Parallelizing Greedy for Submodular Set Function Maximization in Matroids and Beyond[*]

Chandra Chekuri[†]
chekuri@illinois.edu
University of Illinois, Urbana-Champaign
IL, USA

Kent Quanrud[‡]
quanrud2@illinois.edu
University of Illinois, Urbana-Champaign
IL, USA

ABSTRACT

We consider parallel, or low adaptivity, algorithms for submodular function maximization. This line of work was recently initiated by Balkanski and Singer and has already led to several interesting results on the cardinality constraint and explicit packing constraints. An important open problem is the classical setting of matroid constraint, which has been instrumental for developments in submodular function maximization. In this paper we develop a general strategy to parallelize the well-studied greedy algorithm and use it to obtain a randomized $(1/2 - \epsilon)$-approximation in $O\left(\log^2(n)/\epsilon^2\right)$ rounds of adaptivity. We rely on this algorithm, and an elegant amplification approach due to Badanidiyuru and Vondrák to obtain a fractional solution that yields a near-optimal randomized $(1 - 1/e - \epsilon)$-approximation in $O\left(\log^2(n)/\epsilon^3\right)$ rounds of adaptivity. For non-negative functions we obtain a $\left(3 - 2\sqrt{2} - \epsilon\right)$-approximation and a fractional solution that yields a $(1/e - \epsilon)$-approximation. Our approach for parallelizing greedy yields approximations for intersections of matroids and matchoids, and the approximation ratios are comparable to those known for sequential greedy.

CCS CONCEPTS

• **Theory of computation** → *Submodular optimization and polymatroids*; *Approximation algorithms analysis*; *Massively parallel algorithms*.

KEYWORDS

submodular maximization, parallel algorithms, matroids

ACM Reference Format:
Chandra Chekuri and Kent Quanrud. 2019. Parallelizing Greedy for Submodular Set Function Maximization in Matroids and Beyond. In *Proceedings of the 51st Annual ACM SIGACT Symposium on the Theory of Computing (STOC '19), June 23–26, 2019, Phoenix, AZ, USA.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3313276.3316406

1 INTRODUCTION

Matroids and submodular functions are two fundamental objects in combinatorial optimization that help generalize and unify many results. A matroid $\mathcal{M} = (\mathcal{N}, \mathcal{I})$ consists of a finite ground set $\mathcal{N}$ and a downclosed family of independent sets $\mathcal{I} \subseteq 2^{\mathcal{N}}$ that satisfy a simple exchange property. Whitney [55] defined matroids to abstract properties of dependence from linear algebra. Matroids are surprisingly common in combinatorial optimization and capture a wide variety of constraints. Submodular set functions are a class of real-valued set functions $f : 2^{\mathcal{N}} \to \mathbb{R}$ that discretely model decreasing marginal returns. (Formal definitions for both matroids and submodular functions are given in Appendix A.) More than purely mathematical generalizations, matroids and submodular functions capture the computational character of their more concrete instances. One can maximize a linear function over a matroid, the same way that one can compute a maximum or minimum weight spanning tree. For a submodular function f, one can obtain an optimal $\left(1 - e^{-1}\right)$-approximation to maximizing $f(S)$ over a cardinality constraint on S [52], the same way that one can obtain an optimal $\left(1 - e^{-1}\right)$-approximation to maximum coverage subject to a cardinality constraint (where optimality assumes P ≠ NP) [33]. Both of these connections are realized by a simple greedy algorithm.

The above connections are by now classical. A comparably recent result is an optimal $\left(1 - e^{-1}\right)$-approximation to maximizing $f(S)$ over a matroid constraint $S \in \mathcal{I}$ when f is monotone [17], a significant generalization of the cardinality constraint problem. Subsequent developments obtained a e^{-1}-approximation for nonnegative submodular functions subject to a matroid constraint [36] and then improved beyond e^{-1} [13, 26]. The techniques underlying these results take a fractional point of view with one part continuous optimization and another part rounding, somewhat analogous to the use of LP's for approximating integer programs. These and other techniques lead to a number of improved approximations for other set systems, such as combinations of matroids and packing constraints. There has been significant work on submodular function maximization in the recent past based also on classical combinatorial techniques such as greedy and local search and some novel variants such as the double greedy algorithm which led to an optimum $\frac{1}{2}$-approximation for unconstrained maximization of a nonnegative submodular function [16].

[*] This work is partially supported by NSF grant CCF-1526799. The full version of this extended abstract is available at https://arxiv.org/abs/1811.12568.
[†] http://chekuri.cs.illinois.edu
[‡] http://kentquanrud.com

STOC '19, June 23–26, 2019, Phoenix, AZ, USA

ACM ISBN 978-1-4503-6705-9/19/06…$15.00
https://doi.org/10.1145/3313276.3316406

With generality comes applicability. Many modern problems in machine learning and data mining can be cast as submodular function maximization subject to combinatorial or linear constraints. These applications leverage large amounts of informative data that form very large inputs in the corresponding optimization problem. Practical concerns of scalability have lead to theoretical consideration of various "big data" models of computation, such as faster approximations [2, 15, 27, 34, 46, 47], online and streaming models [1, 14, 18–20, 35, 39, 46, 48, 53], and algorithms conforming to map-reduce frameworks [8, 9, 43–45, 49].

To retain the full generality of submodular functions, submodular optimization algorithms are generally specified in a value oracle model, where one is given access to an oracle that returns the value $f(S)$ for any set S. Algorithms in this model are typically measured by the number of oracle calls made in addition to usual computational considerations. From a learning theoretic perspective, the oracle model raises a question of how much one can learn about a submodular function from a limited number of queries to the oracle. Balkanski et al. [4] showed that a polynomial number of queries of random samples (drawn from any distribution) cannot lead to a constant factor approximation for maximizing a submodular function subject to a cardinality constraint. The key point is that the queries are selected independently. All the standard polynomial time algorithms, of course, make only polynomially many queries, but these queries are chosen sequentially, one query informing the next.

Motivated by both practical and theoretical interests, Balkanski and Singer [7] investigated the minimum required **adaptivity** for submodular maximization. Adaptivity can be interpreted (to some extent) as a parallel model of computation with parallel access to oracles. For $k \in \mathbb{N}$, an algorithm with access to an oracle is ***k*-adaptive** if all of its oracle queries can be divided into a sequence of k rounds, where the choice of queries in one round can only depend on the values of queries in strictly preceding rounds. For example, the greedy algorithm for cardinality constraints has adaptivity equal to the specified cardinality. The negative results of [4] can be restated as saying more than 1 round of adaptivity is necessary for a constant approximation. Pushing this model further, Balkanski and Singer [7] showed that no algorithm can obtain better than a $O(1/\log n)$ approximation ratio with less than $O(\log(n)/\log\log n)$ adaptivity. On the positive side, Balkanski and Singer [7] gave a $\frac{1}{3}$-approximation for maximizing a monotone submodular function using $O(\log n)$ adaptive rounds. This was improved by $\left(1 - e^{-1} - \epsilon\right)$-approximation algorithm for maximizing a monotone submodular function subject to a cardinality with $O(\log(n)/\text{poly}(\epsilon))$ adaptivity [6, 28]. A number of follow up works have extended these results to knapsack and packing constraints [22, 30] and nonnegative submodular functions [3, 30, 31], or improved other aspects such as the total number of oracle calls [32]. One could argue that all of these papers essentially build upon the understanding and analysis for the cardinality constraint (even the ones which address significantly more general packing constraints [22, 30]).

One classical setting that is important to address is submodular set function maximization subject to an arbirary matroid constraint. Let $\mathcal{M} = (\mathcal{N}, \mathcal{I})$ be a matroid, and $f : 2^{\mathcal{N}} \to \mathbb{R}_{\geq 0}$ a nonnegative submodular function. The goal is to compute, in parallel, an independent set $I \in \mathcal{I}$ maximizing $f(I)$. We note that the natural and simple greedy algorithm gives a 1/2-approximation when f is monotone [38]. A strong theoretical motivation to study this problem is to understand the extent to which the classical greedy algorithm that gives good approximation for matroid constraints, and several generalizations, can be parallelized. Historically, the matroid constraint problem has been important for developing several new algorithmic ideas including the multilinear relaxation approach [17]. Before giving our results, it is important to establish the model, and in particular how we engage the matroid from a parallel perspective.

As with submodular functions, algorithms optimizing over matroids typically access the matroid structure via oracles. Standard oracles for a matroid $\mathcal{M} = (\mathcal{N}, \mathcal{I})$ are **independence oracles**, which take as input a set $S \subseteq \mathcal{N}$ and return whether or not $S \in \mathcal{I}$; **rank oracles**, which take as input a set $S \subseteq \mathcal{N}$ and return the maximum cardinality of any set in S; and **span oracles**, which take as input a set $S \subseteq \mathcal{N}$ and an element $e \in \mathcal{N}$ and returns whether or not $S + e$ has higher rank than S. Rank oracles are at least as strong as both independence oracles (since a set S is independent iff $\text{rank}(S) = |S|$) and span oracles (by comparing $\text{rank}(S)$ and $\text{rank}(S + e)$). In this work, we assume access to span oracles, and extend the notion of adaptivity to span oracles in the natural way.

Parallel rank oracles are known for most useful matroids. Parallel rank oracles for graphic matroids are given by parallel algorithms for computing spanning trees, such as Borůvka's algorithm. Rank oracles for linearly representable matroids (i.e., independent sets of vectors in some field) are also known [11, 25, 40, 50]. We note that many standard matroids (such as partition matroids and graphic matroids) can be viewed as linearly representable matroids. Parallel oracle models are well established in the literature. For example, the parallel oracle model was assumed by Karp et al. [42], who considered parallel algorithms w/r/t both independence and rank oracles for computing maximal and maximum independent sets, generalizing work on perfect matchings [41]. The oracle model was also considered by Narayanan et al. [51], who obtained parallel algorithms for matroid union and intersection in representable matroids, and asked if similar results can be obtained for general matroids assuming access to rank or independence oracles.

We are now prepared to state our results. We first obtain results for matroid constraints that are competitive with the best known sequential algorithms and are polylogarithmically adaptive. We obtain different approximation ratios depending on whether one desires a discrete independent set or a fractional point in the independent set polytope. The fractional point is evaluated w/r/t the "multilinear extension" $F : \mathbb{R}^{\mathcal{N}}_{\geq 0} \to \mathbb{R}$ of the set function f. The multilinear extension (defined in Appendix A.1) is a continuous extension of f first applied to submodular optimization in [17]. A point in the independent set polytope of a matroid can be rounded to a discrete set without loss, and nearly all competitive approximation algorithms for matroid constraints are obtained by approximating the multilinear extension and then rounding [2, 17, 23, 36] (with the notable exception being [37]). The rounding schemes, however, are not known to be parallelizable for general matroids. For now, the fractional solutions give a certificate that allow us to approximate the optimum value. In the following, let $\mathrm{OPT} = \max_{I \in \mathcal{I}} f(I)$ denote the maximum value of any independent set. The **rank** of a matroid is the maximum cardinality of any independent set. We first give results for maximizing a monotone submodular function subject to a matroid constraint.

