Decomposable Submodular Function Minimization via Maximum Flow

Kyriakos Axiotis [1] **Adam Karczmarz** [2] **Anish Mukherjee** [2] **Piotr Sankowski** [3 4 2] **Adrian Vladu** [5 6]

Abstract

This paper bridges discrete and continuous optimization approaches for decomposable submodular function minimization, in both the standard and parametric settings. We provide improved running times for this problem by reducing it to a number of calls to a maximum flow oracle. When each function in the decomposition acts on $O(1)$ elements of the ground set V and is polynomially bounded, our running time is up to polylogarithmic factors equal to that of solving maximum flow in a sparse graph with $O(|V|)$ vertices and polynomial integral capacities. We achieve this by providing a simple iterative method which can optimize to high precision any convex function defined on the submodular base polytope, provided we can efficiently minimize it on the base polytope corresponding to the cut function of a certain graph that we construct. We solve this minimization problem by lifting the solutions of a parametric cut problem, which we obtain via a new efficient combinatorial reduction to maximum flow. This reduction is of independent interest and implies some previously unknown bounds for the parametric minimum s, t-cut problem in multiple settings.

1. Introduction

A significant amount of work has been dedicated to the study of submodular functions. While this topic has garnered a lot of excitement from the theory community due to its the multiple connections to diverse algorithmic areas (Lovász, 1983; Grötschel et al., 1981), on the practical side minimizing submodular functions has been intensively used to model discrete problems in machine learning. MAP inference in Markov Random Fields (Kohli et al., 2009), image segmentation (Arora et al., 2012; Shanu et al., 2016), clustering (Narasimhan & Bilmes, 2007), corpus extraction problems (Lin & Bilmes, 2011) are just a few success stories of submodular minimization.

Polynomial time algorithms for this problem have been known ever since the 80's (Grötschel et al., 1981), and they have seen major running time improvements in more recent years (Schrijver, 2000; Iwata, 2003; Fleischer & Iwata, 2003; Orlin, 2009; Lee et al., 2015; Chakrabarty et al., 2017). However, the massive scale of the problems that use submodular minimization nowadays drives the need for further developments.

One great advantage offered by the submodular functions that occur in practice is that they are *structured*. For example, in many common cases (hypergraph cuts (Veldt et al., 2020), covering functions (Stobbe & Krause, 2010), MAP inference (Fix et al., 2013; Kohli et al., 2009; Vicente et al., 2009)) these can be decomposed into sums of simple submodular functions defined on small subsets. For these instances, prior work (Jegelka et al., 2013; Nishihara et al., 2014; Ene & Nguyen, 2015; Ene et al., 2017; Li & Milenkovic, 2018; Kumar et al., 2019) has focused on providing efficient algorithms in the regime where the functions in the decomposition admit fast optimization oracles.

Notably, many of these recent developments have leveraged a mix of ideas coming from both discrete and continuous optimization. In particular, Ene et al. (Ene et al., 2017) present algorithms for decomposable function minimization that are based on both continuous methods (i.e. gradient descent) and discrete algorithms, as the authors employ a version of the preflow-push algorithm for maximum flow (Goldberg & Tarjan, 1988). [1]

As this work was paralleled by multiple improvements to the running time for maximum flow (Madry, 2013; 2016; Liu & Sidford, 2020; Kathuria et al., 2020; Brand et al., 2021; Gao et al., 2021), most of which stemmed from innovations in convex optimization, it seemed plausible that the same new optimization techniques could be helpful for

[*]Equal contribution [1]MIT [2]University of Warsaw [3]IDEAS NCBR [4]MIM Solutions [5]CNRS [6]IRIF, Université de Paris. Correspondence to: Kyriakos Axiotis <kaxiotis@mit.edu>, Piotr Sankowski <sank@mimuw.edu.pl>, Adrian Vladu <vladu@irif.fr>.

Proceedings of the 38th International Conference on Machine Learning, PMLR 139, 2021. Copyright 2021 by the author(s).

