Linearly Representable Submodular Functions: An Algebraic Algorithm for Minimization

Rohit Gurjar
Indian Institute of Technology Bombay, India
http://www.cse.iitb.ac.in/~rgurjar
rohitgurjar0@gmail.com

Rajat Rathi
Indian Institute of Technology Bombay, India
rajatrathidgr81@gmail.com

Abstract

A set function $f\colon 2^E \to \mathbb{R}$ on the subsets of a set E is called submodular if it satisfies a natural diminishing returns property: for any $S \subseteq E$ and $x, y \notin S$, we have $f(S \cup \{x, y\}) - f(S \cup \{y\}) \leq f(S \cup \{x\}) - f(S)$. Submodular minimization problem asks for finding the minimum value a given submodular function takes. We give an algebraic algorithm for this problem for a special class of submodular functions that are "linearly representable". It is known that every submodular function f can be decomposed into a sum of two monotone submodular functions, i.e., there exist two non-decreasing submodular functions f_1, f_2 such that $f(S) = f_1(S) + f_2(E \setminus S)$ for each $S \subseteq E$. Our class consists of those submodular functions f, for which each of f_1 and f_2 is a sum of k rank functions on families of subspaces of $\mathbb{F}^n$, for some field $\mathbb{F}$.

Our algebraic algorithm for this class of functions can be parallelized, and thus, puts the problem of finding the minimizing set in the complexity class randomized NC. Further, we derandomize our algorithm so that it needs only $O(\log^2(kn|E|))$ many random bits.

We also give reductions from two combinatorial optimization problems to linearly representable submodular minimization, and thus, get such parallel algorithms for these problems. These problems are (i) covering a directed graph by k a-arborescences and (ii) packing k branchings with given root sets in a directed graph.

2012 ACM Subject Classification Mathematics of computing → Combinatorial optimization; Theory of computation → Parallel algorithms; Theory of computation → Pseudorandomness and derandomization

Keywords and phrases Submodular Minimization, Parallel Algorithms, Derandomization, Algebraic Algorithms

Digital Object Identifier 10.4230/LIPIcs.ICALP.2020.61

Category Track A: Algorithms, Complexity and Games

Acknowledgements We would like to thank Siddharth Barman for helpful comments.

1 Introduction

Submodular functions have been studied in a wide variety of contexts like combinatorics, electrical networks, game theory, and machine learning. For a set E, a submodular function is a set function $f\colon 2^E \to \mathbb{R}$ that satisfies a natural diminishing returns property: for any $T \subseteq S \subseteq E$ and $x \in E \setminus S$, we have

$$f(S \cup \{x\}) - f(S) \leq f(T \cup \{x\}) - f(T).$$

That is, the marginal value of an element with respect to a set decreases as the set grows. Another equivalent way to describe submodularity is: for any $S, T \subseteq E$, we have $f(S)+f(T) \geq f(S \cup T) + f(S \cap T)$. Submodular functions appear in a diverse set of areas. To give a few

47th International Colloquium on Automata, Languages, and Programming (ICALP 2020).
Editors: Artur Czumaj, Anuj Dawar, and Emanuela Merelli; Article No. 61; pp. 61:1–61:15
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

examples, a linear function, maximum number in the given subset, the rank function on a set of vectors or subspaces, the cut function on the set of vertices of a graph, entropy on a set of random variables, the coverage function on a collection of subsets, are all submodular.

There are various natural optimization problems involving submodular functions: *Submodular Minimization* asks for the set that minimizes a given submodular function among all subsets of the ground set. Similarly, *Submodular Maximization* asks for the maximizing set. Note that the these two questions are interesting only for non-monotone submodular functions like graph cut. There are also constrained versions of minimization and maximization, for example, optimizing a submodular function over subsets of a given size. The submodular function might be given by an explicit representation, for example, a given graph can represent the corresponding cut function. However, not all submodular functions are succinctly representable, as their number grows as doubly-exponential in the ground set size [27]. In the most general framework, the function is given via a value oracle, i.e., given any subset $S \subseteq E$, the oracle will provide the function value on S.

Submodular minimization is in some sense a discrete version of convex minimization [20], and thus, admits polynomial time algorithms (even with just the value oracle). The initial algorithms for it were based on the ellipsoid algorithm [11, 12], but later on combinatorial algorithms were also obtained [6, 15, 28]. Submodular maximization, on the other hand, is known to be hard: Max-cut [16] and maximum facility location [5] are instances of submodular maximization which are NP-hard. Moreover, in the oracle model, there is an exponential lower bound known on the number of queries required [9].

When we put cardinality constraints, then in fact, both minimization and maximization problems become hard even for monotone submodular functions. Examples of such maximization problems that are NP-hard include max-k-cover (set-cover, which is NP-hard [16], reduces to it) and sparse approximation [7] (for a set E of vectors and fixed vector v, the function $f_v(S) = \|proj_{span(S)}(v)\|_2$ for $S \subseteq E$ is submodular). Similarly, min-k-vertex-cover is an example of an NP-hard minimization question (see [13]). Moreover, in the oracle model, cardinality constrained submodular minimization has a sub-exponential lower bound on the number of queries (follows from [31]).

Parallel complexity of submodular minimization. In this paper, we investigate the question of parallel complexity of unconstrained submodular minimization. In the oracle model, the parallel complexity question can be phrased as follows: if one is allowed to simultaneously make polynomially many function value queries in one round, how many rounds are required to find the minimum value (and the minimizing set). The number of rounds required is also known the *adaptivity* (see [1]). To the best of our knowledge, the best upper bound on the adaptivity of submodular minimization is $O(n \log(nM))$ [19], where M is the largest absolute value the function takes (they use a separation oracle that can be implemented with one round of n parallel queries to the value oracle). While on the lower bound side, there is a known impossibility result for one round [2], and $\tilde{\Omega}(n^2/k^5)$ query lower bound for k rounds [1]. Very recently, it was shown that there are no adaptive algorithms that run in $o(\frac{\log n}{\log \log n})$ rounds with poly(n) queries per round [3]. In particular, it is not clear whether the adaptivity of submodular minimization can be sublinear.

On the other hand, if we consider explicitly given submodular functions, there are instances for which the minimization problem admits parallel algorithms. Such special cases of submodular minimization include (s,t)-min-cut (small capacities), maximum bipartite matching[1], and its generalization linear matroid intersection. These problems have algebraic algorithms, which just involve randomized reductions to matrix rank computation and thus, fall into the class randomized NC (RNC) [17, 21, 23, 24]. In recent years, these algorithms have also been partially derandomized [10, 14], i.e., they can work with only $O(\log^2 n)$ random bits. A natural question arises: what is the most general class of submodular functions for which such algebraic algorithms can work. One would expect such algebraic algorithms for submodular functions that are linear algebraic in some sense. Towards this, we define a class of *linearly representable* submodular functions.

