Online Disjoint Spanning Trees and Polymatroid Bases[*]

Karthekeyan Chandrasekaran   Chandra Chekuri   Weihao Zhu

September 14, 2025

Abstract

Finding the maximum number of disjoint spanning trees in a given graph is a well-studied problem with several applications and connections. The Tutte-Nash-Williams theorem provides a min-max relation for this problem which also extends to disjoint bases in a matroid and leads to efficient algorithms [13]. Several other packing problems such as element disjoint Steiner trees, disjoint set covers, and disjoint dominating sets are NP-Hard but admit an $O(\log n)$-approximation [6, 3]. Călinescu, Chekuri, and Vondrák [2] viewed all these packing problems as packing bases of a polymatroid and provided a unified perspective. Motivated by applications in wireless networks, recent works have studied the problem of packing set covers in the *online* model [11, 5, 1]. The online model poses new challenges for packing problems. In particular, it is not clear how to pack a maximum number of disjoint spanning trees in a graph when edges arrive online. Motivated by these applications and theoretical considerations we formulate an online model for packing bases of a polymatroid, and describe a randomized algorithm with a polylogarithmic competitive ratio. Our algorithm is based on interesting connections to the notion of quotients of a polymatroid that has recently seen applications in polymatroid sparsification [12]. We generalize the previously known result for the online disjoint set cover problem [5] and also address several other packing problems in a unified fashion. For the special case of packing disjoint spanning trees in a graph (or a hypergraph) whose edges arrive online, we provide an alternative to our general algorithm that is simpler and faster while achieving the same poly-logarithmic competitive ratio.

1 Introduction

Finding the maximum number of disjoint spanning trees in a given graph is a well-studied problem with several applications and connections. The Tutte-Nash-Williams theorem provides a min-max relation for this problem which also extends to the maximum number of disjoint bases in a matroid with efficient algorithms [13]. Several other packing problems such as disjoint set covers, disjoint dominating sets, and element disjoint Steiner trees are NP-Hard but admit an $O(\log n)$-approximation[1] [6, 3]. Călinescu, Chekuri, and Vondrák [2] viewed these problems as packing bases of a polymatroid and provided a unified perspective. Motivated by applications in wireless networks, recent works have studied the problem of packing set covers in the *online* model [11, 5, 1]. The online model poses new challenges for packing problems. In particular, it is not clear how to pack a maximum number of disjoint spanning trees in a graph when edges arrive online. Motivated by these applications and theoretical considerations we consider the problem of packing disjoint bases of a polymatroid in the online model.

A *polymatroid* $f : 2^{\mathcal{N}} \to \mathbb{Z}_{\geq 0}$ on a ground set $\mathcal{N}$ is an integer-valued monotone submodular function with $f(\emptyset) = 0$. We recall that a function $f : 2^{\mathcal{N}} \to \mathbb{Z}_{\geq 0}$ is *monotone* if $f(A) \leq f(B)$ for every $A \subseteq B$ and *submodular* if $f(A) + f(B) \geq f(A \cup B) + f(A \cap B)$ for every $A, B \subseteq \mathcal{N}$. A subset $S \subseteq \mathcal{N}$ is a *base* of f if $f(S) = f(\mathcal{N})$.[2] In the DISJ-POLYMATROID-BASES problem, we are given a polymatroid $f : 2^{\mathcal{N}} \to \mathbb{Z}_{\geq 0}$ via an evaluation oracle. The goal is to find a maximum number of disjoint bases. We define

$$\mathtt{opt}(f) := \max\{k : \exists \, k \text{ disjoint bases of } f\}.$$

Polymatroids generalize matroid rank function, coverage functions, and many others. Consequently, numerous set packing problems can be cast as special cases of DISJ-POLYMATROID-BASES. We will discuss some of these special cases shortly.

Online DISJ-POLYMATROID-BASES Model. We formally describe the online model for DISJ-POLYMATROID-BASES (denoted ONLINE-DISJ-POLYMATROID-BASES). We have an underlying polymatroid $f : 2^{\mathcal{N}} \to \mathbb{Z}_{\geq 0}$ over a large but finite ground set $\mathcal{N}$. Let $n := |\mathcal{N}|$. We index the elements of the ground set $\mathcal{N}$ as $e_1, e_2, \ldots, e_n$ where e_t is the element that arrives at time t for every $t \in [n]$. For each $t \in [n]$, we denote $\mathcal{N}_t := \{e_1, e_2, \ldots, e_t\}$ and let $f_{|\mathcal{N}_t} : 2^{\mathcal{N}_t} \to \mathbb{Z}_{\geq 0}$ be the function obtained from f by restricting the ground set to $\mathcal{N}_t$, that is, $f_{|\mathcal{N}_t}(A) := f(A)$ for every $A \subseteq \mathcal{N}_t$. At each timestep $t \in [n]$, the online algorithm has access to the evaluation oracle[3] of the function restricted to the set of elements that have arrived until time t, i.e., evaluation oracle of the function $f_{|\mathcal{N}_t}$, and has to color element e_t irrevocably. A color is said to be a *base color* if the set S of elements with that color is a base of f. The goal of the online algorithm is to maximize the number of base colors. We remark that we are implicitly assuming that the elements of $\mathcal{N}$ is the input sequence. The competitive ratio of an online algorithm is the ratio between the base colors in an optimal offline algorithm and that of the online algorithm. For randomized online algorithms, we will be interested in the expected competitive ratio. We assume that the online algorithm has prior knowledge of the function value of the ground set $\mathcal{N}$, i.e., $f(\mathcal{N})$. This assumption is motivated by applications to be discussed below.

---

[1] We are using the convention of α-approximation for a maximization problem with $\alpha > 1$ with the understanding that the returned value is at least $\mathtt{opt}/\alpha$.

[2] In most settings, one would define a set S to be a base if it is *inclusionwise minimal* subject to satisfying $f(S) = f(\mathcal{N})$. This is particularly important in the setting of matroids where all bases have the same cardinality which is not necessarily true in the polymatroidal setting. However, since we are interested in (approximating) the maximum number of disjoint bases we adopt the relaxed definition for simplicity.

[3] The evaluation oracle of a function $g : 2^V \to \mathbb{R}$ takes a set $S \subseteq V$ as input and returns $g(S)$.

Applications.   Polymatroids generalize coverage functions and matroid rank functions. We discuss these two special cases, the associated packing problems, and their online model below.

1. In the DISJ-SET-COVER problem, the input is a set system over a finite universe. We will alternatively view the set system as a hypergraph $H = (V, E)$ where V corresponds to the universe and the hyperedges in E correspond to sets in the system. The goal is to find a maximum number of disjoint set covers (a subset $A \subseteq E$ of hyperedges is a set cover if $\cup_{e \in A} e = V$). DISJ-SET-COVER can be cast as a special case of DISJ-POLYMATROID-BASES by considering the coverage function of the hypergraph as the polymatroid, i.e., by considering the polymatroid $f : 2^E \to \mathbb{Z}_{\geq 0}$ defined as $f(A) := |\cup_{e \in A} e|$. Coverage functions are fairly general with several prominent special cases–e.g., the domatic number problem is a special case of DISJ-SET-COVER [6]. In the online setting of DISJ-SET-COVER (termed ONLINE-DISJ-SET-COVER), the vertex set is known in advance while the hyperedges are revealed in an online fashion. The online algorithm has to color each hyperedge immediately upon arrival irrevocably in order to maximize the number of colors that form a set cover.

2. In the DISJ-MATROID-BASES problem, we are given evaluation access to a matroid rank function $r : 2^{\mathcal{N}} \to \mathbb{Z}_{\geq 0}$ over a ground set $\mathcal{N}$ (we recall that a matroid rank function r is a polymatroid that additionally satisfies $r(\{e\}) \leq 1$ for every $e \in \mathcal{N}$). A subset $S \subseteq \mathcal{N}$ is a base of the matroid if $r(S) = r(\mathcal{N})$. The goal is to find a maximum number disjoint bases of the matroid. Matroid rank function is a polymatroid and hence, DISJ-MATROID-BASES is a special case of DISJ-POLYMATROID-BASES. In the online setting of DISJ-MATROID-BASES (termed ONLINE-DISJ-MATROID-BASES), the ground set is revealed in an online fashion while the online algorithm has access to the rank function restricted to the set of elements that have arrived so far. The online algorithm has to color each element immediately upon arrival irrevocably in order to maximize the number of base colors.

Next, we describe three special cases of DISJ-POLYMATROID-BASES, namely DISJ-MATRIX-BASES, DISJ-SPANNING-TREES, and DISJ-CONN-SPANNING-SUBHYPERGRAPHS. To the best of authors' knowledge, these three problems have not been explored in the online setting. One of the motivations of this work is to understand these three problems in the online setting.

1. In the DISJ-MATRIX-BASES problem, we are given a matrix $M \in \mathbb{R}^{n \times d}$ of rank d and the goal is to find a maximum number of disjoint spanning subsets of row-vectors—a subset of row vectors is *spanning* if its linear hull is $\mathbb{R}^d$. This is a special case of DISJ-MATROID-BASES where the matroid is the linear matroid defined by M (and consequently, the rank function of a subset of row-vectors is the dimension of the subspace spanned by them). In the online setting of DISJ-MATRIX-BASES (termed ONLINE-DISJ-MATRIX-BASES), the rows of the matrix are revealed in an online fashion and the online algorithm has to color each row immediately upon arrival irrevocably in order to maximize the number of spanning colors.

2. In the DISJ-SPANNING-TREES problem, we are given a connected graph $G = (V, E)$ and the goal is to find a maximum number of disjoint spanning trees in G. DISJ-SPANNING-TREES is a special case of DISJ-MATROID-BASES where the matroid is the graphic matroid defined by G (and consequently, the rank function of a subset $F \subseteq E$ is $|V| - c(V, F)$, where $c(V, F)$ is the number of components in the graph (V, F)). In the online setting of DISJ-SPANNING-TREES (termed ONLINE-DISJ-SPANNING-TREES), the vertex set of G is known in advance while the edges of G are revealed in an online fashion and the online algorithm has to color each edge immediately upon arrival irrevocably in order to maximize the number of connected colors—a color is connected if the edges of the color form a connected graph over the vertex set V.

3. In the DISJ-CONN-SPANNING-SUBHYPERGRAPHS problem, we are given a connected hypergraph $H = (V, E)$ and the goal is to find a maximum number of disjoint connected spanning subhypergraphs in H. DISJ-CONN-SPANNING-SUBHYPERGRAPHS is a special case of DISJ-POLYMATROID-BASES where the polymatroid $f : 2^E \to \mathbb{Z}_{\geq 0}$ of interest is defined as $f(A) := |V| - c(V, A)$ for every $A \subseteq E$, where $c(V, A)$ is the number of components in the hypergraph (V, F). DISJ-CONN-SPANNING-SUBHYPERGRAPHS arises in the context of packing element-disjoint Steiner trees [3]. In the online setting of DISJ-CONN-SPANNING-SUBHYPERGRAPHS (termed ONLINE-DISJ-CONN-SPANNING-SUBHYPERGRAPHS), the vertex set of G is known in advance while the hyperedges of G are revealed in an online fashion and the online algorithm has to color each hyperedge immediately upon arrival irrevocably in order to maximize the number of connected colors—a color is connected if the hyperedges of the color form a connected hypergraph over the vertex set V.

DISJ-CONN-SPANNING-SUBHYPERGRAPHS generalizes DISJ-SPANNING-TREES (in both offline and online settings): if the input hypergraph is a graph, then the problem corresponds to DISJ-SPANNING-TREES. DISJ-CONN-SPANNING-SUBHYPERGRAPHS is also closely related to DISJ-SET-COVER in the following sense: both problems are defined over hypergraphs—the latter asks for disjoint spanning subhypergraphs (which is equivalent to disjoint set covers) while the former asks for disjoint *connected* spanning subhypergraphs. DISJ-CONN-SPANNING-SUBHYPERGRAPHS generalizes DISJ-SET-COVER (in both offline and online settings) via the following approximation-preserving reduction from the latter to the former: given an instance $H = (V, E)$ of DISJ-SET-COVER, add a new vertex r and stretch every hyperedge of H to include r to obtain a new hypergraph $H' = (V + r, E')$; the maximum number of disjoint set covers in H is equal to the maximum number of disjoint connected spanning subhypergraphs in H'.

Prior work in the offline setting. DISJ-MATROID-BASES is polynomial-time solvable [4]. DISJ-SET-COVER and DISJ-CONN-SPANNING-SUBHYPERGRAPHS are $o(\log |V|)$-inapproximable and $O(\log |V|)$-approximable [6, 3]. Călinescu, Chekuri, and Vondrák [2] introduced DISJ-POLYMATROID-BASES as a unified generalization of DISJ-MATROID-BASES, DISJ-SET-COVER, and DISJ-CONN-SPANNING-SUBHYPERGRAPHS. They designed an $O(\log f(\mathcal{N}))$-approximation for DISJ-POLYMATROID-BASES by showing an approximate min-max relation for $\mathsf{opt}(f)$. Their approximate min-max relation is based on the following minimization problem:

$$k^*(f) := \min_{A \subseteq \mathcal{N}:\ f(A) < f(\mathcal{N})} \left\lfloor \frac{\sum_{e \in \mathcal{N}} (f(A + e) - f(A))}{f(\mathcal{N}) - f(A)} \right\rfloor.$$

It is easy to see that $\mathsf{opt}(f) \leq k^*(f)$ (e.g., see [2]). Moreover, if the polymatroid f is a matroid rank function, then Edmonds [4] showed that $\mathsf{opt}(f) = k^*(f)$. Edmonds' result is constructive and implies a polynomial time algorithm for DISJ-MATROID-BASES. However, for coverage functions, it is known that $\mathsf{opt}(f) \leq (1 + o(1))k^*(f)/\log f(\mathcal{N})$ [6]. Călinescu, Chekuri, and Vondrák showed that this bound is tight by giving a polynomial-time algorithm to construct $(1 - o(1))k^*(f)/\log f(\mathcal{N})$ disjoint bases (and hence, $\mathsf{opt}(f) \geq (1 - o(1))k^*(f)/\log f(\mathcal{N})$). Their algorithm unifies the approximation algorithms for DISJ-SET-COVER [6] and DISJ-CONN-SPANNING-SUBHYPERGRAPHS [3]. We state their algorithm since it will be important for the rest of our discussion: The algorithm computes the parameter $k := \lfloor k^*(f)/(\log f(\mathcal{N}) + \log \log f(\mathcal{N})) \rfloor$ (by guessing/binary search) and colors each element with a uniformly random color chosen from a color palette of size k. Călinescu, Chekuri, and Vondrák showed that the expected number of base colors returned by this random coloring algorithm is at least $(1 - e/\log f(\mathcal{N}))k = (1 - o(1))k^*(f)/\log f(\mathcal{N})$. An alternative algorithm based on a random permutation is also described in [2] that we discuss later.