[1] In light of their result, there seems to be an apparent connection to max flow, but their algorithm is unable to black-box it. This is because their scheme relies on local moves in an auxiliary graph, which need to be carefully controlled in order to maintain feasibility in the submodular base polytope.

improving the running times of other fundamental problems in combinatorial optimization, including submodular function minimization. In this context, a particularly intriguing question emerged:

Can we leverage the techniques used to obtain faster algorithms for maximum flow to provide faster algorithms for submodular function minimization?

We answer this question in the affirmative, by showing how to solve decomposable submodular function minimization using *black-box access* to any routine that can compute the maximum flow in a capacitated directed graph. To compare the running times, in the case where all the functions in the decomposition act on $O(1)$ elements of the ground set and are polynomially bounded (such as the case of a hypergraph cut function, with $O(1)$ sized hyperedges), our algorithm has – up to polylogarithmic factors – the same running time as that of computing maximum flow in a sparse graph with $O(|V|)$ vertices, and polynomial integral capacities (Goldberg & Rao, 1998; Brand et al., 2021; Gao et al., 2021).

As it turns out, to achieve this it is not sufficient to directly use off-the-shelf maximum flow algorithms. Instead, our approach is based on solving submodular minimization in the more general parametric setting, where we further parametrize the problem with an additional time-dependent penalty term on the elements in the set, and want to simultaneously solve *all* the problems in this family. In turn, our reduction requires solving the parametric minimum cut problem, which has been intensely studied in the classical graph theoretic literature (Gallo et al., 1989; McCormick, 1999; Tarjan et al., 2006; Granot et al., 2012). In this setting, which is essentially a particular case of parametric submodular minimization, the capacities of certain arcs in the graph evolve in a monotonic fashion.

While some of the existing work on parametric cuts and flows does provide efficient algorithms via reductions to maximum flow (Tarjan et al., 2006), the type of parametric capacities it supports does not cover the requirements for our more general scenario. Therefore, we develop a new efficient algorithm for computing parametric cuts under a broad range of parametric capacities. Our algorithm is nearly optimal from the perspective of weakly-polynomial time algorithms, since its running time matches (up to polylogarithmic factors involving certain parameters) that of the fastest maximum flow algorithm in a directed graph with integer capacities. In addition, our reduction also provides novel improvements in several other regimes, involving the strongly polynomial case, and that of planar graphs, both of which may be of independent interest.

1.1. Our Results

In this paper we establish further connections between discrete and continuous optimization to provide an efficient algorithm for solving the decomposable submodular function minimization problem in the more general parametric setting. Our algorithm is at its core based on a continuous optimization method, but whose progress steps are driven by a new combinatorial algorithm we devise for the parametric cut problem. In this sense, our approach leverages the paradigm of combinatorial preconditioning from scientific computing literature (Spielman & Teng, 2004; Bern et al., 2006; Koutis et al., 2009; Toledo & Avron, 2010).

To properly state our main result, we need to introduce some notation. Let $V = \{1, \dots, n\}$ and let $F : 2^V \to \mathbb{N}$ be a submodular set function with the special property that

$$F(S) = \sum_{i=1}^{r} F_i(S), \quad \text{for all } S \subseteq V\,,$$

where each $F_i : 2^V \to \mathbb{N}$ is a submodular set function acting on a subset $V_i \subseteq V$ of elements, in the sense that $F_i(S) = F_i(S \cap V_i)$ for all $S \subseteq V$. Let EO_i be the time required to evaluate $F_i(S)$ for any $S \subseteq V_i$, and let $\mathcal{O}_i$ be the time required to minimize $F_i(S) + w(S)$ over V_i, where w is any linear function, and suppose that $\max_{S\subseteq V} F(S) = (n+r)^{O(1)}$. Furthermore, for each $i \in V$ let $\psi_i : \mathbb{R} \to \mathbb{R}$ be a strictly convex function satisfying $(n+r)^{-O(1)} \leq |\psi_i''(x)| \leq (n+r)^{O(1)}$ and $|\psi_i'(0)| \leq (n+r)^{O(1)}$.

Then our main theorem is the following.