Theorem 1.1. *Let $\mathcal{M} = (\mathcal{N}, \mathcal{I})$ be a matroid of rank k and $f : 2^{\mathcal{N}} \to \mathbb{R}_{\geq 0}$ a monotone submodular function.*

- *There is a randomized algorithm that outputs a set $I \in \mathcal{I}$ s.t. $\mathrm{E}[f(I)] \geq \left(\frac{1}{2} - \epsilon\right)\mathrm{OPT}$, with $O\left(\frac{\log(n)\log(k)}{\epsilon^2}\right)$ adaptive rounds in expectation.*
- *There is a randomized algorithm that computes a convex combination of $O(1/\epsilon)$ independent sets $x \in \mathrm{convex}(\mathcal{I})$ s.t. $F(x) \geq \left(1 - e^{-1} - \epsilon\right)\mathrm{OPT}$ with $O\left(\frac{\log(n)\log(k)}{\epsilon^3}\right)$ adaptive rounds in expectation, which implies that one can compute a $\left(1 - e^{-1} - \epsilon\right)$ approximation to* OPT.

Remark 1.2. We state the approximation and adaptivity bounds as expected quantities. We can achieve a high-probability bounds via standard tricks, however, in the interest of clarity we leave these details for a future version.

We also obtain approximations for generally nonnegative submodular functions with low adaptivity.

Theorem 1.3. *Let $\mathcal{M} = (\mathcal{N}, \mathcal{I})$ be a matroid of rank k and $f : 2^{\mathcal{N}} \to \mathbb{R}_{\geq 0}$ a nonnegative submodular function.*

- *There is a randomized algorithm that computes an independent set $I \in \mathcal{I}$ such that $\mathrm{E}[f(I)] \geq (1 - \epsilon)\left(3 - 2\sqrt{2}\right)\mathrm{OPT}$ and has adaptivity $O\left(\frac{\log(n)\log(k)}{\epsilon^2}\right)$ in expectation.*
- *There is a randomized algorithm that computes a convex combination of $O(1/\epsilon)$ independent sets $I_1, \ldots, I_\ell \in \mathcal{I}$ with $O\left(\frac{\log(n)\log(k)}{\epsilon^3}\right)$ adaptive rounds in expectation such that, if J_k samples each element in I_k independently with probability $1/\ell$ for each $k \in [\ell]$, we have $\mathrm{E}[f(J_1 \cup \cdots \cup J_\ell)] \geq \left(e^{-1} - \epsilon\right)\mathrm{OPT}$. This implies that one can compute a $\left(e^{-1} - \epsilon\right)$ approximation to* OPT.

Remark 1.4. Given a fractional point x in the matroid polytope it can be rounded to an independent set I such that $\mathrm{E}[f(I)] = F(x)$ when f is submodular. Pipage rounding [17] and swap rounding [23] achieve this. Swap rounding requires the decomposition of x into a convex combination of independent sets and consists of repeatedly (randomly) merging two independent sets via exchanges. The algorithms in the preceding theorems provide such a decomposition with only $O(1/\epsilon)$ independent sets. For some simple matroids such as partition matroids one can implement the random merging between two independent sets in parallel rather easily. This suggests an interesting open problem: for which matroids can two independent set be merged randomly in parallel?

The techniques extend beyond matroids to combinations of matroids, such as matroid intersections or (more generally) p-matchoids. These systems generalize bipartite matchings, arborescences, and non-bipartite matchings, and are defined formally in Appendix A.2.1. The rank of such a system is the maximum cardinality of any independent set. W/r/t the oracle model, these set systems are defined by some underlying collection of matroids, and we assume access to a rank oracle for each underlying matroid. We also note that, unlike in matroids, rounding a fractional solution in matroid intersections and matchoids incurs additional constant factor loss in the approximation, but we still state bounds for the multilinear relaxation as they are of independent interest and can be used in contention resolution schemes when combining with other constraints [24].

Theorem 1.5. *Let $\mathcal{M} = (\mathcal{N}, \mathcal{I})$ be a p-matchoid for some $p \in \mathbb{N}$ of rank k, and let $f : 2^{\mathcal{N}} \to \mathbb{R}_{\geq 0}$ be a monotone submodular function. There is a randomized algorithm that computes an independent set $I \in \mathcal{I}$ s.t. $\mathrm{E}[f(I)] \geq \left(\frac{1-\epsilon}{p+1}\right)\mathrm{OPT}$ with $O\left(\frac{\log(n)\log(k)}{\epsilon^2}\right)$ adaptive rounds in expectation. There is a randomized algorithm that computes a convex combination of $O(1/\epsilon)$ independent sets $x \in \mathrm{convex}(\mathcal{I})$ s.t. $\mathrm{E}[F(x)] \geq \left(1 - e^{-1/p} - \epsilon\right)\mathrm{OPT}$ with $O\left(\frac{\log(n)\log(k)}{\epsilon^3}\right)$ adaptive rounds in expectation.*

For nonnegative functions we have the following theorem.

Theorem 1.6. *Let $\mathcal{M} = (\mathcal{N}, \mathcal{I})$ be a p-matchoid for some $p \in \mathbb{N}$ of rank k, and let $f : 2^{\mathcal{N}} \to \mathbb{R}_{\geq 0}$ be a nonnegative submodular function. There is a randomized algorithm that computes $I \in \mathcal{I}$ such that $\mathrm{E}[f(I)] \geq (1 - \epsilon)\left(\frac{1 + o(1)}{4(p+1)}\right)\mathrm{OPT}$ with $O\left(\frac{\log(n)\log(k)}{\epsilon^2}\right)$ adaptive rounds in expectation. One can computed $\ell = O(1/\epsilon)$ randomized independent sets $I_1, \ldots, I_\ell \in \mathcal{I}$ with $O\left(\frac{\log(n)\log(k)}{\epsilon^3}\right)$ adaptive rounds in expectation such that, if J_i samples each element in I_i independently with probability $1/\ell$ for each $i \in [\ell]$, we have $\mathrm{E}[f(J_1 \cup \cdots \cup J_\ell)] \geq \frac{1-\epsilon}{p} e^{-1/p}\,\mathrm{OPT}$.*

1.1 Overview of Techniques

We give a brief overview of the techniques that lead to our results. The overall algorithm is relatively simple, and much shorter to describe than to analyze fully. Moreover, it is a composition of modular techniques, each of which may be of independent interest. For the sake of discussion, we focus on the setting of maximizing a monotone submodular function subject to a matroid constraint, noting that the techniques apply to nonnegative submodular functions as well.

Consider the simple greedy algorithm in the sequential setting, given in Figure 1. The greedy algorithm starts with an empty set $S = \emptyset$, and greedily adds to S the element e maximizing $f(S+e)$ subject to $S + e \in \mathcal{I}$. It terminates when there are no longer any elements that can be added to S. The greedy algorithm has an approximation ratio of $1/2$, short of the optimal $(1 - e^{-1})$-approximation scheme. This suboptimal ratio is not the primary concern because *a priori* it is not clear how to get any constant factor approximation for matroids in parallel. (Moreover, a $1/2$ approximation ratio can be amplified to

```
greedy($\mathcal{M} = (\mathcal{N}, \mathcal{I}), f : 2^{\mathcal{N}} \to \mathbb{R}_{\geq 0}$)
  1. $S \leftarrow \emptyset$
  2. while $S$ is not a base
    A. $e \leftarrow \arg\max_{d} \{f_S(d) : I + d \in \mathcal{I}\}$
    B. $S \leftarrow S + e$
  3. return $S$
```

Figure 1: A greedy 2-approximation for maximizing a monotone submodular function subject to a matroid constraint.

$1 - e^{-1}$; more on this later.) The greater problem in our setting is that it is inherently very sequential. The number of iterations, rank($\mathcal{M}$), can be as large as n. One hopes to "flatten" the iterations, but each chosen element e depends on the previously selected elements S at two critical points: (a) the marginal value $f_S(e)$ decreases as S increases due to submodularity, and (b) we cannot take e if $S + e$ is infeasible. Note that when one considers a cardinality constraint the second issue is significantly less important since every element in the ground set can be feasibly added as long as we have not reached a base. In fact all recent papers on adaptivity are focused mainly on the first issue.

It is clear that to obtain $\text{poly}(\log n, 1/\epsilon)$ depth, we cannot take just one element at a time, and would prefer to somehow take, say, $\text{rank}(\mathcal{M})/\text{poly}(\log n, 1/\epsilon)$ elements in each parallel round. In fact, we do not even take a set per iteration, but pairs of sets that we call "greedy blocks". For any set system $\mathcal{M} = (\mathcal{N}, \mathcal{I})$, a **$(1 - \epsilon)$-greedy block** consists of a random pair of sets (I, S) s.t.

(1) $I \subseteq S$ and $I \in \mathcal{I}$.
(2) $\mathrm{E}[f(I)] \geq (1 - \epsilon)\, \mathrm{E}[|S|] \max\left\{0, \max_e f(e)\right\}$.

We are interested in randomized greedy blocks not only w/r/t $\mathcal{M}$ and f, but the contracted matroid $\mathcal{M}/Q$ and marginal values f_Q for various sets $Q \subseteq \mathcal{N}$. When we do not assume f is monotone, f_Q can be negative, hence the need to take the max with 0.

Note that S is not required to be independent, but we still require an independent certificate $I \subseteq S$ that captures most of the value of S. At the end of the day, we will output a union of the independent I-components of the greedy blocks which will form a feasible solution. Allowing S to be dependent is an important degree of freedom that is used to make measurable progress and bound the depth.

We produce greedy blocks for matroids by a simple "greedy sampling" procedure. Let $\lambda \geq \max_{e \in \mathcal{N}} f(e)$ be some upper bound on the margin of any element. For a carefully chosen value $\delta > 0$ we let S sample each element e with nearly-maximum marginal value $f(e) \geq (1 - \epsilon)\lambda$ independently with probability δ. We then prune any sampled element that is either (a) spanned by the other sampled elements or (b) has marginal value $\leq (1 - \epsilon)\lambda$ w/r/t the other sampled elements. The pruning step leaves an independent subset $I \subseteq S$ where each element contributes at least $(1 - \epsilon)\lambda$ to I. We choose δ conservatively so that I retains most of the value of S, but also as large as possible within these constraints. This "greedy" choice of δ ensures that the residual problem (consisting of large margin elements not spanned by S) is smaller by about an ϵ-fraction in expectation. We can both search for the appropriate value of δ and sample with probability δ in parallel. The basic idea of greedy sampling is directly inspired by a much simpler greedy sampling procedure in the cardinality setting in our previous work [22].[1]

We now iterate along greedy blocks, where each iteration is w/r/t the residual system induced by previously selected greedy blocks. We start with empty sets $I, S = \emptyset$. We repeatedly generate $(1 - \epsilon)$-greedy blocks (I', S') w/r/t the contracted matroid $\mathcal{M}/S$ and the marginal values f_S, and then add I' to I and S' to S. The λ's are decreased by multiplicative factors of $1 - \epsilon$ to bound the depth: within a fixed λ, we expect a limited number of greedy samples until there are no elements left of marginal value $\geq (1 - \epsilon)\lambda$; λ can decrease by $1 - \epsilon$ multiplicative factor only a limited number of times before the marginal values of remaining elements are negligibly small. When λ is small enough, we return I, which is an independent set.