Linearly representable (LR) submodular functions. Suppose we have a family of subspaces $\mathcal{V} = \{V_e \subseteq \mathbb{F}^n\}_{e\in E}$, for some field F. Recall that the rank function of the family $\mathcal{V}$ given by $r(S) = \dim(\sum_{e\in S} V_e)$ for $S \subseteq E$ is submodular. The rank function is non-decreasing, and hence, the minimization question for it is not interesting. One can try to consider the difference of two rank functions, but that is not submodular. Interestingly, there is a way to construct a non-increasing submodular function from a non-decreasing submodular function: if a function $f(S)$ is submodular then so is $g(S) := f(E \setminus S)$. So, if we have two non-decreasing submodular functions $f_1(S)$ and $f_2(S)$, we can get a non-monotone submodular function by considering $f_1(S) + f_2(E \setminus S)$ (since the sum of two submodular functions is also submodular). In fact, using this way one can arrive at any submodular function. It is known (see [6]) that every submodular function f can be decomposed into two non-decreasing submodular functions f_1 and f_2 such that for any $S \subseteq E$, $f(S) = f_1(S) + f_2(E \setminus S)$.

Our contributions

The above facts motivate us to define the following natural class of linear algebraic submodular functions that are not necessarily monotone. For a ground set E, let $\bar{S} := E \setminus S$.

▶ **Definition 1** (Linearly representable (LR) submodular functions). *We call a submodular function $f: 2^E \to \mathbb{Z}$ linearly representable (LR) by k families of subspaces $\mathcal{V}_j = \{V_{j,e} \subseteq \mathbb{F}^n\}_{e\in E}$ for $1 \le j \le k$ and a number $\ell \le k$ if*

$$f(S) = \sum_{j=1}^{\ell} r_j(S) + \sum_{j=\ell+1}^{k} r_j(\bar{S}),$$

where $r_j: 2^E \to \mathbb{Z}$ is the rank function for family $\mathcal{V}_j$.

This class includes many interesting submodular functions like directed graph cut, hypergraph cut, coverage function, integral linear function (up to additive normalization), and more interestingly, any combination of them in the above form. Our main results are a randomized algebraic algorithm for minimizing LR submodular functions that puts the minimization problem in RNC, and an almost complete derandomization of the algorithm; see Section 3.

▶ **Theorem 2** (Linearly representable submodular minimization). *Given an LR submodular function $f: 2^E \to \mathbb{Z}$ via families of subspaces $\mathcal{V}_j = \{V_{j,e} \subseteq \mathbb{F}^n\}_{e\in E}$ for $1 \le j \le k$ and a number $\ell \le k$ (Definition 1), we can find a set minimizing $f(S)$ in RNC. Further, the randomized algorithm can be almost completely derandomized so that it uses only $O(\log^2(kn|E|))$ random bits.*

[1] For a bipartite graph $G(V_1 \cup V_2, E)$, the maximum matching size is equal to $|V_1| + \min_{S\subseteq V_1}(|N(S)| - |S|)$ (Hall's theorem), where $N(S) \subseteq V_2$ is the set of neighbor of S. The function $|N(S)| - |S|$ is submodular.

Another way to put the derandomization result is that minimizing an LR submodular function is in quasi-NC (see [10, 14] for the details of class quasi-NC). Our results also imply a randomized parallel algorithm and its almost deterministic version for a problem called linear polymatroid intersection, generalizing the corresponding results for linear matroid intersection [24, 14].

In the *linear polymatroid intersection* problem, we are given two families of subspaces, $\mathcal{V}_j = \{V_{j,e} \subseteq \mathbb{F}^n\}_{e\in E}$ for $j = 1, 2$, with their rank functions r_1, r_2, respectively. And the goal is to find

$$\max\{\sum_{e\in E} x_e \mid x_e \geq 0 \ \forall e \in E, \text{ and } \sum_{e\in S} x_e \leq r_1(S), \sum_{e\in S} x_e \leq r_2(S) \text{ for each } S \subseteq E\}.$$

Min-max relation. It is known that this maximum value is equal to $\min_{S\subseteq E} r_1(S) + r_2(\bar{S})$ (see [29, Corollary 46.1c]). Thus, the maximization problem is captured by LR submodular minimization.

▶ **Corollary 3.** *Linear polymatroid intersection has a randomized NC algorithm that uses only $O(\log^2(n|E|))$ random bits.*

Linear matroid intersection is the special case of linear polymatroid intersection when each of the above subspaces $V_{j,e}$ is of dimension 1. Thus, the above min-max relation with a LR submodular function also holds for linear matroid intersection. Our parallel algorithm has a crucial difference from the known parallel algorithms [24, 14] for linear matroid intersection. They give the minimum value of the corresponding LR submodular function, but they do not lead to a minimizing set, while our algorithm also finds a minimizing set.

Further applications

As mentioned above, LR submodular minimization captures linear matroid intersection and thus, several other combinatorial optimization problems that reduce to linear matroid intersection, like bipartite matching, packing spanning trees, finding arborescences (see [29]), packing a-arborescences [29, Theorem 53.10], and hypergraph min-cut [18]. Since linear matroid intersection already has parallel algorithms [24, 14], so do these problems.

However, there are also combinatorial problems that reduce to submodular minimization but are not captured by linear matroid intersection. We show that two such problems, in fact, reduce to LR submodular minimization (see Section 4.2 for definitions and reductions).

- **Covering by a-arborescences.** For a given directed graph and a number k, decide whether the edge set is covered by k a-arborescences.
- **Packing of branchings.** For a given directed graph and given subsets $R_1, R_2, \ldots, R_k$ of vertices, decide whether there exist k edge-disjoint branchings that are rooted at $R_1, R_2, \ldots, R_k$, respectively.

To the best of our knowledge, there is no straightforward reduction known from these problems to linear matroid intersection. Using Theorem 2, we get the following.

▶ **Theorem 4.** *Covering by a-arborescences and packing of branchings can be solved in RNC using only $O(\log^2 n)$ random bits, n being the size of the input graph.*

Variants

Furthermore, we list out two problems, one of which is an extension of LR submodular minimization and the other one is equivalent to it (see Section 4.1).