Theorem 1.1. *There is an algorithm which for all $\lambda \in \mathbb{R}$ simultaneously optimizes the objective*

$$\min_{S\subseteq V} \left(F(S) + \sum_{i\in S} \psi_i'(\lambda) \right)$$

by returning a vector x such that for any $\lambda \in \mathbb{R}$ the set $S^\lambda = \{u \,:\, x_u \geq \lambda\}$ satisfies

$$F(S^\lambda) + \sum_{u\in S^\lambda} \psi_u'(\lambda) \leq \min_{S\subseteq V} \left(F(S) + \sum_{u\in S} \psi_u'(\lambda) \right) + \varepsilon\,.$$

Furthermore, if $T_{\mathrm{maxflow}}(n, m)$ is the time required to compute the maximum flow in a directed graph with polynomially bounded integral capacities, then our algorithm runs in time

$$\widetilde{O}\Bigg(\max_i |V_i|^2 \Bigg(\sum_{i=1}^{r} |V_i|^2 \mathcal{O}_i + T_{\mathrm{maxflow}}\Bigg(n, n + \sum_{i=1}^{r} |V_i|^2\Bigg)\Bigg) \log \frac{1}{\epsilon} \Bigg)\,.$$

To better understand this result, let us consider the case where each submodular function in the decomposition acts on a small number of elements, i.e. $|V_i| = O(1)$. [2] In this case we have the following corollary:

Corollary 1.2. *If each function F_i in the decomposition acts on $O(1)$ elements of the ground set, then we can the solve parametric submodular minimization problem to ϵ precision in time*

$$\widetilde{O}\left(T_{\text{maxflow}}(n, n+r)\log\frac{1}{\epsilon}\right) .$$

While our statements concern the parametric setting, it is easy to use them to recover the solution to the standard submodular minimization problem. Simply by letting $\psi_i'(t) = t$ for all i, and thresholding the returned vector at 0 we obtain the desired result. Using Goldberg-Rao (Goldberg & Rao, 1998) or the current state of the art algorithms for maximum flow (van den Brand et al., 2021; Gao et al., 2021), we see that this significantly improves over all the previous algorithms for decomposable submodular minimization, in the regime where all sets V_i are small. Following (Ene et al., 2017) it has remained widely open whether algorithms improving upon the $\widetilde{O}(\min\{n^2, nr\}\log^{O(1)}(1/\epsilon))$ running time exist, and it has been conjectured that faster running times could be obtained by leveraging the newer techniques for graph algorithms based on interior point methods.

Using (van den Brand et al., 2021; Gao et al., 2021), we obtain a running time of $\widetilde{O}\left(\min\{n^{3/2} + r, (n+r)^{3/2-1/328}\}\log 1/\epsilon\right)$.

A crucial subroutine our algorithm is based on is a novel efficient algorithm for solving the parametric cut problem using a maximum flow oracle. We give an overview of our reduction and its additional applications in Section 3, and describe it in detail in Appendix C.

1.2. Previous Work

Related Works on Submodular Minimization Submodular function minimization is a classical problem in combinatorial optimization, which goes back to the seminal work of Edmonds (Edmonds, 1970). The first polynomial-time algorithm was obtained by Grötschel et al. (Grötschel et al., 1981) using the ellipsoid method. This was followed by a plethora of improvements, among which the more recent ones (Dadush et al., 2018; Jiang et al., 2020) leveraged related techniques. On a different front, there has been significant work dedicated to obtaining strongly polynomial time algorithms for this problem (Fleischer & Iwata, 2003; Iwata, 2003; Iwata et al., 2001; Orlin, 2009; Schrijver, 2000; Lee et al., 2015; Dadush et al., 2018; Jiang et al., 2020).

