An $\widetilde{O}(n^{3/7})$ Round Parallel Algorithm for Matroid Bases

Sanjeev Khanna
Courant Institute, Warren Weaver Hall, New York University, NY, USA

Aaron Putterman
School of Engineering and Applied Sciences, Harvard University, Cambridge, MA, USA

Junkai Song
Courant Institute, Warren Weaver Hall, New York University, NY, USA

Abstract

We study the parallel (adaptive) complexity of the classic problem of finding a basis in an n-element matroid, given access via an *independence oracle*. In this model, the algorithm may submit polynomially many independence queries in each round, and the central question is: how many rounds are necessary and sufficient to find a basis?

Karp, Upfal, and Wigderson (FOCS 1985, JCSS 1988; hereafter KUW) initiated this study, showing that $O(\sqrt{n})$ adaptive rounds suffice for any matroid, and that $\widetilde{\Omega}(n^{1/3})$ rounds are necessary even for partition matroids. This left a substantial gap that persisted for nearly four decades, until Khanna, Putterman, and Song (FOCS 2025; hereafter KPS) achieved $\widetilde{O}(n^{7/15})$ rounds, the first improvement since KUW.

In this work, we make another conceptual advance beyond KPS, giving a new algorithm that finds a matroid basis in $\widetilde{O}(n^{3/7})$ rounds. We develop a structural and algorithmic framework that brings a new lens to the analysis of random circuits, moving from reasoning about individual elements to understanding how dependencies span multiple elements simultaneously. Specifically, our framework introduces three new ideas:

1. A new *subset-based decomposition* that provides precise guarantees on how random circuits intersect groups of elements, yet remains computable in few adaptive rounds.
2. A new method for identifying and removing redundant elements in bulk, based on short circuit witnesses that certify redundancy across large portions of the matroid.
3. An adaptive early-stopping strategy that uses the evolving structure of the matroid to decide when to contract or delete, preventing wasted rounds.

Each of these contributions, in isolation, already yields meaningful improvements over the round complexity achieved in KPS; their combination enables our main result of $\widetilde{O}(n^{3/7})$ rounds.

As further consequences, incorporating our improved basis-finding algorithm into known reductions yields an $\widetilde{O}(n^{17/21})$-round parallel algorithm for matroid intersection, as well as an $\widetilde{O}(n^{3/7})$-round parallel algorithm for approximate monotone submodular maximization under a matroid constraint.

2012 ACM Subject Classification Theory of computation → Constraint and logic programming

Keywords and phrases parallel algorithms, matroids

Digital Object Identifier 10.4230/LIPIcs.ICALP.2026.124

Category Track A: Algorithms, Complexity and Games

Related Version *Full Version*: https://arxiv.org/abs/2605.03979

Funding *Sanjeev Khanna*: Supported in part by NSF award CCF-2625203 and AFOSR award FA9550-25-1-0107.

Aaron Putterman: Supported in part by the Simons Investigator Awards of Madhu Sudan and Salil Vadhan and AFOSR award FA9550-25-1-0112.

Junkai Song: Supported in part by NSF award CCF-2625203.

© Sanjeev Khanna, Aaron Putterman, and Junkai Song;
licensed under Creative Commons License CC-BY 4.0

53rd International Colloquium on Automata, Languages, and Programming (ICALP 2026).
Editors: Sayan Bhattacharya, Danupon Nanongkai, Michael Benedikt, and Gabriele Puppis;
Article No. 124; pp. 124:1–124:21

Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

In this work, we continue the study of designing efficient parallel algorithms for fundamental combinatorial optimization problems. Parallel efficiency is typically measured through *adaptive round complexity*, the number of rounds of operations/queries required to solve a problem, where each round allows polynomially many operations/queries to be executed in parallel. Many foundational problems have been explored under this lens, beginning with the seminal works on maximal independent sets [34], spanning trees [26, 31], and graph matchings [33, 27, 23, 38], and extending to more recent advances in submodular function minimization [5, 11, 12], submodular function maximization [3, 4, 1, 2, 15, 14, 19, 20, 22, 21, 29, 16, 10, 32], and matroid intersection [24, 25, 8, 9].

In this paper, we focus specifically on understanding the parallel complexity of finding a *basis* of a matroid. Formally, a matroid $\mathcal{M} = (E, \mathcal{I})$ consists of a ground set E of n elements and a collection of independent sets $\mathcal{I} \subseteq 2^E$ satisfying:

- $\varnothing \in \mathcal{I}$;
- if $S \in \mathcal{I}$ and $S' \subseteq S$, then $S' \in \mathcal{I}$; and
- if $S, T \in \mathcal{I}$ and $|S| < |T|$, then there exists $e \in T \backslash S$ such that $S \cup \{e\} \in \mathcal{I}$.

A *basis* is an independent set $S \subseteq E$ that is maximal under inclusion. Because matroids capture independence across a wide range of structures, from cycle-freeness in graphs to linear independence in vector spaces, finding a basis is a fundamental combinatorial primitive. For example, in graphs, a matroid basis corresponds to a spanning forest; in vector spaces, it coincides with the standard notion of a basis.

Given the broad applicability of matroids, understanding the parallel complexity of basis finding remains a central theoretical challenge. Despite decades of work, a substantial gap persists in characterizing how efficiently a matroid basis can be computed in parallel. The difficulty stems from the generality of matroids: the number of distinct matroids grows super-exponentially in the size n of the ground set [6]. This fact points both to the rich combinatorial structure of matroids, and the impossibility of representing them succinctly. Thus in the general setting considered in this work, it is typical to assume that the access to a matroid is given via an oracle. Specifically, we will assume the matroid $\mathcal{M}$ is presented via an *independence oracle* Ind, which reports membership in $\mathcal{I}$ for any queried subset $S \subseteq E$. In each adaptive round, the algorithm may make polynomially many independence queries.

This formalization of the problem was first introduced by Karp, Upfal, and Wigderson [26, 28] (KUW), who asked:

> *Given independence oracle access to an arbitrary matroid $\mathcal{M}$, how many rounds of polynomially many queries are required to find a basis?*

KUW [26] provided the first set of foundational results, showing that $O(\sqrt{n})$ adaptive rounds suffice and that $\widetilde{\Omega}(n^{1/3})$ rounds are necessary even for partition matroids. This left a broad gap that persisted for nearly forty years, until Khanna, Putterman, and Song (KPS) [30] developed a new decomposition-based algorithm achieving $\widetilde{O}(n^{7/15})$ rounds, the first improvement since KUW. Nevertheless, the precise adaptive complexity of matroid basis finding remains unresolved.

Our Contributions. We make further progress on this long-standing question by presenting a new parallel algorithm with improved round complexity:

▶ **Theorem 1.** *There is an algorithm which for any arbitrary matroid $\mathcal{M}$ on n elements, makes polynomially many independence queries per round and recovers a basis of $\mathcal{M}$ in $\widetilde{O}(n^{3/7})$ rounds with high probability.*

KUW established a lower bound of $\widetilde{\Omega}(n^{1/3})$ rounds, so our result narrows the possible exponent to the interval $[\frac{1}{3}, \frac{3}{7}] \approx [0.333, 0.429]$, a significant improvement over the previous upper bound of $\frac{7}{15} \approx 0.467$ from KPS. Our approach introduces a new structural and algorithmic framework that brings a *new analytical lens* to random-circuit structure, moving from reasoning about individual elements to understanding how dependencies spread across groups of elements. This shift enables three new algorithmic components: a refined matroid decomposition, a method for bulk identification of redundant elements, and an adaptive meta-strategy for deciding when to contract or delete, which together yield the $\widetilde{O}(n^{3/7})$ bound. We explain these contributions and their context in more detail in Section 1.2.

Applications. The adaptive complexity model has also been explored in several related matroid optimization problems, where the task of finding a matroid basis or computing the rank of the matroid (equivalently, the size of a basis) often serves as a fundamental subroutine. Consequently, our improved basis-finding algorithm directly yields sharper bounds.

Matroid Intersection. In the matroid intersection problem, we are given two matroids $\mathcal{M}_1 = (E, \mathcal{I}_1)$ and $\mathcal{M}_2 = (E, \mathcal{I}_2)$ on the same ground set, and the goal is to find a largest set $S \subseteq E$ such that $S \in \mathcal{I}_1 \cap \mathcal{I}_2$. In the sequential setting, Edmonds [17, 18] gave the first polynomial time algorithm, and subsequent work has progressively improved the running time, culminating in an $\tilde{O}(n\sqrt{r})$ rank-query algorithm [13] and an $\tilde{O}(nr^{3/4})$ independence-query algorithm [7] when the size of the intersection is r.

In the parallel setting, Chakrabarty, Chen, and Khanna [11] established an $\Omega(n^{1/3})$ round lower bound for rank-query algorithms, improving upon the classical $\Omega(n^{1/3})$ lower bound for independence-query algorithms by [26]. On the upper bound side, Blikstad [8] presented the first sublinear round algorithms in both the rank-query and independence-query models, later improved by [9], who gave an $O(n^{2/3})$ round rank-query algorithm, and an $O(n^{5/6})$ round independence-query algorithm.

Our improved basis finding algorithm has immediate implications for this classic problem: indeed, using our improved basis-finding procedure within the work of [9], we obtain the following round complexity for matroid intersection (proved formally in the full version).