1. **Minimization with containment constraints**: There is a variant of submodular minimization that appears frequently in combinatorial optimization. Given two subsets $S_0 \subseteq S_1 \subseteq E$, the goal is to minimize the given submodular function $f(S)$ subject to $S_0 \subseteq S \subseteq S_1$. We can extend our algorithm to LR submodular minimization with this kind of constraints. This is a generalization of the minimum $(S_0, \bar{S}_1)$ cut problem in a graph $G(V, E)$ with two given disjoint subsets $S_0, \bar{S}_1 \subseteq V$.
2. **Separation oracle for a linear polymatroid**: Given a family $\mathcal{V} = \{V_e \subseteq \mathbb{F}^n\}_{e \in E}$ of subspaces with its rank function $r\colon 2^E \to \mathbb{Z}$, the corresponding polymatroid is a polytope $P_r \subseteq \mathbb{R}^E$ defined as

$$P_r = \{x \in \mathbb{R}^E \mid x \geq 0, \sum_{e \in S} x_e \leq r(S) \; \forall S \subseteq E\}.$$

   Given a rational point $\beta \in \mathbb{R}^E$, one needs to decide if β lies in P_r, and if not then find a violating constraint from the above set. We reduce this problem to LR submodular minimization assuming the coordinates in β are rational numbers with a polynomially bounded common denominator.

2 Preliminaries

Complexity Class NC. NC represents the class of problems that can be solved by polynomially many parallel processors in poly-logartihmic time. RNC, i.e., randomized NC, represents problems that can be solved with the same resources, but with the use of randomness.

2.1 Submodular functions

For a set E, a *submodular* function is a set function $f\colon 2^E \to \mathbb{R}$ for any $S, T \subseteq E$, we have $f(S) + f(T) \geq f(S \cup T) + f(S \cap T)$. If f satisfies this with equality then it is called *modular*. There are various properties of submodular functions that are useful for us and are easy to verify. For $S \subseteq E$, $\bar{S}$ will denote $E \setminus S$.

- If f_1 and f_2 are submodular then so is $f_1 + f_2$.
- If f is submodular then so is $g(S) = f(\bar{S})$.

For a set E and a family of subspaces $\mathcal{V} = \{V_e \subseteq \mathbb{F}^n\}_{e \in E}$, its corresponding rank function $r\colon 2^E \to \mathbb{Z}$ is defined to be as $r(S) = \dim(\sum_{e \in S} V_e)$ for $S \subseteq E$. It is not hard to verify that the rank function is submodular for any family of subspaces.

2.2 Polynomial identity testing

To design our randomized algorithm, we will need a fundamental result about zeros of polynomials, which says that if a polynomial is nonzero then at a random point, its evaluation is nonzero with high probability (see, for example, [25, 30, 33, 8]).

▶ **Lemma 5.** *Let there be an n-variate degree-d nonzero polynomial $P(z_1, z_2, \dots, z_n)$. If for each $1 \leq i \leq n$, the variable x_i is substituted with a random number R_i chosen uniformly and independently from a set of size D then*

$$\Pr\{P(R_1, R_2, \dots, R_n) = 0\} \leq d/D.$$

Note that D should be at least as large as the degree. Throughout the paper, we will assume that the underlying field is large enough. This lemma gives a simple algorithm to test if a given polynomial is nonzero: just evaluate it at a random point and output nonzero if and only if the evaluation is nonzero. To derandomize our submodular minimization algorithm, we will need to derandomize this test of nonzeroness of a polynomial. For general polynomials, there is no non-trivial derandomization known. However, we will need the derandomization result only for polynomials that have certain special structure.

Let U be a square matrix whose entries are all linear polynomials (degree-1). The polynomials of interest in our setting will be determinants of such symbolic matrices, where any particular variable appears in at most one column of the matrix. For this class of polynomials, an almost complete derandomization of nonzero testing is known. A way to do deterministic nonzero testing is to obtain a small hitting-set – a set of points such that any nonzero polynomial in the class of our interest gives a nonzero evaluation on at least one of the points.

The work of [14] gave a quasi-polynomial size hitting-set for polynomials of the following form which subsume the above case: $P(\mathbf{z}) = \det(\sum_{i=1}^{n} A_i z_i)$, where each A_i is a rank-1 matrix. The result of [14] can be easily modified to generate a slightly stronger notion of a hitting-set, though it is not explicitly stated there.

▶ **Lemma 6** ([14]). *There is an NC-computable hitting-set generator, that is, a function $h\colon \{0,1\}^t \to \mathbb{F}^n$, $t = O(\log^2 mn)$, with the following property: for any nonzero polynomial $P(\mathbf{z}) = \det(\sum_{i=1}^{n} A_i z_i)$ where each $A_i \in \mathbb{F}^{m\times m}$ is a rank-1 matrix*

$$\Pr_{R\in\{0,1\}^t}\{P(h(R)) = 0\} \le 1/poly(mn).$$

3 Parallel algorithm for linearly representable submodular minimization

Our first step towards the parallel algorithm is to consider one of the special cases of LR submodular minimization. We give an algebraic algorithm for this special case. The algorithm is essentially a reduction to basic linear algebraic operations like computing rank and inverse of a matrix, which are doable in NC. The reduction is randomized and thus, puts the special case in RNC. Finally, we reduce the LR submodular minimization problem to this special case. We start with describing the special case and a solution for it.

3.1 LR submodular minimization for a special case

The special case we first consider is when the submodular function is the difference of a rank function and a positive linear function. Let $\mathcal{V} = \{V_e \subseteq \mathbb{F}^n\}_{e\in E}$ be a family of subspaces for a ground set E and a field $\mathbb{F}$, and $r\colon 2^E \to \mathbb{Z}$ be the corresponding rank function. Let $w \in \mathbb{Z}_+^E$ be a positive integer vector and define a modular function $w(S) := \sum_{e\in S} w_e$ for any $S \subseteq E$. Consider the function defined as $f(S) = r(S) - w(S)$ for $S \subseteq E$. Note that since w is modular, so is $-w$, and hence, f is submodular because both r and $-w$ are. We show that there is a randomized algebraic algorithm to find a minimizing set for $f(S)$ over $S \subseteq E$.

▶ **Lemma 7.** *Given a family of subspaces $\mathcal{V} = \{V_e \subseteq \mathbb{F}^n\}_{e\in E}$ with rank function r and a vector $w \in \mathbb{Z}_+^E$, there is an RNC algorithm to find the minimizing set $S^* \subseteq E$ for $f(S) = r(S) - w(S)$ that uses only $O(\log^2(nw(E)))$ random bits.*

To prove the Lemma 7, we work with a random/generic vectors that belong to any subspace V_e. Let us first build some terminology towards that. Let $B_e \subseteq \mathbb{F}^n$ be a basis for V_e for $e \in E$. For any set $S \subseteq E$, let us define $S_w = \{(e,i) \mid e \in S,\ 1 \le i \le w_e\}$. Clearly, $|S_w| = w(S)$. We will construct a matrix U whose columns will consist of w_e many generic vectors from the subspace V_e for $e \in E$. Formally, consider a tuple of indeterminates $\boldsymbol{\alpha} = (\alpha_{e,i,v} \mid (e,i) \in E_w,\ v \in B_e)$. Now, construct an $n \times E_w$ matrix U over $\mathbb{F}[\boldsymbol{\alpha}]$ whose columns are as follows:

$$u_{(e,i)} = \sum_{v \in B_e} \alpha_{e,i,v} v \text{ for } (e,i) \in E_w. \tag{1}$$

All notions of rank and linear independence for columns of U will be over the field of fractions $\mathbb{F}(\boldsymbol{\alpha})$. For any set $T \subseteq E_w$, let U_T denote the set of columns of U indexed by elements in T. Our first step is to show a lower bound on $\min_S f(S)$ in terms of rank of U.