3

Prior work in the online setting. ONLINE-DISJ-SET-COVER was introduced and studied by Pananjady, Bagaria, and Vaze [11] driven by applications to sensor networks, supply chain management, and crowd-sourcing platforms. In the context of DISJ-SET-COVER, the quantity $k^*(f)$ associated with the coverage function f of the hypergraph $H = (V, E)$ has a simple interpretation: it is equal to the min-degree of the hypergraph H. Pananjady, Bagaria, and Vaze showed that if the min-degree is known to the online algorithm in advance, then there is an online *deterministic* algorithm with competitive ratio $O(\log n)$, where n is the number of vertices in the input hypergraph; we note that a randomized online algorithm with the same competitive ratio is an easy consequence of the random coloring algorithm of [2] discussed above. On the lower bound side, they showed that if min-degree is *not* known in advance, then every online deterministic algorithm for DISJ-SET-COVER has competitive ratio $\Omega(n)$. Although this lower bound result seems to suggest that knowing the min-degree of the graph in advance is required to achieve any meaningful competitive ratio, two different results have overcome this seeming technical barrier by empowering the online algorithm in other ways: Firstly, Emek, Goldbraikh, and Kantor [5] designed an online randomized algorithm with *expected* competitive ratio $O(\log^2 n)$ (assuming no knowledge of the min-degree but using randomness). On the lower bound side, they showed that every online randomized algorithm has expected *impure* competitive ratio[4] $\Omega(\log n/\log\log n)$ (even with knowledge of min-degree). Secondly, Bienkowski, Byrka, and Jeż [1] designed an online deterministic algorithm with *impure* competitive ratio $O(\log^2 n)$ (assuming no knowledge of the min-degree but settling for impure competitive ratio).

For the more general problem of ONLINE-DISJ-POLYMATROID-BASES, Pananjady, Bagaria, and Vaze [11] observed that if $k^*(f)$ is known in advance, then it is possible to design a randomized online algorithm with expected competitive ratio $O(\log f(\mathcal{N}))$: indeed, the random coloring algorithm of [2] mentioned above can be implemented in the online setting using knowledge of $k^*(f)$ and it will have the stated competitive ratio (via the results of [2]). In this work, we are interested in ONLINE-DISJ-POLYMATROID-BASES in the setting where $k^*(f)$ is *not known in advance*.

Our motivations to consider ONLINE-DISJ-POLYMATROID-BASES are multifold: DISJ-MATRIX-BASES and DISJ-SPANNING-TREES are fundamental problems in linear algebra and graph algorithms respectively. We note that DISJ-MATRIX-BASES in the online arrival model is non-trivial even for 2-dimensional vectors, i.e., for $d = 2$. DISJ-CONN-SPANNING-SUBHYPERGRAPHS generalizes DISJ-SPANNING-TREES and arises in the context of packing element-disjoint Steiner trees [3]. As mentioned earlier, DISJ-CONN-SPANNING-SUBHYPERGRAPHS can also be viewed as a generalization of DISJ-SET-COVER. Although DISJ-SET-COVER has been studied in the online model, there has not been any work on DISJ-CONN-SPANNING-SUBHYPERGRAPHS in the online model.

1.1 Our Results

Throughout this work, we will denote the color palette by the set of natural numbers. There is a natural greedy algorithm for ONLINE-DISJ-POLYMATROID-BASES: initialize color $c = 1$; at each timestep $t \in [n]$: use color c for element e_t and if the set of elements with color c is a base, then increment c. For uniform random arrival order, the competitive ratio of this simple online algorithm is $O(\log f(\mathcal{N}))$ (via the results of [2]); this does not seem to have been explicitly noted in prior work. For arbitrary arrival order, the competitive ratio of this online algorithm is $k^*(f)$: the algorithm will return at least one base while the maximum number of possible bases is at most $k^*(f)$ (since $\mathsf{opt}(f) \leq k^*(f)$). It is known that $k^*(f)$ is the best possible competitive ratio of deterministic online

---

[4]An online (randomized) algorithm for DISJ-SET-COVER has *impure* competitive ratio α if the (expected) number of set covers in the online algorithm is at least $(\mathsf{opt}(H)/\alpha) - \beta$ for some $\beta > 0$ that is a function only of n, where H is the input hypergraph, n is the number of vertices in H, and $\mathsf{opt}(H)$ is the maximum number of disjoint set covers in H. Our work focuses on the case of $\beta = 0$, i.e., pure competitive ratio.

algorithms that do not have prior knowledge of $k^*(f)$ [11]. In this work, we are interested in the setting of arbitrary arrival order without prior knowledge of $k^*(f)$. For this setting, we design a randomized online algorithm with expected competitive ratio $O(\log^2 f(\mathcal{N}))$.

Theorem 1. *For DISJ-POLYMATROID-BASES, there exists a randomized online algorithm with expected competitive ratio $O(\log^2 f(\mathcal{N}))$. The runtime of the algorithm at each timestep $t \in [n]$ is $poly(t, \log f(\mathcal{N}))$.*

We recall that the best-known approximation factor for DISJ-POLYMATROID-BASES in the offline setting is $O(\log f(\mathcal{N}))$, and hence the competitive ratio of our online algorithm nearly matches that of the best possible offline algorithm. We discuss the consequences of Theorem 1 for the applications. Specializing Theorem 1 to coverage functions implies a randomized online algorithm for DISJ-SET-COVER with expected competitive ratio $O(\log^2 n)$, where n is the number of vertices of the input hypergraph, thus recovering the result of [5]. Specializing Theorem 1 to matroid rank functions implies a randomized online algorithm for DISJ-MATROID-BASES with expected competitive ratio $O(\log^2 r)$, where r is the rank of the ground set. Consequently, we obtain a randomized online algorithm with expected competitive ratio $O(\log^2 d)$ for DISJ-MATRIX-BASES where d is the dimension of the span of the input vectors and with expected competitive ratio $O(\log^2 n)$ for DISJ-SPANNING-TREES where n is the number of vertices of the input graph. Specializing Theorem 1 to the polymatroid function that arises in DISJ-CONN-SPANNING-SUBHYPERGRAPHS implies a randomized online algorithm for DISJ-CONN-SPANNING-SUBHYPERGRAPHS with expected competitive ratio $O(\log^2 n)$, where n is the number of vertices of the input hypergraph.

Our randomized online algorithm to prove Theorem 1 is based on the notion of quotients of a polymatroid. Quotients played a central role in the recent result on polymatroidal sparsification [12]. The competitive ratio analysis of our algorithm is based on novel properties of quotients which might be of independent interest. We prove Theorem 1 in two steps: firstly, we design a randomized online algorithm with competitive ratio $O(\log^2 f(\mathcal{N}))$ but as stated it needs to solve an NP-Hard problem. Next, we modify this algorithm to achieve polynomial run-time while achieving the same competitive ratio. For this, we rely on the *strength decomposition* of polymatroids which is computable in polynomial time and show a property connecting the strength decomposition to min-sized quotients. Although these algorithms are general and powerful, they are computationally expensive and difficult to interpret for specific applications. As our second result, we give a very simple and fast online randomized algorithm for DISJ-CONN-SPANNING-SUBHYPERGRAPHS and DISJ-SPANNING-TREES that achieves the same competitive ratio.

Theorem 2. *For DISJ-CONN-SPANNING-SUBHYPERGRAPHS, there exists a randomized online algorithm with competitive ratio $O(\log^2 n)$, where n is number of vertices in the input hypergraph. The runtime of the algorithm at each timestep t is $O(|e_t|^2)$, where e_t is the hyperedge that arrives at time t. In particular, the algorithm for DISJ-SPANNING-TREES can be implemented to run in constant time at each timestep.*

1.2 Technical Background and Overview of Algorithms

We recall that we assume $\mathcal{N}$ is the input sequence. Let $f : 2^{\mathcal{N}} \to \mathbb{Z}_{\geq 0}$ be the polymatroid of interest. Let $r := f(\mathcal{N})$ be the function value of bases and $k^* := k^*(f)$. For the discussion here, we will assume that $f(e) > 0$ for every $e \in \mathcal{N}$, $r \geq 2$, and $k^* = \Omega(\log^2 r)$ since this is the non-trivial case. First consider the setting where the online algorithm knows k^* in advance. In this setting, coloring each element uniformly from the palette $[\Theta(k^*/\log r)]$ gives $\Omega(k^*/\log r)$ base colors in expectation and, consequently achieves $O(\log r)$-competitive ratio.

Suppose k^* is not known in advance. Our approach is to *estimate* k^* at each timestep. At each timestep $t \in [n]$, suppose that we can find a number q_t that is within a poly(r) factor of k^*, i.e., $q_t \in [\frac{k^*}{\text{poly}(r)}, k^* \cdot \text{poly}(r)]$—we call such a q_t as a *coarse estimate*. Let P_t be a uniform random sample from $\{q_t \cdot 2^{-O(\log r)}, \ldots, q_t \cdot 2^{-1}, q_t \cdot 2^0, q_t \cdot 2^1, \ldots, q_t \cdot 2^{O(\log r)}\}$; P_t is a 2-approximation of k^* with probability $\Omega(1/\log r)$. Consequently, coloring the element e_t uniformly from the palette $[\Theta(P_t/\log^2 r)]$ leads to $\Omega(k^*/\log^2 r)$ base colors in expectation.

The key challenge is computing a coarse estimate for k^* which depends on the full input sequence. This may not be feasible at *all* timesteps owing to the limited knowledge of the polymatroid. Instead, we compute an estimate at each timestep such that the elements at timesteps which achieve the coarse property provide *a sufficiently large value of k^**. We formalize this approach now. Suppose we have a subroutine to compute some estimate q_t at each timestep $t \in [n]$. We call a timestep $t \in [n]$ to be *good* if the estimate q_t is a coarse estimate of k^* and *bad* otherwise. We let $\mathcal{N}_{\text{good}}$ be the collection of elements that arrive at good timesteps and focus on the function f restricted to $\mathcal{N}_{\text{good}}$, i.e., on the function $g := f_{|\mathcal{N}_{\text{good}}}$. Suppose that the following two properties hold: (i) Bases of g are also bases of f and (ii) $k^*(g) = \Omega(1)k^*$. These two properties suffice to obtain $\Omega(k^*/\log^2 r)$ base colors in expectation via the fine-tuned random coloring argument in the previous paragraph.

We rely on the notion of *quotients* to obtain an estimator satisfying properties (i) and (ii).

Definition 1. For a polymatroid $h : 2^{\mathcal{V}} \to \mathbb{Z}_{\geq 0}$, a set $Q \subseteq \mathcal{V}$ is a *quotient* of h if $h(e + (\mathcal{V} \setminus Q)) > h(\mathcal{V} \setminus Q)$ $\forall e \in Q$, i.e., if each element $e \in Q$ has strictly positive marginal with respect to $\mathcal{V} \setminus Q$.

Since the definition of quotients might seem contrived, we interpret it for some concrete polymatroids:

1. Let $G = (V, E)$ be a connected hypergraph (graph respectively). Consider the polymatroid $h : 2^E \to \mathbb{Z}_{\geq 0}$ that arises in DISJ-CONN-SPANNING-SUBHYPERGRAPHS (DISJ-SPANNING-TREES respectively) with G being the input instance. Quotients of h correspond to union of cut-set of disjoint subsets of vertices, i.e., each quotient $Q \subseteq E$ of h is of the form $Q = \cup_{S \in \mathcal{C}} \delta(S)$, where the family $\mathcal{C} \subseteq 2^V \setminus \{\emptyset, V\}$ is a disjoint family of subsets of vertices. In particular, the minimum size of a non-empty quotient is equal to the global min-cut value of G.

2. Let $G = (V, E)$ be a hypergraph. Consider the polymatroid $h : 2^E \to \mathbb{Z}_{\geq 0}$ that arises in DISJ-SET-COVER with G being the input instance, i.e., h is the coverage function of G. Quotients of h correspond to union of vertex-isolating cuts of G, i.e., each quotient $Q \subseteq E$ of h is of the form $Q = \cup_{u \in S} \delta(u)$ where $S \subseteq V$. In particular, the minimum size of a non-empty quotient is equal to the min-degree of G.

We observed that the minimum size of a non-empty quotient of f is an r-approximation of k^*—see Appendix A. This inspired us to use the minimum-sized quotient of $f_{|\mathcal{N}_t}$ that *contains e_t* as the estimate at each timestep $t \in [n]$. That is, we use the estimator defined as

$$q_t := \min\{|Q| : e_t \in Q \subseteq \mathcal{N}_t \text{ and } Q \text{ is a quotient of } f_{|\mathcal{N}_t}\}. \tag{1}$$

Our main technical contribution is showing that the preceding estimator satisfies the two properties that we discussed earlier. This result combined with the random coloring process yields the desired $O(\log^2 r)$-competitive ratio. A technical issue is that the estimator q_t is NP-Hard to compute for general polymatroids (even for matroid rank functions). However, we note that a poly(r)-approximation of q_t is sufficient for the desired competitive ratio. We show that an r-approximation of q_t can be computed using a *strength decomposition* with respect to $f_{|\mathcal{N}_t}$ which admits a polynomial-time algorithm.