The above produces a randomized independent set I w/ $2\,\mathrm{E}[f(I)] \geq (1 - \epsilon)\,\text{OPT}$. In fact, it achieves a slightly stronger bound of the form $\mathrm{E}[f(I)] \geq (1 - \epsilon) f_S(T)$ for any independent set $T \in \mathcal{I}$, where S is a set containing I such that $\mathrm{E}[f(S)] \leq (1 + \epsilon)\,\mathrm{E}[f(I)]$. This is similar to (but slightly weaker than) the sequential greedy algorithm, which outputs an independent set I such that $f(I) \geq f_I(T)$ for any independent set $T \in \mathcal{I}$.

To improve the bound of 1/2 we rely on an elegant result of Badanidiyuru and Vondrák [2]. Motivated by the problem of finding a faster approximation algorithms for submodular function maximization, they showed that $O(1/\epsilon)$ iterations of the greedy algorithm via auxiliary functions induced by the multilinear relaxation produces a convex combination of $O(1/\epsilon)$ independent sets with approximation ratio $\left(1 - e^{-1} - \epsilon\right)$. We call this process "multilinear amplification", and show that it can be used, via our 1/2 approximation, to produce $\left(1 - e^{-1} - \epsilon\right)$-approximate fractional solution.

1.2 Further Discussion and Related Work

Submodular function maximization is a classical topic that was explored in several papers in the 70's with influential papers on the performance of the greedy algorithm, in particular by the work of Nemhauser et al. [52] and several subsequent ones. Recent years have seen a surge of activity on this topic. There have many important and interesting theoretical developments ranging from algorithms, lower bounds, connections to learning and game theory, and also a number of new applications in various domains including machine learning and data mining. Greedy and local-search methods have been revisited and improved and approximation algorithms for nonnegative nonmonotone functions were developed. An important development was the introduction of the multilinear relaxation approach which brought powerful continuous optimization methods into play and led to a new algorithmic approach. The literature is too large to discuss in this paper. We are primarily motivated by the work in approximation algorithms and the recent interest in finding parallel algorithms. We refer the reader to [7] for background and motivation studying the notion of adaptivity

[1] There one chooses δ as large as possible subject to preserving the gradient (along high margin elements) of the multilinear extension on average. The initial inspiration for the greedy sampling procedure for matroids was to apply the same logic to the rank function of the matroid, which is also a monotone submodular function.

which extends beyond interest in parallelization. Studying adaptivity allows one to avoid certain low-level details of the traditional PRAM model. However, we note that modulo poly-logarithmic factors and the reliance on an oracle for f and the rank function of the the underlying matroids, all of our algorithms can be implemented in the PRAM model without much effort. Parallel algorithms for Set Cover have been well-studied earlier (see [10] and references) and more recent work has explored Max Coverage and submodular function maximization in modern parallel systems such as the Map-Reduce model [8, 43, 44]. The study of parallelism for abstract problems that we consider here can provide high-level tools that could be specialized and refined for various concrete problems of interest.

Some of these results have been obtained independently. Balkanski et al. [5] obtained similar results for monotonic submodular functions, corresponding to Theorem 1.1 and Theorem 1.5. Ene et al. [29] (updating an earlier manuscript [30]) obtains similar results for optimizing the multilinear relaxation subject to a matroid constraint, both monotone and nonnegative, similar to Theorem 1.1 and Theorem 1.3.

Organization: We employ standard methodology and definitions that can be found in standard references in combinatorial optimization such as [54]. For the sake of completeness, preliminaries are given (at a leisurely pace) in Appendix A. Section 2 describes and analyzes the greedy sampling scheme which is the critical piece in our algorithms. Section 3 describes how greedy sampling can be used iteratively to derive approximation algorithms with low adaptivity. In Section 4 we describe the multilinear amplification idea from [2], extend it, and use it on top of the greedy algorithms to obtain improved bounds. Proofs from Section 4 are omitted due to space constraints. We also omit low-level details regarding sampling and estimation that were abstracted out of the algorithms. We refer the reader to the full version [21] for full details and proofs.

Notation: We use the following notation. For a set S and an element e, we let $S + e$ denote the union $S \cup \{e\}$ and let $S - e$ denote the difference $S \setminus \{e\}$. For a sequence of sets $S_1, \ldots, S_k$, we denote their union by $\overline{S}_k = \bigcup_{i=1}^{k} S_i$.

2 GREEDY SAMPLING

Recall the notion of greedy blocks, introduced in Section 1.1. In this section, we define and analyze a randomized procedure for generating greedy blocks, called `greedy-sample` and given in Figure 2. A brief sketch was given in Section 1.1, which we supplement now with a more thorough description.

We focus on the setting where all elements $e \in \mathcal{N}$ have margin $(1 - \epsilon)\lambda \leq f(e) \leq \lambda$ for some $\lambda > 0$. For a carefully chosen value $\delta > 0$, we sample each element in the ground set independently with probability δ to produce a set S. S may be dependent. We prune any element $e \in S$ that is spanned by $S - e$, leaving an independent subset I.

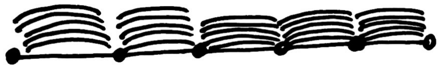
As δ increases, and S grows with it, submodularity pushes down the expected value of S per sampled element, and the expected span of S increases. In turn, the likelihood of pruning increases, and the ratio $f(I)/|S|$ decreases in expectation. Conversely, as we take δ down to 0, I converges to S in expectation, and for sufficiently small δ, (I, S) is a greedy block. We define the "sufficiently small" threshold to be such that in expectation, (a) at least $(1 - \epsilon)$-fraction of $\mathcal{N}$ has marginal value $\geq (1 - \epsilon)\lambda$ w/r/t S, and (b) at least $(1 - \epsilon)$-fraction of $\mathcal{N}$ is not spanned by S. Any δ satisfying both (a) and (b), it is shown, produces a greedy block (I, S).

```
greedy-sample($\mathcal{M} = (\mathcal{N}, \mathcal{I})$, $f$, $\lambda$, $\epsilon$)
    // Assume: $(1 - \epsilon)\lambda \leq f(e) \leq \lambda$ for all $e$
    // Goal: randomized pair of sets $(I, S)$, where $I$ is an independent
       subset of $S$, satisfying Theorem 2.1
    1. choose $\delta > 0$ large as possible s.t. for $S \sim \delta\mathcal{N}$,
       A. $\mathrm{E}[|\{e : f_S(e) \leq (1 - \epsilon)\lambda\}|] \leq \epsilon|\mathcal{N}|$
       B. $\mathrm{E}[|\mathrm{span}(S)|] \leq \epsilon|\mathcal{N}|$
    2. sample $S \sim \delta\mathcal{N}$
    3. $I \leftarrow \{e \in S : f_{S-e}(e) \geq (1 - \epsilon)\lambda, e \notin \mathrm{span}(S - e)\}$
    4. return $(I, S)$
```

Figure 2: A greedy sampling procedure for generating greedy blocks in independence systems.

We greedily maximize δ s.t. the above constraints for the sake of efficiency. Maximality ensures we reach the breaking point of one of the two limiting conditions. If condition (a) is tight, many of the elements have their marginal value drop below $(1 - \epsilon)\lambda$. If (b) is tight, then the sample S spans about an ϵ-fraction of the entire matroid. Either way, we expect to substantially decrease the number of un-spanned elements with marginal value $\geq (1 - \epsilon)\lambda$.

The first condition, (1.A), says we do not sample past the point where the margins are decreasing by a lot. A simpler form appears in previous work in the monotone cardinality setting [22]. The second condition (1.B), is a more significant departure from the cardinality setting, and strikes a balance between trying to span many elements, and not having to prune too many of the sampled elements. To build some intuition for condition (1.B), consider the graphic matroid on a "fat path graph", the multigraph consisting of k copies of each edge.



Then (1.B) (approximately) implies that $\delta \leq \epsilon/k$. For this value of δ, we expect a random sample to span an ϵ-fraction of the legs, hence an ϵ-fraction of the edges. We drop edges from legs that are sampled multiple times, but the probability of double sampling from any particular leg is $\leq O\left(\epsilon^2\right)$.

For a second example where the distribution is less uniform, consider the "fat tail graph", which is a multigraph where the first leg has k copies of that edge for k much larger than $1/\epsilon$.

We consider two regimes for k. On one hand, if k is comparable to n, then (1.B) is something like $\delta \leq \frac{\epsilon}{k} \approx \frac{\epsilon}{n}$. A uniform sample with probability δ will span the fat tail, hence a large fraction of all the edges, with probability about ϵ. The probability of double sampling from the fat tail remains small.

More illuminating is the case where k is much smaller than n. Then (1.B) roughly comes out to $\delta \leq \epsilon$. We expect the uniform sample to span an ϵ-fraction of the single-edge legs, and almost certainly prune from the fat tail. That is, the algorithm deliberately oversamples the fat tail. The edges lost from oversampling fit into "an ϵ of room": the expected number of edges pruned from the fat tail, ϵk, is much smaller than the expected number of edges sampled, ϵn, and the loss is essentially negligible.

Finding δ and other implementation issues: All of our algorithms critically rely on randoimization explicitly or implicitly. For instance it is not obvious how to find the δ in (1) of `greedy-sample`. We rely on random sampling and concentration bounds to estimate various quantities of interest. A formal analysis of the estimation errors and how they influence the approximation and adaptivity clutters the flow of the main ideas. Some of these implementation details are deferred to the full version [21], and we focus here on the key high-level steps.

Theorem 2.1. *Let $\mathcal{M} = (\mathcal{N}, \mathcal{I})$ be a matroid or p-matchoid, let $f : 2^{\mathcal{N}} \to \mathbb{R}$ be a submodular function, and let $\lambda \geq 0$ such that $(1-\epsilon)\lambda \leq f(e) \leq \lambda$ for all $e \in \mathcal{N}$. Then* `greedy-sample`*($\mathcal{M}$, f, λ, ϵ) returns random sets (I, S) such that*

(i) (I, S) is a $(1 - O(\epsilon))$-greedy block,
(ii) $\mathrm{E}[|\{e \notin \mathrm{span}(S) : f_S(e) \geq (1-\epsilon)\lambda\}|] \leq (1-\epsilon)|\mathcal{N}|$.

