \setminus B^F_{Good(j)}$ were added to P before e^* was revealed, we conclude that e^* does not satisfy the condition in Step 2(b)iA. Hence $e^* \notin P$, which contradicts (6). ∎

5 Upper Bounding *loss*

Theorem 23 *For every finite $K \subset \mathbb{Z}$ and $k \in K$, if $rank(B_k) \geq 2^{64}$, then*

$$prob\left(loss\left(B^F_K, B^{U \setminus F}_k\right) \leq uncov\left(B^F_{K \setminus \{k\}}, B^F_k\right) + rank(B_k)^{\frac{2}{3}}\right) > 1 - e^{-rank(B_k)^{\frac{1}{6}}}.$$

We note that the upper bound, that this theorem provides, may be very far from being tight, yet since that only happens when *uncov* is very large, if it happens on a large scale, then the Simple Algorithm is sufficient for dealing with this case.

From here until the end of this subsection, K is a fixed finite subset of $\mathbb{Z}$, k is a fixed member of K and $m = |B_k|$.

The proof of Theorem 23 relies on two random variables Z and $\tilde{Z}$, which we define later on and use in order to construct the sets H and $\tilde{H}$ that are mentioned in Subsection 3.4 and satisfy $Z = |H|$ and $\tilde{Z} = |\tilde{H}|$. For the proof of Theorem 23 it is sufficient to know that these random variables satisfy the following lemmas, which we prove in Subsections 5.1 and 5.2.

Lemma 24 *For every $S \subseteq B_{K \setminus \{k\}}$ and $R \subseteq B_k$,* $Z(S, R) = uncov(S, R).$

Lemma 25 *For every $S \subseteq B_{K \setminus \{k\}}$ and $R \subseteq B_k$,* $\tilde{Z}(S, R) \geq loss(S \cup R, B_k \setminus R).$

Lemma 26 *If $rank(B_k) \geq 2^{64}$, then*

$$prob\left(\tilde{Z}(B^F_{K \setminus \{k\}}, B^F_k) \leq Z(B^F_{K \setminus \{k\}}, B^F_k) + m^{\frac{2}{3}}\right) > 1 - e^{-rank(B_k)^{\frac{1}{6}}}.$$

Proof of Theorem 23 By Lemma 24, $Z(B^F_{K \setminus \{k\}}, B^F_k) = uncov\left(B^F_{K \setminus \{k\}}, B^F_k\right)$ and by Lemma 25, $loss\left(B^F_K, B^{U \setminus F}_k\right) \leq \tilde{Z}(B^F_{K \setminus \{k\}}, B^F_k)$. Consequently, by Lemma 26 the theorem follows. ∎

The random variables $\tilde{Z}$ and Z are defined using an S, k-Closure-Read-Once-Tree, which is a vertex and edge labelled tree that encodes the fixed predetermined order, that is mentioned in Subsection 3.4, in which it is exposed whether each element in B_k is either in F or in $U \setminus F$. The definition of a S, k-Closure-Read-Once-Tree is based upon the definition of a k-Read-Once-Tree. The graph theoretic definitions and notations, that are used for formally defining a k-Read-Once-Tree, are provided in Appendix A.

Definition 27 *[k-Read-Once-Tree, Elements-Set and First-Elements-Set] A **k-Read-Once-Tree** is an arc and vertex labelled balanced-binary-tree such that:*

1. every internal vertex is labelled by an element of B_k and

2. every internal vertex has one outgoing-arc labelled 1 and the other labelled 0.

*The **element-set** of a vertex v of a k-Read-Once-Tree, denoted by T_v, is the set of all labels of vertices on the path from the root r to the parent of v. The **first-elements-set** of a vertex v of a k-Read-Once-Tree, denoted by $\hat{T}_v$, is the set of all labels of vertices $w \in T_v$ such that w's 1 labelled outgoing-arc is on the path from r to v.*

We next define, the specific type of k-Read-Once-Tree we call S, k-Closure-Read-Once-Tree. In an S, k-Closure-Read-Once-Tree the order in which the elements of B_k are exposed depends on the set S. Specifically, precedence is given to elements in B_k that are not in the closure of S union with the elements of B_k^F that have already been exposed.

Definition 28 *[S, k-**Closure-Read-Once-Tree**] For every $S \subseteq B_{K \setminus \{k\}}$, a S, k-**Closure-Read-Once-Tree**, denoted by CDT_k^S, is a fixed arbitrarily selected k-Read-Once-Tree such that,*

1. for every leaf v, $T_v^S = B_k$ and

2. for every internal vertex v, v's label is in $\left(B_k \setminus T_v^S \right) \setminus Cl \left(\hat{T}_v^S \cup S \right)$ if this set is not empty.

Observation 29 *For every $S \subseteq B_{K \setminus \{k\}}$, $\hat{T}_v^S$ is a bijection when viewed as a mapping from the leaves of CDT_k^S to 2^{B_k}.*

The path from r to a leaf v of CDT_k^S is sometimes referred to as the *path associated with the set $\hat{T}_v^S$*. Given that $B_k^F = R$, the order of the vertices in the path associated with R is the order they where exposed. For every internal vertex v on this path: $B_k \setminus T_v^S$ is the set of elements of B_k that were not exposed prior to the element labelling v; and $Cl \left(\hat{T}_v^S \cup S \right)$ is the closure of the union S of the vertices of B_k that have already been exposed to be in R prior to the element labelling v being exposed.

For every $S \subseteq B_{K \setminus \{k\}}$, $R \subseteq B_k$ and $i \in [m]$, we define the random variables $X_i(S, R)$, $Y_i(S, R)$, $Z_i(S, R)$ and $\tilde{Z}_i(S, R)$, for which we omit the parameters when clear from context, as follows: let

1. X_i be the label of the depth i arc in the path associated with R in CDT_k^S,

2. Y_i be 1 if the label of the vertex v at depth i, on the path associated with R in CDT_k^S, is not in $Cl \left(\hat{T}_v \cup S \right)$, and 0 otherwise, and $Y = \sum_{j=1}^m Y_j$,

3. $Z_i = Y_i \cdot X_i$, $\quad Z = \sum_{j \in [m]} Z_j$, $\quad$ and

4. $\tilde{Z}_i = Y_i \cdot (1 - X_i)$ $\quad$ and $\quad$ $\tilde{Z} = \sum_{j \in [m]} \tilde{Z}_j$.

We now define the sets H and $\tilde{H}$ that are mentioned in Subsection 3.4.

Definition 30 *[$H(S, R)$, $\tilde{H}(S, R)$ **and** $u_i^S(R)$] For every $S \subseteq B_{K \setminus \{k\}}$, $R \subseteq B_k$ and $i \in [m]$, let $u_i^S(R)$ be the vertex of depth i on the path associated with R in CDT_k^S and $e_i^S(R)$ be $u_i^S(R)$'s label. Let $H(S, R)$ be the set of all $e_i^S(R)$ such that $Z_i = 1$ and $\tilde{H}(S, R)$ be the set of all $e_i^S(R)$ such that $\tilde{Z}_i = 1$. We omit the parameters when clear from context.*

Observation 31 *For every $S \subseteq B_{K \setminus \{k\}}$ and $R \subseteq B_k$,*

1. $Z = |H|$, $\tilde{Z} = |\tilde{H}|$, $Y = Z + \tilde{Z}$, *and*

2. $Y = 1$ *for every* $i \in [Y]$ *and* $Y_i = 0$ *for every* $i \in (Y, m]$.

5.1 Proofs of Lemma 24 and Lemma 25

These proofs require the following proposition.

Proposition 32 *For every $S \subseteq B_{K \setminus \{k\}}$, $R \subseteq B_k$ and $i \in [m + 1]$,*

$$Cl\left(S \cup \hat{T}_{u_i}\right) \subseteq Cl\left(S \cup H\right) = Cl\left(S \cup R\right).$$

Proof. According to the definition of $\hat{T}_{u_i}$, we have that $\hat{T}_{u_{m+1}} = R$ and $\hat{T}_{u_i} \subseteq \hat{T}_{u_{i+1}} \subseteq R$, for every $i \in [m]$. Hence, $Cl\left(S \cup \hat{T}_{u_i}\right) \subseteq Cl\left(S \cup R\right)$, for every $i \in [m + 1]$. By the definition of H, $Cl\left(S \cup H\right) \subseteq Cl\left(S \cup R\right)$. So, now we only need to show that $Cl\left(S \cup R\right) \subseteq Cl\left(S \cup H\right)$.

According to construction, $X_j = 1$, for every $e_j \in R$. So, by the definition of H, $e_j \in H$, for every $e_j \in R$ such that $Y_j = 1$, and $e_j \in Cl\left(S \cup H\right)$ for every $e_j \in R$ such that $Y_j = 0$. Consequently, $Cl\left(S \cup R\right) \subseteq Cl\left(S \cup H\right)$. ∎

Lemma 24 (restated) *For every $S \subseteq B_{K \setminus \{k\}}$ and $R \subseteq B_k$, $Z = uncov\left(S, R\right)$.*

Proof. By definition, for every $e_i \in H$, $e_i \notin Cl\left(S \cup \hat{T}_{u_i}\right)$. Therefore, by Item 3 of Proposition 4, for every $e_i \in H$,

$$rank\left(S \cup \hat{T}_{u_i} \cup \{e_i\}\right) = rank\left(S \cup \hat{T}_{u_i}\right) + 1.$$

Hence, inductively,

$$rank\left(S \cup R\right) = rank\left(S \cup \hat{T}_{u_{m+1}}\right) = rank\left(S\right) + |H|.$$

Now since, by Proposition 32, $rank\left(S \cup H\right) = rank\left(S \cup R\right)$, according to the definition of *uncov* (Definition 15),

$$|H| = rank\left(S \cup R\right) - rank\left(S\right) = uncov\left(S, R\right).$$

The lemma follows from the preceding equality, because $Z = |H|$, by Observation 31. ∎

Lemma 25 (restated) *For every $S \subseteq B_{K \setminus \{k\}}$ and $R \subseteq B_k$, $\tilde{Z} \geq loss\left(S \cup R, B_k \setminus R\right)$.*

Proof. We observe that, for every $e_i \in (B_k \setminus R) \setminus \tilde{H}$ both $X_i = 0$ and $Y_i = 0$ and hence

$$e_i \in Cl\left(S \cup \hat{T}_{u_i}\right).$$

As a result, by Proposition 32, $e_i \in Cl\left(S \cup R\right)$, for every $e_i \in (B_k \setminus R) \setminus \tilde{H}$. Thus, $(B_k \setminus R) \setminus \tilde{H} \subseteq Cl\left(S \cup R\right)$ and therefore, $(B_k \setminus R) \setminus Cl\left(S \cup R\right) \subseteq \tilde{H}$. Now, since $\tilde{Z} = |\tilde{H}|$, by Item 1 of Proposition 4 and the definition of *loss* (Definition 18),

$$\tilde{Z} = |\tilde{H}| \geq |(B_k \setminus R) \setminus Cl\left(S \cup R\right)| \geq rank\left((B_k \setminus R) \setminus Cl\left(S \cup R\right)\right) = loss\left(S \cup R, B_k \setminus R\right).$$

19

5.2 Proof of Lemma 26

Proposition 33 *For every $i \in [m]$ independently, $X_i(B_{K\setminus\{k\}}^F, B_k^F) = 1$ with probability $1/2$.*

Proof. Let $i \in [m]$. By Observation 29, for every $S \subseteq B_k$, there exists a distinct leaf v in CDT_k^S such that $S = \hat{T}_v$. Therefore, by the definition of X_i, to prove the proposition we only need to show that, with probability exactly $1/2$, a path from the root of CDT_k^S to a uniformly at random selected leaf has a depth i arc labeled 1.