▷ Claim 8. For any set $S \subseteq E$,

$$f(S) = r(S) - w(S) \geq \operatorname{rank}(U) - |E_w| = \operatorname{rank}(U) - w(E). \tag{2}$$

Proof. Consider the sets $S_w \subseteq E_w$ and $\bar{S}_w := E_w \setminus S_w$. One can write

$$\operatorname{rank}(U) \leq \operatorname{rank}(U_{S_w}) + \operatorname{rank}(U_{\bar{S}_w}) \leq r(S) + |\bar{S}_w|.$$

The first inequality is from basic linear algebra. The second inequality holds because every column in S_w is in the space $\sum_{e \in S} V_e$ and rank of $U_{\bar{S}_w}$ can be at most its cardinality. Writing $|\bar{S}_w| = |E_w| - |S_w| = w(E) - w(S)$ and rearranging the above inequality will give us the claim. ◁

Once we obtain this lower bound, a natural approach to find $\min_S f(S)$ is to find a set $S^* \subseteq E$ which satisfies (2) with equality. We describe a construction of such a set. First let us define $T^* \subseteq E_w$ to be the set of elements (e,i) such that the column $u_{(e,i)}$ participates non-trivially in some linear dependency among the columns of U. Equivalently,

$$T^* := \{(e,i) \in E_w \mid \operatorname{rank}(U) = \operatorname{rank}(U_{E_w \setminus (e,i)})\}.$$

Then define

$$S^* = \{e \in E \mid (e,i) \in T^* \text{ for some } i\}.$$

▶ **Lemma 9.** *S^* is a set minimizing $f(S)$ over all subsets $S \subseteq E$.*

Proof. As mentioned above, the strategy is to show that S^* satisfies (2) with equality. Towards this we will first prove that

$$\operatorname{rank}(U_{T^*}) = r(S^*). \tag{3}$$

Recall that the columns U_{T^*} are contained in $\sum_{e \in S^*} V_e$. What we need to show for (3) is that $\sum_{e \in S^*} V_e$ is in the linear span of U_{T^*}. We show this for each $V_{\tilde{e}}$, $\tilde{e} \in S^*$.

▷ Claim 10. For each $\tilde{e} \in S^*$, the subspace $V_{\tilde{e}}$ is in the linear span of columns in U_{T^*}.

Proof. By definition of S^*, there must be some $1 \leq \tilde{i} \leq w_{\tilde{e}}$ so that $(\tilde{e}, \tilde{i}) \in T^*$. By definition of T^*, the column $u_{(\tilde{e},\tilde{i})}$ non-trivially participates in some linear dependency among the columns of U_{T^*}. So, there exists a set $J \subseteq T^* \setminus \{(\tilde{e},\tilde{i})\}$ and a vector $\Gamma \in \mathbb{F}[\boldsymbol{\alpha}]^J$ such that

$$u_{(\tilde{e},\tilde{i})} = U_J \Gamma. \tag{4}$$

Now, recall that $u_{(\tilde{e},\tilde{i})}$ is a generic vector in $V_{\tilde{e}}$, and thus, can be used to express any vector in $V_{\tilde{e}}$. Hence, Equation (4) implies that every vector in $V_{\tilde{e}}$ is in the linear span of the columns in U_J. Below, we argue this point more formally.

Recall (1) that gives $u_{(\tilde{e},\tilde{i})}$ as $\sum_{v \in B_{\tilde{e}}} \alpha_{\tilde{e},\tilde{i},v} v$. One can invert (4) to write the vector Γ as a function of indeterminates $\{\alpha_{\tilde{e},\tilde{i},v}\}_v$ as follows. By basic linear algebra, there is an invertible submatrix $\hat{U}_J$ of U_J and a truncation $\hat{u}_{(\tilde{e},\tilde{i})}$ of $u_{(\tilde{e},\tilde{i})}$ such that

$$\Gamma = \hat{U}_J^{-1} \hat{u}_{(\tilde{e},\tilde{i})}. \tag{5}$$

Consider any vector $v' \in V_{\tilde{e}}$. Suppose v' can be expressed using the basis $B_{\tilde{e}}$ as $\sum_{v \in B_{\tilde{e}}} \delta_v v$. We can substitute $(\alpha_{\tilde{e},\tilde{i},v})_v = (\delta_v)_v$ in the right-hand side of (5) to obtain a vector Γ'. Note that since the matrix U_J is free of indeterminates $\{\alpha_{\tilde{e},\tilde{i},v}\}_v$, this substitution in (5) does not create any issues like division by zero. It will follow that

$$v' = U_J \Gamma'.$$

To conclude, every vector $v' \in V_{\tilde{e}}$ is in the linear span of columns in U_{T^*}. ◁

Claim 10 proves Equation (3). Now, we come back to proving that S^* satisfies (2) with equality. From (3), we have

$$f(S^*) = r(S^*) - w(S^*) = \text{rank}(U_{T^*}) - w(S^*) \leq \text{rank}(U_{T^*}) - |T^*|.$$

The inequality holds because $w(S^*) = |S_w^*| \geq |T^*|$ by the definition of S^*. By construction of T^*, the columns in $U_{E_w \setminus T^*}$ do not participate in any column dependency. Thus, we have $\text{rank}(U) - \text{rank}(U_{T^*}) = |E_w \setminus T^*| = |E_w| - |T^*|$. Putting this in the above inequality, we get

$$f(S^*) = r(S^*) - w(S^*) \leq \text{rank}(U) - |E_w|.$$

This together with (2) implies that S^* is a set that minimizes $f(S)$ over $S \subseteq E$. ◀

Proof of Lemma 7: the parallel algorithm

Let us review the construction of the minimizing set S^* from the previous subsection.

- Construct a matrix U, whose columns are generic vectors from the given subspaces. To be precise U has exactly w_e generic vectors from V_e for each $e \in E$.
- Construct the set $T^* := \{(e,i) \in E_w \mid \text{rank}(U) = \text{rank}(U_{E_w \setminus (e,i)})\}$.
- Construct the set $S^* \subseteq E$ that contains all those elements e such that T^* contains (e,i) for some $1 \leq i \leq w_e$.

The rank computations in the second step can all be done in parallel. Importantly, rank computation for any matrix over the base field $\mathbb{F}$ can be done in NC. However, this computation is not efficient for U (or its submatrices) as it is a matrix with indeterminates $\boldsymbol{\alpha}$. To overcome this, we plan to substitute all the indeterminates with field constants. Observe that as long as our substitution preserves the ranks of all column subsets of U, one can safely run the above procedure on the substituted matrix and expect to get the correct answer.