Next, we briefly discuss the idea behind the simpler algorithm for DISJ-CONN-SPANNING-SUBHYPERGRAPHS and DISJ-SPANNING-TREES of Theorem 2. We will focus on DISJ-SPANNING-TREES here. The polymatroid $f : 2^E \to \mathbb{Z}_{\geq 0}$ of interest here is the rank function of the graphic matroid defined by the undirected graph $G = (V, E)$. For this case, the quotient computation is polynomial-time solvable since q_t is equal to the min u_t-v_t cut value in the graph (V, E_t) where $e_t = (u_t, v_t)$ and $E_t = \{e_1, e_2, \ldots, e_t\}$. However, we show that a much simpler estimator for DISJ-SPANNING-TREES also achieves properties (i) and (ii) mentioned above: we use

$$\eta_t := \text{number of edges between } u_t \text{ and } v_t \text{ in the graph } (V, E_t).$$

This estimator generalizes also to DISJ-CONN-SPANNING-SUBHYPERGRAPHS.

2 Preliminaries

Given two integers $a \leq b$, let $[a, b]$ denote the set of integers x with $a \leq x \leq b$ and $[b]$ denote the set of integers x with $1 \leq x \leq b$. The $\log(\cdot)$ operator with an unspecified base refers to $\log_2(\cdot)$. We recall that a polymatroid is an integer-valued monotone submodular function with its value on the empty set being 0. Let $f : 2^{\mathcal{N}} \to \mathbb{Z}_{\geq 0}$ be a polymatroid. We denote the value of the function f on the ground set $\mathcal{N}$ by $r := f(\mathcal{N})$ and recall that a set $S \subseteq \mathcal{N}$ is a base if $f(S) = r$. We will assume that $f(e) > 0$ for every element $e \in \mathcal{N}$: if we have an element $e \in \mathcal{N}$ with $f(e) = 0$, then $f(S) = f(S + e)$ for every $S \subseteq \mathcal{N}$ by submodularity and consequently, picking an arbitrary color for e does not influence the number of base colors.

For a subset $A \subseteq \mathcal{N}$, we define the *marginal function with respect to A* as the function $f_A : 2^{\mathcal{N}} \to \mathbb{Z}$ defined by $f_A(S) := f(A \cup S) - f(A)$ for every $S \subseteq \mathcal{N}$. If f is submodular, then the function f_A is also submodular for every $A \subseteq \mathcal{N}$. For ease of notation, for an element $e \in \mathcal{N}$, we let e denote the singleton set $\{e\}$. For every set $S \subseteq \mathcal{N}$, we use $S + e$ and $S - e$ to abbreviate $S \cup \{e\}$ and $S \setminus \{e\}$, respectively.

We need the notion of span, closed sets, and quotients. For a set $S \subseteq \mathcal{N}$, the *span* of S is the set of elements with marginal value 0 with respect to S, i.e., $\text{span}(S) := \{e \in \mathcal{N} : f_S(e) = 0\}$. We note that for every two sets $A \subseteq B$, $\text{span}(A) \subseteq \text{span}(B)$ since f is a monotone submodular function. A set $S \subseteq \mathcal{N}$ is *closed* if $S = \text{span}(S)$. A set $Q \subseteq \mathcal{N}$ is a *quotient* of f if $Q = \mathcal{N} \setminus \text{span}(S)$ for some set $S \subseteq \mathcal{N}$. This definition of quotient is equivalent to the one in Definition 1. We note that the empty set is closed (since $f(e) > 0$ for every element $e \in \mathcal{N}$) which implies that $\mathcal{N}$ is a quotient of f. The notion of quotients plays a central role in polymatroid sparsification [12].

For a set $S \subseteq \mathcal{N}$, let $f_{|S} : 2^S \to \mathbb{Z}_{\geq 0}$ be the function obtained from f by restricting the ground set to S, i.e., $f_{|S}(T) := f(T)$ for every $T \subseteq S$. We recall that if f is a polymatroid, then $f_{|S}$ is also a polymatroid for every $S \subseteq \mathcal{N}$. Restricting the ground set preserves quotients as shown in the following lemma.

Lemma 1. *[12] Let $f : 2^{\mathcal{N}} \to \mathbb{Z}_{\geq 0}$ be a polymatroid. Let Q be a quotient of f and $S \subseteq \mathcal{N}$. Then, $Q \cap S$ is a quotient of $f_{|S}$.*

We recall that

$$\text{opt}(f) = \max\{k : \exists k \text{ disjoint bases of } f\} \text{ and}$$

$$k^*(f) = \min_{A \subseteq \mathcal{N}: f(A) < f(\mathcal{N})} \left\lfloor \frac{\sum_{e \in \mathcal{N}} f_A(e)}{f(\mathcal{N}) - f(A)} \right\rfloor. \tag{2}$$

We note that there exists a closed set A that achieves the minimum in the definition of $k^*(f)$: suppose A is a set that achieves the minimum, then $f(\text{span}(A)) = f(A) < f(\mathcal{N})$ by definition

of span while $\sum_{e \in \mathcal{N}} f_A(e) \geq \sum_{e \in \mathcal{N}} f_{\text{span}(A)}(e)$ by submodularity of function f and consequently, $\lfloor \frac{\sum_{e \in \mathcal{N}} f_A(e)}{f(\mathcal{N}) - f(A)} \rfloor \geq \lfloor \frac{\sum_{e \in \mathcal{N}} f_{\text{span}(A)}(e)}{f(\mathcal{N}) - f(\text{span}(A))} \rfloor$. If the function f is clear from context, then we denote $\text{opt}(f)$ and $k^*(f)$ by opt and k^* respectively. We need the following approximate min-max relation between opt and k^*.

Theorem 3. *[2]* $\frac{k^*}{O(\log r)} \leq \text{opt} \leq k^*$.

We need the following lemma showing that for every polymatroid f, sampling each element independently with probability $\Omega\left(\frac{c \cdot \log r}{k^*}\right)$ gives a base of f with constant probability. A variant of the lemma was shown in [2]. We include a proof of the lemma in the appendix for completeness.

Lemma 2. *Let* $f : 2^{\mathcal{N}} \to \mathbb{Z}_{\geq 0}$ *be a polymatroid with* $f(\mathcal{N}) = r \geq 2$. *Let* $p := \min\{1, \frac{2 \log r}{k^*}\}$ *and* $S \subseteq \mathcal{N}$ *be a subset obtained by picking each element in* $\mathcal{N}$ *with probability at least* p *independently at random. Then,* S *is a base of* f *with probability at least* $\frac{1}{2}$.

We recall that the elements of the ground set $\mathcal{N}$ are indexed as $e_1, e_2, \ldots, e_n$ according to the arrival order, i.e., e_t is the element that arrives at time $t \in [n]$ and moreover, $\mathcal{N}_t = \{e_1, e_2, \ldots, e_t\}$ is the set of elements that have arrived until time t for every $t \in [n]$. For a coloring of all elements, we say that a fixed color is a *base color* if the set of elements of that color is a base of the polymatroid f. Let $\text{Alg}(f)$ be the number of base colors obtained by an online algorithm Alg for a given polymatroid f. A deterministic online algorithm Alg is *(purely)* α-*competitive* if for every polymatroid f, we have

$$\text{Alg}(f) \geq \frac{\text{opt}(f)}{\alpha}. \tag{3}$$

A randomized online algorithm is α-*competitive* if the bound in (3) holds in expectation.

3 An $O(\log^2 r)$-Competitive Algorithm

In this section, we present a randomized online algorithm with expected competitive ratio $O(\log^2 r)$ but the run-time of the algorithm at each timestep will be exponential. The purpose of this section is to illustrate that the online arrival order has sufficient information to achieve a reasonable competitive ratio (albeit in exponential runtime) and to highlight the main ideas underlying the eventual polynomial-time algorithm that will prove Theorem 1. We describe the algorithm in Section 3.1. We present a novel property of the sequence of minimum sized quotients containing the last element with respect to a fixed ordering in Section 3.2. We prove the structural property that focusing on elements whose estimates are within a $\text{poly}(r)$ factor of k^* does not decrease k^* by more than a constant factor in Section 3.3. We use these properties to analyze the competitive ratio of the algorithm in Section 3.4. For Sections 3.1, 3.2, 3.3, and 3.4, we assume that $r \geq 2$ and $k^* \geq 120 \log^2 r$. We discuss how to relax the assumption in Section 3.5 in which we combine the designed algorithm with algorithms for other cases to obtain an algorithm with competitive ratio $O(\log^2 r)$. We will modify the algorithm in Section 4 to design an $O(\log^2 r)$-competitive algorithm that runs in polynomial time at each timestep, thereby completing the proof of Theorem 1.

3.1 Algorithm Description

We assume that $k^* \geq 120 \log^2 r$ and $r \geq 2$. The algorithm proceeds as follows: At each timestep $t \in [n]$, the algorithm computes

$$q_t := \min\{|Q| : e_t \in Q \subseteq \mathcal{N}_t \text{ and } Q \text{ is a quotient of } f_{|\mathcal{N}_t}\}.$$

We note that $\mathcal{N}_t$ is a quotient of $f_{|\mathcal{N}_t}$ with $e_t \in \mathcal{N}_t$ and hence, q_t is well-defined. Let $\ell_t := \lceil \log q_t \rceil$. Then, the algorithm samples a value R_t from $[\ell_t - 3\lceil \log r \rceil, \ell_t + 3\lceil \log r \rceil]$ uniformly at random. Finally, the algorithm sets $C(e_t)$ to be an integer picked uniformly at random from $[\lfloor 2^{R_t}/(60 \cdot \log^2 r) \rfloor]$. We give a pseudocode of the algorithm in Algorithm 1.

Algorithm 1: Randomized Online Algorithm for DISJ-POLYMATROID-BASES

    Input: an ordering $e_1, e_2, \ldots, e_n$ of the ground set $\mathcal{N}$ where e_t is the element that arrives at time t for each $t \in [n]$; $r = f(\mathcal{N})$, and evaluation oracle access to $f_{|\mathcal{N}_t}$ at each timestep $t \in [n]$, where $f : 2^{\mathcal{N}} \to \mathbb{Z}_{\geq 0}$ is a polymatroid.
    Output: color assignment $C : \mathcal{N} \to \mathbb{Z}_+$.

1 **for** $t = 1, 2, \ldots, n$ **do**
2     $q_t \leftarrow \min\{|Q| : e_t \in Q \subseteq \mathcal{N}_t$ and Q is a quotient of $f_{|\mathcal{N}_t}\}$
3     $\ell_t \leftarrow \lceil \log q_t \rceil$
4   Sample R_t *u.a.r.* from $[\ell_t - 3\lceil \log r \rceil, \ell_t + 3\lceil \log r \rceil]$
5   Sample $C(e_t)$ *u.a.r.* from $[\lfloor 2^{R_t}/(60 \cdot \log^2 r) \rfloor]$

3.2 Ordered Min-sized Quotients Property

In this section, we show that there can be at most r distinct arrival times t with the same value of q_t.

Lemma 3. *Let $f : 2^{\mathcal{N}} \to \mathbb{Z}_{\geq 0}$ be a polymatroid over the ground set $\mathcal{N}$ with $f(e) > 0$ for every $e \in \mathcal{N}$ and $f(\emptyset) = 0$. Let $e_1, e_2, \ldots, e_n$ be an ordering of the ground set $\mathcal{N}$. For every $t \in [n]$, we define*

$$q_t := \min\left\{|Q| : e_t \in Q \subseteq \mathcal{N}_t \text{ and } Q \text{ is a quotient of } f_{|\mathcal{N}_t}\right\}.$$

Then, for every $j \in \mathbb{Z}_+$,

$$\left| \{t \in [n] : q_t = j\} \right| \leq r.$$

Proof. We note that q_t is well-defined: empty set is closed since $f(e) > 0$ for every $e \in \mathcal{N}$ and hence, $\mathcal{N}_t$ is a quotient of $f_{|\mathcal{N}_t}$ containing e_t for every $t \in [n]$. Let $j \in \mathbb{Z}_+$. Let $t_1 < t_2 < \ldots < t_\ell$ be the timesteps t with $q_t = j$. It suffices to show that $\ell \leq r$. For every $i \in [\ell]$, let Q_{t_i} be a minimum sized quotient of $f_{|\mathcal{N}_{t_i}}$ containing element e_{t_i}. We note that $|Q_{t_i}| = q_{t_i} = j$ for every $i \in [\ell]$. We define $S := \bigcup_{i=1}^{\ell} (Q_{t_i} \setminus e_{t_i})$.

We first show that $e_{t_i} \notin S$ for every $i \in [\ell]$. Suppose there exists an element $e_{t_{i_1}} \in S$. Then, there exists an index $i_2 \in [\ell]$ such that $e_{t_{i_1}} \in Q_{t_{i_2}} \setminus e_{t_{i_2}}$. Hence, $e_{t_{i_1}} \in Q_{t_{i_2}} \setminus e_{t_{i_2}} \subseteq \mathcal{N}_{t_{i_2}}$, which implies that $i_1 \leq i_2$. Since $e_{t_{i_1}} \notin Q_{t_{i_1}} \setminus e_{t_{i_1}}$, we have $i_1 \neq i_2$, which shows that $i_1 < i_2$. By Lemma 1, $Q_{t_{i_2}} \cap \mathcal{N}_{t_{i_1}}$ is a quotient of $f_{|\mathcal{N}_{t_{i_1}}}$. We note that $|Q_{t_{i_2}} \cap \mathcal{N}_{t_{i_1}}| < |Q_{t_{i_2}}| = j$ since $e_{t_{i_2}} \in Q_{t_{i_2}}$ and $e_{t_{i_2}} \notin Q_{t_{i_2}} \cap \mathcal{N}_{t_{i_1}}$. Consequently, $Q_{t_{i_2}} \cap \mathcal{N}_{t_{i_1}}$ contradicts the fact that the smallest quotient of $f_{|\mathcal{N}_{t_{i_1}}}$ containing element $e_{t_{i_1}}$ has size exactly j. Hence, $e_{t_i} \notin S$ for every $i \in [\ell]$.