▶ **Theorem 2.** *There is an $\widetilde{O}(n^{17/21})$-round algorithm that, given any two matroids $\mathcal{M}_1$ and $\mathcal{M}_2$ on the same ground set of n elements, makes polynomially many independence queries per round and, with high probability, outputs a maximum common independent set of $\mathcal{M}_1$ and $\mathcal{M}_2$.*

This improves upon the previous best complexity of $\widetilde{O}(n^{37/45})$ rounds established in [30].

Submodular Function Maximization under a Matroid Constraint. In the *submodular function maximization* problem, we are given a monotone submodular function $f : 2^E \to \mathbb{R}_{\geqslant 0}$ and a matroid $\mathcal{M} = (E, \mathcal{I})$, and the goal is to find an independent set $S \in \mathcal{I}$ that approximately maximizes $f(S)$. In the sequential setting, a $(1 - 1/e)$-approximation algorithm has been known in the case where $\mathcal{M}$ is a uniform matroid [36], and Vondrák [39] extended this guarantee to general matroids. The approximation factor is optimal with polynomially many queries (here queries refer to evaluating the function f on a chosen set $S \subseteq E$) [35, 40].

Balkanski and Singer [3] initiated the study of this problem in the parallel complexity model. When $\mathcal{M}$ is a uniform matroid, they proved that any approximation better than $O(1/\log n)$ requires $\Omega(\log n/\log\log n)$ rounds, and designed a 1/3-approximation algorithm in $O(\log n)$ rounds. Subsequent works achieved $1 - 1/e - \epsilon$ approximation algorithms using

$O(\text{poly}(\log n, 1/\epsilon))$ rounds, both when $\mathcal{M}$ is a uniform matroid [1, 15, 19, 22], and when $\mathcal{M}$ is an arbitrary matroid accessed via rank queries [2, 14, 20]. However, when $\mathcal{M}$ is accessed via independence queries, their algorithms only achieved a round complexity of $\widetilde{O}(\sqrt{n}/\epsilon^3)$, as their computation relies explicitly on finding a basis [2]. Later, a separate work by Li, Liu and Vondrák [32] showed that a polynomial dependence on ϵ in the round complexity is unavoidable.

As in matroid intersection, by combining the algorithm of [2] with our improved matroid basis finding algorithm, we obtain immediate improvements in the round complexity. The resulting guarantee is stated below; a formal proof appears in the full version.

▶ **Theorem 3.** *For any $\epsilon > 0$, there is an $\widetilde{O}(n^{3/7}\epsilon^{-3})$-round algorithm that, given any matroid $\mathcal{M} = (E, \mathcal{I})$ on n elements and any monotone submodular function $f : 2^E \to \mathbb{R}_{\geqslant 0}$, makes polynomially many independence queries per round and outputs, with high probability, a $(1 - 1/e - O(\epsilon))$-approximation to the maximum of f under the matroid constraint $\mathcal{M}$.*

This improves the previous best bound of $\widetilde{O}(n^{7/15}\epsilon^{-3})$ rounds, achievable via the algorithm of [30], though not explicitly stated in their work.

We next provide background on prior work and terminology in Section 1.1, before turning to an overview of our new techniques in Section 1.2.

1.1 Prior Work

We start by recalling some matroid terminology that will be used throughout the introduction and reviewing the approaches of KUW [26, 28] and KPS [30], as these provide the key backdrop for our contribution.

Notation and Terminology. For a matroid $\mathcal{M} = (E, \mathcal{I})$, a **basis** is a set $B \subseteq E$ that is a *maximal independent set*: $B \in \mathcal{I}$, but $B \cup \{e\} \notin \mathcal{I}$ for all $e \in E \backslash B$. A **circuit** is a set $C \subseteq E$ that is a *minimal dependent set*: $C \notin \mathcal{I}$ but $C \backslash \{e\} \in \mathcal{I}$ for all $e \in C$. We write $\text{rank}(S)$ for the size of the largest independent subset of $S \subseteq E$, and define the **span** of S as $\text{span}(S) = \{e \in E : \text{rank}(S) = \text{rank}(S \cup \{e\})\}$. That is, $\text{span}(S)$ consists of all elements whose addition to S does not increase its rank.

We will rely on several standard facts about matroids (see, e.g., [37]). If $S \subseteq E$ satisfies $\text{rank}(S) = \text{rank}(E)$, then any basis of S is also a basis of E. Moreover, if S is independent, then there exists a basis of $\mathcal{M}$ that contains S by the extension property of matroids. This fact motivates the operation of **contraction**: given $\mathcal{M} = (E, \mathcal{I})$ and $S \in \mathcal{I}$, the contracted matroid $\mathcal{M}/S$ is defined on the ground set $E \backslash S$ so that $T \subseteq E \backslash S$ is independent in $\mathcal{M}/S$ if and only if $T \cup S$ is independent in $\mathcal{M}$. Intuitively, contraction corresponds to "committing" to include S in the eventual basis; if one finds a basis T of $\mathcal{M}/S$, then $S \cup T$ forms a basis of the original matroid $\mathcal{M}$.

Formal Problem Statement. As mentioned above, we study the *parallel complexity* of finding a basis of a matroid. Formally, an algorithm A is given access to a matroid $\mathcal{M} = (E, \mathcal{I})$ through an *independence oracle* Ind, which, for any set $S \subseteq E$, returns $\mathbf{1}[S \in \mathcal{I}]$. The computation proceeds in *rounds*: in each round, the algorithm may issue up to $\text{poly}(n)$ oracle queries in parallel, where each query is a subset $S \subseteq E$, and the oracle responds whether S is independent. Importantly, the queries in the ith round are made in *parallel*, meaning that these queries depend only on responses to queries in rounds $1, \ldots i - 1$ (and on the algorithm's internal randomness).

The objective is to find a basis of the matroid $\mathcal{M}$ (with high probability over the randomness of the algorithm) in as few rounds of queries as possible. Lastly, our algorithm should work *for all* matroids $\mathcal{M}$: that is, we define the round complexity to be the *maximum over all matroids* $\mathcal{M}$ on n elements, of the number of rounds the algorithm requires to find a basis of $\mathcal{M}$ with high probability.

Overarching Themes. With these preliminaries in place, we turn to the high-level intuition behind both KUW and KPS. In both works, progress toward finding a matroid basis proceeds through two complementary operations:

1. **Deleting redundant elements:** If one can identify a set S such that every element of S lies in the span of $E\backslash S$, then deleting S does not reduce the matroid's rank. This means $\text{rank}(E) = \text{rank}(E\backslash S)$ and therefore there exists a basis supported entirely on $E\backslash S$. This reduces the search space of the problem; instead of finding a basis over $\mathcal{M} = (E, \mathcal{I})$, we instead search over the matroid $\mathcal{M}|_{E\backslash S} = (E\backslash S, \mathcal{I} \cap 2^{E\backslash S})$.
2. **Contracting on an independent set:** If one can find an independent set S, then by the extension property, there exists a basis containing S. This allows the algorithm to contract on S and focus subsequent queries on the residual matroid $\mathcal{M}/S$.

[26]'s $O(\sqrt{n})$ Round Algorithm. Using these two operations, [26] designed a simple yet powerful algorithm: partition the ground set E into $\sqrt{n}$ groups of size $\sqrt{n}$ each. Within each group $S = \{e_1, \ldots, e_{\sqrt{n}}\}$, query the independence oracle on all prefixes $\{e_1\}, \{e_1, e_2\}, \ldots, S$. Two outcomes are possible:

- If any group S is fully independent, we can contract on it, adding at least $\sqrt{n}$ *independent* elements to the basis.
- Otherwise, in every group, the first element that introduces dependence is redundant: if $\{e_1, \ldots, e_j\}$ is independent but $\{e_1, \ldots, e_{j+1}\}$ is dependent, then $e_{j+1} \in \text{span}(\{e_1, \ldots, e_j\})$. Thus at least one element can be deleted per group, removing $\sqrt{n}$ *redundant* elements in total.

Hence, in each round, the instance size decreases from n to $n - \sqrt{n}$, and after $O(\sqrt{n})$ rounds, the algorithm outputs a basis. To complement this upper bound, [26] also proved an $\widetilde{\Omega}(n^{1/3})$ lower bound, leaving a large gap that persisted for next four decades.

[30]'s $\widetilde{O}(n^{7/15})$ Round Algorithm. The algorithm of [30] takes a fundamentally different approach from the $O(\sqrt{n})$-round algorithm of [26]. Instead of insisting on immediate progress in every round, their algorithm deliberately spends several rounds gathering structural information about the matroid, performing what they refer to as a *matroid decomposition*. Only once this decomposition is established does the algorithm proceed to delete redundant elements and contract independent sets. While this strategy may not yield $\widetilde{\Omega}(n^{8/15})$ progress in each round individually, it guarantees an *average* progress of that order over the full execution. This flexibility is key: it allows the algorithm to partition $\mathcal{M}$ into "well-behaved" regions that can then be processed efficiently for contraction or deletion.

To formalize this notion of "well-behaved", [30] introduces two key parameters:

1. **The $\alpha(S)$ parameter.** For any subset $S \subseteq E$, $\alpha(S)$ is defined as the smallest integer ℓ for which a uniformly random ℓ-subset of S is independent with probability at most $1/2$:

$$\alpha(S) = \min\left\{\ell \in [n] : \Pr_{T \sim \binom{S}{\ell}}[\text{Ind}(T) = 1] \leqslant \tfrac{1}{2}\right\}.$$

Intuitively, $\alpha(S)$ captures the point at which dependence typically appears when the elements of S are revealed in random order. If we fix S and query every prefix under a random permutation π of its elements, $\alpha(S)$ corresponds to the *median* prefix length at which dependence first arises.