By Observation 1, F is uniformly distributed over the subsets of U and hence, by construction, B_k^F is uniformly distributed over the subsets of B_k. By definition, every depth i vertex of CDT_k^S has one outgoing-arc labeled 1 and the other labeled 0. Hence, half of the depth i arcs of CDT_k^S are labeled 1 and the other half are labeled by 0. Thus, since CDT_k^S is a balanced-binary tree, a path from the root of CDT_k^S to a uniformly selected at random leaf has a depth i arc labeled 1, with probability exactly $1/2$. ∎

Lemma 26 (restated) *If $rank\left(B_k\right) \geq 2^{64}$, then*

$$prob\left(\tilde{Z}(B_{K\setminus\{k\}}^F, B_k^F) \leq Z(B_{K\setminus\{k\}}^F, B_k^F) + m^{\frac{2}{3}}\right) > 1 - e^{-rank(B_k)^{\frac{1}{6}}}.$$

Proof. By Observation 31, $\tilde{Z} = Y - Z$ and hence,

$$prob\left(\tilde{Z}(B_{K\setminus\{k\}}^F, B_k^F) > Z(B_{K\setminus\{k\}}^F, B_k^F) + m^{\frac{2}{3}}\right) \leq \sum_{i\in[m]} prob\left(Z < \frac{i - m^{\frac{2}{3}}}{2}\Big|Y = i\right) \cdot prob\left(Y = i\right).$$

and therefore,

$$prob\left(\tilde{Z}(B_{K\setminus\{k\}}^F, B_k^F) > Z(B_{K\setminus\{k\}}^F, B_k^F) + m^{\frac{2}{3}}\right) \leq m \cdot \max_{i\in[m]} prob\left(Z < \frac{i - m^{\frac{2}{3}}}{2}\Big|Y = i\right).$$

We shall prove that, for every $i \in [m]$,

$$m \cdot prob\left(Z < \frac{i - m^{\frac{2}{3}}}{2}\Big|Y = i\right) < e^{-rank(B_k)^{\frac{1}{6}}}, \tag{7}$$

which by the preceding inequalities, implies the lemma. According, to Item 2 of Observation 31, if $Y = i$ then, according to the construction of Z, $Z_j = 0$, for every $j > i$. Consequently, according to the construction of X, Y and Z, the event $Z < \frac{i-m^{\frac{2}{3}}}{2}$, given $Y = i$, occurs only if among $X_1, X_2, \ldots, X_i$, at most $\frac{i-m^{\frac{2}{3}}}{2}$ have a value of 1. Hence,

$$prob\left(Z < \frac{i - m^{\frac{2}{3}}}{2}\Big|Y = i\right) \leq prob\left(\sum_{j=1}^{i} X_j < \frac{i - m^{\frac{2}{3}}}{2}\right).$$

By Proposition 33 and the Chernoff bound,

$$prob\left(\sum_{j=1}^{i} X_j < \frac{i - m^{\frac{2}{3}}}{2}\right) \leq e^{-\frac{(m^{\frac{2}{3}})^2}{2i}}.$$

Since, $m = rank\,(B_k)$, $i \leq m$ and $rank\,(B_k) \geq 2^{64}$, for every $i \in [m]$, $m \cdot e^{-\frac{(m^{\frac{2}{3}})^2}{2i}} < e^{-rank(B_k)^{\frac{1}{6}}}$. Thus, (7) indeed holds. ∎

6 Structural Theorem

In this section F is used after all its elements have been revealed and hence it is treated as fixed. We next formally define the strong sequence that was mentioned in Subsection 3.5.

Definition 34 [Strong sequence] *Let $K \subset \mathbb{Z}$. A sequence H of integers $h_1, h_2, \ldots, h_k$ is a **strong sequence** for B_K^F, if*

1. $h_1, h_2, \ldots, h_k \in K$,

2. H is strictly monotonically decreasing,

3. $LOPT\left(B_H^F\right) \geq \frac{1}{18} \cdot LOPT\left(B_K^F\right)$ and

4. for every $j \in [k-1]$, $0 < rank\left(B_{h_j}^F\right) \leq \frac{1}{32} \cdot rank\left(B_{h_{j+1}}^F\right)$.

Lemma 35 *For every $F \subseteq U$ and $K \subset \mathbb{Z}$, there exists a strong sequence for B_K^F.*

The proof of the preceding lemma is in Appendix C. The observation is a direct result of the definition of *uncov* (Definition 15), the definition of a strong sequence (Definitions 34) and Item 4 of Proposition 4.

Observation 36 *Let $K \subset \mathbb{Z}$, $F \subseteq U$ and H be a strong sequence for B_K^F, $\ell \in H^* \subseteq H' \subset H$ and $j \in H$ such that $j < H^*$, then*

1. $2 \cdot rank\left(B_{\min H'}^F\right) > rank\left(B_{H'}^F\right)$,

2. $uncov\left(B_{H^* \setminus \{\ell\}}^F, B_\ell^F\right) \geq uncov\left(B_{H' \setminus \{\ell\}}^F, B_\ell^F\right)$,

3. $uncov\left(B_{H^* \setminus \{j\}}^F, B_j^F\right) \geq \frac{31}{32} \cdot rank\left(B_j^F\right)$ and

4. $2^j \cdot uncov\left(B_{H^* \setminus \{j\}}^F, B_j^F\right) \geq \frac{31}{32} \cdot LOPT\left(B_j^F\right)$.

We now formally define the Partition that was mentioned in Subsection 3.5.

Definition 37 [Partition] *Let $K \subset \mathbb{Z}$, $F \subseteq U$ and H a sequence of integers $h_1, h_2, \ldots, h_k$ that is a strong sequence for B_K^F. We define $Partition(H)$ to be an arbitrary partition of H into subsequences $H_1, H_2, \ldots, H_g$ such that:*

1. $g \leq \max \left\{ 1, 16 \cdot \log \log rank \left(B_K^F \right) \right\}$

2. for every $i \in [g-1], \quad H_i > H_{i+1}$,

3. for every $i \in [g], rank \left(B_{\max H_i}^F \right) \geq rank \left(B_{\min H_i}^F \right)^{\frac{11}{12}}$.

The next observation follows from Item 1 of Observation 36 and the above definition.

Observation 38 *Let* $K \subset \mathbb{Z}$, $F \subseteq U$ *and* H *a sequence of integers* $h_1, h_2, \ldots, h_k$ *that is a strong sequence for* B_K^F. *For every* $H_i \in Partition(H)$ *and* $\ell \in H_i$, *we have* $2 \cdot rank \left(B_\ell^F \right)^{\frac{12}{11}} \geq rank \left(\bigcup_{j \in [i]} B_{H_j}^F \right)$.

Lemma 39 *For every,* $F \subseteq U$, $K \subset \mathbb{Z}$ *and a sequence* H *of integers* $h_1, h_2, \ldots, h_k$ *that is strong sequence for* B_K^F, $Partition(H)$ *is well defined.*

Proof.   Let $\ell_1 = h_1$ and r_1 be the minimum member of K such that $rank \left(B_{\ell_1}^F \right) \geq rank \left(B_{r_1}^F \right)^{\frac{11}{12}}$ and set $H_1 = [\ell_1, r_1] \cap H$. Now, inductively, for every $i > 1$ let ℓ_i be the maximum member of K that is smaller than r_{i-1}, and r_i be the minimum member of H such that $rank \left(B_{\ell_i}^F \right) \geq rank \left(B_{r_i}^F \right)^{\frac{11}{12}}$ and $H_i = [\ell_i, r_i] \cap H$. Let g be the maximum integer for which H_g is defined.

We observe that, by construction, the sets $H_1, H_2, \ldots, H_g$ satisfy **Items 2 and 3** of Definition 37.

If $g = 1$, then **Item 1** of Definition 37 trivially holds and the lemma follows. So, assume that $g > 1$. By definition, $rank \left(B_{\ell_i}^F \right) < rank \left(B_{\ell_{i+1}}^F \right)^{\frac{11}{12}}$, for every $i \in [g-1]$, and hence

$$rank \left(B_K^F \right) \geq rank \left(B_{\ell_g}^F \right) > rank \left(B_{\ell_2}^F \right)^{\left(\frac{12}{11} \right)^{g-2}}.$$

Since ℓ_1 and ℓ_2 are members of the strong sequence H and $\ell_2 < \ell_1$, by Item 4 of Definition 34, $rank \left(B_{\ell_2}^F \right) \geq 32 \cdot rank \left(B_{\ell_1}^F \right) \geq 32$, and hence the preceding inequality implies that indeed $g \leq 16 \cdot \log \log rank \left(B_K^F \right)$, that is, **Item 1** of Definition 37 holds and the lemma follows. ∎

We next define formally the three types of sets that where mentioned in Subsection 3.5: useful, negligible and splittable. The succeeding definition is essential for this goal.