We now show that $f(\mathcal{N}_{t_i} \setminus S) < f(\mathcal{N}_{t_{i+1}} \setminus S)$ for every $i \in [\ell - 1]$. Let $i \in [\ell - 1]$. Since $Q_{t_{i+1}}$ is a quotient of $\mathcal{N}_{t_{i+1}}$ containing element $e_{t_{i+1}}$, we have $f_{\mathcal{N}_{t_{i+1}} \setminus Q_{t_{i+1}}}(e_{t_{i+1}}) > 0$. Since function f is submodular, we have

$$f(\mathcal{N}_{t_{i+1}} \setminus S) - f(\mathcal{N}_{t_{i+1}-1} \setminus S) = f_{\mathcal{N}_{t_{i+1}-1} \setminus S}(e_{t_{i+1}}) \geq f_{\mathcal{N}_{t_{i+1}} \setminus Q_{t_{i+1}}}(e_{t_{i+1}}) > 0.$$

Hence,

$$f(\mathcal{N}_{t_i} \setminus S) \leq f(\mathcal{N}_{t_{i+1}-1} \setminus S) < f(\mathcal{N}_{t_{i+1}} \setminus S),$$

9

where the first inequality is by monotonicity of the function f. Thus, we have $f(\mathcal{N}_{t_i} \setminus S) < f(\mathcal{N}_{t_{i+1}} \setminus S)$ for every $i \in [\ell - 1]$.

This implies that $\{f(\mathcal{N}_{t_i} \setminus S)\}_{i=1}^{\ell}$ is a strictly increasing integer sequence. We note that $f(\mathcal{N}_{t_\ell} \setminus S) \leq r$. Also, since f is a monotone function, we have $f(\mathcal{N}_{t_1} \setminus S) \geq f(e_{t_1}) > 0$. Hence, $\ell \leq r$. $\qquad\square$

Remark 1. *For better understanding, we interpret Lemma 3 for some concrete polymatroids defined on graphs. Let $G = (V, E)$ be a connected graph with $n = |V|$ vertices and $m = |E|$ edges. Let $e_1, e_2, \ldots, e_m$ be an arbitrary ordering of the edges in E. For $t \in [m]$, we let $E_t := \{e_1, e_2, \ldots, e_t\}$.*

1. *Lemma 3 implies that there are at most $n - 1$ distinct timesteps $t \in [m]$ with minimum $u_t - v_t$ cut value in the graph (V, E_t) being exactly j for every $j \in Z_+$, where u_t, v_t are the end vertices of e_t. This follows by applying the lemma to the rank function $f : 2^E \to \mathbb{Z}_{\geq 0}$ of the graphic matroid defined by G. We recall that quotients of f correspond to union of cut-sets of disjoint subsets of vertices, i.e., each quotient $Q \subseteq E$ is of the form $Q = \cup_{S \in \mathcal{C}} \delta(S)$, where $\mathcal{C} \subseteq 2^V \setminus \{\emptyset, V\}$ is a disjoint family. Hence, q_t is the minimum $u_t - v_t$ cut value in the graph (V, E_t).*

2. *Lemma 3 implies that there are at most $m - n + 1$ distinct timesteps $t \in [m]$ at which the shortest cycle containing e_t in the graph (V, E_t) has length exactly j for every $j \in Z_+$. This follows by applying the lemma to the rank function $f : 2^E \to \mathbb{Z}_{\geq 0}$ of the cographic matroid defined by G. We recall that quotients of f correspond to union of cycles, i.e., each quotient $Q \subseteq E$ is of the form $Q = \cup_{F \subseteq \mathcal{F}} F$, where $\mathcal{F}$ is a collection of cycles in G. Hence, q_t is the length of the shortest cycle containing e_t in the graph (V, E_t).*

3.3 Structural Property

For each $t \in [n]$, we define the element e_t to be *good* if $\frac{k^*}{2r} < q_t < 2rk^*$ and *bad* otherwise. Let $\mathcal{N}_{\text{good}}$ be the set of good elements. The following is the main result of this section. It shows that bases of $f_{|\mathcal{N}_{\text{good}}}$ are also bases of f and moreover, the quantity $k^*(f_{|\mathcal{N}_{\text{good}}})$ is at least a constant fraction of $k^*(f)$.

Lemma 4. *We have that*

1. $f(\mathcal{N}_{\text{good}}) = f(\mathcal{N})$ *and*

2. $k^*(f_{|\mathcal{N}_{\text{good}}}) \geq \frac{1}{2} k^*(f)$.

We prove Lemma 4 in a series of steps. We note that an element e_t could be bad because of two reasons: either q_t is too large or it is too small. We first show that the only reason for certain elements being bad is that their q_t is too small.

Lemma 5. *Let $A \subseteq \mathcal{N}$ be a closed set of f. Let $T \geq 1$ be the smallest integer such that*

$$\sum_{e \in \mathcal{N}_T \setminus A} (f(A + e) - f(A)) \geq k^* \cdot (f(\mathcal{N}) - f(A)). \tag{4}$$

Then, for every element $e_t \in \mathcal{N}_T \setminus A$ with $t \in [T]$, we have that $q_t < 2rk^*$.

Proof. We recall that $k^* = k^*(f)$. By definition of k^*, we have

$$\sum_{e \in \mathcal{N} \setminus A} (f(A + e) - f(A)) \geq k^* \cdot (f(\mathcal{N}) - f(A)).$$

Since T is the smallest integer satisfying inequality (4), we have

$$\sum_{e \in \mathcal{N}_{T-1} \setminus A} (f(A + e) - f(A)) < k^* \cdot (f(\mathcal{N}) - f(A)). \tag{5}$$

Hence,

$$\begin{aligned}
\sum_{e \in \mathcal{N}_T \setminus A} (f(A + e) - f(A)) &= \left(\sum_{e \in \mathcal{N}_{T-1} \setminus A} (f(A + e) - f(A)) \right) + (f(A + e_T) - f(A)) \\
&< k^* \cdot (f(\mathcal{N}) - f(A)) + (f(A + e_T) - f(A)) \quad \text{(by inequality (5))} \\
&\leq k^* \cdot (f(\mathcal{N}) - f(A)) + (f(\mathcal{N}) - f(A)) \quad \text{(since } f(A + e_T) \leq f(\mathcal{N})) \\
&= (k^* + 1) \cdot (f(\mathcal{N}) - f(A)). \tag{6}
\end{aligned}$$

Since A is a closed set, we have $f(A + e) - f(A) \geq 1$ for every element $e \in \mathcal{N}_T \setminus A$. Thus, using inequality (6), we have

$$|\mathcal{N}_T \setminus A| < \sum_{e \in \mathcal{N}_T \setminus A} (f(A + e) - f(A)) \leq (k^* + 1) \cdot (f(\mathcal{N}) - f(A)) \leq (k^* + 1)r < 2rk^*. \tag{7}$$

Since A is a closed set, $\mathcal{N} \setminus A$ is a quotient of f. By Lemma 1, for every $t \in [n]$, we know that $\mathcal{N}_t \setminus A$ is a quotient of $f_{|\mathcal{N}_t}$. Hence, for every element $e_t \in \mathcal{N}_T \setminus A$ with $t \in [T]$, we have that $q_t \leq |\mathcal{N}_t \setminus A| \leq |\mathcal{N}_T \setminus A| < 2rk^*$, where the last inequality is by inequality (7). $\qquad \square$

Next, we show that dropping all bad elements will not decrease the value of k^* by more than a constant factor for all sets. We first show this for closed sets (Lemma 6) and derive it for arbitrary sets as a corollary (Corollary 1).

Lemma 6. *For every closed set $A \subseteq \mathcal{N}$,*

$$\sum_{e \in \mathcal{N}_{\text{good}} \setminus A} (f(A + e) - f(A)) \geq \frac{k^*}{2} \cdot (f(\mathcal{N}) - f(A)).$$

Proof. Let $A \subseteq \mathcal{N}$ be a closed set. Let $T \geq 1$ be the smallest integer such that inequality (4) holds. Lemma 5 implies that for every $t \in [T]$, element $e_t \in \mathcal{N}_T \setminus A$ is bad if and only if $q_t \leq \frac{k^*}{2r}$. Hence, we have

$$\begin{aligned}
|\mathcal{N}_T \setminus (A \cup \mathcal{N}_{\text{good}})| = |\{t \in [T] : e_t \in \mathcal{N}_T \setminus (A \cup \mathcal{N}_{\text{good}})\}| &= \left| \left\{ t \in [T] : e_t \in \mathcal{N}_T \setminus A \ \& \ q_t \leq \frac{k^*}{2r} \right\} \right| \\
&\leq \left| \left\{ t \in [T] : q_t \leq \frac{k^*}{2r} \right\} \right| = \sum_{j=1}^{\lfloor \frac{k^*}{2r} \rfloor} |\{t \in [T] : q_t = j\}| \\
&\leq \left(\frac{k^*}{2r} \right) r \quad \text{(by Lemma 3)} \\
&= \frac{k^*}{2}. \tag{8}
\end{aligned}$$

Thus,

$$\sum_{e\in\mathcal{N}_{\mathsf{good}}\setminus A}(f(A+e)-f(A))$$

$$\geq \sum_{e\in(\mathcal{N}_T\cap\mathcal{N}_{\mathsf{good}})\setminus A}(f(A+e)-f(A)) \quad \text{(since } f(A+e)\geq f(A) \text{ for every } e\in\mathcal{N})$$

$$= \sum_{e\in\mathcal{N}_T\setminus A}(f(A+e)-f(A)) - \sum_{e\in\mathcal{N}_T\setminus(A\cup\mathcal{N}_{\mathsf{good}})}(f(A+e)-f(A))$$

$$\geq k^*\cdot(f(\mathcal{N})-f(A)) - \sum_{e\in\mathcal{N}_T\setminus(A\cup\mathcal{N}_{\mathsf{good}})}(f(A+e)-f(A)) \quad \text{(by inequality (4))}$$

$$\geq k^*\cdot(f(\mathcal{N})-f(A)) - \sum_{e\in\mathcal{N}_T\setminus(A\cup\mathcal{N}_{\mathsf{good}})}(f(\mathcal{N})-f(A)) \quad \text{(by monotonicity of } f)$$

$$\geq k^*\cdot(f(\mathcal{N})-f(A)) - \frac{k^*}{2}\cdot(f(\mathcal{N})-f(A)) \quad \text{(by inequality (8))}$$

$$= \frac{k^*}{2}\cdot(f(\mathcal{N})-f(A)).$$

$\square$

Corollary 1. *For every set $A\subseteq\mathcal{N}$,*

$$\sum_{e\in\mathcal{N}_{\mathsf{good}}\setminus A}(f(A+e)-f(A)) \geq \frac{k^*}{2}\cdot(f(\mathcal{N})-f(A)).$$

Proof. We have

$$\sum_{e\in\mathcal{N}_{\mathsf{good}}\setminus A}(f(A+e)-f(A)) \geq \sum_{e\in\mathcal{N}_{\mathsf{good}}\setminus\mathrm{span}(A)}(f(A+e)-f(A)) \quad \text{(since } A\subseteq\mathrm{span}(A))$$

$$\geq \sum_{e\in\mathcal{N}_{\mathsf{good}}\setminus\mathrm{span}(A)}(f(\mathrm{span}(A)+e)-f(\mathrm{span}(A))$$

$$\text{(since } f_A(e)\geq f_{\mathrm{span}(A)}(e) \text{ by submodularity of } f)$$

$$\geq \frac{k^*}{2}\cdot(f(\mathcal{N})-f(\mathrm{span}(A))) \quad \text{(by applying Lemma 6 for } \mathrm{span}(A))$$

$$= \frac{k^*}{2}\cdot(f(\mathcal{N})-f(A)). \quad \text{(since } f(A)=f(\mathrm{span}(A)))$$

$\square$

We now prove Lemma 4.

Proof. We recall that $k^*=k^*(f)>0$ by assumption. By applying Corollary 1 for $A=\mathcal{N}_{\mathsf{good}}$, we conclude that $f(\mathcal{N})=f(\mathcal{N}_{\mathsf{good}})$. Hence,

$$k^*(f_{|\mathcal{N}_{\mathsf{good}}}) = \min_{A\subseteq\mathcal{N}_{\mathsf{good}}:f(A)<f(\mathcal{N}_{\mathsf{good}})}\frac{\sum_{e\in\mathcal{N}_{\mathsf{good}}}f_A(e)}{f(\mathcal{N}_{\mathsf{good}})-f(A)} = \min_{A\subseteq\mathcal{N}_{\mathsf{good}}:f(A)<f(\mathcal{N}_{\mathsf{good}})}\frac{\sum_{e\in\mathcal{N}_{\mathsf{good}}\setminus A}f_A(e)}{f(\mathcal{N})-f(A)},$$

where the last equality holds since $f_A(e)=0$ for every $e\in A$ and $f(\mathcal{N}_{\mathsf{good}})=f(\mathcal{N})$. By Corollary 1, for every set $A\subseteq\mathcal{N}$ with $f(A)<f(\mathcal{N}_{\mathsf{good}})$, we have

$$\frac{\sum_{e\in\mathcal{N}_{\mathsf{good}}\setminus A}f_A(e)}{f(\mathcal{N})-f(A)} \geq \frac{k^*}{2}.$$

Hence, $k^*(f_{|\mathcal{N}\text{good}}) \geq \frac{1}{2}k^*$. □

3.4 Competitive Ratio Analysis

Now, we analyze the competitive ratio of Algorithm 1. Let $h := \lfloor \log k^* \rfloor$. We note that $\frac{k^*}{2} < 2^h \leq k^*$. We now prove that every color $c \in [\lfloor 2^h/(60 \cdot \log^2 r) \rfloor]$ is a base color with constant probability in Algorithm 1.

Lemma 7. *Let* $c \in [\lfloor 2^h/(60 \cdot \log^2 r) \rfloor]$. *Then,* $\mathbf{Pr}_{Alg_1}[c \text{ is a base color}] \geq \frac{1}{2}$.