2. **Marginal circuit probabilities.** For each element $i \in S$, the *marginal circuit probability* p_i measures how likely i is to appear in the *first circuit* formed during a random permutation process. Specifically, for a random permutation π of S, let C_π be the first circuit[1] that appears when elements are added in order $\pi(1), \pi(2), \ldots$. Then $p_i = \Pr_\pi[\, i \in C_\pi \,]$. When needed, we may write $p_{i,S}$ to emphasize that the probability is taken with respect to the set S.

Using these parameters, [30] establishes the following key structural lemma.

▶ **Lemma 4** (Informal; [30]). *There is a decomposition algorithm for $\mathcal{M} = (E, \mathcal{I})$ that uses* $\mathrm{poly}(n)$ *independence queries per round and, if it terminates after γ rounds:*

1. *Recovers disjoint sets $S_1, \ldots, S_\gamma$ such that for every $j \in [\gamma]$ and $i \in S_j$,*

$$p_{i,S_j} = \widetilde{\Omega}\left(\frac{1}{|S_j|}\right). \tag{1}$$

Note here that we use p_{i,S_j} to denote the marginal probability with respect to S_j, not *the parent matroid $\mathcal{M}$.*

2. *For any $j, k \in [\gamma], j < k$,*

$$\alpha(S_k) = \frac{\alpha(S_j)|S_k|}{|S_j|} + \Omega\left(\sqrt{\frac{\alpha(S_j)|S_k|}{|S_j|}}\right). \tag{2}$$

3. *For each $j \in [\gamma]$, we can recover an independent set of size $\Omega\left(\frac{\alpha(S_j)}{|S_j|} \cdot n\right)$ in $\mathcal{M}\backslash(\bigcup_{i<j} S_i)$.*

This decomposition yields several key consequences. First, the recurrence in Equation (2) implies that the number of parts γ is at most $O(n^{1/3})$. Second, if for any S_j the ratio $\alpha(S_j)/|S_j|$ is large, item (3) guarantees that the residual matroid contains a large independent set that can be efficiently recovered, allowing rapid progress via contraction.

The complementary case, when $\alpha(S_j)/|S_j| \ll 1$, indicates the presence of many redundant elements in S_j. [30] shows that these can be identified and deleted in parallel via the following progress lemma.

▶ **Lemma 5** (Informal; [30]). *Let S_j be a set peeled off in the above decomposition, such that for all $i \in S_j$, $p_{i,S_j} = \widetilde{\Omega}\left(\frac{1}{|S_j|}\right)$, and $\alpha(S_j) \leqslant |S_j|/\log^2 |S_j|$. Then:*

1. *There is an $O(\sqrt{|S_j|})$-round algorithm that recovers $\widetilde{\Omega}(|S_j|)$ redundant elements.*
2. *There is also a 1-round algorithm that recovers*

$$\widetilde{\Omega}\left(\min\left(|S_j|, \frac{|S_j|^2}{\alpha(S_j)^2}\right)\right)$$

redundant elements.

[1] Formally, if i is the first index such that $\{e_{\pi(1)}, \ldots, e_{\pi(i-1)}\}$ is independent but $\{e_{\pi(1)}, \ldots, e_{\pi(i)}\}$ is dependent, then the unique circuit contained in this latter set is denoted C_π.

Together, these lemmas yield a "win–win" structure: when $\alpha(S_j)$ is relatively large, a sizable independent set can be contracted; when $\alpha(S_j)$ is relatively small, many redundant elements can be deleted. By grouping the S_j's by size (into $\log n$ geometric buckets), and focusing on the group with the most sets, [30] shows that there always exists a group where one can achieve $\widetilde{\Omega}(n^{8/15})$ average progress per round (where progress refers to both independent elements that are contracted and redundant elements that are deleted), leading to their $\widetilde{O}(n^{7/15})$-round bound for matroid basis finding.

1.2 Our New Algorithm

We introduce three conceptual advances that change how progress is measured and managed in the matroid-basis problem, leading to a sharper framework and improved round complexity over [30]. We summarize them here before turning to their technical details.

1. **Subset-hitting decomposition.** We strengthen the decomposition guarantee from element-wise control to *subset-level* control: instead of ensuring $p_{i,S} \gtrsim 1/|S|$ for each element i, we require that *for every subset* $T \subseteq S$, the first circuit intersects T with probability $\widetilde{\Omega}(|T|/|S|)$. This subset-hitting property rules out highly correlated pathological behavior, yielding a more uniform distribution of first circuit mass across elements in a peeled set. Surprisingly, a decomposition with this stronger (global) guarantee is still computable within the *same* round complexity as in [30] using only polynomially many queries, as before.
2. **Short-circuit witnesses for bulk deletion.** Leveraging subset-hitting, we design a one-round deletion primitive that exploits *circuit sizes*. If there is a set $R \subseteq S$ such that for every $x \in S \backslash R$ we can certify a circuit C_x with $x \in C_x$ and $|C_x \cap (S \backslash R)| \leqslant \ell$, then we can delete $\Omega(|S \backslash R|/\ell)$ redundant elements in one round. By carefully instantiating the set R, we show a new win-win paradigm for deleting redundant elements: namely we can either (a) delete lots of elements by finding short circuits that they participate in, or (b) *re-use* a redundant element finding algorithm from [30] but now with *much better guarantees*!

   This win–win paradigm relies crucially on subset-hitting; without it (e.g., under the KPS element-wise guarantee), the improved bounds do not materialize.
3. **Adaptive early-stopping via evolving α-profiles.** Lastly, we introduce a much improved strategy for deciding *when* to contract on independent sets versus when to delete redundant elements. This is based on a new quantitative analysis of how the α-parameters of the peeled pieces evolve across the decomposition. This yields an *early-stopping* rule that avoids investing rounds when the average progress target is already met, improving amortized progress.

These new ingredients can be applied in a *modular* yet *complementary* way. Using only (1) and (2) with the original [30] global analysis yields an $\widetilde{O}(n^{0.44})$-round algorithm. Using only (3), the adaptive early-stopping analysis along with [30] primitives, gives an $\widetilde{O}(n^{4/9})$ bound with a substantially simpler proof. When combined, the three components reinforce one another, though the resulting analysis becomes significantly more delicate. It is this combination that ultimately enables our $\widetilde{O}(n^{3/7})$-round algorithm. We now describe each ingredient and how they interact in more detail.

1.2.1 Intuition for Improved Redundant Element Recovery

We now elaborate on our new algorithm for identifying redundant elements (the second advance described above). The key starting point is that the structure of *circuits*, that is, the minimal dependent sets, can be exploited much more deeply than before.

Suppose we are trying to find a basis of a matroid $\mathcal{M} = (E, \mathcal{I})$, and that we have peeled off a set S_j according to Lemma 4. For clarity, assume that the marginal circuit probabilities satisfy $p_i = \widetilde{\Omega}(1/|S_j|)$, as in the weaker decomposition of Lemma 4. Intuitively, this means that under a random permutation of the elements, the first circuit encountered contains each $i \in S_j$ with probability $\widetilde{\Omega}(1/|S_j|)$ – that is, a nontrivial fraction of these random circuits involve i.

Now suppose that for every element $i \in S_j$, we can find a circuit C_i containing i of size at most ℓ. In this ideal case, a simple greedy argument shows that at least $|S_j|/\ell$ elements are redundant. Indeed, we may process the elements of S_j in an arbitrary order: for each i, keep all other elements of $C_i \backslash \{i\}$ (at most $\ell - 1$ of them), since $i \in \mathrm{span}(C_i \backslash \{i\})$. We delete i, commit to keeping these witnesses, and repeat. Each deletion costs at most $(\ell - 1)$ commitments, so we can delete $\Omega(|S_j|/\ell)$ elements in total.

Of course, not every element will participate in such a small circuit. This motivates examining the *distribution* of circuit sizes. Within our experiment of randomly sampling permutations and identifying the first emerging circuit, our first observation is that the *expected* circuit size equals the sum of the marginal circuit probabilities:

$$\sum_{i \in S_j} p_i = \mathbb{E}_\pi[|C_\pi|],$$

since $p_i = \Pr[i \in C_\pi] = \mathbb{E}_\pi[\mathbf{1}[i \in C_\pi]]$, and summing over all elements gives the expected size of C_π.

This identity suggests a natural dichotomy for progress: if $\sum_i p_i$ is large, many elements appear frequently in the first circuit, and existing methods (such as those in [30]) can delete a large number of redundant elements. Conversely, if the average circuit size is small, one might hope that many elements lie in short circuits, opening the door to fast progress through our short-circuit deletion scheme.

However, this intuition requires care: a small *average* circuit size does not guarantee that *most* elements actually belong to small circuits. A small subset of elements might have large p_i values and appear disproportionately often in small circuits, while the majority of elements participate only in rare but very large circuits. In such cases, the algorithm could make little progress on most elements, even when the average circuit size is small.