Definition 40 *[M_H] Let* H *be a strong sequence. For every* $K \subseteq H$, *we define*

$$M_H(K) = \{ i \in H \mid i > K \}.$$

We omit the subscript when clear from context.

Definition 41 *[**useful**] A subset* K^* *of* $K \subseteq H$ *is* **useful** *for* K *if the following hold:*

1. $LOPT \left(B_{K^*}^F \right) > \frac{1}{32} \cdot LOPT \left(B_K^F \right)$

2. $\sum_{j \in K^*} 2^j \cdot \left(uncov \left(B_{M(K^*) \cup K^* \setminus \{j\}}^F, B_j^F \right) - uncov \left(B_{M(K) \cup K \setminus \{j\}}^F, B_j^F \right) \right) \geq \frac{LOPT \left(B_{K^*}^F \right)}{2^{14} \cdot \log \log rank(F)}$.

*A set that has a useful subset is **useful**.*

Definition 42 [negligible] *A set $K \subseteq H$ is **negligible** if the following hold:*

1. $\sum_{j \in K} 2^j \cdot uncov\left(B^F_{M(K) \cup K \setminus \{j\}}, B^F_j\right) > \frac{7}{8} \cdot LOPT\left(B^F_K\right)$ *or*

2. $LOPT\left(B^F_K\right) < \frac{LOPT\left(B^F_H\right)}{64 \cdot \log rank(F)}.$

Definition 43 [splittable] *$K \subseteq H$ is **splittable** if it has a partition $\{K_1, K_2\}$ such that*

1. $K_1 > K_2$ *and*

2. $LOPT\left(B^F_{K_i}\right) > \frac{1}{32} \cdot LOPT\left(B^F_K\right)$, *for every $i \in [2]$.*

Proposition 44 *Every set is $K \subseteq H$ is at least one of the following: useful, negligible and splittable.*

Proof. Let $K \subseteq H$. Recall that H is a sequence of integers and for every $K \subseteq H$, we have that $M(K)$ is a set of integers in H.

Fix $K \subseteq H$ and assume that K is neither negligible nor splittable. We show next that this implies that K is useful, which in turn implies the proposition.

Since K is not splittable, there exists $\gamma \in K$ that satisfies the following inequality, since otherwise, by the definition of splittable (Definition 43), a contradiction is reached.

$$LOPT\left(B^F_K\right) < \frac{16}{15} \cdot LOPT\left(B^F_\gamma\right). \tag{8}$$

Consequently, since K is not negligible, by the definition of *negligible* (Definition 42),

$$\sum_{j \in K} 2^j \cdot uncov\left(B^F_{M(K) \cup K \setminus \{j\}}, B^F_j\right) \le \frac{7}{8} \cdot LOPT\left(B^F_K\right) \le \frac{14}{15} \cdot LOPT\left(B^F_\gamma\right)$$

and hence,

$$2^\gamma \cdot uncov\left(B^F_{M(K) \cup K \setminus \{\gamma\}}, B^F_\gamma\right) \le \sum_{j \in K} 2^j \cdot uncov\left(B^F_{M(K) \cup K \setminus \{j\}}, B^F_j\right) \le \frac{14}{15} \cdot LOPT\left(B^F_\gamma\right). \tag{9}$$

Let $K^* = \{\gamma\}$. By the definition of M (Definition 40), $\gamma < M(K^*)$ and therefore, since also $M(K^*) \cup K^* \subseteq H$, by Item 4 of Observation 36,

$$2^\gamma \cdot uncov\left(B^F_{M(K^*) \cup K^* \setminus \{\gamma\}}, B^F_\gamma\right) \ge \frac{31}{32} \cdot LOPT\left(B^F_{K^*}\right).$$

Thus, together with Inequality (9) implies,

$$\sum_{j \in K^*} 2^j \cdot \left(uncov\left(B^F_{M(K^*) \cup K^* \setminus \{j\}}, B^F_j\right) - uncov\left(B^F_{M(K) \cup K \setminus \{j\}}, B^F_j\right)\right) \ge \frac{LOPT\left(B^F_{K^*}\right)}{2^{14} \cdot \log\log rank(F)}.$$

23

Thus, by (8) and the definition of useful (Definition 41), K^* is useful for K and hence, K is useful. This, concludes the proof. ∎

The following proposition bounds the impact of 'replacing' splittable as mentioned in Subsection 3.5.

Proposition 45 *Let $K \subseteq H$ be splittable with partition $\{K_1, K_2\}$ and not useful, then*

$$\sum_{i \in [2]} \sum_{j \in K_i} 2^j \cdot uncov\left(B^F_{M(K_i) \cup K_i \setminus \{j\}}, B^F_j\right) < \sum_{j \in K} 2^j \cdot uncov\left(B^F_{M(K) \cup K \setminus \{j\}}, B^F_j\right) + \frac{LOPT\left(B^F_K\right)}{2^{14} \cdot \log \log rank\left(F\right)}.$$

Proof. Since K is splittable, by the definition of *splittable* (Definition 43), $LOPT\left(B^F_{K_i}\right) > \frac{1}{32} \cdot LOPT\left(B^F_K\right)$, for every $i \in [2]$. Hence, because K is not useful, by the definition of useful (Definition 41),

$$\sum_{i \in [2]} \sum_{j \in K_i} 2^j \cdot \left(uncov\left(B^F_{M(K_i) \cup K_i \setminus \{j\}}, B^F_j\right) - uncov\left(B^F_{M(K) \cup K \setminus \{j\}}, B^F_j\right)\right) < \sum_{i \in [2]} \frac{LOPT\left(B^F_{K_i}\right)}{2^{14} \cdot \log \log rank\left(F\right)}.$$

Since $\{K_1, K_2\}$ is a partition of K, by Observation 10, $LOPT\left(B^F_K\right) = \sum_{i \in [2]} LOPT\left(B^F_{K_i}\right)$ and therefore, by the preceding inequality, the proposition follows. ∎

Remark: This algorithm is used later on in the proof of Theorem 54. There it is ensured that the input parameter Λ is not empty and hence we assume that indeed Λ is not empty. The output of the algorithm $(Block, Good, Bad)$ is such that, $Block(i) = Good(i) = Bad(i) = \emptyset$, for every $i \notin \bigcup_{j \in \mathbb{Z}} Block(j)$. In addition, it is important to note, that for accounting reasons the algorithm uses the sets $\mathcal{Q}_i$ instead of a single set and that if this was not done, then it would follow the description in Subsection 3.5. From here until the end of this section we assume that, H is as defined in Algorithm 3, $(Block, Good, Bad)$ was the output of Algorithm 3. We define d to be one less than the maximum value the index ℓ reaches when Algorithm 3 halts. Lemma 47 below show that Algorithm 3 always halts.

Proposition 46 *On input F and $\Lambda \subseteq Valuable$, for every finite value of ℓ reached by the execution of Algorithm 3, the sets in $\mathcal{Q}_\ell$ are pairwise comparable and in particular pairwise disjoint.*

Proof. According to Step 2, $\mathcal{Q}_0 = \emptyset$, and hence the sets in $\mathcal{Q}_0$ trivially satisfy the conditions of the proposition. According to Step 2, $\mathcal{Q}_1 = Partition(H)$. Thus, the sets in $\mathcal{Q}_1$ satisfy the conditions of the proposition, by the definition of $Partition(H)$ (Definition 37). Assume by induction that the conditions of the proposition hold for $\mathcal{Q}_{\ell-1}$, where $\ell \geq 2$. We note that, by Step 3b, $\mathcal{Q}_\ell$ is initially empty and set are added to $\mathcal{Q}_\ell$ only in Steps 3(c)ii and 3(c)iii. In addition sets are added to $\mathcal{Q}_\ell$ only as a result of every set in $\mathcal{Q}_{\ell-1}$ being examined individually and for each $K \in \mathcal{Q}_{\ell-1}$ examined exactly one of the following happens: (i) nothing happens; (ii) K is added to $\mathcal{Q}_\ell$ and (iii) the set in a partition of K into pairwise-comparable sets are added to $\mathcal{Q}_\ell$. Thus, it is easy to see that, since all the sets $\mathcal{Q}_{\ell-1}$ are pairwise comparable and in particular pairwise disjoint the same holds for all the sets in $\mathcal{Q}_{\ell-1}$. Therefore, the induction step holds and the proposition follows. ∎

Algorithm 3 critical tuple Algorithm

Input: F **and** $\Lambda \subseteq Valuable$

1. $H \longleftarrow$ strong sequence for B_Λ^F

2. $\ell \longleftarrow 1$, $\mathcal{Q}_0 \longleftarrow \emptyset$, $\mathcal{Q}_1 \longleftarrow Partition(H)$

3. while $\mathcal{Q}_\ell \neq \mathcal{Q}_{\ell-1}$, **do**

   (a) $\ell \longleftarrow \ell + 1$

   (b) $\mathcal{Q}_\ell \longleftarrow \emptyset$

   (c) **for each** $K \in \mathcal{Q}_{\ell-1}$, **do**

       i. **if** K is negligible, **do nothing**
       ii. **else, if** K is useful, **do** $\mathcal{Q}_\ell \longleftarrow \mathcal{Q}_\ell \cup \{K\}$
       iii. **else, if** K is splittable with partition $\{K_1, K_2\}$, **do** $\mathcal{Q}_\ell \longleftarrow \mathcal{Q}_\ell \cup \{K_1, K_2\}$

4. **for each** $K \in \mathcal{Q}_{\ell-1}$, **do**

   (a) **if** there exists a set K^* that is useful for K, **do**

       i. **for each** $j \in K^*$, **do**
          A. $Block(j) \longleftarrow K^*$
          B. $Bad(j) \longleftarrow M(K^*)$
          C. $Good(j) \longleftarrow Bad(j) \cup K$

Output: $(Block, Good, Bad)$

Lemma 47 *On input F and $\Lambda \subseteq Valuable$, Algorithm 3 satisfies the following:*

1. *it halts,*

2. *all the sets in $\mathcal{Q}_d$ are useful,*

3. *the output it returns $(Block, Good, Bad)$ is a critical tuple,*

4. *for every $i \in \bigcup_{j \in \mathbb{Z}} Block(j)$, we have that $Good(i)$, $Bad(i)$ and $Block(i)$ are all subsets of $Valuable$ and $Block(i)$ is a subset of a set in $Partition(H)$, and*

5. *$d \leq 2^8 + 2^5 \cdot \log\log rank\,(F)$.*

Proof.   We start by proving that Items 1 and 2. For every j such that $\mathcal{Q}_j$ is defined, let m_j be the maximal size of a set in $\mathcal{Q}_j$ that is splittable but neither useful nor negligible, if such a set exists and 0 otherwise. By Steps 3(c)i, 3(c)ii and 3(c)iii, as set in $\mathcal{Q}_{j+1}$ is splittable but neither useful nor negligible only if it is a strict subset of a set in $\mathcal{Q}_j$ that is splittable but neither useful nor negligible. Consequently, $m_{j+1} < \min\{2, m_j\}$, or every j such that $\mathcal{Q}_{j+1}$ is defined. We note that, by the definition of *splittable* (Definition 43), $m_j > 1$, for every j such that $\mathcal{Q}_j$ is defined. Hence, by the above there exists a finite i such that every set in $\mathcal{Q}_i$ is useful. Thus, by the preceding case Algorithm 3 halts and therefore, **Item 1** holds, d is finite, and by the above three cases all the sets in $\mathcal{Q}_d$ are useful. Consequently, **Item 2** also holds.