How do we find the right substitution? We argue that a random substitution from a large enough set of field elements does the job. It is known that the rank of a subset of columns remains the same with high probability if each indeterminate is replaced with a field element randomly chosen from a set of size $\text{poly}(\text{size}(U)) = \text{poly}(n \times w(E))$. One can see this by applying Lemma 5 on the largest nonzero minor. But, note that we need one substitution that preserves ranks for U and each submatrix $U_{E_w \setminus (e,i)}$ simultaneously. One can use union bound to argue that with high probability, all the desired submatrices preserve their rank. Note that this algorithm needs to use polynomially many random bits. Next we show how to reduce this number of random bits.

Derandomization

To reduce the number of random bits, we use results from deterministic polynomial identity testing. Recall Lemma 6 that gives a pseudorandom substitution that preserves nonzeroness of polynomials of the form $\det(\sum_{i=1}^{n} A_i z_i)$ for $\text{rank}(A_i) = 1$, with high probability. Note that any minor of U is also a polynomial of this form because any variable $\alpha_{e,i,u}$ appears in exactly one column of U. Again, one can use union bound to argue that with high probability, the substitution preserves nonzeroness for all the desired minors of U and each $U_{E_w \setminus (e,i)}$ simultaneously.

To conclude, the pseudorandom substitution from Lemma 6 uses $O(\log^2(nw(E)))$ random bits and with that substitution our algorithm will give the correct minimizing set S^* with high probability.

3.2 Reduction to the special case

Recall Definition 1 which defines LR submodular functions to be those which can be written as a sum of a collection of rank functions together with another collection of rank functions applied on the complement set. We will first argue that the same class of functions is also captured by just taking sum of a rank function and another rank function applied on the complement.

▶ **Observation 11.** *Let $f\colon 2^E \to \mathbb{Z}$ be an LR submodular function given as*

$$f(S) = \sum_{j=1}^{\ell} r_j(S) + \sum_{j=\ell+1}^{k} r_j(\bar{S}),$$

for some $\ell \leq k$, where r_j is the rank function for a family of subspaces $\mathcal{V}_j = \{V_{j,e} \subseteq \mathbb{F}^n\}_{e \in E}$ for $1 \leq j \leq k$. Then f can also be written as

$$f(S) = r'_1(S) + r'_2(\bar{S}),$$

where r'_1 and r'_2 are the rank functions of the families $\mathcal{V}'_1 = \{\oplus_{j=1}^{\ell} V_{j,e} \subseteq \mathbb{F}^{\ell n}\}_{e \in E}$ and $\mathcal{V}'_2 = \{\oplus_{j=\ell+1}^{k} V_{j,e} \subseteq \mathbb{F}^{(k-\ell)n}\}_{e \in E}$, respectively.

Next, we show that we can, in fact, take one of the rank functions to be modular. That is, any LR submodular function can be written as a sum of a rank function and a modular function. In context of general submodular functions, this is a known fact and was used in the first pseudo-polynomial time submodular minimization [6, Lemma 2.1]. Here, we show a more specific result for LR submodular functions that says that the new rank function is also linearly representable and the corresponding family of subspaces can be constructed efficiently.

▶ **Lemma 12.** *Given an LR submodular function $f(S)$ by k families of subspaces of $\mathbb{F}^n$ as in Definition 1, one can compute in* NC *a family of subspaces $\mathcal{V} = \{V_e \subseteq \mathbb{F}^{kn|E|}\}_{e \in E}$ with rank function r, a vector $w \in \{1, 2, \ldots, kn\}^E$, and a constant C such that for each $S \subseteq E$,*

$$f(S) = r(S) - w(S) + C.$$

To prove Lemma 12, the first step is to use Observation 11 to get the LR submodular function in the form $r'_1(S) + r'_2(\bar{S})$. Then the next step is to write $r'_2(\bar{S})$ as a sum of a rank function on S and a modular function, which is what the following lemma does. Final step is to combine the new rank function with $r'_1(S)$ to get a single rank function, again using Observation 11.

▶ **Lemma 13.** *Let $\mathcal{V} = \{V_e \subseteq \mathbb{F}^n\}_{e \in E}$ be a family of subspaces with its rank function $r\colon 2^E \to \mathbb{Z}$. Then one can construct in NC another family of subspaces $\mathcal{V}^* = \{V_e^* \subseteq \mathbb{F}^{n'}\}_{e\in E}$ with rank function r^* for some $n' \le n|E|$, and a vector $b \in \{1, 2, \dots, n\}^E$ such that for each $S \subseteq E$,*

$$r(\bar{S}) = r^*(S) - b(S) + r(E).$$

Proof. For each $e \in E$, let b_e be the dimension of the subspace V_e and $B_e = \{u_{e,i} \mid 1 \le i \le b_e\}$ be a basis for it. Let $E' = \{(e, i) \mid e \in E,\ 1 \le i \le b_e\}$ be a new ground set. Note that since $b_e \le n$, we have $|E'| \le n|E|$. Consider an $n \times |E'|$ matrix M whose set of columns is $\{u_{e,i}\}_{e,i}$. Without loss of generality, we can assume that M has full row-rank, i.e., $n = r(E)$ (otherwise we could drop some rows). Let M^* be a $(|E'| - n) \times |E'|$ matrix whose row-space is the orthogonal complement of the row-space of M (for a construction in NC, see [4, 22]). The following claim is well known in matroid theory and is used for representation of a dual matroid (see [26, 2.1.9 and 2.2.8]). For any $T \subseteq E'$ and matrix M, let M_T stand for the set of columns of M corresponding to the set of indices T.

▷ Claim 14. For any set $T \subseteq E'$,

$$\operatorname{rank}(M_{E' \setminus T}) = \operatorname{rank}(M^*_T) - |T| + n.$$

Let $\{u^*_{e,i}\}_{e,i}$ be the set of columns of M^*. Consider the family of subspaces $\mathcal{V}^* = \{V_e^* \subseteq \mathbb{F}^{|E'|-n}\}_{e\in E}$, where $V_e^* = \operatorname{span}\{u^*_{e,i} \mid 1 \le i \le b_e\}$. Let $r^*\colon 2^E \to \mathbb{Z}$ be the rank function of $\mathcal{V}^*$. Then for any $S \subseteq E$, take $T = \{(e, i) \mid e \in S,\ 1 \le i \le b_e\}$ in Claim 14, and we get

$$r(E \setminus S) = r^*(S) - b(S) + n. \qquad ◀$$

Proof of Theorem 2. Lemma 12 gives an NC-reduction from LR submodular minimization to minimizing functions of the form $f(S) = r(S) - w(S) + C$. To minimize $f(S)$, it is sufficient to minimize $r(S) - w(S)$, which is what Lemma 7 does. This concludes the RNC algorithm for LR submodular functions as claimed in Theorem 2. ◀

4 Variants and Applications

In this section, we first consider two variants of submodular minimization which can be reduced to submodular minimization. Here we basically show that this kind of reductions can also made to work in the setting of LR submodular functions. Later on, we also show reductions from two combinatorial problems to LR submodular minimization.