For the more structured regime of *decomposable submodular function minimization*, algorithms based on both discrete and continuous methods have been developed. Kolmogorov (Kolmogorov, 2012) has shown that this problem reduces to computing maximum submodular flows, and gave an algorithm for this problem based on augmenting paths. This was followed by further algorithms based on discrete methods (Arora et al., 2012; Fix et al., 2013). The continuous methods are based on convex optimization on the submodular base polytope, which is also used here. Notably, Stobbe and Krause (Stobbe & Krause, 2010) tackled this problem using gradient descent, Nishihara et al. (Nishihara et al., 2014) used alternating projections to obtain an algorithm with linear convergence, Ene and Nguyen (Ene & Nguyen, 2015) achieved an improved algorithm with a linear convergence rate based on accelerated coordinate descent, while Ene et al. provided further improvements both via gradient descent and combinatorial techniques (Ene et al., 2017).

Related Works on Parametric Min Cut The seminal work of Gallo et al. (Gallo et al., 1989) studied the generalization of the maximum flow problem where some edge-capacities, instead of being fixed, are allowed to be (possibly different) monotonic functions of a single parameter. They showed how to modify certain versions of the push-relabel algorithm for ordinary maximum flow to the parametric problem with the same asymptotic time complexity. In particular, using the Goldberg-Tarjan max-flow algorithm (Goldberg & Tarjan, 1988) they gave an $O(nm\log(n^2/m))$ time bound for the parametric version. Their algorithm can compute the min cuts either when a set of parameter values are given (Gusfield & Martel, 1992) or the capacity functions are all affine functions of the parameter λ.

Several other max-flow algorithms were also shown to fit into their framework (see e.g., (Granot et al., 2012)) though all requiring $\Omega(mn)$ time in the worst case. Further generalizations of the parametric min-cut problems have also been considered (McCormick, 1999; Granot et al., 2012). When all parameterized capacities are equal to the parameter λ, Tarjan et al. (Tarjan et al., 2006) give a divide and conquer approach that can use any maximum flow algorithm as a black box, but is slower by a factor of $\min\{n, \log(nU)\}$.

[2] We emphasize the regime of $O(1)$ sized supports because it covers MAP inference with k-order potentials, for small k, which appears in image segmentation tasks, such as those described in (Ene & Nguyen, 2015; Ene et al., 2017). The latter explicitly presents the running time of their combinatorial algorithm in terms of the maximum support size. Our algorithm yields an improvement for moderately small values of $\max_i |V_i|$ (roughly at most $n^{1/3}$), and this situation can improve if faster maximum flow algorithms are developed.

2. Background and Preliminaries

2.1. Notation

We let $[n] \overset{\text{def}}{=} \{1, \ldots, n\}$. We write $\| \cdot \|_p$ for the ℓ_p norm, i.e. $\|x\|_p = \left(\sum_i |x_i|^p\right)^{1/p}$, with $\|x\|_\infty = \max_i |x_i|$.

2.2. Submodular Set Functions and Convex Analysis

Let V be a finite ground set of size n, and we assume w.l.o.g. that $V = \{1, \ldots, n\}$. A set function $F : 2^V \to \mathbb{R}$ is submodular if $F(A) + F(B) \geq F(A \cup B) + F(A \cap B)$ for any two sets $A, B \subseteq V$. We are concerned with minimizing submodular set functions of the form $F = \sum_{i=1}^r F_i$, where each F_i is a submodular set function:

$$\min_{A \subseteq V} F(A) = \min_{A \subseteq V} \sum_{i=1}^{r} F_i(A) \ .$$

For the rest of the paper we will assume that F_i are non-negative, integral, and that $\max_{S \subseteq V} F(S) \leq F_{\max}$. The non-negativity constraint holds without loss of generality, as we can simply shift each F_i by a constant until it becomes non-negative. For rational functions that are represented on bounded bit precision, the integrality can be enforced simply by scaling them, at the expense of increasing $F_{\max}$. As we will see, some of our subroutines depend on the magnitude of $F_{\max}$, so we will generally assume that this is polynomially bounded.