Proof.
(i) Clearly, I is an independent subset of S. Indeed, if I is not independent, then there is $e \in I$ s.t. $e \in \mathrm{span}(I - e)$. But $\mathrm{span}(I - e) \subseteq \mathrm{span}(S - e)$ and $e \notin \mathrm{span}(S - e)$ by choice of e. We relate $\mathrm{E}[f(I)]$ and $|S|$ via two intermediate sets. Let

$$\begin{aligned} Q &= \{e \in S : f_{S-e}(e) \geq (1-\epsilon)\lambda\} \\ \text{and } P &= \{e \in S : e \notin \mathrm{span}(S-e)\}. \end{aligned}$$

Then $I = P \cap Q$. We claim that
(a) $\mathrm{E}[f(Q)] \geq (1-\epsilon)\lambda\,\mathrm{E}[|S|]$.
(b) $\mathrm{E}[|S \setminus P|] \leq \epsilon\,\mathrm{E}[|S|]$.
Assuming claims (a) and (b) hold, we have

$$\begin{aligned} \mathrm{E}[f(I)] &\overset{(a)}{=} \mathrm{E}[f(Q)] - \mathrm{E}[f_I(Q)] \overset{(b)}{\geq} \mathrm{E}[f(Q)] - \sum_{e \in Q} \mathrm{E}[f_I(e)] \\ &\overset{(c)}{\geq} \mathrm{E}[f(Q)] - \sum_{e \in Q} \mathrm{P}[e \in Q \setminus I] f(e) \\ &\overset{(d)}{\geq} \mathrm{E}[f(Q)] - \lambda \sum_{e \in Q} \mathrm{P}[e \in Q \setminus I] \\ &= \mathrm{E}[f(Q)] - \lambda\,\mathrm{E}[|Q \setminus I|] \overset{(e)}{\geq} \mathrm{E}[f(Q)] - \lambda\,\mathrm{E}[|S \setminus P|] \\ &\overset{(f)}{\geq} (1-2\epsilon)\lambda\,\mathrm{E}[|S|] \end{aligned}$$

by (a) $I \subseteq Q$, (b,c) submodularity, (d) choice of λ, (e) $Q \setminus I \subseteq S \setminus P$, and (f) plugging in (a) and (b). It remains to prove claims (a) and (b).
(a) For each element e, we have $e \in Q$ iff $e \in S$ and $f_{S-e}(e) \geq (1-\epsilon)\lambda$. Moreover, the events $[e \in S]$ and $[f_{S-e}(e) \geq (1-\epsilon)\lambda]$ are independent because S is an independent sample. Thus

$$\begin{aligned} \mathrm{P}[e \in Q] &= \mathrm{P}[e \in S]\mathrm{P}[f_{S-e}(e) \geq (1-\epsilon)\lambda] \\ &= \delta\mathrm{P}[f_{S-e}(e) \geq (1-\epsilon)\lambda] \end{aligned} \tag{1}$$

We have

$$\begin{aligned} \mathrm{E}[f(Q)] &\overset{(g)}{\geq} \sum_{e \in \mathcal{N}} \mathrm{P}[e \in Q]\,\mathrm{E}\left[f_{Q-e}(e) \,\middle|\, e \in Q\right] \\ &\overset{(h)}{=} \delta \sum_{e \in \mathcal{N}} \mathrm{P}[f_{S-e}(e) \geq (1-\epsilon)\lambda]\,\mathrm{E}\left[f_{Q-e}(e) \,\middle|\, e \in Q\right] \\ &\overset{(i)}{\geq} \delta \sum_{e \in \mathcal{N}} \mathrm{P}[f_{S-e}(e) \geq (1-\epsilon)\lambda]\,\mathrm{E}[f_{S-e}(e) \mid e \in Q] \\ &\overset{(j)}{\geq} \delta(1-\epsilon)\lambda \sum_{e \in \mathcal{N}} \mathrm{P}[f_{S-e}(e) \geq (1-\epsilon)\lambda] \\ &\overset{(k)}{\geq} \delta(1-\epsilon)^2\lambda|\mathcal{N}| = (1-\epsilon)^2\lambda\,\mathrm{E}[|S|] \end{aligned}$$

by (g) submodularity, (h) substituting for $\mathrm{P}[e \in Q]$ by equation (1) above, (i) submodularity and $Q - e \subseteq S - e$, (j) definition of Q, and (k) choice of δ per (1.A).
(b) For each element e, $e \in S \setminus P$ iff $e \in S$ and $e \in \mathrm{span}(S - e)$. Moreover, the events $[e \in S]$ and $[e \in \mathrm{span}(S - e)]$ are independent because S is an independent sample. Thus

$$\mathrm{P}[e \in S \setminus P] = \delta\mathrm{P}[e \in \mathrm{span}(S - e)] \text{ for all } e \in \mathcal{N}. \tag{2}$$

We have

$$\begin{aligned} \mathrm{E}[|S \setminus P|] &\overset{(l)}{=} \sum_{e \in \mathcal{N}} \mathrm{P}[e \in S \setminus P] \\ &\overset{(m)}{=} \delta \sum_{e \in \mathcal{N}} \mathrm{P}[e \in \mathrm{span}(S - e)] \\ &\overset{(n)}{\leq} \delta \sum_{e \in \mathcal{N}} \mathrm{P}[e \in \mathrm{span}(S)] \\ &\overset{(o)}{=} \delta\,\mathrm{E}[|\mathrm{span}(S)|] \overset{(p)}{\leq} \delta\epsilon|\mathcal{N}| = \epsilon\,\mathrm{E}[|S|]. \end{aligned}$$

by (l) linearity of expectation, (m) equation (2) above, (n) monotonicity of span, (o) linearity of expectation, and (p) choice of δ.
(ii) By maximality of δ, either (1.A) or (1.B) is tight. If (1.A) is tight, then

$$\mathrm{E}[|\{e : f_S(e) \geq (1-\epsilon)\lambda\}|] \leq (1-\epsilon)|\mathcal{N}|.$$

If (1.B) is tight, then

$$\mathrm{E}[|\mathcal{N} \setminus \mathrm{span}(S)|] = |\mathcal{N}| - \mathrm{E}[|\mathrm{span}(S)|] \leq (1-\epsilon)|\mathcal{N}|.$$

Thus

$$\begin{aligned} &\mathrm{E}[|\{e \notin \mathrm{span}(S) : f_S(e) \geq (1-\epsilon)\lambda\}|] \\ &\leq \min\left\{ \begin{array}{c} \mathrm{E}[|\mathcal{N} \setminus \mathrm{span}(S)|], \\ \mathrm{E}[|\{e : f_S(e) \geq (1-\epsilon)\lambda\}|] \end{array} \right\} \leq (1-\epsilon)|\mathcal{N}|. \end{aligned}$$

■

Remark 2.2. Theorem 2.1 does not require f to be nonnegative but does require $\lambda \geq 0$. Theorem 2.1 holds for any independence system $\mathcal{M} = (\mathcal{N}, \mathcal{I})$ equipped with a function span $: 2^{\mathcal{N}} \to 2^{\mathcal{N}}$ such that:

(a) If $S \subseteq T \subseteq \mathcal{N}$, then $\mathrm{span}(S) \subseteq \mathrm{span}(T)$.

(b) If $S \subseteq \mathcal{N}$ and $e \notin \operatorname{span}(S - e)$ for all $e \in S$, then $S \in \mathcal{I}$.

For matroids, this is the standard span function. For matroid intersections, this is given by the union of the individual span functions.

Remark 2.3. Claim (b) in the proof implies that $\delta|\mathcal{N}| \leq O(\operatorname{rank}(\mathcal{M}))$. Indeed, $|S \setminus P| \geq |S| - \operatorname{rank}(\mathcal{M})$ deterministically, and (b) asserts that $\mathrm{E}[|S \setminus P|] \leq \epsilon\, \mathrm{E}[|S|]$. On the other hand, we know that $\delta \geq \epsilon/n$, since otherwise $S = \emptyset$ with probability at least $1 - \epsilon$ and so (1.A) and (1.B) are easily satisfied.

3 ITERATION BY GREEDY BLOCKS

We now iterate along greedy blocks, where each iteration is w/r/t the residual of previously selected greedy blocks. We start with empty sets $I, S = \emptyset$. We repeatedly generate $(1-\epsilon)$-greedy blocks (I', S') w/r/t the contracted matroid $\mathcal{M}/S$ and the marginal values f_S, and then add I' to I and S' to S. Recursively selecting greedy blocks like so leads to approximation factors resembling the standard greedy analysis (Theorem 3.3 below).

The λ's are generated by the standard technique of thresholding, which is employed for the sake of efficiency. We maintain a value λ such that $f_S(e) \leq \lambda$ for all $e \in \mathcal{N}$. We do not update λ until $f_S(e) \leq (1-\epsilon)\lambda$ for all $e \in \mathcal{N}$, at which point we replace λ with $(1-\epsilon)\lambda$. By part (ii) of Theorem 2.1, we only expect to call `greedy-sample` $O(\log(n)/\epsilon)$ times for a fixed value of λ. λ can only decrease $O(\log(\operatorname{rank}(\mathcal{M}))/\epsilon)$ times before the marginal values w/r/t S are small enough to be negligible. We expect to select $O\left(\frac{\log(n)\log(\operatorname{rank}(\mathcal{M}))}{\epsilon^2}\right)$ greedy blocks before the algorithm terminates.

First give a general analysis concerning finite sequences of greedy blocks.

Lemma 3.1. *Let $\mathcal{M} = (\mathcal{N}, \mathcal{I})$ be a matroid and let $f : 2^{\mathcal{N}} \to \mathbb{R}$ be a submodular function. Let $I_1, \dots, I_k \subseteq \mathcal{N}$ and $S_1, \dots S_k \subseteq \mathcal{N}$ be two sequences of random sets such that:*

(a) For each i, (I_i, S_i) is an $(1-\epsilon)$-greedy block w/r/t $f_{\overline{S}_{i-1}}$ and $\mathcal{M}/\overline{S}_{i-1}$.

(b) $f_{\overline{S}_k}\left(T \setminus \operatorname{span}\left(\overline{S}_k\right)\right) \leq \beta$ for all $T \in \mathcal{I}$.

Then

(i) $\mathrm{E}\left[f\left(\overline{I}_k\right)\right] \geq (1-\epsilon)\,\mathrm{E}\left[f\left(\overline{S}_k\right)\right]$.

(ii) For a fixed independent set $T \in \mathcal{I}$, there is a partition $T_1, \dots, T_{k+1}$ of T (depending on $S_1, \dots, S_k$) such that $T_i \cap \overline{S}_{i-1} = \emptyset$ for each i and

$$\mathrm{E}\left[f\left(\overline{I}_k\right)\right] \geq (1-\epsilon)\sum_{i=1}^{k+1} \mathrm{E}\left[f_{\overline{S}_{i-1}}(T_i)\right] - (1-\epsilon)\beta.$$

Proof.

(i) We have

$$\mathrm{E}\left[f(\overline{I}_k)\right] \overset{\text{(a)}}{=} \sum_{i=1}^{k} \mathrm{E}\left[f_{\overline{I}_{i-1}}(I_i)\right] \overset{\text{(b)}}{\geq} \sum_{i=1}^{k} \mathrm{E}\left[f_{\overline{S}_{i-1}}(I_i)\right]$$

$$\overset{\text{(c)}}{\geq} (1-\epsilon)\sum_{i=1}^{k} \mathrm{E}\left[|S_i| \max\left\{0, \max_{e \notin \operatorname{span}\left(\overline{S}_{u-1}\right)} f_{\overline{S}_{i-1}}(e)\right\}\right]$$

$$\overset{\text{(d)}}{\geq} (1-\epsilon)\sum_{i=1}^{k} \mathrm{E}\left[f_{\overline{S}_{i-1}}(S_i)\right] \overset{\text{(e)}}{=} (1-\epsilon)\,\mathrm{E}\left[f\left(\overline{S}_k\right)\right]$$

by (a) telescoping, (b) submodularity and $\overline{I}_{i-1} \subseteq \overline{S}_{i-1}$, and (c) assumption (a) that (I_i, S_i) is a greedy block, (d) submodularity, and (e) telescoping.