4.1 Variants

Submodular minimization with containment constraints. We first consider an extension of submodular minimization that asks for a minimizing set with containment constraints. Given a submodular function $f\colon 2^E \to \mathbb{R}$ and two sets $S_0 \subseteq S_1 \subseteq E$, suppose the goal is to minimize $f(S)$ subject to $S_0 \subseteq S \subseteq S_1$. It is known that there is a submodular function g on the ground set $S_1 \setminus S_0$ such that for any $S_0 \subseteq S \subseteq S_1$, $f(S) = g(S \setminus S_0) + C'$ for some constant C'. If f is linearly representable then we would like to come up with such a function g that is also linearly representable. Towards this, we will need the following claim.

▷ Claim 15. Let $\mathcal{V} = \{V_e \subseteq \mathbb{F}^n\}_{e\in E}$ be a family of subspaces and r be the corresponding rank function. Let there be two sets $S_0 \subseteq S_1 \subseteq E$. Then we can construct a family of subspaces $\mathcal{V}' = \{V'_e\}_{e \in S_1 \setminus S_0}$ with rank function r' such that for each $S_0 \subseteq S \subseteq S_1$

$$r(S) - r(S_0) = r'(S \setminus S_0).$$

Proof. Let V_{S_0} be the subspace $\sum_{e\in S_0} V_e$. For each $e \in S_1 \setminus S_0$, we define V'_e to be the quotient space $V'_e = V_e/V_{S_0}$. Now, for any $S_0 \subseteq S \subseteq S_1$, we have

$$r'(S \setminus S_0) = \dim(\sum_{e\in S\setminus S_0} V'_e) = \dim\left((\sum_{e\in S\setminus S_0} V_e)/V_{S_0} \right) = \dim(V_S/V_{S_0}) = r(S) - r(S_0).$$

◁

From Lemma 12, any linearly representable submodular function can be given as $f(S) = r(S) - w(S) + C$, for a rank functions $r(S)$ on a family of subspaces $\mathcal{V}$, a modular function $w(S)$ and a constant C. For given two sets $S_0 \subseteq S_1 \subseteq E$, let r' be the function constructed in Claim 15 from r. For any $T \subseteq S_1 \setminus S_0$ define $g(T) = r'(T) - w(T) + C$. Using Claim 15, one can verify that for any $S_0 \subseteq S \subseteq S_1$,

$$g(S\setminus S_0) = r'(S\setminus S_0) - w(S\setminus S_0) + C = r(S) - r(S_0) - w(S) + w(S_0) + C = f(S) - f(S_0) + C.$$

Choose $C' = C - f(S_0)$ and we get the desired relation. Now, to minimize f under containment constraints, one can just minimize g on the smaller ground set.

Submodular minimization over non-empty sets. In applications, one often needs to minimize a submodular function over non-empty sets, for example, min-cut in an undirected graph. To do this, one can go over all elements $e \in E$ and minimize $f(S)$ with the containment constraint $\{e\} \subseteq S$ (as discussed above). This will give us a minimum value for each choice of e. The minimum among these values will be the minimum over non-empty subsets.

Separation oracle for a linear polymatroid. Given a family of subspaces $\mathcal{V} = \{V_e \subseteq \mathbb{F}^n\}_{e\in E}$ with its rank function $r\colon 2^E \to \mathbb{Z}$, the corresponding polymatroid is a polytope $P_r \subseteq \mathbb{R}^E$ defined as

$$P_r = \{x \in \mathbb{R}^E \mid x \geq 0, \sum_{e\in S} x_e \leq r(S)\ \forall S \subseteq E.\}$$

Given a rational point $\beta \in \mathbb{R}^E$, one needs to decide if β lies in P_r, and if not then find a violating constriant. The non-negativity constraints are easy to check. The other rank constraints are equivalent to

$$\min_{S\subseteq E} (r(S) - x(S)) \geq 0.$$

Thus, to check if β satisfies the rank constraints, it suffices to minimize the function $f(S) = r(S) - \beta(S)$. Moreover, if there is a violating constraint, then the set S^* minimizing $f(S)$ will give a violating constraint. If β is an integer vector, we have already seen how to find S^* in Lemma 7. When β has rational coordinates, then one can assume them to have a common denominator q, i.e., $\beta_e = p_e/q$ for integers p_e, q. Now, the minimization function becomes $q \times r(S) - p(S)$. Now, $p(S)$ is an integral function. To get the multiplicative factor q in the rank, for each subspace V_e in the family, one can take the direct sum with its copies as $\oplus_{j=1}^{q} V_e$. Note that this is efficient as long as the number q is polynomially bounded.

4.2 Applications

In this section, we show that the two combinatorial problems mentioned in Section 1 reduce to LR submodular minimization. To the best of our knowledge, these problems do not have any known reduction to linear matroid intersection. We start with defining the necessary terminology. Branchings and arborescences are directed analogues of forests and spanning trees.

▶ **Definition 16** (Branching and Arborescence). *For a directed graph $G(V,E)$, a subset $B \subseteq E$ of edges is a branching if it contains no undirected cycles (i.e, there are no cycles induced by B if edges in B are considered to be undirected) and for every vertex v, there is at most one edge in B that is incoming to v. A vertex is a root of B if it has no incoming edges in B. An arborescence is a branching with exactly one root, that is, it is a rooted tree. If the root vertex of an arborescence is a, then we call it an a-arborescence.*

Covering by a-arborescences

A directed graph $G(V,E)$ is said to be covered by k a-arborescences if there exists subsets $B_1, B_2, \dots, B_k \subseteq E$, each of which is an a-arborescence and $E = B_1 \cup B_2 \cup \cdots \cup B_k$. Vidyasankar [32] gave the following characterization for the graph to covered by k a-arborescences. For any set of vertices $S \subseteq V$, let $H[S] := \{u \in S \mid \exists v \in V \setminus S,\ (v,u) \in E\}$ and let $\delta^{in}(S) := \{(v,u) \in E \mid \exists v \in V \setminus S,\ u \in S\}$ be the set of incoming edges in S. Let $\deg^{\text{in}}(v)$ be the number of incoming edges to v.