Proof. Let $k^*_{\text{good}} := k^*(f_{|\mathcal{N}_{\text{good}}})$. We recall that $k^* = k^*(f)$. By Lemma 4, we have that $f(\mathcal{N}_{\text{good}}) = f(\mathcal{N})$ and hence, bases of $f_{|\mathcal{N}_{\text{good}}}$ correspond to bases of f. By Lemma 4, we also have that $k^*_{\text{good}} \geq \frac{k^*}{2} \geq 2^{h-1}$. Let $e_t \in \mathcal{N}_{\text{good}}$. We have $\frac{k^*}{2r} < q_t < 2rk^*$ and hence, $\lceil \log q_t \rceil - 3\lceil \log r \rceil \leq h \leq \lceil \log q_t \rceil + 3\lceil \log r \rceil$. Therefore,

$$\mathbf{Pr}[R_t = h] = \frac{1}{6\lceil \log r \rceil + 1} > \frac{1}{15 \log r}. \tag{9}$$

Hence, for every element $e_t \in \mathcal{N}_{\text{good}}$, the probability that e_t is colored with c is

$$\mathbf{Pr}[C(e_t) = c] \geq \mathbf{Pr}[R_t = h] \cdot \mathbf{Pr}[C(e_t) = c | r_t = h]$$
$$= \mathbf{Pr}[R_t = h] \cdot \frac{1}{\lfloor 2^h/(60 \cdot \log^2 r) \rfloor}$$
$$> \frac{1}{15 \log r} \cdot \frac{60 \cdot \log^2 r}{2^h} \quad \text{(by inequality (9))}$$
$$\geq \frac{1}{15 \log r} \cdot \frac{60 \cdot \log^2 r}{2k^*_{\text{good}}} = \frac{2 \log r}{k^*_{\text{good}}}.$$

We recall that $k^*_{\text{good}} = k^*(f_{|\mathcal{N}_{\text{good}}})$. By applying Lemma 2 for the polymatroid $f_{|\mathcal{N}_{\text{good}}}$, we conclude that the elements in $\mathcal{N}_{\text{good}}$ with color c form a base of $f_{|\mathcal{N}_{\text{good}}}$ with probability at least $\frac{1}{2}$. We recall that bases of $f_{|\mathcal{N}_{\text{good}}}$ are also bases of f. Hence, c is a base color with probability at least $\frac{1}{2}$. □

Lemma 7 implies a lower bound on the expected number of base colors obtained by Algorithm 1.

Corollary 2. *The expected number of base colors obtained by Algorithm 1 is at least*

$$\mathbb{E}[Alg_1(f)] \geq \frac{1}{2} \cdot \lfloor 2^h/(60 \cdot \log^2 r) \rfloor.$$

Proof. By Lemma 7, we have that

$$\mathbb{E}[Alg_1(f)] = \sum_{c \in [\lfloor 2^h/(60 \cdot \log^2 r) \rfloor]} \mathbf{Pr}_{Alg_1}[c \text{ is a base color}] \geq \frac{1}{2} \cdot \lfloor 2^h/(60 \cdot \log^2 r) \rfloor.$$

□

3.5 Combined Algorithm

We recall that our analysis of Algorithm 1 assumed that $r \geq 2$ and $k^* \geq 120 \log^2 r$. We now combine Algorithm 1 with other algorithms to address all ranges of r and k^*.

Consider the online algorithm Alg_1^* that runs Algorithm 1 with probability $\frac{1}{3}$, assigns $C(e_t) = 1$ for every element e_t with probability $\frac{1}{3}$, and assigns $C(e_t) = t$ for every element e_t with probability $\frac{1}{3}$. We now show that the resulting online algorithm Alg_1^* has competitive ratio $O(\log^2 r)$.

Theorem 4. *Algorithm* $\mathtt{Alg}_1^*$ *has competitive ratio* $O(\log^2 r)$.

Proof. Suppose $r = 1$. Then, each singleton forms a base, which implies that $\mathtt{opt}(f) = n$. We recall that the algorithm assigns $C(e_t) = t$ for every element e_t with probability $\frac{1}{3}$. This implies that

$$\mathbb{E}[\mathtt{Alg}_1^*(f)] \geq \frac{1}{3} \cdot \mathtt{opt}(f).$$

Suppose $r \geq 2$ and $k^* < 120 \log^2 r$. Then, the optimum is also smaller than $120 \log^2 r$ by Theorem 3. We recall that the algorithm assigns $C(e_t) = 1$ for every element e_t with probability $\frac{1}{3}$. This implies that

$$\mathbb{E}[\mathtt{Alg}_1^*(f)] \geq \frac{1}{3} > \frac{1}{360 \log^2 r} \cdot \mathtt{opt}(f).$$

Now, we may assume that $k^* \geq 120 \log^2 r$ and $r \geq 2$. We recall that the algorithm $\mathtt{Alg}_1^*$ runs Algorithm 1 with probability $\frac{1}{3}$. We have

$$2^h/(60 \cdot \log^2 r) > \frac{1}{60} \cdot \frac{k^*}{2 \log^2 r} \geq 1, \tag{10}$$

which implies that

$$\lfloor 2^h/(60 \cdot \log^2 r) \rfloor \geq \frac{1}{2} \cdot 2^h/(60 \cdot \log^2 r). \tag{11}$$

Hence, we have

$$\begin{aligned}
\mathbb{E}[\mathtt{Alg}_1^*(f)] &\geq \frac{1}{3} \cdot \mathbb{E}[\mathtt{Alg}_1(f)] \\
&\geq \frac{1}{6} \cdot \lfloor 2^h/(60 \cdot \log^2 r) \rfloor \quad \text{(by Corollary 2)} \\
&\geq \frac{1}{12} \cdot 2^h/(60 \cdot \log^2 r) \quad \text{(by inequality (11))} \\
&\geq \frac{1}{1440 \cdot \log^2 r} \cdot k^* \\
&\geq \frac{1}{1440 \cdot \log^2 r} \cdot \mathtt{opt}(f). \quad \text{(by Theorem 3)}
\end{aligned}$$

$\square$

Remark 2. *We recall that we assumed $k^* \geq 120 \log^2 r$ and $r \geq 2$ in Sections 3.1, 3.2, 3.3, and 3.4. The first assumption is to ensure the correctness of inequality (10), which is used in the last case of the proof for Theorem 4. The second assumption is used for inequality (9) in the proof for Lemma 7.*

4 Approximation of q_t in Polynomial Time

In Section 3, we presented a randomized online algorithm with competitive ratio $O(\log^2 r)$. The algorithm takes exponential time, since the parameter $q_t = \min\{|Q| : e_t \in Q \subseteq \mathcal{N}_t$ and Q is a quotient of $f_{|\mathcal{N}_t}\}$ is NP-hard to compute[5]. In this section, we show how to get an approximation of q_t in polynomial time via *strength decomposition* of the function f and complete the proof of Theorem 1.

---

[5]The problem of computing q_t for f being a matroid rank function is NP-hard. We note that the problem is equivalent to computing a min-sized cocircuit of a matroid containing a specified element. Computing a min-sized cocircuit of a matroid is a NP-hard problem [14] that reduces to finding a min-sized cocircuit of a matroid containing a specified element (an algorithm for the latter problem can be applied for each choice of the element to solve the former problem).

4.1 Strength Decomposition

We first introduce the definition of strength decomposition. Let $f : 2^{\mathcal{N}} \to \mathbb{Z}_{\geq 0}$ be a polymatroid over the ground set $\mathcal{N}$. For a subset T of S, the *strength-ratio* of T in S, denoted $\varphi(T|S)$, is the value

$$\varphi(T|S) := \frac{|S| - |T|}{f(S) - f(T)},$$

with the convention that $x/0 = +\infty$ for every $x \geq 0$.

Definition 2. [12] Let $f : 2^{\mathcal{N}} \to \mathbb{Z}_{\geq 0}$ be a polymatroid over the ground set $\mathcal{N}$. A *strength decomposition* of $\mathcal{N}$ with respect to f is a sequence of sets $S_0 \supseteq S_1 \supseteq \ldots \supseteq S_w$ such that:

(1) $S_0 = \mathcal{N}$,

(2) $S_w = \emptyset$ and $S_i \neq \emptyset$ for every $i \in [0, w-1]$,

(3) For every $i \in [w]$, $S_i = \underset{S \subseteq S_{i-1}}{argmin} \; \varphi(S|S_{i-1})$, and

(4) The strength ratios $\varphi(S_i|S_{i-1})$ are nondecreasing in i.

For every polymatroid $f : 2^{\mathcal{N}} \to \mathbb{Z}_{\geq 0}$ over a ground set $\mathcal{N}$, there exists a strength decomposition of $\mathcal{N}$ with respect to f and it can be computed in polynomial time via reduction to submodular minimization [9, 10, 8] (also see [12]). We summarize this result below.

Theorem 5. *Given a polymatroid $f : 2^{\mathcal{N}} \to \mathbb{Z}_{\geq 0}$ over ground set $\mathcal{N}$ via an evaluation oracle, a strength decomposition of $\mathcal{N}$ with respect to f exists and can be computed in polynomial time.*

We show that the function value of sets in a strength decomposition sequence is strictly decreasing.

Lemma 8. *Let $f : 2^{\mathcal{N}} \to \mathbb{Z}_{\geq 0}$ be a polymatroid over the ground set $\mathcal{N}$. Let $\mathcal{N} = S_0 \supseteq S_1 \supseteq \ldots \supseteq S_w = \emptyset$ be a strength decomposition with respect to f. Then, $\{f(S_i)\}_{i=0}^{w}$ is a strictly decreasing integer sequence.*

Proof. Integrality of the sequence follows since f is a polymatroid. We now show that the sequence is strictly decreasing. We fix an index $i \in [w]$ and prove that $f(S_{i-1}) > f(S_i)$. Since $S_{i-1} \neq \emptyset$ and $f(e) > 0$ for every element $e \in \mathcal{N}$, we have $f(S_{i-1}) > 0$. We recall that $S_i = \underset{S \subseteq S_{i-1}}{argmin} \; \varphi(S|S_{i-1})$, which implies that $\varphi(S_i|S_{i-1}) \leq \varphi(\emptyset|S_{i-1})$. Hence,

$$\frac{|S_{i-1}| - |S_i|}{f(S_{i-1}) - f(S_i)} = \varphi(S_i|S_{i-1}) \leq \varphi(\emptyset|S_{i-1}) = \frac{|S_{i-1}|}{f(S_{i-1})} < +\infty. \tag{12}$$

The last inequality is by $f(S_{i-1}) > 0$. If $f(S_{i-1}) = f(S_i)$, then $\frac{|S_{i-1}| - |S_i|}{f(S_{i-1}) - f(S_i)} = +\infty$, which contradicts inequality (12). Thus, $f(S_{i-1}) \neq f(S_i)$. We recall that f is a monotone function and $S_i \subseteq S_{i-1}$ by definition of the strength decomposition. Therefore, $f(S_{i-1}) > f(S_i)$. $\qquad\square$

We use Lemma 8 to conclude that all sets in a strength decomposition are closed.

Lemma 9. *Let $f : 2^{\mathcal{N}} \to \mathbb{Z}_{\geq 0}$ be a polymatroid over the ground set $\mathcal{N}$. Let $\mathcal{N} = S_0 \supseteq S_1 \supseteq \ldots \supseteq S_w = \emptyset$ be a strength decomposition with respect to f. Then, S_i is a closed set for every $i \in [0, w]$.*

15

Proof. We fix an integer $i \in [0, w]$ and consider the set S_i. Let $e \in \mathcal{N} \setminus S_i$. Then, there exists an index $j \in [0, i]$ with $e \in S_{j-1} \setminus S_j$. If $f_{S_j}(e) = f(S_j + e) - f(S_j) = 0$, then

$$\begin{aligned}
\varphi(S_j | S_{j-1}) &= \frac{|S_{j-1}| - |S_j|}{f(S_{j-1}) - f(S_j)} \\
&> \frac{|S_{j-1}| - |S_j + e|}{f(S_{j-1}) - f(S_j)} \quad (\text{since } f(S_{j-1}) > f(S_j) \text{ by Lemma } 8) \\
&= \frac{|S_{j-1}| - |S_j + e|}{f(S_{j-1}) - f(S_j + e)} \quad (\text{since } f(S_j) = f(S_j + e)) \\
&= \varphi((S_j + e) | S_{j-1}),
\end{aligned}$$

which leads to a contradiction that $S_j = \underset{S \subseteq S_{j-1}}{argmin} \; \varphi(S|S_{j-1})$. Hence, $f_{S_j}(e) > 0$ for every $e \in \mathcal{N} \setminus S_i$. By submodularity of the function f, we have $f_{S_i}(e) \geq f_{S_j}(e) > 0$ for every $e \in \mathcal{N} \setminus S_i$, which implies that $\texttt{span}(S_i) = S_i$. $\qquad\square$

4.2 Algorithm Description

Now, we present our online algorithm with competitive ratio $O(\log^2 r)$ in polynomial time. Let us first assume that $k^* \geq 120 \log^2 r$ and $r \geq 2$. When the element e_t arrives, instead of computing q_t, the algorithm computes a strength decomposition $\mathcal{N}_t = S_0 \supseteq S_1 \supseteq \ldots \supseteq S_w = \emptyset$ with respect to $f_{|\mathcal{N}_t}$. Let $i \in [w]$ be the index such that $e_t \in S_{i-1} \setminus S_i$. The algorithm computes parameter $\eta_t := \frac{|\mathcal{N}_t \setminus S_i|}{f(\mathcal{N}_t) - f(S_i)}$. Let $\ell_t := \lceil \log \eta_t \rceil$. Then, it samples variable R_t from $[\ell_t - 3\lceil \log r \rceil, \ell_t + 3\lceil \log r \rceil]$ uniformly at random. Finally, the algorithm sets $C(e_t)$ to be an integer picked uniformly at random from $[\lfloor 2^{R_t}/(60 \cdot \log^2 r) \rfloor]$. We give a pseudocode of the algorithm in Algorithm 2.

Algorithm 2: Randomized Online Algorithm for DISJ-POLYMATROID-BASES

Input: an ordering $e_1, e_2, \ldots, e_n$ of the ground set $\mathcal{N}$ where e_t is the element that arrives at time t for each $t \in [n]$; $r = f(\mathcal{N})$, and evaluation oracle access to $f_{|\mathcal{N}_t}$ at each timestep $t \in [n]$, where $f : 2^{\mathcal{N}} \to \mathbb{Z}_{\geq 0}$ is a polymatroid.

Output: color assignment $C : \mathcal{N} \to \mathbb{Z}_+$.