By Proposition 46, all the sets in $\mathcal{Q}_d$ are pairwise-comparable and hence, by all the Steps in the "for each" of Step 4, $(Block, Good, Bad)$ is a critical tuple. Thus, **Item 3** holds. By Steps 1, 3(c)i, 3(c)ii and 3(c)iii, **Item 4** holds. We next prove that Item 5 holds.

Let $i \geq 4$ be an integer not exceeding d and suppose that $K'_i \in \mathcal{Q}_i$ is splittable and neither negligible nor useful. Then, by Steps 3(c)i, 3(c)ii and 3(c)iii, there exists a set $K'_{i-1} \in \mathcal{Q}_{i-1}$ that is a superset of K'_i and, for every $j \in [i-1]$ inductively, there exists a set $K'_j \in \mathcal{Q}_j$ that is a superset of K'_{j+1}. We note that, for every $j \in [i-1]$, K'_j splittable and neither negligible nor useful since otherwise, either $K'_i \notin \mathcal{Q}_i$ or K'_i satisfies at least one of the conditions of Steps 3(c)i and 3(c)ii which is a contradiction. Consequently, by the definition of splittable (43),

$$LOPT\left(B^F_{K'_i}\right) \leq \left(\frac{15}{16}\right)^{i-1} \cdot LOPT\left(B^F_{K'_1}\right) \leq \left(\frac{15}{16}\right)^{i-1} \cdot LOPT\left(B^F_H\right) \tag{10}$$

Since K'_{i-1} is not negligible, by the definition of *negligible* (Definition 42),

$$\frac{LOPT\left(B^F_H\right)}{64 \cdot \log rank\,(F)} \leq LOPT\left(B^F_{K'_{i-1}}\right).$$

This together with (10), implies that

$$\frac{LOPT\left(B^F_H\right)}{64 \cdot \log rank\,(F)} \leq \left(\frac{15}{16}\right)^{i-4} \cdot LOPT\left(B^F_H\right).$$

Hence, $i - 1 \leq 2^8 - 4 + 2^5 \cdot \log\log rank\,(F)$. Thus, for any $i > 2^8 + 2^5 \cdot \log\log rank\,(F) - 3$, $\mathcal{Q}_i$ does not have any splittable set. Consequently, by Steps 3(c)i and 3(c)ii, $\mathcal{Q}_i = \mathcal{Q}_{i+1}$, for some

$i \leq 2^8 + 2^5 \cdot \log \log rank(F) - 1$. By Step 3, this implies that $d \leq 2^8 + 2^5 \cdot \log \log rank(F)$ when Algorithm 3 halts. Thus, **Item 5** holds. ∎

Lemma 48 *On input F and $\Lambda \subseteq Valuable$, Algorithm 3 satisfies the following: H computed by Algorithm 3 is a strong sequence for B_Λ^F, and **if***

$$\text{for every } H_i \in Partition(H), \quad \sum_{j \in H_i} 2^j \cdot uncov\left(B_{H_i \setminus \{j\}}^F, B_j^F\right) \leq \frac{LOPT\left(B_H^F\right)}{2^{14} \cdot \log \log rank(F)}, \quad (11)$$

then Algorithm 3 returns a critical tuple $(Block, Good, Bad)$ such that

$$\sum_{j \in \mathbb{Z}} 2^j \cdot \left(uncov\left(B_{Bad(j) \cup Block(j) \setminus \{j\}}^F, B_j^F\right) - uncov\left(B_{Good(j) \setminus \{j\}}^F, B_j^F\right)\right) \geq \frac{LOPT\left(B_H^F\right)}{2^{20} \cdot \log \log rank(F)}.$$

Proof. By Step 1, H computed by Algorithm 3 is a strong sequence for B_Λ^F. By Lemma 47, on input F and Λ, d is a finite integer and Algorithm 3 halts and returns a critical tuple $(Block, Good, Bad)$. Suppose first that,

$$\sum_{K \in \mathcal{Q}_d} LOPT\left(B_K^F\right) \geq \frac{1}{2} \cdot LOPT\left(B_H^F\right). \quad (12)$$

By Lemma 47, all the sets in $\mathcal{Q}_d$ are *useful* and therefore, according to the steps executed in the "for each" of Step 4 and the definition of useful (Definition 41),

$$\sum_{K \in \mathcal{Q}_d} \sum_{j \in K} 2^j \cdot \left(uncov\left(B_{Bad(j) \cup Block(j) \setminus \{j\}}^F, B_j^F\right) - uncov\left(B_{Good(j) \setminus \{j\}}^F, B_j^F\right)\right) \geq \frac{1}{32} \sum_{K \in \mathcal{Q}_d} \frac{LOPT\left(B_K^F\right)}{2^{14} \cdot \log \log rank(F)}$$

and hence, by (12) the lemma follows. We now prove that (11) implies (12).

Let $\mathcal{Q}^{small}$ be the family of all the negligible sets in $\bigcup_{i \in [d]} \mathcal{Q}_i$, that satisfy Item 2 of Definition 42, and $\mathcal{Q}^{uncov}$ be the family of all the negligible sets in $\bigcup_{i \in [d]} \mathcal{Q}_i$, that are not in $\mathcal{Q}^{small}$. According to construction:

$$\sum_{K \in \mathcal{Q}_d} LOPT\left(B_K^F\right) = \sum_{K \in \mathcal{Q}_1} LOPT\left(B_K^F\right) - \sum_{K \in \mathcal{Q}^{uncov}} LOPT\left(B_K^F\right) - \sum_{K \in \mathcal{Q}^{small}} LOPT\left(B_K^F\right). \quad (13)$$

We will individually bound each sum in the preceding equality, so that (12) follows.
According to Step 2 of Algorithm 3, the definition of $Partition(H)$ (Definition 37) and the definition of $LOPT$ (Definition 9),

$$\sum_{K \in \mathcal{Q}_1} LOPT\left(B_K^F\right) = \sum_{K \in Partition(H)} LOPT\left(B_K^F\right) = LOPT\left(B_H^F\right). \quad (14)$$

Since $H \subseteq Valuable$, by Observation 17, we have $|H| \le 2 \log rank\,(F)$ and therefore

$$\sum_{K \in \mathcal{Q}^{small}} LOPT\left(B_K^F\right) \le |H| \cdot \frac{LOPT\left(B_H^F\right)}{64 \cdot \log rank\,(F)} \le \frac{1}{32} \cdot LOPT\left(B_H^F\right). \tag{15}$$

We next prove the succeeding inequality which, by (13), (14), (15) and (16), implies that (12) holds.

$$\sum_{K \in \mathcal{Q}^{uncov}} LOPT\,(K) \le \frac{1}{14} \cdot LOPT\left(B_H^F\right). \tag{16}$$

For every $i \in [d]$, let $\mathcal{Q}_i' = \mathcal{Q}_i \cup \left(\mathcal{Q}^{uncov} \cap \bigcup_{j \in [i-1]} \mathcal{Q}_j\right)$. By Lemma 47, $\mathcal{Q}^{uncov} \cap \mathcal{Q}_d = \emptyset$ and hence, $\mathcal{Q}^{uncov} \subseteq \mathcal{Q}_d'$. As a result,

$$\sum_{K \in \mathcal{Q}^{uncov}} \sum_{j \in K} 2^j \cdot uncov\left(B_{M(K) \cup K \setminus \{j\}}^F, B_j^F\right) \le \sum_{K \in \mathcal{Q}_d'} \sum_{j \in K} 2^j \cdot uncov\left(B_{M(K) \cup K \setminus \{j\}}^F, B_j^F\right).$$

By the definition of negligible (Definition 42), this implies that,

$$\sum_{K \in \mathcal{Q}^{uncov}} LOPT\left(B_K^F\right) \le \frac{8}{7} \cdot \sum_{K \in \mathcal{Q}_d'} \sum_{j \in K} 2^j \cdot uncov\left(B_{M(K) \cup K \setminus \{j\}}^F, B_j^F\right). \tag{17}$$

We prove next that

$$\frac{8}{7} \cdot \sum_{K \in \mathcal{Q}_d'} \sum_{j \in K} 2^j \cdot uncov\left(B_{M(K) \cup K \setminus \{j\}}^F, B_j^F\right) \le \frac{1}{14} \cdot LOPT\left(B_H^F\right), \tag{18}$$

which, by (17), implies that (16) indeed holds. By Item 2 of Observation 36,

$$\sum_{K \in \mathcal{Q}_1} \sum_{j \in K} 2^j \cdot uncov\left(B_{M(K) \cup K \setminus \{j\}}^F, B_j^F\right) \le \sum_{K \in \mathcal{Q}_1} \sum_{j \in K} 2^j \cdot uncov\left(B_{K \setminus \{j\}}^F, B_j^F\right).$$

Hence, by (11),

$$\sum_{K \in \mathcal{Q}_1} \sum_{j \in K} 2^j \cdot uncov\left(B_{M(K) \cup K \setminus \{j\}}^F, B_j^F\right) \le |Partition(H)| \cdot \frac{LOPT\left(B_H^F\right)}{2^{14} \cdot \log \log rank\,(F)}.$$