To illustrate, consider a set S_j consisting of two disjoint uniform matroids, P_1 and P_2. Let P_1 contain $\sqrt{n}$ elements with rank $\sqrt{n}/2$ (meaning any subset of $\leqslant \sqrt{n}/2$ elements is independent), and P_2 contain $n - \sqrt{n}$ elements with rank $(n - \sqrt{n})/2 + \eta$, for a parameter η. Under a random permutation of all elements, when $\eta = 0$ the first circuit C_π is equally likely to arise in either P_1 or P_2. As η increases, circuits become increasingly unlikely to form in P_2 first. Choosing $\eta \approx \sqrt{n \log n}$ ensures that circuits appear in P_2 only a $1/n$ fraction of the time. Then:

1. The expected circuit size is $\approx \sqrt{n}/2$, since with probability $1 - 1/n$, the first dependence occurs in P_1.
2. Almost all elements are in P_2, and they participate only in circuits of size $\geqslant (n - \sqrt{n})/2$.

Thus, even though the *expected circuit size* is small, there is no way to make better progress than [30]. The main reason is that many elements with small marginal circuit probabilities often co-occur in the same circuits, creating *correlated low-probability clusters*. To overcome this, our new decomposition explicitly prevents such correlations from emerging, ensuring that elements with small marginal probabilities do not systematically appear together.

1.2.2 An Improved Decomposition

The decomposition guarantee achieved in [30] (see Lemma 4) only ensures that for each individual element $i \in S_j$, we have that $p_{i,S_j} = \Pr_\pi[i \in C_\pi] = \widetilde{\Omega}\big(1/|S_j|\big)$. While this prevents any single element from being too rare, it still admits highly correlated configurations in which many low-marginal elements systematically co-occur in the same circuits (as in the preceding example).

Formally, the issue is that although each element may satisfy $p_{i,S_j} \gtrsim 1/n$, there may exist a *large subset* T (e.g., P_2 in the example above) with $\Pr_\pi[C_\pi \cap T \neq \varnothing] \approx 1/n$. This occurs because elements in T are highly correlated: either none appear in the first circuit, or $\Omega(|T|)$ of them do.

Our first step is therefore to rule out such set-level correlations, not just rare elements. To eliminate this obstacle, we strengthen the decomposition to enforce the following *subset-hitting property*:

$$\forall\, T \subseteq S_j, \quad p_{T,S_j} = \Pr_\pi[T \cap C_\pi \neq \varnothing] = \widetilde{\Omega}\big(|T|/|S_j|\big).$$

That is, *every* subset T of elements is hit by the first circuit with probability (up to polylogarithmic factors) proportional to its size. In the example above, P_2 would violate this condition, since the property demands a $\widetilde{\Omega}(1)$ intersection probability rather than the $\Theta(1/n)$ that arises from correlation.

Although this requirement is strictly stronger than element-wise marginals (it constrains exponentially many subsets), we show that such parts *do* exist and can be computed within the same round and query bounds as in [30]. Moreover, our decomposition continues to satisfy all guarantees of Lemma 4 while adding the new subset-hitting property. Operationally, this property prevents mass from concentrating on a few large, highly correlated circuits and ensures that circuit mass is distributed in a stable manner across each peeled part, allowing for more powerful routines for deleting redundant elements.

1.2.3 Core vs. Non-core Elements

With this strengthened decomposition theorem established, it still remains to show how we can use the decomposition to explicitly make progress by deleting redundant elements. The next step is to stratify elements by their marginal probabilities. Intuitively, elements with small marginals are conducive for finding short-circuit witnesses (thus enabling many deletions), whereas elements with large marginals can be handled by a deletion procedure from [30].

To this end, we partition the elements of each peeled set S_j into two groups:

(a) *Core elements:* elements $i \in S_j$ with $p_{i,S_j} > \frac{\alpha(S_j)^2}{|S_j|^2}$, so that $p_{i,S_j} \cdot (|S_j|^2/\alpha(S_j)^2) > 1$, and

(b) *Non-core elements:* the remaining elements, each with $p_{i,S_j} \leqslant \frac{\alpha(S_j)^2}{|S_j|^2}$.

We will denote by CORE, the set of core elements. The choice of the threshold $\frac{\alpha(S_j)^2}{|S_j|^2}$ in defining core vs. non-core elements is deliberate: a lemma (implicitly) established in [30] shows that one can delete

$$\widetilde{\Omega}\Big(\sum_{i\in S_j} \min\{1,\, p_{i,S_j}\cdot(|S_j|^2/\alpha(S_j)^2)\}\Big) \tag{3}$$

redundant elements using a single round of queries.

Restricting (3) to the core immediately yields $\widetilde{\Omega}\big(|\text{CORE}|\big)$ deletions in one round (the second term in the min is 1 by definition of core). Hence if CORE is a constant fraction of S_j, we delete $\widetilde{\Omega}(|S_j|)$ elements in one round.

We therefore focus on the complementary case where |CORE| is small. In this event, let $\ell = \sum_{i\in S_j\setminus\text{CORE}} p_{i,S_j}$ denote the sum of all the marginal probabilities of the non-core elements. Applying Equation (3) to these non-core elements implies we can delete $\widetilde{\Omega}\Big(\frac{|S_j|^2}{\alpha(S_j)^2}\,\ell\Big)$ elements in a single round. Thus if ℓ is large, we once again ensure many deletions.

The remaining case is when ℓ is small. In this regime we *prove* (using the subset-hitting property) that most non-core elements admit short-circuit witnesses of size $\widetilde{O}(\ell)$ (or more formally, admit circuits who only have a small number of *non-core* elements), which allows us to delete $\widetilde{\Omega}(|S_j|/\ell)$ redundant elements in one round via a modification of the aforementioned short-circuit deletion procedure. We establish this formally below.

Finding redundant elements when non-core circuit mass is small. As before, we let ℓ denote the total marginal probability mass of the non-core elements: $\ell = \sum_{i\in S_j\setminus\text{CORE}} p_{i,S_j}$. Now, observe that this quantity equals the expected number of non-core elements in the first circuit:

$$\ell = \sum_{i\in S_j\setminus\text{CORE}} p_{i,S_j} = \mathbb{E}_\pi[|(S_j\setminus\text{CORE})\cap C_\pi|]. \tag{4}$$

This implies that, in expectation, the number of elements in $S_j\setminus$CORE in each circuit that appears is bounded by ℓ. The question is whether this expectation can be algorithmically leveraged to certify many short circuits for non-core elements.

It turns out that the subset-hitting guarantee of our new decomposition yields an affirmative answer. Informally, sample many (a large polynomial number) of random permutations π over S_j, and find the first circuits C_π that form. Suppose there is a large set $T \subseteq S_j\setminus$CORE (say, $|T| = \Omega(|S_j|)$) such that for every $x \in T$ we *fail* to find a short circuit containing x, namely, no circuit C_x with $x \in C_x$ and $|C_x \cap (S_j\setminus\text{CORE})| \leqslant \ell\log n$. Then every time C_π intersects T we must have $|C_\pi \cap (S_j\setminus\text{CORE})| \geqslant \ell\log n$. By the subset-hitting property, $\Pr_\pi[C_\pi \cap T \neq \varnothing] \gtrsim |T|/|S_j| = \Omega(1)$. Hence

$$\mathbb{E}_\pi\big[\,|C_\pi\cap(S_j\setminus\text{CORE})|\,\big] \;\geqslant\; \Omega(1)\cdot\ell\log n \;>\; \ell,$$

contradicting (4). Therefore $|T| = o(|S_j|)$: for *most* non-core elements we do find short circuits, enabling one-round deletion of $\widetilde{\Omega}(|S_j|/\ell)$ elements.

To summarize, if the set of core elements forms only a small fraction of S_j, the analysis above yields a one-round win–win: either (a) the [30] deletion bound gives $\widetilde{\Omega}\left(\frac{|S_j|^2}{\alpha(S_j)^2}\cdot\ell\right)$ deletions, or (b) short-circuit witnesses allow $\widetilde{\Omega}(|S_j|/\ell)$ deletions. Balancing these outcomes gives an unconditional bound of $\widetilde{\Omega}\left(\frac{|S_j|^{3/2}}{\alpha(S_j)}\right)$ deletions, strictly improving upon [30]'s $\widetilde{\Omega}\left(\frac{|S_j|^2}{\alpha(S_j)^2}\right)$. We next show how this stronger local progress integrates into the global round analysis.

1.2.4 Global Analysis by Leveraging Evolving α-Values

In the preceding discussion, we introduced a new subroutine for making progress on each set S_j produced by our decomposition. To summarize, we now have two primary ways of making progress:

- Contraction: contracting an independent set of size $\Omega\left(\frac{\alpha(S_j)}{|S_j|} \cdot n\right)$ (Lemma 4, Item 3).
- Deletion: deleting $\tilde{\Omega}\left(\frac{|S_j|^{3/2}}{\alpha(S_j)}\right)$ redundant elements, improving on the $\tilde{\Omega}\left(\frac{|S_j|^2}{\alpha(S_j)^2}\right)$ bound from [30] (Lemma 5, Item 2). (For clarity we omit the $\min\{\cdot, |S_j|\}$ truncation since it does not affect asymptotics here.)