▶ **Theorem 17** ([32]). *For a vertex $a \in V$ and a positive number k, the directed graph $G(V,E)$ is be covered by k a-arborescences if and only if*
- $\deg^{\text{in}}(a) = 0$ *and* $\deg^{\text{in}}(v) \leq k$ *for each* $v \in V$ *and*
- $\sum_{v \in H[S]} (k - \deg^{\text{in}}(v)) \geq k - |\delta^{in}(S)|$, *for each non-empty subset S of $V \setminus \{a\}$.*

Reduction to submodular minimization. We show that testing the conditions required in Theorem 17 can be reduced to LR submodular minimization. Testing the first condition is trivial. We come to the second one. Let us define a function $f \colon 2^{V \setminus \{a\}} \to \mathbb{Z}$ as follows:

$$f(S) = \sum_{v \in H[S]} (k - \deg^{\text{in}}(v)) + |\delta^{in}(S)|.$$

We will just show that $f(S)$ is a linearly representable submodular function. Clearly, one can check the required condition by finding $\min_{S \subseteq V \setminus \{a\}} f(S)$ and verifying that it is at least k.

For any vertex $u \in V$, let $\chi_u \in \{0,1\}^V$ be the characteristic vector of v. Let us define two families of subspaces $\mathcal{L}_1$ and $\mathcal{L}_2$ with rank functions r_1 and r_2.
- $\mathcal{L}_1 = \{L_{1,u} \subseteq \mathbb{R}^{k \times |V|}\}_{u \in V}$, where $L_{1,u} = \sum_{\substack{v=u \text{ or} \\ (u,v) \in E}} \oplus_{j=1}^{k-\deg^{\text{in}}(v)} \operatorname{span}(\chi_v)$ and
- $\mathcal{L}_2 = \{L_{2,u} \subseteq \mathbb{R}^{k \times |V|}\}_{u \in V}$, where $L_{2,u} = \oplus_{j=1}^{k-\deg^{\text{in}}(u)} \operatorname{span}(\chi_u)$

One can observe that for any set $S \subseteq V$, $r_1(S)$ is just the sum of the quantity $k - \deg^{\text{in}}(v)$ over all vertices v that are either in S or out-neighbors of S, and $r_2(S)$ is sum of the same quantity over all the vertices of S. Thus, we can write

$$r_1(S) - r_2(S) = \sum_{v \in H[\bar{S}]} (k - \deg^{\text{in}}(v)), \tag{6}$$

where $\bar{S} = V \setminus S$. Note that $-r_2(S)$ is same as $r_2(\bar{S}) - r_2(V)$. Thus,

$$\sum_{v \in H[S]} (k - \deg^{\text{in}}(v)) = r_1(\bar{S}) + r_2(S) - r_2(V). \tag{7}$$

Now, we will express the second part of $f(S)$, that is $|\delta^{in}(S)|$, as a LR submodular function. For any edge $e \in E$, let $\chi_e \in \{0,1\}^E$ be the characteristic vector of e. Let us define three families of subspaces $\mathcal{L}_3$, $\mathcal{L}_4$ and $\mathcal{L}_5$ with rank functions r_3, r_4 and r_5, respectively.

- $\mathcal{L}_3 = \{L_{3,u} \subseteq \mathbb{R}^E\}_{u \in V}$, where $L_{3,u} = \text{span}\{\chi_e \mid e = (u,v) \text{ for some } v \in V\}$.
- $\mathcal{L}_4 = \{L_{4,u} \subseteq \mathbb{R}^E\}_{u \in V}$, where $L_{4,u} = \text{span}\{\chi_e \mid e = (v,u) \text{ for some } v \in V\}$.
- $\mathcal{L}_5 = \{L_{5,u} \subseteq \mathbb{R}^E\}_{u \in V}$, where $L_{5,u} = \text{span}\{\chi_e \mid e = (u,v) \text{ or } e = (v,u) \text{ for some } v \in V\}$.

One can observe that for any set $S \subseteq V$, $r_3(S)$ is the total number of edges which are outgoing from some vertex in S, $r_4(S)$ is the total number of edges which are incoming to some vertex in S, and $r_5(S)$ is the total number of edges that are incident (outgoing or incoming) to some vertex in S. Thus, one can write

$$2|\delta^{in}(S)| = r_3(\bar{S}) + r_4(S) + r_5(S) + r_5(\bar{S}) - 2|E|. \tag{8}$$

Together with (7) and (8), we can write,

$$2f(S) = 2 \times (r_1(\bar{S}) + r_2(S) - r_2(V)) + r_3(\bar{S}) + r_4(S) + r_5(S) + r_5(\bar{S}) - 2|E|.$$

The terms $r_2(V)$ and $2|E|$ are constants here. The other terms give us a linearly representable submodular function.

Recall that we have to minimize the function $f(S)$ over subsets S that do not contain the vertex a. We had reduced such a constrained minimization to general minimization in the previous subsection.

Packing of branchings

For a given directed graph and given subsets $R_1, R_2, \ldots, R_k$ of vertices, we need to decide if there exist k edge-disjoint branchings that are rooted at $R_1, R_2, \ldots, R_k$, respectively. Edmonds (see [29, Theorem 53.1]) gave the following characterization.

▶ **Theorem 18.** *Let $G = (V, E)$ be a directed graph with $R_1, R_2, \ldots, R_k$ being subsets of V. Then there exist disjoint branchings $B_1, B_2, \ldots, B_k$ such that B_i has root set R_i for $1 \le i \le k$ if and only if $|\delta^{in}(S)| \ge |i : R_i \cap S = \emptyset|$ for each non-empty subset S of V.*

Let $f(S) = |\delta^{in}(S)| - |i : R_i \cap S = \emptyset|$. To check the condition in the theorem, it is sufficient to minimize $f(S)$ over non-empty subsets of V. We have already expressed $|\delta^{in}(S)|$ as a linearly representable submodular function in (8). We need to now express the other part of the function. Let us define

$$g_i(S) := \begin{cases} 1 & \text{if } S \cap R_i \neq \emptyset \\ 0 & \text{otherwise.} \end{cases}$$

One can verify that g_i is the rank function of the following family of subspaces: $\mathcal{L}_i = \{L_{i,u} \subseteq \mathbb{R}\}_{u \in V}$, where $L_{i,u} = \{1\}$ if $u \in R_i$ and $L_{i,u} = \{0\}$ otherwise. Now, one can express the desired function in terms of g_i's.

$$-|i : R_i \cap S = \emptyset| = \sum_{i=1}^{k} g_i(S) - k.$$

Together with (8), this gives us a linear representation for $f(S)$ (up to additive constants).

5 Discussion

We have given a parallel algorithm for submodular minimization in the special case of linearly representable submodular functions. It would be interesting to know if there are other classes of submodular functions that admit parallel algorithms. More generally, it is not clear if there can be an efficient parallel algorithm for submodular minimization in the oracle model.

We have given two examples of combinatorial problems that are captured by LR submodular minimization, but are not known to be reducible to linear matroid intersection. One needs to investigate what are other examples of such problems.