By Step 2 of Algorithm 3 and the definition of $Partition(H)$ (Definition 37), $|\mathcal{Q}_1| = |Partition(H)| \le 16 \cdot \log \log rank\left(B_K^F\right)$, and hence, by the preceding inequality,

$$\sum_{K \in \mathcal{Q}_1} \sum_{j \in K} 2^j \cdot uncov\left(B_{M(K) \cup K \setminus \{j\}}^F, B_j^F\right) \le \frac{1}{2^{10}} \cdot LOPT\left(B_H^F\right). \tag{19}$$

By Proposition 45 and Steps 3(c)i, 3(c)ii and 3(c)iii, for every $i \in [2, d]$,

$$\sum_{K \in \mathcal{Q}'_{\mathbf{i}}} \sum_{j \in K} 2^j \cdot uncov\left(B^F_{M(K) \cup K \setminus \{j\}}, B^F_j\right) \leq \sum_{K \in \mathcal{Q}'_{\mathbf{i-1}}} \sum_{j \in K} 2^j \cdot uncov\left(B^F_{M(K) \cup K \setminus \{j\}}, B^F_j\right) + \frac{LOPT\left(B^F_H\right)}{2^{14} \cdot \log \log rank\,(F)}.$$

By resolving the recurrence we get,

$$\sum_{K \in \mathcal{Q}'_d} \sum_{j \in K} 2^j \cdot uncov\left(B^F_{M(K) \cup K \setminus \{j\}}, B^F_j\right) \leq \sum_{K \in \mathcal{Q}_1} \sum_{j \in K} 2^j \cdot uncov\left(B^F_{M(K) \cup K \setminus \{j\}}, B^F_j\right) + \frac{(d-1) \cdot LOPT\left(B^F_H\right)}{2^{14} \cdot \log \log rank\,(F)}$$

Consequently, inequality (18) follows, by (19) and Item 5 of Lemma 47. ∎

Theorem 49 *Let $\Lambda \subseteq Valuable$, if $rank\,(F) > 2^8$, then one of the following exists:*

1. *a set $J \subseteq \Lambda$, where $\sum_{j \in J} 2^j \cdot uncov\left(B^F_{J \setminus \{j\}}, B^F_j\right) \geq \frac{LOPT(B^F_\Lambda)}{2^{19} \cdot \log \log rank(F)}$ and for every $j \in J$, we have that $2 \cdot rank\left(B^F_j\right)^{\frac{12}{11}} \geq rank\left(B^F_J\right)$, or*

2. *a critical tuple $(Block, Good, Bad)$, where for every $i \in \mathbb{Z}$, we have that $Good(i)$, $Bad(i)$ and $Block(i)$ are subsets of Λ and $2 \cdot rank\left(B^F_i\right)^{\frac{12}{11}} \geq rank\left(B^F_{Good(i)}\right)$, and*

$$\sum_{j \in \mathbb{Z}} 2^j \cdot \left(uncov\left(B^F_{Bad(j)} \cup B^F_{Block(j) \setminus \{j\}}, B^F_j\right) - uncov\left(B^F_{Good(j) \setminus \{j\}}, B^F_j\right)\right) \geq \frac{LOPT\left(B^F_\Lambda\right)}{2^{25} \cdot \log \log rank\,(F)}.$$

Proof.   Suppose that there exists $J \in Partition(H)$, where H be a strong sequence for B^F_Λ, such that $\sum_{j \in J} 2^i \cdot uncov\left(B^F_{J \setminus \{j\}}, B^F_j\right) \geq \frac{LOPT(B^F_H)}{2^{20} \cdot \log \log rank(F)}$. Then, by Item 3 of the definition of a strong sequence (Definition 34), the first equation of the theorem holds for J. By Observation 38, J satisfies the condition of the first item of the theorem. So, suppose that a set as J does not exist. Then, by Lemma 48 and Item 3 of the definition of a strong sequence (Definition 34), on input F, Algorithm 3 returns a critical tuple $(Block, Good, Bad)$, such that

$$\sum_{j \in \mathbb{Z}} 2^j \cdot \left(uncov\left(B^F_{Bad(j)} \cup B^F_{Block(j) \setminus \{j\}}, B^F_j\right) - uncov\left(B^F_{Good(j) \setminus \{j\}}, B^F_j\right)\right) \geq \frac{LOPT\left(B^F_\Lambda\right)}{2^{25} \cdot \log \log rank\,(F)}.$$

By Item 4 of Lemma 47, for every $i \in \mathbb{Z}$, we have that $Good(i)$, $Bad(i)$ and $Block(i)$ are subsets of Λ and in $Block(i)$ is a subset of a set in $Partition(H)$. Thus, by Observation 38, $(Block, Good, Bad)$ satisfies the second item of theorem and the whole theorem follows. ∎

7   Main Result

The proof of the main result combines the algorithmic, probabilistic and structural theorems. We start by combining the probabilistic and structural theorems. The structural theorem is proved for subsets of $Valuable$, which depends on F. The probabilistic part is proved for a set of Bucket indices, that is called $Super$, is defined next and depends solely on the matroid.

Definition 50 *[Super] We define **Super** to be the set of all $i \geq 1 + LOPT(U) - 2\log rank(U)$ such that $rank(B_i) \geq \max\left\{1, \sqrt{\frac{LOPT(U)}{2^{i+12}}}\right\}.$*

The following lemma states, among other things, a condition in which $Valuable \subseteq Super$. We show later on that this condition is met with high probability and hence, with high probability, the probabilistic part applies to all subsets of $Valuable$.

Lemma 51

1. *for every $i \in Super$, $rank(B_i) \geq 2^{2^{32}}$,*

2. *if $LOPT(F) \geq \frac{1}{8} \cdot LOPT(U)$, then $Valuable \subseteq Super$,*

3. *$LOPT(B_{Super}) \geq \frac{7}{8} \cdot LOPT(U)$ and*

4. *if $rank(F) > 2^8$, then $LOPT\left(B^F_{Valuable}\right) > \frac{1}{2} \cdot LOPT(F)$.*

The proof of the preceding lemma is in Appendix C.1. The following theorem merges all the probabilistic parts of this result.

Theorem 52 *With probability at least $\frac{3}{4}$, for every $k \in Super$, $J \subseteq Super$, $j \in J$, critical tuple $(Block, Good, Bad)$ and $i \in \bigcup_{\ell^* \in \mathbb{Z}} Block(\ell^*)$ the following holds: if for every $\ell \in \mathbb{Z}$, we have that $Good(\ell)$, $Bad(\ell)$ and $Block(\ell)$ are subsets of $Super$, then*

1. *$uncov\left(B^{U \setminus F}_{J \setminus \{j\}}, B^{U \setminus F}_j\right) \geq uncov\left(B^F_{J \setminus \{j\}}, B^F_j\right) - 4 \cdot rank\left(B^F_J\right)^{\frac{3}{4}},$*

2.
$$uncov\left(B^F_{Bad(i)} \cup B^{U \setminus F}_{Block(i) \setminus \{i\}}, B^{U \setminus F}_i\right)$$
$$\geq$$
$$uncov\left(B^F_{Bad(i)} \cup B^F_{Block(i) \setminus \{i\}}, B^F_i\right) - 4 \cdot rank\left(B^F_{Good(i)}\right)^{\frac{3}{4}}$$

3. *$loss\left(B^F_{Good(i)}, B^{U \setminus F}_i\right) \leq uncov\left(B^F_{Good(i) \setminus \{i\}}, B^F_i\right) + 2 \cdot rank\left(B^F_{Good(i)}\right)^{\frac{5}{6}}.$*

4. *for every $\ell \in Super$, we have $rank\left(B^F_\ell\right) \geq \frac{1}{4} \cdot rank(B_\ell)$ and*

5. *$LOPT(F) \geq \frac{1}{5} \cdot LOPT(U).$*

The proof of the preceding theorem is in Appendix B. The next proposition is used in order to bound the influence of the rightmost terms of the inequalities in the first three items of the preceding theorem.

Proposition 53 *Let $K \subseteq Valuable$. If $rank(F) \geq \frac{1}{4} \cdot 2^{2^{32}}$, $LOPT\left(B^F_K\right) > \frac{1}{16} \cdot LOPT(U)$ and $K < \log LOPT(F) - 64 \cdot \log \log rank(F)$, then*

$$\frac{LOPT\left(B^F_K\right)}{2^{25} \cdot \log \log rank(F)} - 8 \cdot \sum_{i \in K} 2^i \cdot rank\left(B^F_i\right)^{\frac{10}{11}} \geq \frac{OPT(U)}{2^{30} \cdot \log \log rank(U)}.$$

The proof of the preceding proposition is in Appendix C.2. The following theorem combines the probabilistic and structural parts of the result.

Theorem 54 *There exists an algorithm that, with probability at least $\frac{3}{4}$ over the choice of F, will return one of the following:*

1. $J \subset \mathbb{Z}$, where $\sum_{j \in J} 2^j \cdot uncov\left(B_{J \setminus \{j\}}^{U \setminus F}, B_j^{U \setminus F}\right) \geq \frac{OPT(U)}{2^{30} \cdot \log \log rank(U)}$ and

2. a critical tuple $(Block, Good, Bad)$ such that

$$\sum_{j \in \bigcup_{i \in \mathbb{Z}} Block(i)} 2^j \cdot \left(uncov\left(B_{Bad(j)}^F \cup B_{Block(j) \setminus \{j\}}^{U \setminus F}, B_j^{U \setminus F}\right) - loss\left(B_{Good(j)}^F, B_j^{U \setminus F}\right)\right) \geq \frac{OPT(U)}{2^{30} \cdot \log \log rank(U)}.$$

Proof. We prove that conditioned on the events of Thm 52 the assertion of the theorem holds with probability 1. Since the events of Theorem 52 hold, with probability at least $\frac{3}{4}$, the theorem follows.