The key parameter that controls the efficiency of these routines is $\alpha(S_j)$, which governs both the likelihood of quickly finding a large independent set or deleting in parallel a large number of redundant elements. To achieve an overall round complexity of $\widetilde{O}(n^{3/7})$, we need to ensure that on *average*, each round leads to contraction or deletion of $\widetilde{\Omega}(n^{4/7})$ elements. We set $f := n^{4/7}$ as this target average progress per round. While some rounds may fall short (depending on $\alpha(S_j)$), we prove that a prolonged small-progress regime is impossible; the process must transition to a regime that achieves amortized progress at least f when averaged over all sets peeled off in the decomposition.

An adaptive strategy. In [30], the matroid decomposition algorithm is always executed until the total size of the peeled off parts reaches at least $n/2$ elements, with each peeling step costing one round. Our adaptive strategy is to terminate *as soon as* the average progress meets f. In particular, when the algorithm peels off the jth part S_j, it can terminate immediately if:

$$\sum_{j' \leqslant j} \frac{|S_{j'}|^{3/2}}{\alpha(S_{j'})} \geqslant j \cdot f \text{ (deletion)} \quad \text{or} \quad \frac{\alpha(S_j)}{|S_j|} \cdot n \geqslant j \cdot f \text{ (contraction)},$$

At that point, the progress guarantees imply that the average progress per round is already $\tilde{\Omega}(f)$.

Intuitively, small $\alpha(S_j)$ favors deletion; large $\alpha(S_j)$ favors contraction. Because α increases along the decomposition (for comparable part sizes), there is a point at which contraction dominates. Continuing to peel beyond this point yields diminishing average progress (the independent set size grows only by a constant factor, while rounds accrue), so we stop as soon as we achieve average progress f.

Improved global analysis. The potentially problematic scenario is when the decomposition produces a sequence of peeled sets $S_1, \ldots, S_\gamma$ where the average progress per round is always much smaller than f. To rule this out, we choose f as large as possible subject to the condition that no such sequence exists. Formally, we must forbid sets $S_1, \ldots, S_\gamma$ satisfying:

1. $\sum_{j=1}^{\gamma} |S_j| \leqslant n$.
2. $\alpha(S_j)$'s satisfy the recurrence given in Lemma 4, Equation (2).
3. For every $j \in [\gamma]$, $\sum_{j' \leqslant j} \frac{|S_{j'}|^{3/2}}{\alpha(S_{j'})} < j \cdot f$ and $\frac{|S_j|}{\alpha(S_j)} \cdot n < j \cdot f$.

The benefit of our adaptive strategy is that the constraints in Item 3 must hold for all prefixes $j \in [\gamma]$, since the decomposition algorithm will terminate as soon as the average progress reaches f, not just for the final index $j = \gamma$ as in [30]. This prefix condition is *strictly stronger* than the terminal condition in [30], and permits a larger feasible f.

A surprising consequence is that even without the new decomposition theorem and $\widetilde{\Omega}\left(\frac{|S_j|^{3/2}}{\alpha(S_j)}\right)$ deletion subroutine, using only the existing $\widetilde{\Omega}\left(\frac{|S_j|^2}{\alpha(S_j)^2}\right)$ deletion algorithm from [30] (Lemma 5, Item 2), the adaptive strategy alone improves the round complexity to $\tilde{O}(n^{4/9})$. We give a simple self-contained proof of this fact in the appendix of the full version.

Incorporating the stronger deletion subroutine. Once the new $\widetilde{\Omega}\left(\frac{|S_j|^{3/2}}{\alpha(S_j)}\right)$ deletion routine is used, the analysis becomes substantially more delicate. In obtaining the $\tilde{O}(n^{4/9})$ round result, homogeneity in $|S_j|$ and $\alpha(S_j)$ allows reduction to the single ratio $\alpha(S_j)/|S_j|$. The bound $\frac{|S_j|^{3/2}}{\alpha(S_j)}$ breaks this homogeneity, so we must track both parameters explicitly, leading to a more complex analysis.

To analyze it, we group the peeled sets $S_1, \ldots, S_\gamma$ by size into $\log n$ categories where each category contains sets whose sizes differ by a factor of at most 2. We then focus on the *dominant category*, i.e., the size range currently containing the most such sets. We keep track of when the dominant category changes, and what the new dominant category becomes. Let a_j denote the number of rounds spent between the $(j-1)$st and jth change of dominant category, and let b_j denote the representative size of the sets in the dominant category during this interval. Using Item 2 and Item 3, we are able to derive a tight recursive relationship on how these parameters evolve. In particular, we show in in the full version that for any $j \geqslant 1$,

$$b_j = \frac{f^{7\cdot 2^j - 6}}{n^{4\cdot 2^j - 4}} \cdot \log^{O(2^j)}(n), \qquad a_j = \widetilde{O}\left(\frac{f \cdot b_j}{n}\right).$$

Importantly, if we select f to be slightly smaller than $n^{4/7}$, then in the above recursive formulation, these bounds on b_j, and hence also a_j, will be strongly decaying. This immediately implies that our decomposition must terminate in a small number of rounds, which is only possible if either (1) the average progress exceeds f, or (2) we have peeled off $\Omega(n)$ elements in our decomposition. Clearly, if case (1) occurs, we have succeeded, and so the only remaining case is (2). The final piece of our argument shows that, after recovering $\Omega(n)$ elements in our decomposition, there is always a way to recover $\widetilde{\Omega}(n^{3/4})$ redundant or independent elements. Dividing this by our bound on the number of rounds invested so far (using the recurrence relation above) then guarantees our average progress is $\tilde{\Omega}(f)$.

A more detailed analysis and discussion of this argument appears in the full version.

Organization. In the remainder of this paper, we introduce preliminaries, and then in Section 3 we prove explain our new decomposition algorithm, which guarantees the stronger "subset-hitting" property described above. In the full version of the paper, we formalize the tradeoff between circuit sizes and marginal probabilities, yielding our improved subroutine that recovers $\widetilde{\Omega}\left(\frac{|S_j|^{3/2}}{\alpha(S_j)}\right)$ redundant elements from each part S_j in a single round. The full version then leverages this refined subroutine to establish the global $\widetilde{O}(n^{3/7})$ round bound through a careful amortized analysis.

2 Preliminaries

▶ **Definition 6** (Matroids). *A* matroid $\mathcal{M} = (E, \mathcal{I})$ *is a pair where* E *is a finite ground set and* $\mathcal{I} \subseteq 2^E$ *is a collection of independent sets with the following properties: (i)* $\varnothing \in \mathcal{I}$ *(non-triviality), (ii) for every* $S \in \mathcal{I}$ *and* $S' \subset S$, $S' \in \mathcal{I}$ *(downward-closedness), and (iii) for every* $S, S' \in \mathcal{I}$ *and* $|S'| < |S|$, *there exists some* $x \in S \backslash S'$ *such that* $S + x \in \mathcal{I}$ *(exchange property).*

▶ **Definition 7** (Independent Sets, Circuits, Bases). *For a matroid $\mathcal{M} = (E, \mathcal{I})$, we say a set $S \subseteq E$ is* independent *if $S \in \mathcal{I}$ and* dependent *otherwise. We call a set B a* basis *if it is a maximal independent set, i.e. for any $x \notin B$, $B + x \notin \mathcal{I}$. We call a set C a* circuit *if it is a minimal dependent set, i.e. for any $x \in C$, $C - x \in \mathcal{I}$.*

▶ **Definition 8** (Rank). *For a matroid $\mathcal{M} = (E, \mathcal{I})$, we define the* rank *of $\mathcal{M}$ as $rank(\mathcal{M}) = \max_{S \in \mathcal{I}} |S|$. Further, for any $S \subseteq E$, we define $rank_{\mathcal{M}}(S) = \max_{T \subseteq S, T \in \mathcal{I}} |T|$. The rank function of a matroid is submodular.*

▶ **Definition 9** (Span). *In a matroid $\mathcal{M} = (E, \mathcal{I})$, we define $span(S)$ as*

$$span(S) = \{x \in E \mid rank(S \cup \{x\}) = rank(S)\}.$$

▶ **Definition 10** (Restriction, Contraction). *Let $\mathcal{M} = (E, \mathcal{I})$ be a matroid and $S \subseteq E$. We write $\mathcal{M}|_S$ for the restriction of $\mathcal{M}$ to S, and we use $M - S$ to mean the restriction of M to $E \backslash S$, and $\mathcal{M}/S$ for the contraction of $\mathcal{M}$ by S, whose rank function is $rank_{\mathcal{M}/S}(T) = rank_{\mathcal{M}}(S \cup T) - rank_{\mathcal{M}}(T)$ for any $T \subseteq E \backslash S$.*

▶ **Definition 11** (Permutation-Induced Circuit). *Let $\mathcal{M} = (E, \mathcal{I})$ be a matroid, and let π be an arbitrary permutation over E. We let C_π denote the unique first circuit which appears when adding elements in the order of π. I.e., if $\{e_{\pi(1)}, \dots e_{\pi(j)}\} \in \mathcal{I}$, but $\{e_{\pi(1)}, \dots e_{\pi(j)}, e_{\pi(j+1)}\} \notin \mathcal{I}$, we let C_π denote the unique circuit in $\{e_{\pi(1)}, \dots e_{\pi(j)}, e_{\pi(j+1)}\}$.*

3 A New Decomposition Algorithm

In this section, we introduce the key notion of globally optimal sets and present our new decomposition and its implications. Section 3.1 introduces globally optimal sets and how they can be efficiently identified using the independence oracle, and in Section 3.2, we present our iterative decomposition process and highlight properties of the evolving $\alpha()$ values during the decomposition.