As in previous works (Ene et al., 2017; Li & Milenkovic, 2018), in this paper we are concerned with the regime where each function F_i in the decomposition acts on few elements of the ground set V.[3] More precisely, for each $i \in \{1, \ldots, r\}$ there is a *small* set $V_i \subseteq V$ such that $F_i(A) = F_i(A \cap V_i)$ for all $A \subseteq V$. We assume w.l.o.g. that $F_i(\emptyset) = F_i(V_i)$, which we discuss in more detail in Section D. The running time of our algorithm depends on $\max_{1 \leq i \leq r} |V_i|$. This assumption is important as, furthermore, the final running time of our algorithm depends on (i) the time $\mathcal{O}_i$ to optimize functions of the form $F_i(S) + w(S)$ over V_i, where w is a linear function and (ii) the time EO_i to evaluate F_i for subsets of V_i. In the case where $|V_i| = O(1)$, this is also constant time.

Given an arbitrary vector $w \in \mathbb{R}^n$ and a subset $A \subseteq V$, we use the notation $w(A) = \sum_{i \in A} w_i$.

Definition 2.1. *Given a submodular set function $F : 2^V \to \mathbb{R}$, such that $F(\emptyset) = 0$, its submodular base polytope $B(F)$ is defined as follows:*

$$B(F) = \{w \in \mathbb{R}^n : w(A) \leq F(A) \text{ for all } A \subseteq V, \\ w(V) = F(V)\} \ .$$

Definition 2.2. *Given a submodular set function $F : 2^V \to \mathbb{R}$, $F(\emptyset) = 0$, its Lovász extension $f : \mathbb{R}^n \to \mathbb{R}$ is defined over $[0, 1]^n$ as the convex closure of F. However, it will be more convenient to consider its extension of $\mathbb{R}^n$, given by*

$$f(x) = \int_0^\infty F(\{i : x_i \geq t\})\, dt \\ + \int_{-\infty}^0 \left(F(\{i : x_i \geq t\}) - F(V)\right) dt \ .$$

Fact 2.3. *It is well known (Bach, 2011) that the Lovász extension of a submodular set function F can be equivalently characterized in terms of its submodular base polytope $B(F)$. More precisely, if $F(\emptyset) = 0$, then:*

$$f(x) = \max_{w \in B(F)} \langle w, x \rangle \ .$$

For parametric submodular function minimization we consider a family of functions parameterized by $\alpha \in \mathbb{R}$:

$$F_\alpha(A) = F(A) + \sum_{i \in A} \psi_i'(\alpha) \ , \tag{1}$$

where $\psi_j : \mathbb{R} \to \mathbb{R}$ are strictly convex differentiable functions, satisfying $\lim_{\alpha \to -\infty} \psi_i'(\alpha) = -\infty$ and $\lim_{\alpha \to \infty} \psi_i'(\alpha) = \infty$, for all i. A common example is $\psi_j'(\alpha) = \alpha$, which imposes an ℓ_1 penalty on the size of the set A. It is shown in (Chambolle & Darbon, 2009; Bach, 2011) that minimizing $F_\alpha(A)$ for the entire range of scalars α amounts to minimizing a regularized version of the Lovász extension.

Lemma 2.4. *Let F_α be the family of parameterized submodular set functions defined as in (1), where ψ_i are strictly convex functions. Let f be the Lovász extension of F, and consider the optimization problem*

$$\min_{x \in \mathbb{R}^n} f(x) + \sum_{i \in V} \psi_i(x_i) \ . \tag{2}$$

Let $A^\alpha = \arg\min_{A \subseteq V} F_\alpha(A)$, and let x^* be the minimizer of (2). Then

$$A^\alpha = \{i : x_i^* \geq \alpha\} \ . \tag{3}$$

For completeness we reproduce the proof of Lemma 2.4 in Section B.

Via convex duality one can prove that minimizing (2) is equivalent to a dual optimization problem on the submodular base polytope $B(F)$:

$$\min_{w \in B(F)} \sum_{i \in V} \psi_i^*(-w_i) \, , \tag{4}$$

where ψ_i^* is the *Fenchel dual* of ψ_i.

[3] To be more specific, many previous works (Nishihara et al., 2014; Ene & Nguyen, 2015; Ene et al., 2017; Li & Milenkovic, 2018; Kumar et al., 2019) only assumed that each F_i has access to an optimization oracle. (Ene et al., 2017) used the additional assumption of limited support to obtain an improved combinatorial algorithm. Similarly, (Li & Milenkovic, 2018) used this assumption to provide a more efficient continuous method.