(ii) By Lemma A.2, we can partition $T \cap \operatorname{span}\left(\overline{S}_k\right) = T_1 \cup \dots \cup T_k$ such that for each i,

$$\begin{gathered} T_i \subseteq \mathcal{N} \setminus \operatorname{span}\left(\overline{S}_{i-1}\right) \text{ and} \\ |T_i| \leq \operatorname{rank}\left(\overline{S}_i\right) - \operatorname{rank}\left(\overline{S}_{i-1}\right) \leq |S_i|. \end{gathered} \tag{3}$$

Let $T_1, \dots, T_k$ be such a partition for each realization of $S_1, \dots, S_k$, and let $T_{k+1} = T \setminus \operatorname{span}\left(\overline{S}_k\right)$. The partition $T_1, \dots, T_{k+1}$ of T is randomized, and a deterministic function of $S_1, \dots, S_k$. We have

$$\mathrm{E}\left[f(\overline{I}_k)\right] \overset{\text{(f)}}{=} \sum_{i=1}^{k} \mathrm{E}\left[f_{\overline{I}_{i-1}}(I_i)\right] \overset{\text{(g)}}{\geq} \sum_{i=1}^{k} \mathrm{E}\left[f_{\overline{S}_{i-1}}(I_i)\right]$$

$$\overset{\text{(h)}}{\geq} (1-\epsilon)\sum_{i=1}^{k} \mathrm{E}\left[|S_i| \max\left\{0, \max_{e \notin \operatorname{span}\left(\overline{S}_{i-1}\right)} f_{\overline{S}_{i-1}}(e)\right\}\right]$$

$$\overset{\text{(i)}}{\geq} (1-\epsilon)\sum_{i=1}^{k} \mathrm{E}\left[|T_i| \max\left\{0, \max_{e \notin \operatorname{span}\left(\overline{S}_{i-1}\right)} f_{\overline{S}_{i-1}}(e)\right\}\right]$$

$$\overset{\text{(j)}}{\geq} (1-\epsilon)\sum_{i=1}^{k} \mathrm{E}\left[f_{\overline{S}_{i-1}}(T_i)\right] \overset{\text{(k)}}{\geq} (1-\epsilon)\sum_{i=1}^{k} \mathrm{E}\left[f_{\overline{S}_k}(T_i)\right]$$

$$\overset{\text{(l)}}{\geq} (1-\epsilon)\sum_{i=1}^{k+1} \mathrm{E}\left[f_{\overline{S}_{i-1}}(T_i)\right] - (1-\epsilon)\beta$$

by (f) telescoping, (g) submodularity and $\overline{I}_{i-1} \subseteq \overline{S}_{i-1}$, (h) (I_i, S_i) being a greedy block w/r/t $f_{\overline{S}_{i-1}}$ and $\mathcal{M}/\overline{S}_{i-1}$, and (i) $|T_i| \leq |S_i|$, (j) $T_i \subseteq \mathcal{N} \setminus \operatorname{span}\left(\overline{S}_{i-1}\right)$, (k) submodularity, and (l) adding the (negative) term $(1-\epsilon)\,\mathrm{E}\left[f_{\overline{S}_k}(T_{k+1})\right] - (1-\epsilon)\beta$.

■

```
block-greedy($\mathcal{M} = (\mathcal{N}, \mathcal{I}), f : 2^{\mathcal{N}} \to \mathbb{R}, \epsilon$)
  1. $I, S \leftarrow \emptyset$;  $\lambda \leftarrow \max_{e \in \mathcal{N}} f(e)$;  $\lambda_0 \leftarrow \frac{\epsilon\,\mathrm{OPT}}{\operatorname{rank}(\mathcal{M})}$
  // or any $\lambda_0 \leq \frac{\epsilon\,\mathrm{OPT}}{\operatorname{rank}(\mathcal{M})}$
  2. while $\lambda \geq \lambda_0$
    A. while $\mathcal{N}' = \{e \in \mathcal{N} : f_S(e) \geq (1-\epsilon)\lambda\}$ is not empty
      i. $(I', S') \leftarrow$ $(1-\epsilon)$-greedy blocks w/r/t $f_S$ and
         $(\mathcal{M}/S) \wedge \mathcal{N}'$
     ii. $I \leftarrow I \cup I'$,  $S \leftarrow S \cup S'$
    B. $\lambda \leftarrow (1-\epsilon)\lambda$
  3. return $(I, S)$
```

Figure 3: An extension of the greedy algorithm to greedy blocks with a polylogarithmic number of iterations in expectation.

Lemma 3.2. *Let $f : 2^{\mathcal{N}} \to \mathbb{R}$ be a normalized submodular function and let (I, S) be a greedy block where* $\mathrm{P}[S \neq \emptyset] > 0$. *Then (I, S) is a greedy block conditional on $S \neq \emptyset$.*

Proof. We have

$$\mathrm{E}[|S|] = \mathrm{P}[S \neq \emptyset]\,\mathrm{E}[|S| \mid S \neq \emptyset],$$

and since (a) $S = \emptyset \implies I = \emptyset$ and (b) f is normalized, we have

$$\begin{aligned} \mathrm{E}[f(I)] &\overset{(a)}{=} \mathrm{P}[S = \emptyset] f(\emptyset) + \mathrm{P}[S \neq \emptyset]\,\mathrm{E}[f(I) \mid S \neq \emptyset] \\ &\overset{(b)}{=} \mathrm{P}[S \neq \emptyset]\,\mathrm{E}[f(I) \mid S \neq \emptyset] \end{aligned}$$

Thus

$$\begin{aligned} \mathrm{E}[f(I) \mid S \neq \emptyset] &= \frac{\mathrm{E}[f(I)]}{\mathrm{P}[S \neq \emptyset]} \\ &= (1-\epsilon)\frac{\mathrm{E}[|S|]}{\mathrm{P}[S \neq \emptyset]} \max\left\{0, \max_e f(e)\right\} \\ &= (1-\epsilon)\,\mathrm{E}[|S| \mid S \neq \emptyset] \max\left\{0, \max_e f(e)\right\}, \end{aligned}$$

as desired. ■

Theorem 3.3. *Let $\mathcal{M} = (\mathcal{N}, \mathcal{I})$ be a matroid and let $f : 2^{\mathcal{N}} \to \mathbb{R}_{\geq 0}$ be a nonnegative submodular function. With $O\left(\frac{\log(n)\log(\mathrm{rank}(\mathcal{M}))}{\epsilon^2}\right)$ calls to* `greedy-sample` *in expectation, one can compute a randomized independent set $I \in \mathcal{I}$ and a randomized sequence of $n = |\mathcal{N}|$ sets $S_1, \ldots, S_n$ such that*

(1) $I \subseteq \overline{S}_n$.

(2) $\mathrm{E}[f(I)] \geq (1-\epsilon)\,\mathrm{E}\left[f\left(\overline{S}_n\right)\right]$.

(3) For any set $T \in \mathcal{I}$, there is a partition $T_1, \ldots, T_{n+1}$ of T (depending on $S_1, \ldots, S_n$) such that

$$\mathrm{E}[f(I)] \geq (1-\epsilon) \sum_{i=1}^{n+1} \mathrm{E}\left[f_{\overline{S}_{i-1}}(T_i)\right].$$

Proof. Consider `block-greedy`. Each iteration of (2.A.*) expects to decrease $\mathcal{N}'$ by a $(1-\epsilon)$-multiplicative factor, and $\mathcal{N}'$ is at most n. So we expect at most $O(\log(n)/\epsilon)$ iterations of (2.A.*) per λ. Moreover, we have $O(\log(\mathrm{rank}(\mathcal{M}))/\epsilon)$ choices of λ. Thus we expect at most $O\left(\log(n)\log(\mathrm{rank}(\mathcal{M}))/\epsilon^2\right)$ iterations total.

Of the greedy samples computed by `block-greedy`, let $(I_1, S_1), (I_2, S_2), \ldots$ be the subsequence of nonempty greedy blocks with $S_i \neq \emptyset$. Since the S_i's are disjoint, there are at most n pairs in the sequence. Appending empty greedy blocks if necessary, we have n random pairs $(I_1, S_1), \ldots, (I_n, S_n)$ such that:

(a) for each $i \in [n]$, by Lemma 3.2, (I_i, S_i) is an $(1-\epsilon)$-greedy block w/r/t $f_{\mathrm{span}\left(\overline{S}_{i-1}\right)}$ and $\mathcal{M}/\overline{S}_{i-1}$.

(b) deterministically we have $\mathrm{E}\left[f_{\overline{S}_n}(e)\right] \leq \frac{c\epsilon \max_e f(e)}{\mathrm{poly}(\mathrm{rank}(\mathcal{M}))}$ for any desired constant c.

The result now follows from applying Theorem 3.3 with $\beta \leq \epsilon\,\mathrm{E}\left[f\left(\overline{I}_n\right)\right]$. ■

To help interpret Theorem 3.3, we note that the sequential greedy algorithm gives the above guarantee deterministically, with $I = \overline{S}_n$ and $\epsilon = 0$. That is, greedy returns an independent set $I \in \mathcal{I}$ and a sequence of sets $S_1, \ldots, S_k \subseteq \mathcal{N}$ (for $k = \mathrm{rank}(\mathcal{M})$) such that:

(a) $I \subseteq \overline{S}_n$.

(b) For any set $T \in \mathcal{I}$, there is a partition $T_1, \ldots, T_{n+1}$ of T (depending $S_1, \ldots, S_{n+1}$) such that

$$f(I) \geq \sum_{i=1}^{n+1} f_{\overline{S}_{i-1}}(T_i).$$

3.1 Beyond Matroids

It is well-known that the greedy analysis extends past matroids, to intersections of matroids, and more generally, matchoids. In this section, we show that greedy blocks extend to p-matchoids as well, obtaining the appropriate approximation ratios inversely proportional to p. Proofs are omitted due to space constraints and we refer the reader to the full version [21]. In this section, let $\mathrm{rank}(\mathcal{M}) = \max\{|I| : I \in \mathcal{I}\}$ denote the maximum cardinality of any independent set.

Theorem 3.4. *Let $\mathcal{M} = (\mathcal{N}, \mathcal{I})$ be a p-matchoid, and let $f : 2^{\mathcal{N}} \to \mathbb{R}$ be a submodular function. With $O(\log(n)\log(\mathrm{rank}(\mathcal{M}))/\epsilon)$ calls to* `greedy-sample` *in expectation, one can compute a randomized independent set $I \in \mathcal{I}$ and a sequence of $n = |\mathcal{N}|$ sets $S_1, \ldots, S_n$ such that*

(i) $I \subseteq S$ *and* $\mathrm{E}[f(I)] \geq (1-\epsilon)\,\mathrm{E}[f(S)]$.

(ii) For any $T \in \mathcal{I}$, there is a partition $T_1, \ldots, T_{n+1}$ of T (depending on S) such that $T_i \cap \mathrm{span}\left(\overline{S}_{i-1}\right) = \emptyset$ for each i and

$$\mathrm{E}[f(I)] \geq \frac{1-\epsilon}{p} \sum_{i=1}^{n+1} \mathrm{E}\left[f_{\overline{S}_{i-1}}(T_i)\right].$$

To help interpret Theorem 3.4, we note that the sequential greedy algorithm can be interpreted as a sequence of exact (i.e., $\epsilon = 0$) and deterministic greedy blocks. This gives the above guarantee deterministically, with $I = \overline{S}_n$ and with $\epsilon = 0$. That is, greedy returns an independent set $I \in \mathcal{I}$ and a sequence of sets $S_1, \ldots, S_k \subseteq \mathcal{N}$ (for $k = |I|$) such that:

(a) $I \subseteq \overline{S}_n$.