3.1 Globally-Optimal Sets

To start, we present a modification of the decomposition procedure presented in [30]. Whereas this previous work constructed *greedily-optimal* sets, here we instead introduce *globally-optimal* sets. Before defining these notions, we introduce some key parameters that we will utilize.

▶ **Definition 12.** *For a matroid $\mathcal{M} = (E, \mathcal{I})$ and $S \subseteq E$, we let $\alpha(S)$ denote the median number of elements sampled from the S before a circuit forms. I.e.,*

$$\alpha(S) = \min \left\{ k \in \mathbb{N} : \Pr_{T \sim \binom{S}{k}} [\mathrm{Ind}(T) = 1] \leqslant \frac{1}{2} \right\}.$$

▶ **Definition 13.** *Let $\mathcal{M} = (E, \mathcal{I})$ be a matroid over n elements. For $i \in E$, we say that*

$$p_{i,\mathcal{M}} = \Pr_{\pi}[i \in C_\pi],$$

where C_π is the unique first circuit that appears when adding elements in the order of π. More generally, for an arbitrary set $T \subseteq E$, we define the ***hitting probability*** *of the set T as*

$$p_{T,\mathcal{M}} = \Pr_{\pi}[T \cap C_\pi \neq \varnothing].$$

▶ **Definition 14.** *Let $\mathcal{M} = (E, \mathcal{I})$ be a matroid over n elements. For $T \subseteq E$, we define T's* ***circuit mass in $\mathcal{M}$*** *to be*

$$q_{T,\mathcal{M}} = \Pr_{\pi}[C_\pi \subseteq T].$$

Often we will denote this by q_T when the parent matroid $\mathcal{M}$ is clear by context.

Before defining globally-optimal and greedily-optimal sets, we first recall that in a single round (and using only polynomially many queries), we can estimate q_S to small error *for every* S:

▷ Claim 15 (Claim 4.4 of [30]). There is a one round algorithm using polynomially many independence queries, which for a matroid $\mathcal{M} = (E, \mathcal{I})$ and every subset $S \subseteq E$, yields $\hat{q}_S$ such that

$$|\widehat{q}_S - q_S| \leqslant \frac{1}{n^2},$$

with probability $1 - 2^{-\Omega(n)}$.

With this, we can now introduce the notion of a *greedily-optimal* set:

▶ **Definition 16** (Definition 4.9, Claim 4.11 of [30])**.** *For a matroid $\mathcal{M} = (E, \mathcal{I})$, we say a set $S \subseteq E$ is* greedily-optimal *if*

1. $q_S \geqslant 1 - 2^{-20}$
2. *For every $x \in S$,*

$$p_{x,\mathcal{M}|_S} \geqslant \frac{1}{2^{21}|S|\log(n)}.$$

As discussed in Section 1.2.2, we strengthen the definition by requiring that the second property holds for arbitrary sets $T \subseteq S$, in the sense that including multiple elements *scales* the hitting probability, rather than just holding individually for each element $x \in S$:

▶ **Definition 17.** *Let $\mathcal{M} = (E, \mathcal{I})$ be a matroid over n elements. We say that a set $S \subseteq E$ is* ***globally optimal*** *if:*

1. $q_{S,\mathcal{M}} \geqslant 1 - 2^{-20}$.
2. $\forall T \subseteq S$, *it is the case that* $p_{T,\mathcal{M}|_S} \geqslant \frac{|T|}{2^{21}|S|\log(n)}$.

Note that here that 2^{20} and 2^{21} are just sufficiently large constants to ensure the probabilistic arguments go through.

▶ Remark 18. Observe that globally-optimal sets (Definition 17) are *also* greedily-optimal sets as defined in Definition 16, as we can consider the singleton sets where $T = \{x\}$ for elements $x \in S$. Importantly, this means that going forward, all properties that [30] established for greedily-optimal sets also *automatically* hold for globally-optimal sets.

Ultimately, our goal is to have an algorithm for *decomposing* the matroid $\mathcal{M}$ into a sequence of sets which are all *globally optimal*, while still maintaining guarantees on how the α-values of these sets grow:

▶ **Lemma 19.** *Let $\mathcal{M}$ be a matroid. There is a decomposition algorithm, making polynomially many queries, such that if the algorithm runs for k rounds, it recovers sets $S_1, \ldots S_k$, where every $S_i : i \in [k]$ is* globally-optimal *with respect to $\mathcal{M} - S_1 - \ldots S_{i-1}$. Further, for any $\ell \in [\log n]$, if we let $T = \{i \in [k] : |S_i| \in [2^\ell, 2^{\ell+1} - 1]\}$, $\gamma = |T|$, and let $a_1, \ldots a_\gamma$ denote the indices in T, then with probability $1 - 2^{-\Omega(n)}$ it must be the case that:*

1. $\alpha(S_{a_i}) = \Omega(i^2)$ *for every* $i \in [\gamma]$.
2. $\gamma = O\left(\sqrt{2^\ell}\right)$.
3. $k = O(n^{1/3})$.

Additionally, we have the property that for any $i < j \in [k]$, $\frac{\alpha(S_j)}{|S_j|} = \Omega\left(\frac{\alpha(S_i)}{|S_i|}\right)$.

Now, before proving this lemma, we require a few building blocks. To start, we show in the rest of this subsection that a globally optimal set can be constructed by a simple algorithm: we start by setting S to be the ground set E, and continue to remove sets T from S that do not alter the probability mass of $\hat{q}_S$ by too much:

Algorithm 1 GloballyOptimalConstructor($\mathcal{M} = (E, \mathcal{I})$).

```
Let S = E.
Use Claim 15 to calculate q̂_T for every T ⊆ S.
while ∃T ⊆ S such that q̂_S − q̂_{S\T} ⩽ |T| / (2^20 |S| log(n)) do
    S ← S\T
end
return S
```

Immediately, we have the following claim:

▷ Claim 20. Suppose we invoke Algorithm 1 on a matroid $\mathcal{M} = (E, \mathcal{I})$, yielding a set S. Then, for every set $T \subseteq S$, it must be the case that

$$p_{T,\mathcal{M}|_S} \geqslant \frac{|T|}{2^{21}|S|\log(n)}.$$

Proof. Whenever a set S is returned, it must be the case that $\forall T \subseteq S$ that

$$\hat{q}_S - \hat{q}_{S\setminus T} \geqslant \frac{|T|}{2^{20}|S|\log(n)}.$$

In particular, because $|\widehat{q}_S - q_S| \leqslant \frac{1}{n^2}$ (with high probability), we also know that

$$q_S - q_{S\setminus T} \geqslant \frac{|T|}{2^{20}|S|\log(n)} - \frac{2}{n^2} \geqslant \frac{|T|}{2^{21}|S|\log(n)}.$$

Next, observe that

$$q_S - q_{S\setminus T} = \Pr_\pi[C_\pi \subseteq S] - \Pr_\pi[C_\pi \subseteq S\setminus T].$$

Thus, the only way for a circuit C_π to contribute to q_S and not $q_{S\setminus T}$ is if C_π contains at least one element from T. Formally,

$$q_S - q_{S\setminus T} = \Pr_\pi[C_\pi \subseteq S \wedge C_\pi \cap T \neq \varnothing].$$

Thus, we obtain that

$$\frac{|T|}{2^{21}|S|\log(n)} \leqslant \Pr_\pi[C_\pi \subseteq S \wedge C_\pi \cap T \neq \varnothing],$$

Finally, we can observe that

$$\Pr_\pi[C_\pi \subseteq S \wedge C_\pi \cap T \neq \varnothing] \leqslant p_{T,\mathcal{M}|_S}.$$

This is because whenever we sample in accordance to a permutation π over E (the ground set of $\mathcal{M}$), and recover a circuit C_π such that $T \cap C_\pi \neq \varnothing$ and $C_\pi \subseteq S$, the same permutation, if restricted to S and used to sample elements of S, would have still given a circuit such that $T \cap S \neq \varnothing$. Together then, this means that

$$p_{T,\mathcal{M}|_S} \geqslant \frac{|T|}{2^{21}|S|\log(n)},$$

as we desire. ◁

Next, we have the following claim:

▷ Claim 21. Suppose we invoke Algorithm 1 on a matroid $\mathcal{M}$, yielding a set S. Then,

$$q_{S,\mathcal{M}} \geqslant 1 - 2^{-20}.$$

Proof. Consider an iteration of Algorithm 1 starting with a set S. We then recover a set T such that

$$\hat{q}_S - \hat{q}_{S\backslash T} \leqslant \frac{|T|}{2^{20}|S|\log(n)},$$

and set $S \leftarrow S\backslash T$. Inductively, we claim that when S has ℓ elements remaining, that

$$\hat{q}_S \geqslant 1 - 2^{-20} + \sum_{i=1}^{\ell} \frac{1}{2^{20} i \log(n)}.$$

As a base case, we can consider when $S = E$, i.e., $|S| = n$. Then, we have that

$$1 - 2^{-20} + \sum_{i=1}^{\ell} \frac{1}{2^{20} i \log(n)} \leqslant 1 - 2^{-20} + \frac{1 + \ln(n)}{2^{20}\log(n)} \leqslant 1 = \hat{q}_S.$$