1 **for** $t = 1, 2, \ldots, n$ **do**
2   compute a strength decomposition $\mathcal{N}_t = S_0 \supseteq S_1 \supseteq \ldots \supseteq S_w = \emptyset$ with respect to $f_{|\mathcal{N}_t}$
3      $\eta_t \leftarrow \frac{|\mathcal{N}_t \setminus S_i|}{f(\mathcal{N}_t) - f(S_i)}$ where $e_t \in S_{i-1} \setminus S_i$
4      $\ell_t \leftarrow \lceil \log q_t \rceil$
5   Sample R_t *u.a.r.* from $[\ell_t - 3\lceil \log r \rceil, \ell_t + 3\lceil \log r \rceil]$
6   Sample $C(e_t)$ *u.a.r.* from $[\lfloor 2^{R_t}/(60 \cdot \log^2 r) \rfloor]$
7   **return** $C(e_t)$

4.3 Relating η_t and q_t

We recall that the estimate used in Algorithm 1 is

$$q_t = \min\{|Q| : e_t \in Q \subseteq \mathcal{N}_t \text{ and } Q \text{ is a quotient of } f_{|\mathcal{N}_t}\}.$$

In contrast, the estimate used in Algorithm 2 is η_t. We show that the parameter η_t is an r-approximation of q_t.

Lemma 10. *For every $t \in [n]$, $\frac{q_t}{r} \leq \eta_t \leq q_t$.*

Proof. We first show the lower bound on η_t. Let $\mathcal{N}_t = S_0 \supseteq S_1 \supseteq \ldots \supseteq S_w = \emptyset$ be a strength decomposition with respect to $f_{|\mathcal{N}_t}$ computed by Algorithm 2. Let $i \in [0, w]$ be the index such that $e_t \in S_{i-1} \setminus S_i$. We recall that $\eta_t = \frac{|\mathcal{N}_t \setminus S_i|}{f(\mathcal{N}_t) - f(S_i)} \geq \frac{|\mathcal{N}_t \setminus S_i|}{r}$ since $f(\mathcal{N}_t) - f(S_i) \leq f(\mathcal{N}) \leq r$. By Lemma 9, S_i is a closed set. Consequently, $\mathcal{N}_t \setminus S_i$ is a quotient of $f_{|\mathcal{N}_t}$ containing element e_t since $e_t \in S_{i-1} \setminus S_i$. This implies that $q_t \leq |\mathcal{N}_t \setminus S_i|$ and hence, $\eta_t \geq \frac{q_t}{r}$.

We now show the upper bound on η_t. Let Q be a minimum sized quotient of $f_{|\mathcal{N}_t}$ containing e_t with $|Q| = q_t$. We recall that

$$\eta_t = \frac{|\mathcal{N}_t \setminus S_i|}{f(\mathcal{N}_t) - f(S_i)} = \frac{\sum_{j=1}^{i} |S_{j-1} \setminus S_j|}{\sum_{j=1}^{i} (f(S_{j-1}) - f(S_j))}.$$

For every $j \in [i]$, $\frac{|S_{j-1} \setminus S_j|}{f(S_{j-1}) - f(S_j)} = \varphi(S_j | S_{j-1}) \leq \varphi(S_i | S_{i-1})$, where the last inequality is because the strength ratios $\varphi(S_i | S_{i-1})$ in a strength decomposition are nondecreasing in i. Therefore, we have

$$
\begin{aligned}
\eta_t &= \frac{\sum_{j=1}^{i} |S_{j-1} \setminus S_j|}{\sum_{j=1}^{i} (f(S_{j-1}) - f(S_j))} \\
&\leq \frac{\sum_{j=1}^{i} (\varphi(S_i | S_{i-1}) \cdot (f(S_{j-1}) - f(S_j)))}{\sum_{j=1}^{i} (f(S_{j-1}) - f(S_j))} \quad \left(\text{since } \frac{|S_{j-1} \setminus S_j|}{f(S_{j-1}) - f(S_j)} \leq \varphi(S_i | S_{i-1})\right) \\
&= \varphi(S_i | S_{i-1}) \\
&\leq \varphi((S_{i-1} \setminus Q) | S_{i-1}) \quad \left(\text{since } S_i = \underset{S \subseteq S_{i-1}}{argmin}\ \varphi(S | S_{i-1})\right) \\
&= \frac{|S_{i-1}| - |S_{i-1} \setminus Q|}{f(S_{i-1}) - f(S_{i-1} \setminus Q)}.
\end{aligned}
$$

Since Q is a quotient of $f_{|\mathcal{N}_t}$, we have $f_{\mathcal{N}_t \setminus Q}(Q) \geq f_{\mathcal{N}_t \setminus Q}(e_t) > 0$. Hence, by submodularity of the function f, $f_{S_{i-1} \setminus Q}(Q) \geq f_{\mathcal{N}_t \setminus Q}(Q) > 0$ since $S_{i-1} \subseteq \mathcal{N}_t$. Consequently, $f(S_{i-1}) - f(S_{i-1} \setminus Q) = f_{S_{i-1} \setminus Q}(Q) \geq 1$. This shows that

$$\eta_t \leq \frac{|S_{i-1}| - |S_{i-1} \setminus Q|}{f(S_{i-1}) - f(S_{i-1} \setminus Q)} \leq |S_{i-1}| - |S_{i-1} \setminus Q| = |S_{i-1} \cap Q| \leq |Q| = q_t.$$

$\square$

4.4 Competitive Ratio Analysis

Now, we analyze the competitive ratio of Algorithm 2. We recall that $h = \lfloor \log k^* \rfloor$ and $\frac{k^*}{2} < 2^h \leq k^*$. We now prove that every color $c \in [\lfloor 2^h/(60 \cdot \log^2 r) \rfloor]$ is a base color with constant probability.

Lemma 11. *Let $c \in [\lfloor 2^h/(60 \cdot \log^2 r) \rfloor]$. Then, $\mathbf{Pr}_{Alg_2}[c \text{ is a base color}] \geq \frac{1}{2}$.*

Proof. Let $k^*_{\text{good}} := k^*(f_{|\mathcal{N}_{\text{good}}})$. We recall that $k^* = k^*(f)$. By Lemma 4, we have that $f(\mathcal{N}_{\text{good}}) = f(\mathcal{N})$ and hence, bases of $f_{|\mathcal{N}_{\text{good}}}$ correspond to bases of f. By Lemma 4, we also have that $k^*_{\text{good}} \geq \frac{k^*}{2} \geq 2^{h-1}$. Let $e_t \in \mathcal{N}_{\text{good}}$. We have $\frac{k^*}{2r} < q_t < 2rk^*$. By Lemma 10, we have $\frac{k^*}{2r^2} < \frac{q_t}{r} \leq \eta_t \leq q_t < 2rk^*$ and hence, $\lceil \log \eta_t \rceil - 3\lceil \log r \rceil \leq h \leq \lceil \log \eta_t \rceil + 3\lceil \log r \rceil$, which implies that

$$\mathbf{Pr}[R_t = h] = \frac{1}{6\lceil \log r \rceil + 1} > \frac{1}{15 \log r}. \tag{13}$$

17

Hence, for every element $e_t \in \mathcal{N}_{\mathsf{good}}$, the probability that e_t is colored with c is

$$
\begin{aligned}
\mathbf{Pr}[C(e_t) = c] &\geq \mathbf{Pr}[R_t = h] \cdot \mathbf{Pr}[C(e_t) = c | R_t = h] \\
&= \mathbf{Pr}[R_t = h] \cdot \frac{1}{\lfloor 2^h / (60 \cdot \log^2 r) \rfloor} \\
&> \frac{1}{15 \log r} \cdot \frac{60 \cdot \log^2 r}{2^h} \quad \text{(by inequality (13))} \\
&\geq \frac{1}{15 \log r} \cdot \frac{60 \cdot \log^2 r}{2 k^*_{\mathsf{good}}} = \frac{2 \log r}{k^*_{\mathsf{good}}}.
\end{aligned}
$$

By applying Lemma 2 for the polymatroid $f_{|\mathcal{N}_{\mathsf{good}}}$, we conclude that elements in $\mathcal{N}_{\mathsf{good}}$ with color c form a base of $f_{|\mathcal{N}_{\mathsf{good}}}$ with probability at least $\frac{1}{2}$. We recall that bases of $f_{|\mathcal{N}_{\mathsf{good}}}$ are also bases of f. Hence, c is a base color with probability at least $\frac{1}{2}$. $\qquad\square$

The lemma above implies a lower bound on the expected number of base colors obtained by Algorithm 2.

Corollary 3. *The expected number of base colors obtained by Algorithm 2 is at least*

$$
\mathbb{E}[\mathit{Alg}_2(f)] \geq \frac{1}{2} \cdot \lfloor 2^h / (60 \cdot \log^2 r) \rfloor.
$$

Proof. By Lemma 11, we have that

$$
\mathbb{E}[\mathtt{Alg}_2(f)] = \sum_{c \in [\lfloor 2^h / (60 \cdot \log^2 r) \rfloor]} \mathbf{Pr}_{\mathtt{Alg}_2}[c \text{ is a base color}] \geq \frac{1}{2} \cdot \lfloor 2^h / (60 \cdot \log^2 r) \rfloor.
$$

$\qquad\square$

4.5 Combined Algorithm

We recall that our analysis of Algorithm 2 assumed that $r \geq 2$ and $k^* \geq 120 \log^2 r$. We now combine Algorithm 2 with other algorithms to address all ranges of r and k^*.

Consider the online algorithm $\mathtt{Alg}_2^*$ that runs Algorithm 2 with probability $\frac{1}{3}$, assigns $C(e_t) = 1$ for every element e_t with probability $\frac{1}{3}$, and assigns $C(e_t) = t$ for every element e_t with probability $\frac{1}{3}$. We now show that the online algorithm $\mathtt{Alg}_2^*$ has competitive ratio $O(\log^2 r)$.

Theorem 6. *Algorithm Alg_2^* has competitive ratio $O(\log^2 r)$.*

Proof. Suppose $r = 1$. Then, each singleton forms a base, which implies that $\mathsf{opt}(f) = n$. We recall that the algorithm assigns $C(e_t) = t$ for every element e_t with probability $\frac{1}{3}$. This implies that

$$
\mathbb{E}[\mathtt{Alg}_2^*(f)] \geq \frac{1}{3} \cdot \mathsf{opt}(f).
$$

Suppose $r \geq 2$ and $k^* < 120 \log^2 r$. Then, the optimum is also smaller than $120 \log^2 r$ by Theorem 3. We recall that the algorithm assigns $C(e_t) = 1$ for every element e_t with probability $\frac{1}{3}$. This implies that

$$
\mathbb{E}[\mathtt{Alg}_2^*(f)] \geq \frac{1}{3} > \frac{1}{360 \log^2 r} \cdot \mathsf{opt}(f).
$$

Now, we may assume that $k^* \geq 120 \log^2 r$ and $r \geq 2$. We recall that the algorithm $\mathtt{Alg}^*$ runs Algorithm 2 with probability $\frac{1}{3}$. We have

$$2^h/(60 \cdot \log^2 r) > \frac{1}{60} \cdot \frac{k^*}{2 \log^2 r} \geq 1,$$

which implies that

$$\lfloor 2^h/(60 \cdot \log^2 r) \rfloor \geq \frac{1}{2} \cdot 2^h/(60 \cdot \log^2 r). \tag{14}$$

Hence, we have

$$\begin{aligned} \mathbb{E}[\mathtt{Alg}_2^*(f)] &\geq \frac{1}{3} \cdot \mathbb{E}[\mathtt{Alg}_2(f)] \\ &\geq \frac{1}{6} \cdot \lfloor 2^h/(60 \cdot \log^2 r) \rfloor \quad \text{(by Corollary 3)} \\ &\geq \frac{1}{12} \cdot 2^h/(60 \cdot \log^2 r) \quad \text{(by inequality (14))} \\ &\geq \frac{1}{1440 \cdot \log^2 r} \cdot k^* \\ &\geq \frac{1}{1440 \cdot \log^2 r} \cdot \mathtt{opt}(f). \quad \text{(by Theorem 3)} \end{aligned}$$

$\square$

Theorem 6 completes the proof of Theorem 1 since Algorithm 2 and Algorithm $\mathtt{Alg}_2^*$ can be implemented in polynomial time by Theorem 5.

5 Simpler Algorithm for Online DISJ-CONN-SPANNING-SUBHYPERGRAPHS

In this section, we consider DISJ-CONN-SPANNING-SUBHYPERGRAPHS. The input here is a connected hypergraph $G = (V, E)$ where each hyperedge $e \in E$ is a non-empty subset of V. We use $n := |V|$ and $m := |E|$ to denote the number of vertices and hyperedges in the hypergraph respectively. We note that E is a multi-set, i.e., it can contain multiple copies of the same hyperedge. We recall that the hypergraph $G = (V, E)$ can equivalently be represented as a bipartite graph $G' = (V \cup E, E')$, where we have a node for each vertex $u \in V$ and each hyperedge $e \in E$ and an edge between nodes $u \in V$ and $e \in E$ if the hyperedge e contains the vertex u. A hypergraph is *connected* if its bipartite representation is connected. A subhypergraph of G is a subset of hyperedges $F \subseteq E$ and it is said to be *spanning* if $\cup_{e \in F} e = V$ and *connected* if (V, F) is connected. The goal is to find a maximum number of disjoint connected spanning subhypergraphs. We define $\mathtt{opt}(G) := \max\{k : \exists\, k \text{ disjoint spanning hypergraphs}\}$. We note that DISJ-CONN-SPANNING-SUBHYPERGRAPHS is a special case of DISJ-POLYMATROID-BASES with the polymatroid $f : 2^E \to \mathbb{Z}_{\geq 0}$ of interest being defined as $f(A) := |V| - c(V, A)$ for every $A \subseteq E$, where $c(V, A)$ is the number of components in the bipartite representation of the hypergraph (V, F).

In the online model for DISJ-CONN-SPANNING-SUBHYPERGRAPHS, the set V of nodes is known in advance while the hyperedges E arrive in an online fashion. For a color assignment $C : E \to \mathbb{Z}_+$, we say that a color $c \in \mathbb{Z}_+$ is a *connected spanning color* if the set of hyperedges with color c is spanning and connected. An online algorithm has to color each hyperedge immediately upon arrival irrevocably in order to maximize the number of connected spanning colors. For an online algorithm $\mathtt{Alg}$, we denote the number of connected spanning colors obtained by an algorithm on the hypergraph G by $\mathtt{Alg}(G)$.