(b) For any set $T \in \mathcal{I}$, there is a partition $T_1, \ldots, T_{n+1}$ of T (dependent on $S_1, \ldots, S_{n+1}$) such that

$$f(I) \geq \frac{1}{p} \sum_{i=1}^{n+1} f_{\overline{S}_{i-1}}(T_i).$$

Monotone approximation. Theorem 3.3 and Theorem 3.4 directly lead to approximations in the monotone case essentially matching the greedy algorithms.

Theorem 3.5. *Let $\mathcal{M} = (\mathcal{N}, \mathcal{I})$ be a p-matchoid, and let $f : 2^{\mathcal{N}} \to \mathbb{R}_{\geq 0}$ be a monotonic submodular function. With $O(\log(n)\log(\mathrm{rank}(\mathcal{M}))/\epsilon)$ calls to* `greedy-sample` *in expectation, one can compute a randomized independent set $I \in \mathcal{I}$ such that*

$$\mathrm{E}[f(I)] \geq \frac{1-\epsilon}{p+1} f(T)$$

for any $T \in \mathcal{I}$.

Nonnegative approximation. Let $\mathcal{M} = (\mathcal{N}, \mathcal{I})$ by a p-matchoid and $f : 2^{\mathcal{N}} \to \mathbb{R}_{\geq 0}$ be a nonnegetive submodular function. If f is not monotone, then the approximation guarantees (i) and (ii) of Theorem 3.3 do not immediately imply an actual approximation factor, as the sum on the RHS of (ii) may be much smaller than $f(T)$. Nonetheless one can prove the following.

Corollary 3.6. *Let $\mathcal{M} = (\mathcal{N}, \mathcal{I})$ be a p-matchoid and $f : 2^{\mathcal{N}} \to \mathbb{R}_{\geq 0}$ a nonnegative submodular function. For sufficiently small $\epsilon > 0$, with $O\left(\log(n)\log(\operatorname{rank}(\mathcal{M}))/\epsilon^2\right)$ adaptive rounds in expectation, one can compute an independent set J such that*

$$\begin{aligned} \mathrm{E}[f(I)] &\geq (1-\epsilon)\left(2p + 1 - 2\sqrt{p(p+1)}\right)\mathrm{OPT} \\ &= (1-\epsilon)\left(\frac{1+o(1)}{4(p+1)}\right)\mathrm{OPT}. \end{aligned}$$

4 MULTILINEAR AMPLIFICATION

By combining greedy matroid sampling of Section 2 with the iterative analysis of greedy blocks in Section 3, we can produce independent sets that resemble those of the greedy algorithm in performance. For monotone functions, this implies a near-2 approximation, short of the desired near-$(1 - 1/e)$ approximation factor achieved originally by the continuous greedy algorithm in [17]. Badanidiyuru and Vondrák [2] showed how to amplify a near-2 approximation into a near-$(1 - 1/e)$ approximate fractional solution, as a convex combination of $O(1/\epsilon)$ near-2 approximations w/r/t submodular functions induced by the multilinear relaxation. We call this process "multilinear amplification".

We want to highlight that multilinear amplification applies to any set system, and for any approximation guarantee for the oracle, where the amplified approximation factor varies with the approximation ratio of the oracle.

The primary caveat with multilinear amplification is that the fractional solution still needs to be rounded. Many constraints can be rounded with some bounded loss, but in many cases the loss offsets the gains from the amplification.[2] An important exception is matroids, which can be rounded without loss. For other constraints, the amplification step may still prove valuable if the rounding schemes improve in the future.

It is not clear if general matroids can be rounded with low depth. Some explicit matroids, such as partition matroids, can be handled fairly easily and directly. Rounding algorithms for general matroids, such as swap rounding, appeal to properties (such as Brualdi's strong exchange) that are inherently sequential.

The amplification process still gives an improved approximation ratio to the *value* of the optimization problem, since there is no integrality gap for matroids. Moreover, rounding to a matroid in parallel may be less important than solving the optimization problem in parallel, because the $O(1/\epsilon)$ independent sets produced by the amplification may be much smaller than the original input, and existing rounding schemes are oblivious and do not require f. We think that parallelizable rounding schemes for matroids is an interesting open question.

```
monotone-ML-amp($\mathcal{M} = (\mathcal{N}, \mathcal{I})$, $f : 2^{\mathcal{N}} \to \mathbb{R}_{\geq 0}$, $\epsilon > 0$)
  1. $x \leftarrow \mathbb{0}$
  2. repeat $\ell = O(1/\epsilon)$ times
    A. define $g(S) = F(x + S/\ell) - F(x)$
    B. invoke oracle to get $S \in \mathcal{I}$ s.t.
             $\mathrm{E}[g(S)] \geq (1-\epsilon)\alpha\,\mathrm{E}[g_S(T)]$ for all $T \in \mathcal{I}$
    C. $x \leftarrow x + \frac{S}{\ell}$
  3. return $x$
```

Figure 4: A procedure based on [2] amplifying an α-approximation algorithm for monotone submodular maximization to a fractional nearly $\left(1 - e^{-\alpha/1-\alpha}\right)$-approximation algorithm via the multilinear extension.

```
nonnegative-ML-amp($\mathcal{M} = (\mathcal{N}, \mathcal{I})$, $f : 2^{\mathcal{N}} \to \mathbb{R}_{\geq 0}$, $\epsilon$, $\alpha$)
  1. $x \leftarrow \mathbb{0}$
  2. for $i = 1$ up to $\ell = O(1/\epsilon)$
    A. define $g(S) = \mathrm{E}\left[f_{\overline{J}_{i-1}}(S')\right]$ where $S' \sim S/\ell$ and
       $J_j \sim \alpha I_j/\ell$ for each $j$
    B. invoke oracle to get $I_i \in \mathcal{I}$ satisfying
       conditions (a) and (b) of Theorem 4.2
  3. return $(I_1, \ldots, I_\ell)$
```

Figure 5: A procedure amplifying greedy-type approximation algorithms for nonegative submodular functions via the multilinear extension.

4.1 Monotone Multilinear Amplification

The monotone case is essentially given by Badanidiyuru and Vondrák [2] and sketched in Figure 4.

Theorem 4.1. *Let $\mathcal{M} = (\mathcal{N}, \mathcal{I})$ be a set system, and let $f : 2^{\mathcal{N}} \to \mathbb{R}_{\geq 0}$ be a monotone submodular function. Suppose one has access to an oracle that, given a submodular function g, returns a (possibly randomized) set $S \in \mathcal{I}$ such that*

$$\mathrm{E}[g(S)] \geq \alpha\,\mathrm{E}[g_S(T)] \text{ for all } T \in \mathcal{I},$$

for some $\alpha \in (0, 1]$. With $O(1/\epsilon)$ calls to this oracle, one can compute a convex combination of $O(1/\epsilon)$ independent sets $x \in \operatorname{convex}(\mathcal{I})$ such that

$$F(x) \geq (1-\epsilon)\left(1 - e^{-\alpha}\right)f(T) \text{ for all } T \in \mathcal{I},$$

where F is the multilinear extension of f.

4.2 Nonnegative Multilinear Amplification

In this section, we consider another multilinear amplification scheme that attains weaker bounds for the more general class of nonnegative submodular functions. The nonnegative multilinear amplification scheme is to measured greedy what the monotone multilinear amplification scheme is to continuous greedy, as discussed further in the full version [21].

Theorem 4.2. *Let $\mathcal{M} = (\mathcal{N}, \mathcal{I})$ be a set system, and let $f : 2^{\mathcal{N}} \to \mathbb{R}_{\geq 0}$ be a nonnegative submodular function. Suppose one has access to an oracle that, given a submodular function g, computes a (possibly*

[2] Perhaps this is why [2] does not explicitly explore multilinear amplification in its full generality.

randomized) set $I \subseteq \mathcal{N}$ and for which there exists a sequence of disjoint sets $S_1, S_2, \ldots, S_k \subseteq \mathcal{N}$ such that

(a) $\mathrm{E}[g(I)] \geq (1-\epsilon)\,\mathrm{E}\left[g(\overline{S}_k)\right]$

(b) For any $T \in \mathcal{I}$, there exists a partition $T = T_1 \cup \cdots \cup T_{k+1}$ (depending on $S_1, \ldots, S_k$) such that $T_j \cap \overline{S}_{j-1} = \emptyset$ for each j and $\mathrm{E}[g(I)] \geq \alpha \sum_j \mathrm{E}\left[g_{\overline{S}_{j-1}}(T_j)\right]$ for some $\alpha \in [0,1]$.

With $\ell = O(1/\epsilon)$ calls to this oracle, one can compute ℓ independent sets $I_1, \ldots, I_\ell \in \mathcal{I}$ such that if $J_i \sim \alpha I_i/\ell$ independently for each $i \in [\ell]$, then

$$\mathrm{E}[f(J_1 \cup \cdots \cup J_\ell)] \geq (1-\epsilon)\alpha e^{-\alpha} f(T)$$

for all $T \in \mathcal{I}$.

Remark 4.3. One motivation for Badanidiyuru and Vondrák [2] was to improve the oracle complexity of maximizing a monotone submodular function subject to a matroid constraint in the sequential setting. The nonnegative amplification scheme appears to offer some improvement in the oracle complexity of maximizing a generally nonnegative submodular function subject to a matroid constraint in the sequential setting. It is cleaner to analyze the sequential setting so we plan to address this in a separate writeup.

A PRELIMINARIES

This paper is primarily concerned with two abstract objects, submodular functions and matroids, which we define formally in Appendix A.1 and Appendix A.2 respectively.

A.1 Submodular Functions

Let $f : 2^{\mathcal{N}} \to \mathbb{R}$ be a real-valued set function. f is:

1. **normalized** if $f(\emptyset) = 0$.
2. **nonnegative** if $f(S) \geq 0$ for all $S \subseteq \mathcal{N}$.
3. **monotone** $f(S) \leq f(T)$ for all $S \subseteq T \subseteq \mathcal{N}$.
4. **submodular** if $f(S) + f(T) \geq f(S \cup T) + f(S \cap T)$ for all $S \subseteq T \subseteq \mathcal{N}$.

All set valued functions in this paper are normalized and submodular, and submodular functions will always be assumed to be normalized. The given function that we are optimizing over is nonnegative, but other set-valued functions arise that are not necessarily nonnegative.

Submodularity can be understood intuitively in terms of "decreasing marginal returns". To this end, we denote

$$f_S(U) \overset{\text{def}}{=} f(S \cup U) - f(S) \text{ for } S, U \subseteq \mathcal{N}.$$

$f_S(U)$ is represents the increase in value gained by adding U to S, and more succinctly called the **marginal value of U to S**. Submodularity (as defined above) is equivalent to saying that the marginal value of U is decreasing in S in the following sense:

$$f_S(U) \geq f_T(U) \text{ for all } S, T, U \subseteq \mathcal{N} \text{ with } S \subseteq T.$$

For any submodular function f and set S, the marginal values $f_S : \mathcal{N} \to \mathbb{R}$ form a normalized submodular function. If f is monotone, then f_S is monotone and nonnegative. However, f being nonnegative does *not* imply that f_S is nonnegative.