References

1 Sepehr Assadi, Yu Chen, and Sanjeev Khanna. Polynomial pass lower bounds for graph streaming algorithms. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2019, page 265–276, New York, NY, USA, 2019. Association for Computing Machinery. doi:10.1145/3313276.3316361.

2 Eric Balkanski and Yaron Singer. Minimizing a submodular function from samples. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 814–822, Red Hook, NY, USA, 2017. Curran Associates Inc.

3 Eric Balkanski and Yaron Singer. A lower bound for parallel submodular minimization. In *Proceedings of the 52nd Annual ACM Symposium on Theory of Computing (STOC) June 22–26, 2020 in Chicago, IL*, 2020. URL: http://people.seas.harvard.edu/~yaron/papers/lower_bound_submodular_minimization.pdf.

4 Allan Borodin, Joachim von zur Gathen, and John Hopcroft. Fast parallel matrix and GCD computations. *Information and Control*, 52(3):241–256, 1982.

5 G. Cornuéjols, G.L. Nemhauser, and L.A. Wolsey. The uncapacitated facility location problem. In *In: Discrete Location Theory (P. Mirchandani, R. Francis, eds.)*, pages 119–171. Wiley, New York, 1990.

6 William H. Cunningham. On submodular function minimization. *Combinatorica*, 5(3):185–192, 1985. doi:10.1007/BF02579361.

7 G. Davis, S. Mallat, and M. Avellaneda. Adaptive greedy approximations. *Constructive Approximation*, 13(1):57–98, 1997. doi:10.1007/BF02678430.

8 Richard A. Demillo and Richard J. Lipton. A probabilistic remark on algebraic program testing. *Information Processing Letters*, 7(4):193–195, 1978.

9 Uriel Feige, Vahab S. Mirrokni, and Jan Vondrák. Maximizing non-monotone submodular functions. *SIAM J. Comput.*, 40(4):1133–1153, 2011. doi:10.1137/090779346.

10 Stephen A. Fenner, Rohit Gurjar, and Thomas Thierauf. Bipartite perfect matching is in quasi-nc. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA*, pages 754–763, 2016.

11 Martin Grötschel, László Lovász, and Alexander Schrijver. The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, 1(2):169–197, 1981. doi:10.1007/BF02579273.

12 Martin Grötschel, László Lovász, and Alexander Schrijver. *Geometric Algorithms and Combinatorial Optimization*, volume 2 of *Algorithms and Combinatorics*. Springer, 1988. doi:10.1007/978-3-642-97881-4.

13 Jiong Guo, Rolf Niedermeier, and Sebastian Wernicke. Parameterized complexity of vertex cover variants. *Theory Comput. Syst.*, 41:501–520, October 2007. doi:10.1007/s00224-007-1309-3.

14 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, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*, pages 821–830, 2017. doi:10.1145/3055399.3055440.

15 Satoru Iwata, Lisa Fleischer, and Satoru Fujishige. A combinatorial strongly polynomial algorithm for minimizing submodular functions. *J. ACM*, 48(4):761–777, 2001. doi:10.1145/502090.502096.

16 Richard M. Karp. Reducibility among combinatorial problems. In *Proceedings of a symposium on the Complexity of Computer Computations, held March 20-22, 1972, at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York, USA*, pages 85–103, 1972. doi:10.1007/978-1-4684-2001-2_9.

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

18 E. L. Lawler. Cutsets and partitions of hypergraphs. *Networks*, 3(3):275–285, 1973. doi:10.1002/net.3230030306.

19 Yin Tat Lee, Aaron Sidford, and Sam Chiu-Wai Wong. A faster cutting plane method and its implications for combinatorial and convex optimization. In *Proceedings of the 2015 IEEE 56th Annual Symposium on Foundations of Computer Science (FOCS)*, FOCS '15, page 1049–1065, USA, 2015. IEEE Computer Society. doi:10.1109/FOCS.2015.68.

20 L. Lovász. Submodular functions and convexity. In Achim Bachem, Bernhard Korte, and Martin Grötschel, editors, *Mathematical Programming The State of the Art: Bonn 1982*, pages 235–257. Springer Berlin Heidelberg, Berlin, Heidelberg, 1983. doi:10.1007/978-3-642-68874-4_10.

21 László Lovász. On determinants, matchings, and random algorithms. In *Fundamentals of Computation Theory*, pages 565–574, 1979.

22 Ketan Mulmuley. A fast parallel algorithm to compute the rank of a matrix over an arbitrary field. *Combinatorica*, 7(1):101–104, 1987.

23 Ketan Mulmuley, Umesh V. Vazirani, and Vijay V. Vazirani. Matching is as easy as matrix inversion. *Combinatorica*, 7:105–113, 1987. doi:10.1007/BF02579206.

24 H. Narayanan, Huzur Saran, and Vijay V. Vazirani. Randomized parallel algorithms for matroid union and intersection, with applications to arboresences and edge-disjoint spanning trees. *SIAM J. Comput.*, 23(2):387–397, 1994. doi:10.1137/S0097539791195245.

25 Øystein Ore. Über höhere Kongruenzen. *Norsk Mat. Forenings Skrifter Ser. I*, 7(15):27, 1922.

26 James G. Oxley. *Matroid Theory (Oxford Graduate Texts in Mathematics)*. Oxford University Press, Inc., New York, NY, USA, 2006.

27 M. J. Piff and D. J. A. Welsh. On the number of combinatorial geometries. *Bulletin of the London Mathematical Society*, 3(1):55–56, 1971. doi:10.1112/blms/3.1.55.

28 Alexander Schrijver. A combinatorial algorithm minimizing submodular functions in strongly polynomial time. *J. Comb. Theory, Ser. B*, 80(2):346–355, 2000. doi:10.1006/jctb.2000.1989.

29 Alexander Schrijver. *Combinatorial optimization : polyhedra and efficiency. Vol. B. , Matroids, trees, stable sets. chapters 39-69.* Algorithms and combinatorics. Springer-Verlag, Berlin, Heidelberg, New York, N.Y., et al., 2003. URL: http://opac.inria.fr/record=b1124843.

30 Jacob T. Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *Journal of the ACM*, 27(4):701–717, October 1980.

31 Zoya Svitkina and Lisa Fleischer. Submodular approximation: Sampling-based algorithms and lower bounds. *SIAM J. Comput.*, 40(6):1715–1737, December 2011. doi:10.1137/100783352.

32 K. Vidyasankar. Covering the edge set of a directed graph with trees. *Discrete Mathematics*, 24(1):79–85, 1978. doi:10.1016/0012-365X(78)90174-7.

33 Richard Zippel. Probabilistic algorithms for sparse polynomials. In *Proceedings of the International Symposium on Symbolic and Algebraic Computation (EUROSAM)*, pages 216–226. Springer-Verlag, 1979.