Now, consider an iteration where we have an intermediate set S' of size ℓ', and subsequently remove a set T, yielding a set S with ℓ elements. By induction, we suppose that

$$\hat{q}_{S'} \geqslant 1 - 2^{-20} + \sum_{i=1}^{\ell'} \frac{1}{2^{20} i \log(n)}.$$

Now, for the set T of elements that we remove (which is being removed from S'), we know that

$$\hat{q}_{S'} - \hat{q}_S \leqslant \frac{|T|}{2^{20}|S'|\log(n)}.$$

This means that

$$\hat{q}_S \geqslant \hat{q}_{S'} - \frac{|T|}{2^{20}|S'|\log(n)} \geqslant 1 - 2^{-20} + \left(\sum_{i=1}^{\ell'} \frac{1}{2^{20} i \log(n)}\right) - \frac{|T|}{2^{20}|S'|\log(n)}.$$

Importantly, because $S = S'\backslash T$, we know that $|T| = \ell' - \ell$, so we can now re-write

$$\frac{|T|}{2^{20}|S'|\log(n)} = \sum_{i=\ell+1}^{\ell'} \frac{1}{2^{20}\ell'\log(n)}.$$

Plugging this in, we see that

$$\begin{aligned}\hat{q}_S &\geqslant 1 - 2^{-20} + \left(\sum_{i=1}^{\ell} \frac{1}{2^{20} i \log(n)}\right) + \left(\sum_{i=\ell+1}^{\ell'} \frac{1}{2^{20} i \log(n)} - \frac{1}{2^{20}\ell' \log(n)}\right) \\ &\geqslant 1 - 2^{-20} + \left(\sum_{i=1}^{\ell} \frac{1}{2^{20} i \log(n)}\right) + \left(\sum_{i=\ell+1}^{\ell'} \frac{1}{2^{20}\ell' \log(n)} - \frac{1}{2^{20}\ell' \log(n)}\right) \\ &\geqslant 1 - 2^{20} + \left(\sum_{i=1}^{\ell} \frac{1}{2^{20} i \log(n)}\right),\end{aligned}$$

thus yielding our inductive claim. Finally, we use Claim 15 to conclude that $q_S \geqslant 1 - 2^{-20}$, given that $\hat{q}_S \geqslant 1 - 2^{-20} + \frac{1}{2^{20}\log(n)}$ (and assuming n is sufficiently large). This yields the claim. ◁

To summarize the above discussion, we now have the following lemma:

▶ **Lemma 22.** *Algorithm 1 returns a set S which is globally-optimal with probability $1-2^{-\Omega(n)}$.*

Proof. Claim 21 shows that $q_S \geqslant 1 - 2^{-20}$, and Claim 20 shows that $p_{T,\mathcal{M}|_S} \geqslant \frac{|T|}{2^{21}|S|\log(n)}$. The probability bound follows from Claim 15. ◀

In the following subsection, we show how we can repeatedly invoke the above lemma by doing so-called *iterative peeling*. In each round of peeling, we recover a new globally optimal set, thus creating the sequence of globally optimal sets guaranteed in Lemma 19.

3.2 Iterative Peeling

As in [30], our next step is to *repeat* this procedure of peeling off sets. We present this algorithm below:

Algorithm 2 RepeatedGlobalPeeling($\mathcal{M} = (E, \mathcal{I})$).

```
M ← RemoveSmallCircuits(M) // Algorithm 7 in [30]
n ← |E|
k ← 0
while M ≠ ∅ do
    k ← k + 1
    S_k ← GloballyOptimalConstructor(M).
    M ← M\S_k.
    if α(S_k)/|S_k| ⩾ 1/log(n) or |S_k| > n/2 then
        return S_1, ... S_{k-1}
    end
end
return S_1, ... S_k
```

▶ Remark 23. At the beginning of Algorithm 2, we invoke the RemoveSmallCircuits procedure (Algorithm 7 in [30]) to eliminate circuits of size $\leqslant 50$. This is for a minor technical reason in the probabilistic argument, the starting circuit size to be a sufficiently large constant. We omit the details here as they are identical to [30].

Notationally, we also let $\mathcal{M}_j = \mathcal{M} - S_1 - \cdots - S_{j-1}$. While our globally-optimal sets are different than the notion of greedily-optimal sets used in [30], our decomposition still enjoys all of the same favorable properties. Importantly, we have the following lemma which governs the growth of the α-value within the sets that we peel off:

▶ **Theorem 24** (Theorem 4.6 in [30]). *Let $\mathcal{M}$ be a matroid, and let $S_1, \dots S_k$ be a sequence of greedily-optimal sets that are peeled off (analogously to Algorithm 2). Now, let $\ell \in [\log n]$ be an integer, let $T = \{i \in [k] : |S_i| \in [2^\ell, 2^{\ell+1} - 1]\}$, let $\gamma = |T|$, and let $a_1, \dots a_\gamma$ denote the indices in T. Then, with probability $1 - 2^{-\Omega(n)}$ it must be the case that*

$$\alpha(S_{a_i}) = \Omega(i^2) \text{ for every } i \in [\gamma], \quad \gamma = O\left(\sqrt{2^\ell}\right), \quad k = O(n^{1/3}).$$

Additionally, for any $i < j \in [k]$,

$$\frac{\alpha(S_j)}{|S_j|} = \Omega\left(\frac{\alpha(S_i)}{|S_i|}\right).$$

Note that the proof of the final part of the claim above was actually *not* presented in [30]. For this reason, we provide a complete proof of the above in the full version.

Finally, with this machinery in place, we can now prove Lemma 19:

Proof. First, we must show that each set S_i that is recovered is globally-optimal with respect to $\mathcal{M} - S_1 - \dots S_{i-1}$. This follows by invoking Lemma 22, as the algorithm Algorithm 2 calls GloballyOptimalConstructor on $\mathcal{M} - S_1 - \dots S_{i-1}$ to construct S_i.

The remaining properties hold by first using Remark 18 to observe that our *globally* optimal sets are also *greedily* optimal sets, and then invoking Theorem 24 on this sequence of greedily optimal sets that is removed. ◀

3.3 Finding Independent Sets via Globally-Optimal Sets

With our notion of globally-optimal sets established, and an efficient algorithm for peeling them off, we also briefly mention here that the α value of each S_j has a strong relationship with the α value of the parent matroid $\mathcal{M}_j$. Specifically:

▷ Claim 25. Let $\mathcal{M} = (E, \mathcal{I})$ be a matroid on n elements, and let $S \subseteq E$ be a globally-optimal set in M. Then, for $\ell = \frac{\alpha(S)}{10|S|} n$, we have

$$\Pr_\pi[\text{Ind}(\{\pi(1), \dots, \pi(\ell)\}) = 1] \geqslant \frac{1}{4}.$$

Proof. This follows by noting that a globally-optimal set is *also* a greedily-optimal set as per Remark 18. Then, we can simply invoke Claim 5.1 of [30]. ◁

Importantly, this gives the following lemma:

▶ **Lemma 26.** *There is a 1-round, polynomial-query algorithm which, given a matroid $\mathcal{M}$ on n elements and globally-optimal set S, recovers $\Omega\left(\frac{\alpha(S)}{|S|} n\right)$ independent elements with probability $1 - 2^{-n}$.*

Proof. Simply sample polynomially many random permutations in accordance with Claim 25. With exponentially high probability, one of these random permutations will find $\ell = \frac{\alpha(S)}{10|S|} n$ independent elements. ◀

For brevity, we omit the discussion of how this decomposition is used to improve basis finding. We invite the interested reader to view the full version of the paper for these details.