We appeal to a continuous extension of f to $\mathbb{R}^n_{\geq 0}$ called the "multilinear extension", for which we introduce the following notation. For a vector $x \in \mathbb{R}^n_{\geq 0}$, we write $S \sim x$ to denote the random set $S \subseteq \mathcal{N}$ that samples each $e \in \mathcal{N}$ independently with probability $\min\{x_e, 1\}$. That is, we interpret x (after truncation) as the margins of an independent sample. The **multilinear extension of f**, denoted $F : \mathbb{R}^{\mathcal{N}}_{\geq 0} \to \mathbb{R}$, is the expected value of a random set drawn according to x:

$$F(x) = \mathrm{E}[f(S) \mid S \sim x].$$

The name "multilinear extension" can be explained as follows. We identify each set S with its incidence vector in $\{0,1\}^{\mathcal{N}}$. Abusing notation, we let S also denote its incident vector (when the meaning is clear). Then we have $F(S) = f(S)$ for every set S, and F is an extension of f as a function of $\{0,1\}^{\mathcal{N}}$. The "multilinear" comes from the fact that $F(x)$ is multilinear in x when $x \in [0,1]^{\mathcal{N}}$.

As an expectation of f, F inherits many of the properties as f. f is normalized, nonnegative, and monotone iff F is normalized, nonnegative and monotone, respectively. Submodularity translates to a particular kind of concavity, as follows. We say that F is **monotone concave** if for any $x, v \in \mathbb{R}^{\mathcal{N}}_{\geq 0}$ and $\delta > 0$, the map

$$\delta \mapsto F(x + \delta v) \text{ is concave in } \delta.$$

Then f is submodular iff F is monotone concave. For example, if f is submodular and monotonic, then

$$F(\epsilon x) \geq \epsilon F(x) \text{ for all } x \in \mathbb{R}^{\mathcal{N}}_{\geq 0} \text{ and } \epsilon \in (0,1).$$

A.2 Combinatorial Constraints

A set system $\mathcal{M} = (\mathcal{N}, \mathcal{I})$ consists of a **ground set** $\mathcal{N}$ and a family of subsets $\mathcal{I} \subseteq 2^{\mathcal{N}}$. A set system $\mathcal{M} = (\mathcal{N}, \mathcal{I})$ is an **independence system** if $\mathcal{I}$ is nonempty and closed under subsets:

(a) $\emptyset \in \mathcal{I}$

(b) For $S \subseteq T \subseteq \mathcal{N}$, $T \in \mathcal{I}$ implies $S \in \mathcal{I}$.

A set $S \in \mathcal{I}$ is called an **independent set**. An independence system is a **matroid** if it also satisfies the following augmentation property.

(c) If $S, T \in \mathcal{I}$ and $|S| < |T|$, then there is an element $e \in T \setminus S$ such that $S + e \in \mathcal{I}$.

A base is a maximal independent set. By property (c), every base in a matroid has the same cardinality, called the **rank**. More generally, for any set $S \subseteq \mathcal{N}$ in a matroid $\mathcal{M} = (\mathcal{N}, \mathcal{I})$, every maximal independent set in S has the same cardinality, called the **rank of S** and denoted $\mathrm{rank}(S)$. The rank is a nonnegative, monotone submodular function. The **span** of an independent set $I \in \mathcal{I}$ is the set of elements $e \in \mathcal{N}$ such that either $e \in I$ or $I + e \notin \mathcal{I}$, where $I + e$ is a shorthand for $I \cup \{e\}$. In general, the span of a set S is the set of elements that do not increase the rank:

$$\mathrm{span}(S) = \{e \in \mathcal{N} : \mathrm{rank}(S + e) = \mathrm{rank}(S)\}.$$

Given a matroid $\mathcal{M} = (\mathcal{N}, \mathcal{I})$, there are two different ways to modify $\mathcal{M}$ of interest. Given a set $S \subseteq \mathcal{N}$, the **restriction of $\mathcal{M}$ to S**, denoted $\mathcal{M} \wedge S = (S, \mathcal{I} \wedge S)$, has ground set S and independent sets consisting of the independent subsets of S, $\mathcal{I} \wedge S = \{I \subseteq S : I \in \mathcal{I}\}$. The rank of $\mathcal{M} \wedge S$ is precisely the rank of S.

The second modification is contraction. Give a set $S \subseteq \mathcal{N}$, the **contraction of $\mathcal{M}$ to S**, denoted $\mathcal{M}/S = (\mathcal{N}/S, \mathcal{I}/S)$, has ground set $\mathcal{N}/S = \mathcal{N} \setminus \mathrm{span}(S)$, lemmand independence is defined by the rank function $\mathrm{rank}_S(T) = \mathrm{rank}(S + T) - \mathrm{rank}(S)$. Alternatively, one can choose any base B of S, and define $\mathcal{I}/S$ by $\mathcal{I}_S = \{T \subseteq \mathcal{N}/S : B \cup T \in \mathcal{I}\}$.

A good working example of a matroid is the **graphic matroid**. Here the ground set corresponds to the edges of some graph $G = (V, E)$, and a set of edges is independent if they form a forest. The bases of this matroid are the spanning forests in the graph. Restricting a graphic matroid to a set of edges is the graphic matroid over the subgraph induced by these edges. Contracting a set of edges corresponds to the graphic matroid over the minor obtained by contracting each of these edges.

A particular useful property of matroids, first observed by Brualdi [12], is the following.

Lemma A.1. *Let I, J be two independent sets with $|I| \geq |J|$. Then there exists an injection $\pi : J \setminus I \to I \setminus J$ such that for all $e \in J$,*

$$I - \pi(e) + e \in \mathcal{I}.$$

Brualdi's exchange mapping easily implies the following, which is in a slightly more convenient form for us.

Lemma A.2. *Let $S_1, S_2, \cdots, S_k$ be a sequence of sets in a matroid $\mathcal{M} = (\mathcal{N}, \mathcal{I})$ such that $S_i \subseteq \mathcal{N} \setminus \operatorname{span}\left(\overline{S}_{i-1}\right)$ for each i. For any independent set $I \in \mathcal{I}$, one can partition $T \cap \operatorname{span}\left(\overline{S}_k\right)$ into sets $\{T_1, \ldots, T_k\}$ such that for each i, $T_i \subseteq \mathcal{N} \setminus \operatorname{span}\left(\overline{S}_i\right)$ and $|T_i| \leq |S_i|$.*

A.2.1 Combinations of Matroids. We optimize over matroids and combinations of matroids, such as intersections of matroids and more generally matchoids. A **matroid intersection** is an independence system $\mathcal{M} = (\mathcal{N}, \mathcal{I})$ where $\mathcal{I} = \bigcap_i \mathcal{I}_i$ for a collections matroids $\mathcal{M}_i = (\mathcal{N}, \mathcal{I}_i)$ with the same set. Matroid intersections generalize bipartite matchings and arboresences. A **matchoid** $\mathcal{M} = (\mathcal{N}, \mathcal{I})$ is an independence system defined by a collection of matroids $\mathcal{M}_i = (\mathcal{N}_i, \mathcal{I}_i)$, where $\mathcal{N}_i \subseteq \mathcal{N}$ for each i, with independent sets

$$\mathcal{I} = \{S : S \cap \mathcal{N}_i \in \mathcal{I}_i \text{ for all } i\}.$$

For $k \in \mathbb{N}$, a k-matroid intersection is an intersection of k matroids, and a k-matchoid is a matchoid where each element $e \in \mathcal{N}$ participates in at most k of the underlying matroids. A k-matroid is of course a k-matchoid. A k-matchoid can be recast as a matroid intersection by extending each underlying $\mathcal{M}_i = (\mathcal{N}_i, \mathcal{I}_i)$ to all of $\mathcal{N}$ by allowing any extra element $e \in \mathcal{N} \setminus \mathcal{N}_i$ to only be spanned by itself. The number of matroids in the matroid intersection may be much larger than k, which matters only because the approximation ratios depend on k.

Matroid intersections and matchoids still carry some of the structure and notions of matroids. Suppose a matchoid or matroid intersection is defined by the matroids $\mathcal{M}_1, \ldots, \mathcal{M}_k$. One can define a function span : $2^{\mathcal{N}} \to 2^{\mathcal{N}}$ by

$$\operatorname{span}(S) = \bigcup_{i=1}^{k} \operatorname{span}_i(S)$$

where for each i, span_i is the span function associated with the ith matroid. This span function still has the following properties of span functions for matroids:

(1) $\operatorname{span}(S) \subseteq \operatorname{span}(T)$ for $S \subseteq T$.
(2) If $S \subseteq \mathcal{N}$ and $e \notin \operatorname{span}(S - e)$ for all $e \in S$, then $S \in \mathcal{I}$.

There notions of restricting and contract a matroid intersection or matchoid are still well-defined, by taking the restriction or contraction in each of the underlying matroids, and recombining the restricted or contracted matroids into a matroid intersection or matchoid.

Canonical examples of 2-matroid intersection are bipartite matchings and arborescences. An example of a 2-matchoid is a matching, and an example of a k-matchoid is a matching in a hypergraph of rank k. In a matching (bipartite or general), the span of an edge set S is the set of all edges incident to some edge in S. Contracting an edge corresponds to removing both endpoints and all incident edges.

REFERENCES

[1] Ashwinkumar Badanidiyuru, Baharan Mirzasoleiman, Amin Karbasi, and Andreas Krause. 2014. Streaming submodular maximization: massive data summarization on the fly. In *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014*. 671–680.

[2] Ashwinkumar Badanidiyuru and Jan Vondrák. 2014. Fast algorithms for maximizing submodular functions. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5-7, 2014*. 1497–1514.

[3] Eric Balkanski, Adam Breuer, and Yaron Singer. 2018. Non-monotone Submodular Maximization in Exponentially Fewer Iterations. *CoRR* abs/1807.11462 (2018). http://arxiv.org/abs/1807.11462 To appear in NIPS, 2018.

[4] Eric Balkanski, Aviad Rubinstein, and Yaron Singer. 2017. The limitations of optimization from samples. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*. 1016–1027.

[5] Eric Balkanski, Aviad Rubinstein, and Yaron Singer. 2018. An Optimal Approximation for Submodular Maximization under a Matroid Constraint in the Adaptive Complexity Model. *CoRR* abs/1811.03093 (Nov. 2018). https://arxiv.org/abs/1811.03093

[6] Eric Balkanski, Aviad Rubinstein, and Yaron Singer. 2019. An Exponential Speedup in Parallel Running Time for Submodular Maximization without Loss in Approximation. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*. 283–302.

[7] Eric Balkanski and Yaron Singer. 2018. The adaptive complexity of maximizing a submodular function. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*. 1138–1151.

[8] Rafael da Ponte Barbosa, Alina Ene, Huy L. Nguyen, and Justin Ward. 2015. The Power of Randomization: Distributed Submodular Maximization on Massive Datasets. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*. 1236–1244.

[9] Rafael da Ponte Barbosa, Alina Ene, Huy L. Nguyen, and Justin Ward. 2016. A New Framework for Distributed Submodular Maximization. In *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, 9-11 October 2016, Hyatt Regency, New Brunswick, New Jersey, USA*. 645–654.