19

Theorem 1 implies a randomized online algorithm for this problem with competitive ratio $O(\log^2 n)$ that runs in polynomial time. We recall that Algorithm 2 needs to compute a strength decomposition of the current hypergraph upon the arrival of each hyperedge. This step is computationally expensive since strength decompositions are obtained via reductions to submodular minimization (in the context of DISJ-CONN-SPANNING-SUBHYPERGRAPHS, we need repeated application of max-flow). We also recall that Algorithm 1 needs to compute a minimum s-t cut of the current hypergraph upon the arrival of each hyperedge, so we need to solve max-flow at each timestep. In this section, we design a simple and fast randomized online algorithm with the same competitive ratio for DISJ-CONN-SPANNING-SUBHYPERGRAPHS.

We introduce necessary notations and background in Section 5.1. Then, we design an $O(\log^2 n)$ competitive algorithm under the assumption that $\texttt{opt}(G) \geq 80 \cdot \log^2 n$. We describe the algorithm and prove the structural property in Section 5.2. We prove the structural property that focusing on hyperedges with good estimate does not decrease the minimum cut value by more than a constant factor in Section 5.3. We analyze the competitive ratio of the algorithm in Section 5.4. For Sections 5.2, 5.3, and 5.4, we assume that $\texttt{opt}(G) \geq 80 \cdot \log^2 n$. We discuss in Section 5.5 how to relax the assumption by combining the designed algorithm with algorithms for the other case to complete the proof of Theorem 2.

5.1 Preliminaries

Let $G = (V, E)$ be a connected hypergraph. Let $n := |V|$ and $m := |E|$. For a subset $S \subseteq V$ of vertices, let $\delta(S)$ denote the set of hyperedges intersecting both S and $V \setminus S$. The *value of a global minimum cut of G*, denoted $\lambda(G)$, is the minimum number of hyperedges whose removal disconnects G, i.e., $\lambda(G) := \min_{\emptyset \neq S \subsetneq V} |\delta(S)|$. We note that $\texttt{opt}(G) \leq \lambda(G)$ since for every $\emptyset \neq S \subsetneq V$, every connected spanning color c should contain at least one hyperedge in $\delta(S)$. We recall that DISJ-CONN-SPANNING-SUBHYPERGRAPHS for the input hypergraph G is a special case of DISJ-POLYMATROID-BASES with the polymatroid $f : 2^E \to \mathbb{Z}_{\geq 0}$ of interest being defined as $f(A) := |V| - c(V, A)$ for every $A \subseteq E$, where $c(V, A)$ is the number of components in the bipartite representation of the hypergraph (V, F). We recall that $f(E) = n - 1$ since G is connected. Moreover, bases of f correspond to connected spanning subhypergraphs of G. We define the *weak partition connectivity of G* as

$$k_G^* := k^*(f) = \min_{A \subseteq E:\ f(A) < f(E)} \left\lfloor \frac{\sum_{e \in E}(f(A + e) - f(A))}{f(E) - f(A)} \right\rfloor.$$

Weak partition connectivity was studied in hypergraph literature owing to its applications to hypergraph orientation problems [7]. The terminology of weak partition connectivity defined in the literature is equivalent to the one defined here. By applying Lemma 2 to the polymatroid f, we have the following lemma.

Lemma 12. *Let $G = (V, E)$ be a connected hypergraph, $n := |V|$, and $p := \min\left\{1, \frac{2\log n}{k_G^*}\right\}$. Let $H = (V, F)$ be a random subhypergraph of G whose hyperedges F are obtained by picking each hyperedge in E with probability at least p independently at random. Then, H is a connected spanning subhypergraph of G with probability at least $\frac{1}{2}$.*

We also need the following relationship between weak partition connectivity and global min-cut in hypergraphs:

Lemma 13. *[2] $\lambda(G)/2 \leq k_G^* \leq \lambda(G)$.*

For every integer $t \in [m]$, let e_t be the hyperedge arriving at time t. Define $E_t := \{e_1, \ldots, e_t\}$ as the set of hyperedges that have arrived until time t. For a subset $S \subseteq V$ of nodes, let

$$E_t(S) := \{e \in E_t : S \subseteq e\}$$

be the set of hyperedges containing all vertices in S that have arrived until time t. Let $\delta_t(S)$ be the set of hyperedges in $\delta(S)$ that have arrived until time t.

5.2 Algorithm Description

We recall that $\mathtt{opt}(G) \geq 80 \cdot \log^2 n$. The algorithm works as follows: For every $t \in [m]$, when the hyperedge e_t arrives, the algorithm computes

$$\eta_t := \min_{\{u,v\} \in e_t} |E_t(\{u, v\})|.$$

Equivalently, η_t is the min over all pairs of vertices in the current hyperedge of the number of hyperedges containing the pair in the current hypergraph. Let $\ell_t := \lceil \log \eta_t \rceil$. Then, the algorithm samples variable R_t from $[\ell_t, \ell_t + 2\lceil \log n \rceil]$ uniformly at random. Finally, the color of the hyperedge $C(e_t)$ is assigned to be an integer picked uniformly at random from $[\lfloor 2^{R_t}/(40 \cdot \log^2 n) \rfloor]$. We give a pseudocode of the algorithm in Algorithm 3. We note that $|E_t(\{u, v\})|$ for every $u, v \in V$ can be maintained in $O(|e_t|^2)$ time in the t-th timestep, which implies that Algorithm 3 takes $O(|e_t|^2)$ time in the t-th timestep.

Algorithm 3: Randomized Online Algorithm for DISJ-CONN-SPANNING-SUBHYPERGRAPHS

    Input: a hypergraph $G = (V, E)$ with n vertices and m hyperedges; an ordering
        $e_1, e_2, \ldots, e_m$ of set E where e_t is the hyperedge that arrives at time t for each
        $t \in [n]$.
    Output: color assignment $C : E \to \mathbb{Z}_+$.
1 **When** *the hyperedge e_t arrives* **:**
2 $\eta_t \leftarrow \min_{\{u,v\} \subseteq e_t} |E_t(\{u, v\})|$
3 $\ell_t \leftarrow \lceil \log \eta_t \rceil$
4   Sample R_t *u.a.r.* from $[\ell_t, \ell_t + 2\lceil \log n \rceil]$
5   Sample $C(e_t)$ *u.a.r.* from $[\lfloor 2^{R_t}/(40 \cdot \log^2 n) \rfloor]$
6   **return** $C(e_t)$

5.3 Structural Property

For every $t \in [m]$, we define a hyperedge e_t to be *good* if $\frac{\lambda(G)}{n^2} \leq \eta_t \leq \lambda(G)$ and *bad* otherwise. Let $E_{\mathsf{good}} \subseteq E$ be the collection of all good hyperedges in G. Let $G_{\mathsf{good}} := (V, E_{\mathsf{good}})$ be the subhypergraph of G containing the good hyperedges. We show that G_{good} is connected and moreover, the minimum cut value of G_{good} is at least a constant fraction of of the minimum cut value of G.

Lemma 14. *We have that*

    1. G_{good} is connected and

    2. $\lambda(G_{\mathsf{good}}) \geq \frac{1}{2}\lambda(G)$.

Proof. We will show that $\lambda(G_{\mathsf{good}}) \geq \frac{1}{2}\lambda(G)$. This would imply that G_{good} is connected since

$$\lambda(G_{\mathsf{good}}) \geq \frac{\lambda(G)}{2} \geq \frac{\mathtt{opt}(G)}{2} \geq 40 \cdot \log^2 n \geq 1.$$

The second inequality above is by Lemma 13 and the third inequality is because $\mathtt{opt}(G) \geq 80 \log^2 n$.

Let $\emptyset \neq S \subsetneq V$. We will show that $|\delta_{G_{\mathsf{good}}}(S)| \geq \lambda(G)/2$. Let $e_{t_1}, e_{t_2}, \ldots, e_{t_k}$ be the hyperedges in $\delta(S)$ with $t_1 < t_2 < \ldots < t_k$, where $k = |\delta(S)| \geq \lambda(G)$. We let $F := \{e_{t_i} : 1 \leq i \leq \lambda(G)\}$ be the first $\lambda(G)$ hyperedges from $\delta(S)$. We now give an upper bound on the number of bad hyperedges in F.

For every $i \in [k]$, we recall that $\eta_{t_i} = \min_{\{u,v\} \subseteq e_{t_i}} |E_{t_i}(\{u,v\})|$, which is at most i. Hence, every hyperedge in F has its parameter $\eta_{t_i} \leq i \leq k \leq \lambda(G)$. This implies that every bad hyperedge $e_t \in F$ satisfies $\eta_t < \frac{\lambda(G)}{n^2}$. For every two different vertices $u, v \in V$ and positive integer $\mathtt{val} < \frac{\lambda(G)}{n^2}$, we call the pair $(\{u,v\}, \mathtt{val})$ a *witness* of a bad hyperedge $e_t \in F$ if $\{u,v\} \subseteq e_t$ and $|E_t(\{u,v\})| = \eta_t = \mathtt{val}$. Hence, the total number of possible distinct witnesses is smaller than

$$|\{\{u,v\} : u, v \in V\}| \cdot \frac{\lambda(G)}{n^2} \leq \frac{n^2}{2} \cdot \frac{\lambda(G)}{n^2} = \frac{\lambda(G)}{2}.$$

We note that every bad hyperedge in F has at least one witness. For every two different bad hyperedges $e_{t_i}, e_{t_j} \in F$ with $t_i < t_j$, if they share a same witness $(u, v, \mathtt{val})$, then we have

$$|E_{t_i}(\{u,v\})| = \mathtt{val} = |E_{t_j}(\{u,v\})|,$$

which leads to a contradiction since $|E_{t_i}(\{u,v\})|$ must be strictly smaller than $|E_{t_j}(\{u,v\})|$ by definition. Thus, no two bad hyperedges can share the same witness, which implies that the number of bad hyperedges in F does not exceed the total number of possible distinct witnesses, which is smaller than $\frac{\lambda(G)}{2}$. Therefore, the number of good hyperedges in F is at least $|F| - \frac{\lambda(G)}{2} = \frac{\lambda(G)}{2}$. This shows that $|\delta_{G_{\mathsf{good}}}(S)| \geq \lambda(G)/2$, which further implies that $\lambda(G_{\mathsf{good}}) \geq \frac{1}{2}\lambda(G)$. $\square$

5.4 Competitive Ratio Analysis

Now, we analyze the competitive ratio of Algorithm 3. Let $q := \lfloor \log \lambda(G) \rfloor$. We note that $\lambda(G) \geq 2^q > \lambda(G)/2$. We now prove that every color $c \in [\lfloor 2^q/(40 \cdot \log^2 n) \rfloor]$ is a connected spanning color with constant probability.

Lemma 15. *Let $c \in [\lfloor 2^q/(40 \cdot \log^2 n) \rfloor]$. Then, $\mathbf{Pr}_{Alg_3}[c \text{ is a connected spanning color}] \geq \frac{1}{2}$.*

Proof. We recall that $G_{\mathsf{good}} = (V, E_{\mathsf{good}})$ is the subhypergraph of G containing the good hyperedges. By Lemma 14, the hypergraph G_{good} is connected. Moreover, by Lemma 14, we have that $\lambda(G_{\mathsf{good}}) \geq \lambda(G)/2$. We recall that $k^*_{G_{\mathsf{good}}}$ is the weak partition connectivity of the hypergraph G_{good}. By applying Lemma 13 for the hypergraph G_{good}, we have $\lambda(G_{\mathsf{good}})/2 \leq k^*_{G_{\mathsf{good}}} \leq \lambda(G_{\mathsf{good}})$, which implies that

$$k^*_{G_{\mathsf{good}}} \geq \frac{\lambda(G_{\mathsf{good}})}{2} \geq \frac{\lambda(G)}{4} \geq \frac{2^q}{4}.$$

Let e_t be a hyperedge in G_{good}, i.e., $e_t \in E_{\mathsf{good}}$. Then, we have $\frac{\lambda(G)}{n^2} \leq \eta_t \leq \lambda(G)$ and hence, $\lceil \log \eta_t \rceil \leq q \leq \lceil \log \eta_t \rceil + 2\lceil \log n \rceil$. Therefore,

$$\mathbf{Pr}[R_t = q] = \frac{1}{2\lceil \log n \rceil + 1} > \frac{1}{5 \log n}. \tag{15}$$

22

Then, for every hyperedge $e_t \in E_{\mathsf{good}}$, the probability that e_t is colored with c is

$$\mathbf{Pr}[C(e_t) = c] \geq \mathbf{Pr}[R_t = q] \cdot \mathbf{Pr}[C(e_t) = c | R_t = q]$$
$$\geq \mathbf{Pr}[R_t = q] \cdot \frac{1}{\lfloor 2^q / (40 \cdot \log^2 n) \rfloor}$$
$$> \frac{1}{5 \log n} \cdot \frac{40 \cdot \log^2 n}{2^q} \quad \text{(by inequality (15))}$$
$$\geq \frac{1}{5 \log n} \cdot \frac{10 \cdot \log^2 n}{k^*_{G_{\mathsf{good}}}} = \frac{2 \log n}{k^*_{G_{\mathsf{good}}}}.$$

By applying Lemma 12 to the connected hypergraph G_{good}, we obtain that c is a connected spanning color with probability at least $\frac{1}{2}$. $\qquad\square$

Lemma 15 implies a lower bound on the expected number of connected spanning colors obtained by Algorithm 3.

Corollary 4. *The expected number of connected spanning colors obtained by Algorithm 3 is at least*

$$\mathbb{E}[\mathit{Alg}_3(G)] \geq \frac{1}{2} \cdot \lfloor 2^q / (40 \cdot \log^2 n) \rfloor.$$

Proof. By Lemma 15, we have that

$$\mathbb{E}[\mathtt{Alg}_3(G)] = \sum_{c \in [\lfloor 2^q / (40 \cdot \log^2 n) \rfloor]} \mathbf{Pr}_{\mathtt{Alg}_3}[c \text{ is a connected spanning color}] \geq \frac{1}{2} \cdot \lfloor 2^q / (40 \cdot \log^2 n) \rfloor.$$

$\qquad\square$

5.5   Combined Algorithm

We recall that our analysis of Algorithm 3 assumed that $\mathtt{opt}(G) \geq 80 \cdot \log^2 n$. We now combine Algorithm 3 with another algorithm to address all ranges of $\mathtt{opt}(G)$.