By Item 5 of Theorem 52 and Item 2 of Lemma 51, $Valuable \subseteq Super$ and hence, by Item 1 of Lemma 51,

$$\text{for every } i \in Valuable, \quad rank(F) \geq rank\left(B_i^F\right) \geq \frac{1}{4} \cdot 2^{2^{32}}. \tag{20}$$

Let $\{K_1, K_2\}$ be a partition of $Valuable$ such that $K_1 \geq \log LOPT(F) - 64 \cdot \log \log rank(F)$ and $K_2 < K_1$. We observe that $|K_1| \leq 64 \cdot \log \log rank(F)$ since, by definition $\max Valuable \leq \log LOPT(F)$.

Suppose that $LOPT\left(B_{K_1}^F\right) \geq \frac{1}{8} \cdot LOPT(F)$. Since $|K_1| \leq 64 \cdot \log \log rank(F)$, by the Pigeon Hole Principle and the definition of $LOPT$ (Definition 9), there exists $k \in K_1$ such that $LOPT\left(B_k^F\right) \geq \frac{LOPT(F)}{2^{22} \cdot \log \log rank(F)}$. Hence, $LOPT\left(B_k^{U \setminus F}\right) \geq \frac{OPT(U)}{2^{30} \cdot \log \log rank(U)}$, by (20) and Item 4 of Theorem 52. Consequently, $J = \{k\}$ satisfies the first item of the theorem.

Suppose that $LOPT\left(B_{K_1}^F\right) < \frac{1}{8} \cdot LOPT(F)$. By Item 4 of Lemma 51, $LOPT\left(B_{Valuable}^F\right) > \frac{1}{2} \cdot LOPT(F)$ and hence, according to Item 5 of Theorem 52,

$$LOPT\left(B_{K_2}^F\right) > \frac{3}{8} \cdot LOPT(F) > \frac{1}{16} \cdot LOPT(U). \tag{21}$$

Since $K_2 \subseteq Valuable$ and (20), by the assumption that the items of Theorem 52 hold, at least one of the following exists:

1. a set $J \subseteq K_2$ of bucket indices such that

$$\sum_{j \in J} 2^i \cdot uncov\left(B_{J \setminus \{j\}}^{U \setminus F}, B_j^{U \setminus F}\right) \geq \frac{LOPT\left(B_{K_2}^F\right)}{2^{19} \cdot \log \log rank(F)} - 8 \cdot \sum_{j \in J} 2^j \cdot rank\left(B_j^F\right)^{\frac{9}{11}}$$

2. a critical tuple $(Block, Good, Bad)$, where for every $\ell \in \mathbb{Z}$, $Good(\ell)$, $Bad(\ell)$ and $Block(\ell)$ are

31

subsets of K_2, such that

$$\sum_{j \in \bigcup_{i \in \mathbb{Z}} Block(i)} 2^j \cdot \left(uncov\left(B_{Bad(j)}^F \cup B_{Block(j) \setminus \{j\}}^{U \setminus F}, B_j^{U \setminus F} \right) - loss\left(B_{Good(j)}^F, B_j^{U \setminus F} \right) \right)$$
$$\geq$$
$$\frac{LOPT\left(B_{K_2}^F \right)}{2^{25} \cdot \log\log rank(F)} - 8 \cdot \sum_{j \in \bigcup_{i \in \mathbb{Z}} Block(i)} 2^j \cdot rank\left(B_j^F \right)^{\frac{10}{11}}$$
.

Since K_2 is finite, whichever of the above exists, it can be found and returned, using only exhaustive search and the knowledge obtained about the elements of F via the oracle. This together with (20), (21), and Proposition 53 implies the theorem. ∎

The following theorem is the main result of this paper.

Theorem 55 *The Main Algorithm is Order-Oblivious, Known-Cardinality and has a competitive-ratio of $O(\log\log rank(U))$.*

Proof. Suppose first that $\max\{val(e) \mid e \in U\} \geq 2^{-2^{34}} \cdot OPT(U)$. By Observation 1, with probability $\frac{1}{4}$, F has the element with the second largest value among the elements of U (or largest if U has more than a single element with value $\max\{val(e) \mid e \in U\}$) and $U \setminus F$ has an element of value $\max\{val(e) \mid e \in U\}$. Thus, with probability at least $\frac{1}{4}$, the value of the element selected is $\max\{val(e) \mid e \in U\}$. Since, with probability $1/2$, $\max\{val(e) \mid e \in F\}$, is the input to Stage 3 of Main Algorithm, the competitive-ratio holds in this case.

Assume that $\max\{val(e) \mid e \in U\} < 2^{-2^{34}} \cdot OPT(U)$ and $\max\{val(e) \mid e \in F\}$, is not the input to Stage 3 of Main Algorithm. According to Theorem 54, there exists an algorithm that, with probability at least $\frac{3}{4}$, finds a $J \subset \mathbb{Z}$, that satisfies the first item of Theorem 54 or a critical tuple $(Block, Good, Bad)$ that satisfies the second item of Theorem 54. If the algorithm returns a finite set J as above then, by Theorem 21, the required competitive-ratio is achieved. If the algorithm returned a critical tuple $(Block, Good, Bad)$ as above then, by Theorem 22, the required competitive-ratio is achieved.

We note that the Main Algorithm is Known-Cardinality, since (i) the computation in Gathering stage is independent of the matroid elements; (ii) the computation in the Preprocessing stage; and (iii) the Selection stage uses only elements of the matroid that have been revealed.

We also note that the Main Algorithm is Order-Oblivious, because by construction it follows definition 13 and the analysis depends on the elements in the sets F and $U \setminus F$ but not on their order. ∎

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. Submodular 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] Nedialko 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.

A   Graph Definitions and Notations

A *directed graph* is an ordered pair $G = (V, E)$, where V is a set of vertices and E is a set of ordered pairs of vertices of V called *arcs*. We use the notation $V(G)$ for the set of vertices of G and $E(G)$ for the set of arcs of G. An arc $(u, v) \in E(G)$ is an *incoming-arc* of the vertex v and an *outgoing-arc* of the vertex u. The *in-degree* of a vertex $v \in V(G)$ is the number of its incoming-arcs and its *out-degree* is the number of its outgoing-arcs. A *directed path* P in G from u_1 to u_k is a tuple $(u_1, u_2, \ldots, u_k) \in V^k$ of k distinct entries such that $(u_i, u_{i+1}) \in E(G)$ for every $i \in [k-1]$. The *length* of P is $k-1$. A *directed cycle* in G is a tuple $(u_1, u_2, \ldots, u_{k+1}) \in V^{k+1}$ such that $u_{k+1} = u_1$, $(u_k, u_1) \in E(G)$ and $(u_1, u_2, \ldots, u_k)$ is a directed path. The *distance* from u to v in $V(G)$ is the minimum length of a path from u to v if such a path exists, and ∞ otherwise.

A *binary tree* is an ordered triple $T = (V, E, r)$, where (V, E) is a directed graph that has exactly three types of vertices: (i) $r \in V(G)$, called the *root* of T, it has in-degree 0 and out-degree 2, (ii) *internal vertices*, each having in degree 1 and out-degree 2 and (iii) *leaves*, each having in-degree 1 and out-degree 0. For every $v \in V(T)$, we define the *depth* of v to be the distance from r to v. We define the depth of an arc $(u, v) \in E(T)$ to be the depth of u. A *binary tree* $T = (V, E, r)$ is *balanced*, if all leaves have the same depth. We say that $u \in V(T)$ is the *parent* of $v \in V(T)$ if $(u, v) \in E(T)$ and a *child* of v if $(v, u) \in E(T)$.

B   Proof of Theorem 52

The actual proof appears at the end of Appendix B.2. The succeeding subsection contains the concentration inequalities, that are required for the proof.

B.1   Talagrand based concentrations

This subsection of the appendix is very similar to one that appears in [8], we add it for the sake of completeness.

The following definition is an adaptation of the Lipschitz condition to our setting.

Definition 56 [*Lipschitz*] *Let $h \in \mathbb{N}^+$ and $f : U \longrightarrow \mathbb{N}$. If $|f(S_1) - f(S_2)| \leq c$ 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 **c-Lipschitz**.*

Definition 57 [Definition 3 of [1]] Let $f : \mathbb{N} \longrightarrow \mathbb{N}$. h is f-certifiable if whenever $h(x) \geq s$ there exists $I \subseteq \{1, \ldots, n\}$ 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 58 For every finite $K \subset \mathbb{Z}$, the rank function over subsets of B_K is 1-Lipschitz and f-certifiable with $f(s) = rank(B_K)$.

Proof. The *rank* function is 1-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 $rank(S) \leq rank(R) \leq rank(B_K)$. Thus, the *rank* function over subsets of B_K is f-certifiable with $f(s) = rank(B_K)$. ∎

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

Theorem 59 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 60 Let $t \geq 2$, K be a finite subset of $\mathbb{Z}$, $K' \subseteq K$ and
$$S = B_{K'}^F \cup B_{K \setminus K'}^{U \setminus F},$$

then
$$prob\left(\left|rank(S) - med(rank(S))\right| \geq t \cdot \sqrt{rank(B_K)}\right) \leq e^{2 - \frac{t^2}{4}}.$$

Proof. By Observation 58, the *rank* function is 1-Lipschitz and *rank*-certifiable. Hence, by the union bound together with Theorem 59 once with $b = med(rank(S)) + t \cdot \sqrt{rank(B_K)}$ and once with $b = med(rank(S))$, and by noting that $prob(rank(S) \geq med(rank(S))) \geq \frac{1}{2}$ and $prob(rank(S) \leq med(rank(S))) \geq \frac{1}{2}$, the lemma follows. ∎

Because the random sets F and $U \setminus F$ have exactly the same distribution and the buckets are pairwise disjoint, the following observation holds.