Definition 2.5 (Fenchel dual)**.** *Let $g : \mathbb{R}^n \to \mathbb{R} \cup \{-\infty, +\infty\}$ be a convex function. Its Fenchel dual or convex conjugate $g^* : \mathbb{R}^n \to \mathbb{R} \cup \{-\infty, +\infty\}$ is defined as*

$$g^*(w) = \sup_x \langle w, x\rangle - g(x)\,.$$

We will refer to (2) as the primal problem and (4) as the dual problem. The algorithm described in this paper will focus on optimizing (4) while strongly leveraging the decomposable structure of F. We assume that all functions ψ_i have "nice" second derivatives, which will play an important role in the algorithm, since this will also ensure that the minimizers of (2) and (4) are unique.

Assumption 2.6. *The function ψ_i is L-smooth and σ-strongly convex for all $i \in V$. Equivalently for each i, its second derivative satisfies $0 < \sigma \le \psi_i''(x) \le L$, for all $x \in \mathbb{R}$. Furthermore, $|\psi_i'(0)| \le (n+r)^{O(1)}$, for all $i \in V$.*

This condition also helps us ensure that we can efficiently convert between the primal and dual spaces onto which the ψ_i and its Fenchel dual ψ_i^* act. Also, whenever it is convenient, we will use the notation $\psi(x) = \sum_{i\in V} \psi_i(x_i)$, $\psi^*(y) = \sum_{i\in V} \psi_i^*(y_i)$. A relevant example to consider is $\psi_i(x) = x_i^2/2$, which corresponds to the standard parametric problem of minimizing $F(S) + \alpha \cdot |S|$ for all values of $\alpha > 0$.

2.3. Overview of Approach

Decomposable Submodular Minimization Here we provide an overview of our approach for minimizing decomposable submodular functions. Our approach for the parametric setting yields a strictly stronger result without sacrificing running time, so we will focus on this more general problem.

Our approach is based on minimizing a convex function on the submodular base polytope $B(F)$. As it has been seen in previous works (Bach, 2011), in order to solve the parametric problem (1), it suffices to solve the dual problem (4), which is a convex optimization problem over $B(F)$. For convenience let us denote by $h(w) = \sum_{i\in V} \psi_i^*(-w_i)$, so that our objective becomes computing $\min_{w\in B(F)} h(w)$.

We use an iterative method, which maintains a point $w \in B(F)$ and updates it in such a way that the objective value improves significantly in each step. To do so, we find a polytope P such that

$$w + \frac{1}{\alpha}\cdot P \subseteq B(F) \subseteq w + P \tag{5}$$

and such that we can efficiently minimize ψ over $w + \frac{1}{\alpha}\cdot P$. If we can find the minimizer w' over $w + \frac{1}{\alpha}\cdot P$, then moving our iterate to w' also guarantees that

$$h(w') - h(w^*) \le \left(1 - \frac{1}{\alpha}\right)(h(w) - h(w^*))\,,$$

where w^* is the minimizer of h over $B(F)$. This is true due to the convexity of h. Indeed, let $\widetilde{w} = w + t(w^* - w)$ where $t = \max\{t \le 1 : w + t(w^* - w) \in w + \frac{1}{\alpha}P\}$; in other words $\widetilde{w}$ represents the furthest point on the segment connecting w and w^* such that $\widetilde{w}$ still lies inside the small polytope $w + \frac{1}{\alpha}P$. Due to the sandwiching property of the polytopes (5), we have that $t \ge 1/\alpha$. Hence, using the convexity of h, we obtain that

$$\begin{aligned}&h(\widetilde{w}) - h(w^*) = h(w + t(w^* - w)) - h(w^*)\\ &\le (1-t)\,(h(w) - h(w^*)) \le (1 - 1/\alpha)(h(w) - h(w^*))\,.\end{aligned}$$

Since w' minimizes h over $w + \frac{1}{\alpha}\cdot B(F)$, we must have $h(w') \le h(\widetilde{w})$, and we obtain the desired progress in function value. Thus iterating $\widetilde{O}(\alpha)$ times we obtain a high precision solution, which we then convert back to a combinatorial solution to the original problem using some careful error analysis.