References

1. Eric Balkanski, Aviad Rubinstein, and Yaron Singer. An exponential speedup in parallel running time for submodular maximization without loss in approximation. In Timothy M. Chan, editor, *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 283–302. SIAM, 2019. doi:10.1137/1.9781611975482.19.
2. Eric Balkanski, Aviad Rubinstein, and Yaron Singer. An optimal approximation for submodular maximization under a matroid constraint in the adaptive complexity model. In Moses Charikar and Edith Cohen, editors, *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC 2019, Phoenix, AZ, USA, June 23-26, 2019*, pages 66–77. ACM, 2019. doi:10.1145/3313276.3316304.
3. Eric Balkanski and Yaron Singer. The adaptive complexity of maximizing a submodular function. In Ilias Diakonikolas, David Kempe, and Monika Henzinger, editors, *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, pages 1138–1151. ACM, 2018. doi:10.1145/3188745.3188752.
4. Eric Balkanski and Yaron Singer. Approximation guarantees for adaptive sampling. In Jennifer G. Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 393–402. PMLR, 2018. URL: http://proceedings.mlr.press/v80/balkanski18a.html.
5. Eric Balkanski and Yaron Singer. A lower bound for parallel submodular minimization. In Konstantin Makarychev, Yury Makarychev, Madhur Tulsiani, Gautam Kamath, and Julia Chuzhoy, editors, *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC 2020, Chicago, IL, USA, June 22-26, 2020*, pages 130–139. ACM, 2020. doi:10.1145/3357713.3384287.
6. Nikhil Bansal, Rudi A Pendavingh, and Jorn G Van der Pol. On the number of matroids. *Combinatorica*, 35:253–277, 2015. doi:10.1007/S00493-014-3029-Z.
7. Joakim Blikstad. Breaking o(nr) for matroid intersection. In Nikhil Bansal, Emanuela Merelli, and James Worrell, editors, *48th International Colloquium on Automata, Languages, and Programming, ICALP 2021, July 12-16, 2021, Glasgow, Scotland (Virtual Conference)*, volume 198 of *LIPIcs*, pages 31:1–31:17. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPIcs.ICALP.2021.31.
8. Joakim Blikstad. Sublinear-round parallel matroid intersection. In Mikolaj Bojanczyk, Emanuela Merelli, and David P. Woodruff, editors, *49th International Colloquium on Automata, Languages, and Programming, ICALP 2022, July 4-8, 2022, Paris, France*, volume 229 of *LIPIcs*, pages 25:1–25:17. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPIcs.ICALP.2022.25.
9. Joakim Blikstad and Ta-Wei Tu. Efficient matroid intersection via a batch-update auction algorithm. In Ioana Oriana Bercea and Rasmus Pagh, editors, *2025 Symposium on Simplicity in Algorithms, SOSA 2025, New Orleans, LA, USA, January 13-15, 2025*, pages 226–237. SIAM, 2025. doi:10.1137/1.9781611978315.18.
10. Adam Breuer, Eric Balkanski, and Yaron Singer. The FAST algorithm for submodular maximization. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 1134–1143. PMLR, 2020. URL: http://proceedings.mlr.press/v119/breuer20a.html.
11. Deeparnab Chakrabarty, Yu Chen, and Sanjeev Khanna. A polynomial lower bound on the number of rounds for parallel submodular function minimization. In *62nd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2021, Denver, CO, USA, February 7-10, 2022*, pages 37–48. IEEE, 2021. doi:10.1109/FOCS52979.2021.00013.
12. Deeparnab Chakrabarty, Andrei Graur, Haotian Jiang, and Aaron Sidford. Improved lower bounds for submodular function minimization. In *63rd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2022, Denver, CO, USA, October 31 - November 3, 2022*, pages 245–254. IEEE, 2022. doi:10.1109/FOCS54457.2022.00030.

13 Deeparnab Chakrabarty, Yin Tat Lee, Aaron Sidford, Sahil Singla, and Sam Chiu-wai Wong. Faster matroid intersection. In David Zuckerman, editor, *60th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2019, Baltimore, Maryland, USA, November 9-12, 2019*, pages 1146–1168. IEEE Computer Society, 2019. doi:10.1109/FOCS.2019.00072.

14 Chandra Chekuri and Kent Quanrud. Parallelizing greedy for submodular set function maximization in matroids and beyond. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 78–89, 2019. doi:10.1145/3313276.3316406.

15 Chandra Chekuri and Kent Quanrud. Submodular function maximization in parallel via the multilinear relaxation. In Timothy M. Chan, editor, *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 303–322. SIAM, 2019. doi:10.1137/1.9781611975482.20.

16 Lin Chen, Moran Feldman, and Amin Karbasi. Unconstrained submodular maximization with constant adaptive complexity. In Moses Charikar and Edith Cohen, editors, *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC 2019, Phoenix, AZ, USA, June 23-26, 2019*, pages 102–113. ACM, 2019. doi:10.1145/3313276.3316327.

17 Jack Edmonds. Submodular functions, matroids, and certain polyhedra. In *Combinatorial Structures and Their Applications*, pages 69–87. Gordon and Breach, 1970.

18 Jack Edmonds. Matroid partition. In *50 Years of Integer Programming 1958-2008: From the Early Years to the State-of-the-Art*, pages 199–217. Springer, 2009. doi:10.1007/978-3-540-68279-0_7.

19 Alina Ene and Huy L. Nguyen. Submodular maximization with nearly-optimal approximation and adaptivity in nearly-linear time. In Timothy M. Chan, editor, *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 274–282. SIAM, 2019. doi:10.1137/1.9781611975482.18.

20 Alina Ene, Huy L. Nguyen, and Adrian Vladu. Submodular maximization with matroid and packing constraints in parallel. In Moses Charikar and Edith Cohen, editors, *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC 2019, Phoenix, AZ, USA, June 23-26, 2019*, pages 90–101. ACM, 2019. doi:10.1145/3313276.3316389.

21 Matthew Fahrbach, Vahab S. Mirrokni, and Morteza Zadimoghaddam. Non-monotone submodular maximization with nearly optimal adaptivity and query complexity. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 1833–1842. PMLR, 2019. URL: http://proceedings.mlr.press/v97/fahrbach19a.html.

22 Matthew Fahrbach, Vahab S. Mirrokni, and Morteza Zadimoghaddam. Submodular maximization with nearly optimal approximation, adaptivity and query complexity. In Timothy M. Chan, editor, *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 255–273. SIAM, 2019. doi:10.1137/1.9781611975482.17.

23 Stephen A. Fenner, Rohit Gurjar, and Thomas Thierauf. Bipartite perfect matching is in quasi-nc. In Daniel Wichs and Yishay Mansour, editors, *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 754–763. ACM, 2016. doi:10.1145/2897518.2897564.

24 Sumanta Ghosh, Rohit Gurjar, and Roshan Raj. A deterministic parallel reduction from weighted matroid intersection search to decision. In Joseph (Seffi) Naor and Niv Buchbinder, editors, *Proceedings of the 2022 ACM-SIAM Symposium on Discrete Algorithms, SODA 2022, Virtual Conference / Alexandria, VA, USA, January 9 - 12, 2022*, pages 1013–1035. SIAM, 2022. doi:10.1137/1.9781611977073.44.

25 Rohit Gurjar and Thomas Thierauf. Linear matroid intersection is in quasi-nc. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 821–830, 2017. doi:10.1145/3055399.3055440.

26 Richard M. Karp, Eli Upfal, and Avi Wigderson. The complexity of parallel computation on matroids. In *26th Annual Symposium on Foundations of Computer Science, Portland, Oregon, USA, 21-23 October 1985*, pages 541–550. IEEE Computer Society, 1985. doi:10.1109/SFCS.1985.57.

27 Richard M. Karp, Eli Upfal, and Avi Wigderson. Constructing a perfect matching is in random NC. *Comb.*, 6(1):35–48, 1986. doi:10.1007/BF02579407.

28 Richard M. Karp, Eli Upfal, and Avi Wigderson. The complexity of parallel search. *J. Comput. Syst. Sci.*, 36(2):225–253, 1988. doi:10.1016/0022-0000(88)90027-X.

29 Ehsan Kazemi, Marko Mitrovic, Morteza Zadimoghaddam, Silvio Lattanzi, and Amin Karbasi. Submodular streaming in all its glory: Tight approximation, minimum memory and low adaptive complexity. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 3311–3320. PMLR, 2019. URL: http://proceedings.mlr.press/v97/kazemi19a.html.

30 Sanjeev Khanna, Aaron Putterman, and Junkai Song. On the parallel complexity of finding a matroid basis. *arXiv preprint arXiv:2507.08194*, 2025. doi:10.48550/arXiv.2507.08194.

31 Sanjeev Khanna, Aaron Putterman, and Junkai Song. Optimal parallel basis finding in graphic and related matroids. *arXiv preprint arXiv:2511.04826*, 2025. doi:10.48550/arXiv.2511.04826.

32 Wenzheng Li, Paul Liu, and Jan Vondrák. A polynomial lower bound on adaptive complexity of submodular maximization. In Konstantin Makarychev, Yury Makarychev, Madhur Tulsiani, Gautam Kamath, and Julia Chuzhoy, editors, *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC 2020, Chicago, IL, USA, June 22-26, 2020*, pages 140–152. ACM, 2020. doi:10.1145/3357713.3384311.

33 László Lovász. On determinants, matchings, and random algorithms. In Lothar Budach, editor, *Fundamentals of Computation Theory, FCT 1979, Proceedings of the Conference on Algebraic, Arthmetic, and Categorial Methods in Computation Theory, Berlin/Wendisch-Rietz, Germany, September 17-21, 1979*, pages 565–574. Akademie-Verlag, Berlin, 1979.

34 Michael Luby. A simple parallel algorithm for the maximal independent set problem. *SIAM J. Comput.*, 15(4):1036–1053, 1986. doi:10.1137/0215074.

35 George L. Nemhauser and Laurence A. Wolsey. Best algorithms for approximating the maximum of a submodular set function. *Math. Oper. Res.*, 3(3):177–188, 1978. doi:10.1287/MOOR.3.3.177.

36 George L. Nemhauser, Laurence A. Wolsey, and Marshall L. Fisher. An analysis of approximations for maximizing submodular set functions - I. *Math. Program.*, 14(1):265–294, 1978. doi:10.1007/BF01588971.

37 James Oxley. *Matroid Theory*. Oxford University Press, February 2011. doi:10.1093/acprof:oso/9780198566946.001.0001.

38 Ola Svensson and Jakub Tarnawski. The matching problem in general graphs is in quasi-nc. In Chris Umans, editor, *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 696–707. IEEE Computer Society, 2017. doi:10.1109/FOCS.2017.70.

39 Jan Vondrák. Optimal approximation for the submodular welfare problem in the value oracle model. In Cynthia Dwork, editor, *Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada, May 17-20, 2008*, pages 67–74. ACM, 2008. doi:10.1145/1374376.1374389.

40 Jan Vondrák. Symmetry and approximability of submodular maximization problems. *SIAM J. Comput.*, 42(1):265–304, 2013. doi:10.1137/110832318.