Observation 61 For every $K \subset \mathbb{Z}$, and subsets K_1 and K_2 of K, we have
$$med\left(B_{K_1}^F \cup B_{K \setminus K_1}^{U \setminus F}\right) = med\left(B_{K_2}^F \cup B_{K \setminus K_2}^{U \setminus F}\right).$$

B.2 Union bound based results

Proposition 62 With probability less than
$$\sum_{K \subseteq Super} \sum_{K_1 \subseteq K} e^{4 - \frac{1}{4} \cdot rank(B_K)^{\frac{1}{2}}},$$

there exists $K \subseteq Super$, $K_1 \subseteq K$ and $R_1, R_2 \in \{F, U \setminus F\}$, such that $|K| \geq 2$ and
$$|rank\left(B_{K_1}^{R_1} \cup B_{K \setminus K_1}^{R_2}\right) - med\left(rank\left(B_{K_1}^{R_1} \cup B_{K \setminus K_1}^{R_2}\right)\right)| \geq rank(B_K)^{\frac{3}{4}}.$$

Proof. Let $K \subseteq Super$, $R_1 \in \{F, U \setminus F\}$, $R_2 \in \{F, U \setminus F\}$, $K_1 \subseteq K$,
$$S(K, K_1, R_1, R_2) = B_{K_1}^{R_1} \cup B_{K \setminus K_1}^{R_2}$$

and

$$s(K, K_1, R_1, R_2) = med\left(rank\left(B_{K_1}^{R_1} \cup B_{K \setminus K_1}^{R_2}\right)\right).$$

By Lemma 60,

$$prob\left(\left|rank\left(S(K, K_1, R_1, R_2)\right) - s(K, K_1, R_1, R_2)\right| \geq t \cdot \sqrt{rank\left(B_K\right)}\right) \leq e^{2 - \frac{t^2}{4}}.$$

Since $K \subseteq Super$, by Assumption 14 and Item 1 of Lemma 51,

$$rank\left(B_K\right) \geq \min\left\{rank\left(B_i\right) \mid i \in Super\right\} \geq 16.$$

Consequently, by taking $t = \left(rank\left(B_K\right)\right)^{\frac{1}{4}}$ and using the union bound, the proposition follows. ∎

Lemma 63 *With probability at least $\frac{3}{4}$, for every $K \subseteq Super$, $K^* \subseteq K$, $R_1, R_2 \in \{F, U \setminus F\}$ and $k \in K$, we have*

$$\left|rank\left(B_{K^*}^{R_1} \cup B_{K \setminus K^*}^{R_2}\right) - med\left(rank\left(B_{K^*}^{R_1} \cup B_{K \setminus K^*}^{R_2}\right)\right)\right| \geq rank\left(B_K\right)^{\frac{3}{4}}.$$

and

$$loss\left(B_K^F, B_k^{U \setminus F}\right) \leq uncov\left(B_{K \setminus \{k\}}^F, B_k^F\right) + rank\left(B_k\right)^{\frac{2}{3}}.$$

Proof. By Propositions 62, Theorem 23 and the union bound, the probability, that the event in the lemma does not occur, does not exceed $\eta = \sum_{K \subseteq Super} \sum_{K^* \subseteq K} e^{5 - \frac{1}{4} \cdot rank(B_K)^{\frac{1}{6}}}$. Note that we over count by using K^* instead of k. By the definition of $Super$ (Definition 50), $rank\left(B_K\right) \geq rank\left(B_{\min K}\right) \geq \sqrt{\frac{LOPT(U)}{2^{\min K + 12}}}$ and therefore,

$$\eta < \sum_{K \subseteq Super} \sum_{K^* \subseteq K} e^{5 - \frac{1}{4} \cdot \left(\frac{LOPT(U)}{2^{\min K + 12}}\right)^{\frac{1}{12}}}$$

For every $z \in Super$, there are at most $\max Super - z$ members $i \in Super$ such that $i \geq z$. Hence, there are at most $2^{2(\max Super - z)}$ pairs $K, K_1 \subseteq Super$ such that $K_1 \subseteq K$ and $\min K = z$. Consequently by the above inequality,

$$\eta < \sum_{z \in Super} 2^{2(\max Super - z)} \cdot e^{5 - \frac{1}{4} \cdot \left(\frac{LOPT(U)}{2^{z + 12}}\right)^{\frac{1}{12}}} < \sum_{z \in Super} 2^{2(\max Super - z) + 10 - \frac{1}{4} \cdot \left(\frac{LOPT(U)}{2^{z + 12}}\right)^{\frac{1}{12}}}.$$

By Assumption 14, we have $\max Super < \log LOPT\left(U\right) - 2^{34}$ and therefore,

$$\eta < \sum_{z \in \mathbb{N}} 2^{2z + 10 - \frac{1}{4} \cdot \left(\frac{LOPT(U)}{2^{\max Super - z}}\right)^{\frac{1}{12}}} < \sum_{z \in \mathbb{N}} 2^{2z + 10 - \frac{1}{4} \cdot \left(\frac{LOPT(U)}{2^{\log LOPT(U) - 2^{34} + 12 - z}}\right)^{\frac{1}{12}}} < \frac{1}{4}.$$

∎

Theorem 52 (restated) *With probability at least $\frac{3}{4}$, for every $k \in Super$, $J \subseteq Super$, $j \in J$, critical tuple $(Block, Good, Bad)$ and $i \in \bigcup_{\ell^* \in \mathbb{Z}} Block(\ell^*)$ the following holds: if for every $\ell \in \mathbb{Z}$,*

36

we have that $Good(\ell)$, $Bad(\ell)$ and $Block(\ell)$ are subsets of $Super$, then

1. $uncov\left(B_{J\setminus\{j\}}^{U\setminus F}, B_j^{U\setminus F}\right) \geq uncov\left(B_{J\setminus\{j\}}^F, B_j^F\right) - 4 \cdot rank\left(B_J^F\right)^{\frac{3}{4}}$,

2. $$uncov\left(B_{Bad(i)}^F \cup B_{Block(i)\setminus\{i\}}^{U\setminus F}, B_i^{U\setminus F}\right)$$
$$\geq$$
$$uncov\left(B_{Bad(i)}^F \cup B_{Block(i)\setminus\{i\}}^F, B_i^F\right) - 4 \cdot rank\left(B_{Good(i)}^F\right)^{\frac{3}{4}}$$

3. $loss\left(B_{Good(i)}^F, B_i^{U\setminus F}\right) \leq uncov\left(B_{Good(i)\setminus\{i\}}^F, B_i^F\right) + 2 \cdot rank\left(B_{Good(i)}^F\right)^{\frac{5}{6}}$.

4. *for every $\ell \in Super$, we have $rank\left(B_\ell^F\right) \geq \frac{1}{4} \cdot rank\left(B_\ell\right)$ and*

5. $LOPT\left(F\right) \geq \frac{1}{5} \cdot LOPT\left(U\right)$.

Proof. **Items 1, 2 and 3** follow from the definition of *uncov* (Definitions 15), Observation 61, Lemma 63 and Item 4 of Proposition 4. **Item 4** follows from Item 1 of Lemma 51 and the first inequality of Lemma 63. According to Item 4, the definition of *LOPT* (Definition 9) and Item 3 of Lemma 51.

$$LOPT\left(F\right) \geq LOPT\left(B_{Super}^F\right) \geq \frac{1}{4} \cdot LOPT\left(B_{Super}\right) \geq \frac{1}{5} \cdot LOPT\left(U\right).$$

Thus, **Item 5** holds and the theorem follows. ∎

C Matroid structural results

C.1 Proof of Lemma 51

Each item of the lemma is asserted by a dedicated proposition. The following proposition implies **Item 1** of Lemma 51.

Proposition 64 *For every $i \in Super$, $rank\left(B_i\right) \geq 2^{2^{32}}$.*

Proof. Let $i \in Super$. By Assumption 5, $val(e) = 2^i$, for every $i \in \mathbb{Z}$ and $e \in B_i$, therefore by Assumption 14, $2^i \leq \max\{val(e) \mid e \in U\} < 2^{-2^{34}} \cdot OPT\left(U\right)$ and hence,

$$i < \log OPT\left(U\right) - 2^{34} < \log LOPT\left(U\right) - 2^{34},$$

where the last inequality follows from Item 1 of Observation 10. By the definition of *Super* (Definition 50),

$$rank\left(B_i\right) > \sqrt{\frac{LOPT\left(U\right)}{2^{i+12}}} > \sqrt{\frac{LOPT\left(U\right)}{2^{\log LOPT(U) - 2^{34} + 12}}}.$$

Thus, the proposition follows. ∎

The next proposition implies **Item 2** of Lemma 51.

Proposition 65 *If $LOPT\left(F\right) \geq \frac{1}{8} \cdot LOPT\left(U\right)$, then $Valuable \subseteq Super$.*

Proof. By the definition of $Valuable$ (Definition 16), for every $i \in Valuable$,

$$i \geq 4 + \log LOPT(F) - 2\log rank(F) \geq 1 + \log LOPT(U) - 2\log rank(U),$$

$rank\left(B_i^F\right) \geq 1$ and

$$rank\left(B_i^F\right) \geq \sqrt{\frac{LOPT(F)}{2^{i+8}}} \geq \sqrt{\frac{LOPT(U)}{2^{i+11}}} > \sqrt{\frac{LOPT(U)}{2^{i+12}}}.$$

Thus, by the definition of $Super$ (Definition 50), the proposition follows. ∎

The next proposition implies **Item 3** of Lemma 51.

Proposition 66 $LOPT(B_{Super}) \geq \frac{7}{8} \cdot LOPT(U).$

Proof. Let J_1 be the set of all integers smaller than $1 + \log LOPT(U) - 2\log rank(U)$ and J_2 be the set of all $i \in \mathbb{Z}$ such that $1 \leq rank(B_i) < \sqrt{\frac{LOPT(U)}{2^{i+12}}}$. Thus, by the definition of $LOPT$ (Definition 9),

$$LOPT(B_{J_2}) < 2^{-6} \cdot \sum_{i \in J_2} 2^i \cdot \sqrt{\frac{LOPT(U)}{2^i}} = 2^{-6} \cdot \sum_{i \in J_2} \sqrt{2^i \cdot LOPT(U)}.$$

Recall that $val(e) = 2^i$, for every $i \in \mathbb{Z}$ and $e \in B_i$, therefore, by Observation 10, $2^i \leq LOPT(U)$, for every $i \in J_2$. Consequently,

$$LOPT(B_{J_2}) < 2^{-6} \cdot LOPT(U) \cdot \sum_{i \in \mathbb{N}} 2^{-\frac{i}{2}} < \frac{1}{16} \cdot LOPT(U). \tag{22}$$

According to the construction of J_1 and the definition of $LOPT$ (Definition 9),

$$LOPT(B_{J_1}) = \sum_{i \in J_1} 2^i \cdot rank(B_i) \leq \sum_{i \in \mathbb{N}} 2^{-i} \cdot \frac{2 \cdot LOPT(U)}{rank(U)^2} \cdot rank\left(B_i^U\right)$$

and hence, since, by Proposition 64, $rank(U) > 2^{2^{32}}$, we get

$$LOPT(B_{J_1}) \leq \frac{4 \cdot LOPT(U)}{rank(U)} < \frac{1}{16} \cdot LOPT(U). \tag{23}$$

We note that $Super \cup J_1 \cup J_2$ contains all the indices of non-empty buckets. Consequently, by (22), (23) and the definition of $LOPT$ (Definition 9),

$$LOPT(B_{Super}) \geq LOPT(U) - \sum_{i \in [2]} LOPT(B_{J_i}) \geq \frac{7}{8} \cdot LOPT(U).$$

∎

The next proposition implies **Item 4** of Lemma 51.