More importantly, we need to address the question of finding a polytope P satisfying (5).

To do so, for each i, we define the "residual" submodular functions $F_i'(A) = F_i(A) - w_i(A)$ for all $A \subseteq V$, where $w_i \in B(F_i)$ such that $\sum_{i=1}^r w_i = w$. The existence of such a decomposition of $w \in B(\sum_{i=1}^r F_i)$ is well-known, and goes back to Edmonds (Edmonds, 1971). Very importantly, we note that since F_i were non-negative, F_i' remain non-negative submodular set functions.

It is known (Devanur et al., 2013) that non-negative submodular set functions can be approximated by graph cuts. Following the proof from (Devanur et al., 2013), for each i we construct a graph on $O(|V_i|)$ vertices whose cuts approximate the value of F_i' within a factor of $O(|V_i|^2)$. Combining all these graphs into a single one, we obtain a weighted directed graph on $O(|V|)$ vertices and $O(|V| + \sum_{i=1}^r |V_i|^2)$ arcs such that its cut function G approximates F' within a factor of $O(\max_i |V_i|^2)$.

Crucially, we can show that if G is the cut function which approximates F', then we also have that

$$\frac{1}{\max_i |V_i|^2}\cdot B(G) \subseteq B(F_i') \subseteq B(G)\,,$$

and therefore it suffices to implement a routine that minimizes h over $w + \frac{1}{\max_i |V_i|^2}\cdot B(G)$ in order to obtain an algorithm that terminates in $\widetilde{O}(\max_i |V_i|^2)$ such iterations.

To implement this routine, we devise a new combinatorial algorithm for solving the parametric flow problem, with general parameterized capacities. By comparison to previous literature, our algorithm efficiently leverages a maximum flow oracle on a sequence of graphs obtained via contracting edges, and whose running time is up to polylogarithmic factors equal to that of computing a maximum flow in a capacitated directed graph with $O(|V|)$ vertices and $O(|V|+\sum_{i=1}^{r}|V_i|^2)$ arcs.

Following this, we convert the combinatorial solution to the parametric flow problem into a solution to its corresponding dual problem on the submodular base polytope, which returns the new iterate w'.

Throughout the algorithm we need to control the errors introduced by the fact that both the solution we receive for the parametric flow problem and the one we return as an approximate minimizer of h over $B(F)$ are *approximate*, but these are easily tolerable since our main routines return high precision solutions.

3. Parametric Min s,t-Cut

In the general parametric min s,t-cut problem (Gallo et al., 1989), the capacities of the source's outgoing edges (s,v) are (possibly different) nonnegative real nondecreasing functions of a parameter $\lambda \in \mathbb{D}$, where $\mathbb{D} \subseteq \mathbb{R}$ is some domain, whereas the capacities of the sink's incoming edges vt are nonincreasing functions of λ. The goal is to compute the representation of the *cut function* $\kappa : \mathbb{D} \to \mathbb{R}$ such that $\kappa(\lambda)$ equals the capacity of the minimum s,t-cut in G_λ obtained from G by evaluating the parameterized capacity functions at λ. It is known that κ consists of $O(n)$ pieces, where each piece equals the parameterized capacity of some fixed cut in G as a function of λ.