[10] Guy E. Blelloch, Richard Peng, and Kanat Tangwongsan. 2011. Linear-work greedy parallel approximate set cover and variants. In *SPAA 2011: Proceedings of the 23rd Annual ACM Symposium on Parallelism in Algorithms and Architectures, San Jose, CA, USA, June 4-6, 2011 (Co-located with FCRC 2011)*, Rajmohan Rajaraman and Friedhelm Meyer auf der Heide (Eds.). ACM, 23–32.

[11] Allan Borodin, Joachim von zur Gathen, and John E. Hopcroft. 1982. Fast Parallel Matrix and GCD Computations. *Information and Control* 52, 3 (1982), 241–256. Preliminary version in FOCS, 1982.

[12] Richard A. Brualdi. 1969. Comments on bases in dependence structures. *Bulletin of the Australian Mathematical Society* 1, 2 (1969), 161–167.

[13] Niv Buchbinder and Moran Feldman. 2016. Constrained Submodular Maximization via a Non-symmetric Technique. *CoRR* abs/1611.03253 (2016). arXiv:1611.03253 http://arxiv.org/abs/1611.03253

[14] Niv Buchbinder, Moran Feldman, and Roy Schwartz. 2015. Online Submodular Maximization with Preemption. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015, San Diego, CA, USA, January 4-6, 2015*. 1202–1216.

[15] Niv Buchbinder, Moran Feldman, and Roy Schwartz. 2017. Comparing Apples and Oranges: Query Trade-off in Submodular Maximization. *Math. Oper. Res.* 42, 2 (2017), 308–329. Preliminary version in SODA, 2015.

[16] Niv Buchbinder, Moran Feldman, Joseph Seffi, and Roy Schwartz. 2015. A tight linear time (1/2)-approximation for unconstrained submodular maximization.

SIAM J. Comput. 44, 5 (2015), 1384–1402.

[17] Gruia Călinescu, Chandra Chekuri, Martin Pál, and Jan Vondrák. 2011. Maximizing a Monotone Submodular Function Subject to a Matroid Constraint. *SIAM J. Comput.* 40, 6 (2011), 1740–1766.

[18] Amit Chakrabarti and Sagar Kale. 2015. Submodular maximization meets streaming: matchings, matroids, and more. *Math. Program.* 154, 1-2 (2015), 225–247. Preliminary version in IPCO, 2014.

[19] T.-H. Hubert Chan, Zhiyi Huang, Shaofeng H.-C. Jiang, Ning Kang, and Zhihao Gavin Tang. 2018. Online Submodular Maximization with Free Disposal. *ACM Trans. Algorithms* 14, 4 (2018), 56:1–56:29. Preliminary version in SODA, 2017.

[20] Chandra Chekuri, Shalmoli Gupta, and Kent Quanrud. 2015. Streaming Algorithms for Submodular Function Maximization. In *Automata, Languages, and Programming - 42nd International Colloquium, ICALP 2015, Kyoto, Japan, July 6-10, 2015, Proceedings, Part I.* 318–330.

[21] Chandra Chekuri and Kent Quanrud. 2018. Parallelizing greedy for submodular set function maximization in matroids and beyond. *CoRR* abs/1811.12568 (2018). arXiv:1811.12568 http://arxiv.org/abs/1811.12568

[22] Chandra Chekuri and Kent Quanrud. 2019. Submodular Function Maximization in Parallel via the Multilinear Relaxation. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019.* 303–322.

[23] Chandra Chekuri, Jan Vondrák, and Rico Zenklusen. 2010. Dependent Randomized Rounding via Exchange Properties of Combinatorial Structures. In *51th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2010, October 23-26, 2010, Las Vegas, Nevada, USA.* 575–584.

[24] Chandra Chekuri, Jan Vondrák, and Rico Zenklusen. 2014. Submodular function maximization via the multilinear relaxation and contention resolution schemes. *SIAM J. Comput.* 43, 6 (2014), 1831–1879.

[25] Alexander L. Chistov. 1985. Fast parallel calculation of the rank of matrices over a field of arbitrary characteristic. In *Fundamentals of Computation Theory, FCT '85, Cottbus, GDR, September 9-13, 1985.* 63–69.

[26] Alina Ene and Huy L. Nguyen. 2016. Constrained Submodular Maximization: Beyond 1/e. In *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, 9-11 October 2016, Hyatt Regency, New Brunswick, New Jersey, USA.* 248–257.

[27] Alina Ene and Huy L. Nguyen. 2017. A Nearly-linear Time Algorithm for Submodular Maximization with a Knapsack Constraint. *CoRR* abs/1709.09767 (2017). http://arxiv.org/abs/1709.09767

[28] Alina Ene and Huy L. Nguyen. 2019. Submodular Maximization with Nearly-optimal Approximation and Adaptivity in Nearly-linear Time. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019.* 274–282.

[29] Alina Ene, Huy L. Nguyen, and Adrian Vladu. 2018. Submodular Maximization with Matroid and Packing Constraints in Parallel. *CoRR* abs/1808.09987v2 (Nov. 2018). http://arxiv.org/abs/1808.09987v2

[30] Alina Ene, Huy L. Nguyen, and Adrian Vladu. 2018. Submodular Maximization with Packing Constraints in Parallel. *CoRR* abs/1808.09987v1 (Aug. 2018). http://arxiv.org/abs/1808.09987v1

[31] Matthew Fahrbach, Vahab S. Mirrokni, and Morteza Zadimoghaddam. 2018. Non-monotone Submodular Maximization with Nearly Optimal Adaptivity Complexity. *CoRR* abs/1808.06932 (2018). arXiv:1808.06932 http://arxiv.org/abs/1808.06932

[32] Matthew Fahrbach, Vahab S. Mirrokni, and Morteza Zadimoghaddam. 2018. Submodular Maximization with Optimal Approximation, Adaptivity and Query Complexity. *CoRR* abs/1807.07889 (2018). arXiv:1807.07889 http://arxiv.org/abs/1807.07889

[33] U. Feige. 1998. A Threshold of ln n for Approximating Set Cover. 45, 4 (July 1998), 634–652. Preliminary version in STOC, 1996.

[34] Moran Feldman, Christopher Harshaw, and Amin Karbasi. 2017. Greed Is Good: Near-Optimal Submodular Maximization via Greedy Optimization. In *Proceedings of the 30th Conference on Learning Theory, COLT 2017, Amsterdam, The Netherlands, 7-10 July 2017.* 758–784.

[35] Moran Feldman, Amin Karbasi, and Ehsan Kazemi. 2018. Do Less, Get More: Streaming Submodular Maximization with Subsampling. *CoRR* abs/1802.07098 (2018). arXiv:1802.07098 http://arxiv.org/abs/1802.07098

[36] Moran Feldman, Joseph Naor, and Roy Schwartz. 2011. A Unified Continuous Greedy Algorithm for Submodular Maximization. In *IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS 2011, Palm Springs, CA, USA, October 22-25, 2011.* 570–579.

[37] Yuval Filmus and Justin Ward. 2014. Monotone Submodular Maximization over a Matroid via Non-Oblivious Local Search. *SIAM J. Comput.* 43, 2 (2014), 514–542. Preliminary version in FOCS, 2012.

[38] Marshall L Fisher, George L Nemhauser, and Laurence A Wolsey. 1978. An analysis of approximations for maximizing submodular set functions—II. In *Polyhedral combinatorics.* Springer, 73–87.

[39] Chien-Chung Huang, Naonori Kakimura, and Yuichi Yoshida. 2017. Streaming Algorithms for Maximizing Monotone Submodular Functions under a Knapsack Constraint. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2017, August 16-18, 2017, Berkeley, CA, USA.* 11:1–11:14.

[40] Oscar H. Ibarra, Shlomo Moran, and Louis E. Rosier. 1980. A Note on the Parallel Complexity of Computing the Rank of Order n Matrices. *Inf. Process. Lett.* 11, 4/5 (1980), 162.

[41] Richard M. Karp, Eli Upfal, and Avi Wigderson. 1986. Constructing a perfect matching is in random NC. *Combinatorica* 6, 1 (1986), 35–48.

[42] Richard M. Karp, Eli Upfal, and Avi Wigderson. 1988. The Complexity of Parallel Search. *J. Comput. Syst. Sci.* 36, 2 (1988), 225–253. Some results appeared in preliminary work in STOC, 1985 and FOCS, 1985.

[43] Ravi Kumar, Benjamin Moseley, Sergei Vassilvitskii, and Andrea Vattani. 2015. Fast Greedy Algorithms in MapReduce and Streaming. *TOPC* 2, 3 (2015), 14:1–14:22. Preliminary version in SPAA, 2013.

[44] Paul Liu and Jan Vondrák. 2018. Submodular optimization in the MapReduce model. *CoRR* abs/1810.01489 (2018). http://arxiv.org/abs/1810.01489 To appear in SOSA 2019.

[45] Vahab S. Mirrokni and Morteza Zadimoghaddam. 2015. Randomized Composable Core-sets for Distributed Submodular Maximization. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015.* 153–162.

[46] Baharan Mirzasoleiman, Ashwinkumar Badanidiyuru, and Amin Karbasi. 2016. Fast Constrained Submodular Maximization: Personalized Data Summarization. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016.* 1358–1367.

[47] Baharan Mirzasoleiman, Ashwinkumar Badanidiyuru, Amin Karbasi, Jan Vondrák, and Andreas Krause. 2015. Lazier Than Lazy Greedy. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA.* 1812–1818.

[48] Baharan Mirzasoleiman, Stefanie Jegelka, and Andreas Krause. 2018. Streaming Non-Monotone Submodular Maximization: Personalized Video Summarization on the Fly. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018.* 1379–1386.

[49] Baharan Mirzasoleiman, Amin Karbasi, Rik Sarkar, and Andreas Krause. 2016. Distributed Submodular Maximization. *Journal of Machine Learning Research* 17 (2016), 238:1–238:44. Preliminary version in NIPS, 2013.

[50] Ketan Mulmuley. 1987. A fast parallel algorithm to compute the rank of a matrix over an arbitrary field. *Combinatorica* 7, 1 (1987), 101–104. Preliminary version in STOC, 1986.

[51] H. Narayanan, Huzur Saran, and Vijay V. Vazirani. 1994. Randomized Parallel Algorithms for Matroid Union and Intersection, With Applications to Arboresences and Edge-Disjoint Spanning Trees. *SIAM J. Comput.* 23, 2 (1994), 387–397. Preliminary version in SODA, 1992.

[52] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. 1978. An analysis of approximations for maximizing submodular set functions – I. *Math. Prog.* 14, 1 (1978), 265–294.

[53] Ashkan Norouzi-Fard, Jakub Tarnawski, Slobodan Mitrovic, Amir Zandieh, Aidasadat Mousavifar, and Ola Svensson. 2018. Beyond 1/2-Approximation for Submodular Maximization on Massive Data Streams. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018.* 3826–3835.

[54] Alexander Schrijver. 2003. *Combinatorial optimization: polyhedra and efficiency.* Vol. 24. Springer Science & Business Media.

[55] Hassler Whitney. 1935. On the abstract properties of linear dependence. *American Journal of Mathematics* 57, 3 (1935), 509–533.