Consider the online algorithm $\mathtt{Alg}_3^*$ that runs Algorithm 3 with probability $\frac{1}{2}$ and assigns $C(e) = 1$ for every hyperedge e with probability $\frac{1}{2}$. We now show that the online algorithm $\mathtt{Alg}_3^*$ has competitive ratio $O(\log^2 n)$.

Theorem 7. *Algorithm Alg_3^* has pure competitive ratio $O(\log^2 n)$.*

Proof. Suppose $\mathtt{opt}(G) < 80 \cdot \log^2 n$. We recall that the algorithm assigns $C(e) = 1$ for every hyperedge e with probability $\frac{1}{2}$. This implies that

$$\mathbb{E}[\mathtt{Alg}_3^*(G)] \geq \frac{1}{2} \cdot \min\{\mathtt{opt}(G), 1\} \geq \frac{1}{160 \log^2 n} \cdot \mathtt{opt}(G).$$

Now, we may assume that $\mathtt{opt}(G) \geq 80 \cdot \log^2 n$. We recall that the algorithm runs Algorithm 3 with probability $\frac{1}{2}$. We have

$$\lfloor 2^q / (40 \cdot \log^2 n) \rfloor \geq \frac{\lambda(G)}{80} \cdot \log^2 n \geq \frac{\mathtt{opt}(G)}{80} \cdot \log^2 n \geq 1,$$

which implies that

$$\lfloor 2^q / (40 \cdot \log^2 n) \rfloor \geq \frac{1}{2} \cdot 2^q / (40 \cdot \log^2 n). \tag{16}$$

Hence, we have

$$\begin{aligned}
\mathbb{E}[\mathtt{Alg}_3^*(G)] &\geq \frac{1}{2} \cdot \mathbb{E}[\mathtt{Alg}_3(G)] \\
&\geq \frac{1}{4} \cdot \lfloor 2^q/(40 \cdot \log^2 n) \rfloor \quad \text{(by Corollary 4)} \\
&\geq \frac{1}{8} \cdot 2^q/(40 \cdot \log^2 n) \quad \text{(by inequality (16))} \\
&> \frac{1}{640 \cdot \log^2 n} \cdot \lambda(G) \quad \text{(since } \lambda(G) < 2^{q+1}) \\
&\geq \frac{1}{640 \cdot \log^2 n} \cdot \mathtt{opt}(G). \quad \text{(since } \mathtt{opt}(G) \leq \lambda(G))
\end{aligned}$$

$\square$

6 Conclusion

Packing bases of a polymatroid generalizes numerous set packing problems including disjoint bases of a matroid, disjoint spanning trees of a graph, disjoint set covers of a set system, and disjoint connected spanning subhypergraphs of a hypergraph. In this work, we introduced an online model for packing bases of a polymatroid and gave an algorithm that achieves a polylogarithmic competitive ratio. Our algorithm leads to a polylogarithmic competitive ratio for all these packing problems in a unified fashion. Our algorithm is based on novel properties of the notion of quotients of polymatroids. For the special cases of disjoint spanning trees of a graph and disjoint connected spanning subhypergraphs of a hypergraph, we gave a simpler and more elegant online algorithm that is also easy to implement while achieving the same polylogarithmic competitive ratio.

Our work leads to several interesting open questions. We mention three prominent ones: Firstly, we recall that DISJ-SPANNING-TREES (and more generally, DISJ-MATROID-BASES) in the offline setting is polynomial-time solvable via matroidal techniques. Is it possible to design an online algorithm for DISJ-SPANNING-TREES (and more generally, for DISJ-MATROID-BASES) with constant competitive ratio? Secondly, it is known that there is no randomized algorithm with expected impure competitive ratio $o(\log n/\log \log n)$ for DISJ-SET-COVER, where n is the size of the ground set. Can we show that there is no randomized algorithm with expected impure competitive ratio $o(\log f(\mathcal{N}))$ or pure competitive ratio $o(\log^2 f(\mathcal{N}))$ for DISJ-POLYMATROID-BASES? Thirdly, we recall that our online model for DISJ-POLYMATROID-BASES assumes knowledge of $f(\mathcal{N})$. Is it possible to achieve polylogarithmic competitive ratio without knowledge of $f(\mathcal{N})$? This seems to be open even in the special case of coverage functions (i.e., for online DISJ-SET-COVER) [5].

References

[1] M. Bienkowski, M. Byrka, and L. Jeż. Online disjoint set covers: Randomization is not necessary. arXiv: 2404.15554, 2024.

[2] G. Călinescu, C. Chekuri, and J. Vondrák. Disjoint bases in a polymatroid. *Random Struct. Algorithms*, 35(4):418–430, 2009.

[3] J. Cheriyan and M. R. Salavatipour. Packing element-disjoint steiner trees. *ACM Trans. Algorithms*, 3(4):47, 2007.

[4] J. Edmonds. Minimum partition of a matroid into independent subsets. *J. Res. Nat. Bur. Standards B*, 69:73–77, 1965.

[5] Y. Emek, A. Goldbraikh, and E. Kantor. Online Disjoint Set Cover Without Prior Knowledge. In *27th Annual European Symposium on Algorithms (ESA)*, pages 44:1–44:16, 2019.

[6] U. Feige, M. M. Halldórsson, G. Kortsarz, and A. Srinivasan. Approximating the domatic number. *SIAM Journal on Computing*, 32(1):172–195, 2002.

[7] A. Frank. *Connections in combinatorial optimization.* Oxford University Press, Oxford, 2011.

[8] S. Fujishige. Theory of principal partitions revisited. *Research Trends in Combinatorial Optimization*, pages 127–162, 2009.

[9] H. Narayanan. The principal lattice of partitions of a submodular function. *Linear Algebra and its Applications*, 144:179–216, 1991.

[10] H. Narayanan. *Submodular Functions and Electrical Networks (2nd edition).* 1997.

[11] A. Pananjady, V. K. Bagaria, and R. Vaze. The online disjoint set cover problem and its applications. In *2015 IEEE Conference on Computer Communications (INFOCOM)*, pages 1221–1229, 2015.

[12] K. Quanrud. Quotient sparsification for submodular functions. In *Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 5209–5248, 2024.

[13] A. Schrijver. *Combinatorial Optimization: Polyhedra and Efficiency.* Springer, 2003.

[14] A. Vardy. The intractability of computing the minimum distance of a code. *IEEE Transactions on Information Theory*, 43(6):1757–1766, 1997.

A Min-size of a non-empty Quotient and k^*

Let q^* be the minimum size of non-empty quotients of f, i.e.,

$$q^* := \min\{|Q| : \emptyset \neq Q \subseteq \mathcal{N} \text{ is a quotient of } f\}.$$

We recall that

$$k^* = k^*(f) = \min_{A \subseteq \mathcal{N} : f(A) < f(\mathcal{N})} \left\lfloor \frac{\sum_{e \in \mathcal{N}} f_A(e)}{f(\mathcal{N}) - f(A)} \right\rfloor.$$

The following lemma shows that q^* is an r-approximation of k^*.

Lemma 16. $\frac{q^*}{r} - 1 < k^* \leq q^*$.

Proof. We first show the lower bound on k^*. Consider a closed set $A \subseteq \mathcal{N}$ with $f(A) < f(\mathcal{N})$ and $k^* = \left\lfloor \frac{\sum_{e \in \mathcal{N}} f_A(e)}{f(\mathcal{N}) - f(A)} \right\rfloor$. Since A is closed, we have that $f_A(e) \geq 1$ for every $e \in \mathcal{N} \setminus A$ and $\mathcal{N} \setminus A$ is a quotient of f. Therefore, we have

$$
\begin{aligned}
k^* = \left\lfloor \frac{\sum_{e \in \mathcal{N}} f_A(e)}{f(\mathcal{N}) - f(A)} \right\rfloor &> \frac{\sum_{e \in \mathcal{N}} f_A(e)}{f(\mathcal{N}) - f(A)} - 1 \quad (\text{since } \lfloor x \rfloor > x - 1) \\
&\geq \frac{\sum_{e \in \mathcal{N}} f_A(e)}{r} - 1 \quad (\text{since } f(\mathcal{N}) - f(A) \leq f(\mathcal{N}) \leq r) \\
&\geq \frac{|\mathcal{N} \setminus A|}{r} - 1 \quad (\text{since } f_A(e) \geq 1 \text{ for every } e \in \mathcal{N} \setminus A) \\
&\geq \frac{q^*}{r} - 1. \quad (\text{since } \mathcal{N} \setminus A \text{ is a quotient of } f)
\end{aligned}
$$

We now show the upper bound on k^*. Consider a minimum sized non-empty quotient Q. Let $A := \mathcal{N} \setminus Q$. Since A is a closed set and $A \neq \mathcal{N}$, we have $f(\mathcal{N}) > f(A)$. We note that for every $e \in Q$, $f_A(e) = f(A + e) - f(A) \leq f(\mathcal{N}) - f(A)$, where the inequality is by monotonicity of the function f. Therefore,

$$
\begin{aligned}
q^* = |Q| &\geq \sum_{e \in Q} \left(\frac{f_A(e)}{f(\mathcal{N}) - f(A)} \right) \quad (\text{since } f_A(e) \leq f(\mathcal{N}) - f(A)) \\
&= \frac{\sum_{e \in \mathcal{N}} f_A(e)}{f(\mathcal{N}) - f(A)} \quad (\text{since } f_A(e) = 0 \text{ for every } e \in A) \\
&\geq k^*. \quad (\text{by definition of } k^*)
\end{aligned}
$$

$\square$

B Random Sampling gives a Base

In this section, we prove Lemma 2. A variant of the lemma was shown in [2].

Lemma 2. *Let $f : 2^{\mathcal{N}} \to \mathbb{Z}_{\geq 0}$ be a polymatroid with $f(\mathcal{N}) = r \geq 2$. Let $p := \min\{1, \frac{2 \log r}{k^*}\}$ and $S \subseteq \mathcal{N}$ be a subset obtained by picking each element in $\mathcal{N}$ with probability at least p independently at random. Then, S is a base of f with probability at least $\frac{1}{2}$.*

Proof. If $k^* \leq 2 \log r$, then we have $p = 1$, which implies that $S = \mathcal{N}$ which is a base of f. Henceforth, we may assume that $k^* > 2 \log r$ and consequently, $p = \frac{2 \log r}{k^*}$. We recall that $n = |\mathcal{N}|$.

Let $\sigma = (e_1, e_2, \ldots, e_n)$ be a uniformly random permutation of the elements in $\mathcal{N}$. For every $i \in [0, n]$, we define $\mathcal{N}_i := \{e_j : j \in [i]\}$ and $S_i := S \cap \mathcal{N}_i$. We note that $S_0 = \mathcal{N}_0 = \emptyset$ and $f_{S_0}(\mathcal{N}) = r$.

Let $i \in [n]$. We consider the distribution of element e_i conditioned on S_{i-1}. Since σ is a uniformly random permutation, e_i can be an arbitrary element in $\mathcal{N} \setminus S_{i-1}$, of which each has the same probability. That is, for every element $e \in \mathcal{N}$,

$$\mathbf{Pr}_{\sigma, S}[e_i = e | S_{i-1}] = \begin{cases} 0, & e \in S_{i-1} \\ \frac{1}{|\mathcal{N}| - |S_{i-1}|}. & e \in \mathcal{N} \setminus S_{i-1} \end{cases}$$

We note that element e_i is picked with probability at least p. If element e_i is picked, then $S_i = S_{i-1} + e_i$. Otherwise, $S_i = S_{i-1}$. Hence, we have

$$\begin{aligned}
\mathbb{E}_{\sigma, S}[f_{S_{i-1}}(\mathcal{N}) - f_{S_i}(\mathcal{N}) | S_{i-1}] &= \sum_{e \in \mathcal{N} \setminus S_{i-1}} \frac{1}{|\mathcal{N}| - |S_{i-1}|} \cdot \mathbb{E}_{\sigma, S}[f_{S_{i-1}}(\mathcal{N}) - f_{S_i}(\mathcal{N}) | S_{i-1} \text{ and } e_i = e] \\
&\geq \sum_{e \in \mathcal{N} \setminus S_{i-1}} \frac{p}{|\mathcal{N}| - |S_{i-1}|} \cdot f_{S_{i-1}}(e) \\
&\geq \frac{p}{n} \cdot \sum_{e \in \mathcal{N} \setminus S_{i-1}} f_{S_{i-1}}(e) \\
&\geq \frac{p}{n} \cdot k^* \cdot f_{S_{i-1}}(\mathcal{N}),
\end{aligned}$$

where the last inequality is by the definition of k^*. This implies that

$$\begin{aligned}
\mathbb{E}_{\sigma, S}[f_{S_i}(\mathcal{N})] &= \mathbb{E}_{\sigma, S}[f_{S_{i-1}}(\mathcal{N}) - \mathbb{E}_{\sigma, S}[f_{S_{i-1}}(\mathcal{N}) - f_{S_i}(\mathcal{N}) | S_{i-1}]] \\
&\leq \mathbb{E}_{\sigma, S}\left[\left(1 - \frac{p}{n} \cdot k^*\right) \cdot f_{S_{i-1}}(\mathcal{N})\right] \\
&= \left(1 - \frac{p}{n} \cdot k^*\right) \cdot \mathbb{E}_{\sigma, S}[f_{S_{i-1}}(\mathcal{N})].
\end{aligned}$$

By setting $i = n$, we have

$$\begin{aligned}
\mathbb{E}_{\sigma, S}[f_{S_n}(\mathcal{N})] &\leq \left(1 - \frac{p}{n} \cdot k^*\right)^n \cdot \mathbb{E}_{\sigma, S}[f_{S_0}(\mathcal{N})] \\
&= \left(1 - \frac{p}{n} \cdot k^*\right)^n \cdot r \\
&< \exp(-p \cdot k^*) \cdot r = \exp(-2 \log r) \cdot r < \frac{1}{2}.
\end{aligned}$$

According to Markov's inequality, we have

$$\mathbf{Pr}_{\sigma, S}[f_{S_n}(\mathcal{N}) \geq 1] \leq \mathbb{E}_{\sigma, S}[f_{S_n}(\mathcal{N})] < \frac{1}{2},$$

which shows that S is a base with probability at least $\frac{1}{2}$. □