Proposition 67 *If* $rank(F) > 2^8$, *then* $\quad LOPT\left(B_{Valuable}^F\right) > \frac{1}{2} \cdot LOPT(F).$

Proof. Let J_1 be the set of all integers smaller than $4 + \log LOPT(F) - 2 \log rank(F)$ and J_2 be the set of all integers i such that $1 \leq rank\left(B_i^F\right) < \sqrt{\frac{LOPT(F)}{2^{i+8}}}$. Thus,

$$LOPT\left(B_{J_2}^F\right) < 2^{-4} \cdot \sum_{i \in J_2} 2^i \cdot \sqrt{\frac{LOPT(F)}{2^i}} = 2^{-4} \cdot \sum_{i \in J_2} \sqrt{2^i \cdot LOPT(F)}.$$

Recall that $val(e) = 2^i$, for every $i \in \mathbb{Z}$ and $e \in B_i^F$, therefore, by Observation 10, $2^i \leq LOPT(F)$, for every $i \in J_2$. Hence,

$$LOPT\left(B_{J_2}^F\right) < 2^{-4} \cdot LOPT(F) \cdot \sum_{i \in \mathbb{N}} 2^{-\frac{i}{2}} < \frac{1}{4} \cdot LOPT(F). \tag{24}$$

According to the construction of J_1 and the definition of $LOPT$ (Definition 9),

$$LOPT\left(B_{J_1}^F\right) = \sum_{i \in J_1} 2^i \cdot rank\left(B_i^F\right) \leq \sum_{i \in \mathbb{N}} 2^{-i} \cdot \frac{16 \cdot LOPT(F)}{rank(F)^2} \cdot rank\left(B_i^F\right)$$

and hence, because $rank(F) > 2^8$,

$$LOPT\left(B_{J_1}^F\right) \leq \frac{32 \cdot LOPT(F)}{rank(F)} < \frac{1}{4} \cdot LOPT(F), \tag{25}$$

We note that $Valuable \cup J_1 \cup J_2$ contains all the indices of non-empty buckets. Consequently, by (24), (25) and the definition of $LOPT$ (Definition 9),

$$LOPT\left(B_{Valuable}^F\right) \geq LOPT(F) - \sum_{i \in [2]} LOPT\left(B_{J_i}^F\right) > \frac{1}{2} \cdot LOPT(F).$$

$\blacksquare$

C.2 Proof of Proposition 53

Proposition 53 (restated) *Assume* $K \subseteq Valuable$ *and* $rank(F) \geq \frac{1}{4} \cdot 2^{2^{32}}$. *If* $LOPT\left(B_K^F\right) > \frac{1}{16} \cdot LOPT(U)$ *and* $K < \log LOPT(F) - 64 \cdot \log \log rank(F)$, *then*

$$\frac{LOPT\left(B_K^F\right)}{2^{25} \cdot \log \log rank(F)} - 8 \cdot \sum_{i \in K} 2^i \cdot rank\left(B_i^F\right)^{\frac{10}{11}} \geq \frac{OPT(U)}{2^{30} \cdot \log \log rank(U)}.$$

Proof. By Observation 10, $LOPT\left(B_K^F\right) > \frac{1}{16} \cdot OPT(U)$, and hence, it is sufficient to show that

$$8 \cdot \sum_{i \in K} 2^i \cdot rank\left(B_i^F\right)^{\frac{10}{11}} < \frac{OPT\left(B_K^F\right)}{2^{26} \cdot \log \log rank(F)}. \tag{26}$$

Let $k \in K$ be such that $rank\left(B_k^F\right) = \min\left\{rank\left(B_i^F\right) \mid i \in K\right\}$. By using the definition of $LOPT$

(Definition 9), we get that

$$\sum_{i \in K} 2^i \cdot rank\left(B_i^F\right)^{\frac{10}{11}} \leq \sum_{i \in K} \frac{2^i \cdot rank\left(B_i^F\right)}{rank\left(B_k^F\right)^{\frac{1}{11}}} = \frac{LOPT\left(B_K^F\right)}{rank\left(B_k^F\right)^{\frac{1}{11}}}$$

and hence, we only need to show that we have

$$\frac{1}{8} \cdot rank\left(B_k^F\right)^{\frac{1}{11}} > 2^{26} \cdot \log\log rank\left(F\right).$$

Since $k \in Valuable$, by the definition of $Valuable$ (Definition 16), $rank\left(B_k^F\right)^{\frac{1}{11}} \geq \left(\frac{LOPT(F)}{2^{k+8}}\right)^{\frac{1}{22}}$.
Hence, because $k < \log LOPT\left(F\right) - 64 \cdot \log\log rank\left(F\right)$,

$$rank\left(B_i^F\right)^{\frac{1}{11}} \geq \left(\frac{LOPT\left(F\right)}{2^{\log LOPT(F) - 64 \cdot \log\log rank(F) + 8}}\right)^{\frac{1}{22}} > (\log rank\left(F\right))^2.$$

Finally, since $rank\left(F\right) \geq \frac{1}{4} \cdot 2^{2^{32}}$,

$$\frac{1}{8} \cdot rank\left(B_i^F\right)^{\frac{1}{11}} > \frac{1}{8} \cdot (\log rank\left(F\right))^2 > 2^{26} \cdot \log\log rank\left(F\right),$$

and therefore, (26) holds and thus, the proposition follows. ∎

C.3 Proof of Lemma 35

The proof, that is given at the end of this subsection, is almost a direct result of the following
proposition, which is purely number theoretic.

Proposition 68 *Let $w : \mathbb{Z} \longrightarrow \mathbb{N}$, where $\{i \mid w(i) > 0\}$ is finite, and $m = \sum_{j \in \mathbb{Z}} w(j) \cdot 2^j$. Then,
there exists a strictly monotonically decreasing sequence of integers $h_1, h_2, \ldots, h_k$ such that:*

1. $0 < w(h_i) \leq \frac{1}{32} \cdot w(h_{i+1})$, for every $i \in [k'-1]$, and

2. $\sum_{i \in [k']} w(h_i) \cdot 2^{h_i} > \frac{m}{18}$.

Proof. Let $\ell_1, \ell_2, \ldots, \ell_{k'} \in \mathbb{Z}$ be a maximal strictly decreasing sequence such that, ℓ_1 is maximum
so that $w(\ell_1) > 0$ and for every $j \in [k'-1]$, inductively, ℓ_{j+1} is the maximum integer such that
$\ell_{j+1} < \ell_j$ and $w(\ell_j) \leq \frac{1}{2} \cdot w(\ell_{j+1})$. We note that $\ell_1, \ell_2, \ldots, \ell_{k'}$ satisfy,

$$\text{for every } i \in [k'-1], \quad 0 < w(\ell_i) \leq \frac{1}{2} \cdot w(\ell_{i+1}) \tag{27}$$

Let $J_{k'}$ be the set of all integers $x < \ell_{k'}$ and, for every $i \in [k'-1]$, let $J_i = \{x \in \mathbb{Z} \mid \ell_{i+1} < x < \ell_i\}$.
According to construction, for every $i \in [k']$ and $j \in J_i$ we have that $w(\ell_i) > \frac{1}{2} \cdot w(j)$. Hence, for
every $i \in [k']$,

$$w(\ell_i) \cdot 2^{\ell_i} \geq w(\ell_i) \cdot \sum_{j < \ell_i} 2^j > \frac{1}{2} \cdot \sum_{j \in J_i} w(j) \cdot 2^j$$

40

Consequently,

$$2 \cdot \sum_{i=1}^{k'} w(\ell_i) \cdot 2^{\ell_i} > \sum_{i=1}^{k'} \sum_{j \in J_i} w(j) \cdot 2^j.$$

According to construction,

$$\left(\sum_{i=1}^{k'} w(\ell_i) \cdot 2^{\ell_i} \right) + \sum_{i=1}^{k'} \sum_{j \in J_i} w(j) \cdot 2^j = m.$$

This together with preceding inequality implies that

$$\sum_{i \in [k']} w(\ell_i) \cdot 2^{\ell_i} > \frac{m}{3}. \tag{28}$$

For every $i \in [6]$, define $K_i = \{\ell_j \mid (j \in [k']) \wedge (j = i \mod 6)\}$. Hence, by (27),

$$\text{for every } i \in [6], \text{ and } j_1, j_2 \in K_i \text{ such that } j_1 > j_2, \quad 0 < w(j_1) \le \frac{1}{32} \cdot w(j_2). \tag{29}$$

Let q be such that

$$\sum_{j \in K_q} w(j) \cdot 2^j > \frac{m}{3 \cdot 6} = \frac{m}{18}. \tag{30}$$

By (28) and the pigeon hole principle, such a q exists. Let $h_1, h_2, \ldots, h_k$ be the monotonically decreasing sequence consisting of the elements of K_q. Thus, by (29) and (30), the proposition follows. ∎

Lemma 35 (restated)   *For every $F \subseteq U$ and $K \subset \mathbb{Z}$, there exists a strong sequence for B_K^F.*

Proof.   Define, $w : \mathbb{Z} \longrightarrow \mathbb{N}$ as follows: for every $i \in \mathbb{Z}$, $w(i) = rank\left(B_i^F\right)$ if $i \in K$, and otherwise $w(i) = 0$. Let $m = \sum_{j \in \mathbb{Z}} w(j) \cdot 2^j = LOPT\left(B_K^F\right)$. By Proposition 68 and Definition 34, there exists a strong sequence for B_K^F. ∎