More formally, let $\lambda_{\min} \in \mathbb{D}$ be such that the minimal min s,t-cuts of $G_{\lambda_{\min}}$ and $G_{\lambda'}$ are equal for all $\lambda' \in \mathbb{D}$, $\lambda' < \lambda_{\min}$. Similarly, let $\lambda_{\max} \in \mathbb{D}$ be such that the minimal min s,t-cuts of $G_{\lambda_{\max}}$ and $G_{\lambda'}$ are equal for all $\lambda' \in \mathbb{D}$ with $\lambda' > \lambda_{\max}$. We will consider $\lambda_{\min}$ and $\lambda_{\max}$ inputs to our problem. Then, there exist $O(n)$ *breakpoints* $\Lambda = \{\lambda_1, \dots, \lambda_k\}$, $\lambda_{\min} = \lambda_0 < \lambda_1 < \dots < \lambda_k$ and an embedding of vertices $\tau : V \to \Lambda \cup \{\lambda_{\min}, \infty\}$ such that for all $i = 0, \dots, k-1$, $\lambda' \in [\lambda_i, \lambda_{i+1}) \cap \mathbb{D}$, $\kappa(\lambda')$ equals the capacity of the cut $S(\lambda_i) = \{v \in V : \tau(v) \le \lambda_i\}$ in $G_{\lambda'}$, and also $S(\lambda_k)$ is a min s,t-cut of $G_{\lambda_{\max}}$.

Motivated by our submodular minimization application, our algorithm in the most general setting solves the ε-approximate parametric min s,t-cut problem.

Definition 3.1 (ε-approximate parametric min s,t-cut). *Let Λ, τ, and $S : \mathbb{D} \to 2^V$ be as defined above. A pair (Λ, τ) is called an ε-approximate parametric min s,t-cut of G if:*

1. *For $i = 0, \dots, k-1$, $S(\lambda_i)$ is a min s,t-cut of $G_{\lambda'}$ for all $\lambda' \in [\lambda_i, \lambda_{i+1} - \varepsilon) \cap \mathbb{D}$.*
2. *$S(\lambda_k)$ is a min s,t-cut of $G_{\lambda_{\max}}$.*
3. *For $i = 0, \dots, k-1$, $S(\lambda_i) \subsetneq S(\lambda_{i+1})$.*

We prove that an ε-approximate parametric min s,t-cut yields breakpoints within ε additive error wrt. to the breakpoints of the exact parametric min s,t-cut. Our algorithm solves the above problem assuming only constant-time black-box access to the capacity functions.

Theorem 3.2. *Let $R = \lambda_{\max} - \lambda_{\min}$ be an integral multiple of $\varepsilon > 0$. Let $T_{\text{maxflow}}(n', m') = \Omega(m' + n')$ be a convex function bounding the time needed to compute maximum flow in a graph with n' vertices and m' edges obtained from G_λ by edge/vertex deletions and/or edge contractions (with merging parallel edges by summing their capacities) for any $\lambda = \lambda_{\min} + \ell\varepsilon$ and any integer $\ell \in [0, R/\varepsilon]$. Then, ε-approximate parametric min s,t-cut in G can be computed in $O(T_{\text{maxflow}}(n, m \log n) \cdot \log \frac{R}{\varepsilon} \cdot \log n)$ time.*

The algorithm is recursive. In order to ensure uniqueness of the minimum cuts considered, it always computes cuts with *minimal* s-side.

Roughly speaking, given initial guesses $\lambda_{\min}, \lambda_{\max}$ the algorithm finds, using $O(\log((\lambda_{\max} - \lambda_{\min})/\varepsilon))$ maximum flow computations, the most balanced split λ_1, λ_2 of the domain such that (1) the s-sides for all the min-cuts of $G_{\lambda'}$ for $\lambda' > \lambda_2$ have size at least $n/2$, (2) the t-sides of all the min-cuts of $G_{\lambda'}$ for $\lambda' < \lambda_1$ have size at least $n/2$, (3) $\lambda_2 - \lambda_1 \le \varepsilon$. We then recurse on the intervals $[\lambda_{\min}, \lambda_1]$ and $[\lambda_2, \lambda_{\max}]$ on *minors* of G with at least $n/2$ vertices contracted. Even though the contraction requires merging parallel edges in order to have at most $m+n$ (as opposed to $2m$) edges in the recursive calls, we are able to guarantee that the capacity functions in the recursive calls are all obtained by shifting the original capacity functions by a real number, and thus can be evaluated in constant time as well. Since the number of vertices decreases by a factor of two in every recursive call, one can prove that for each level of the recursion tree, the sum of numbers of vertices in the calls at that level is $O(n)$, whereas the sum of sizes of edge sets is $O(m + n \log n) = O(m \log n)$.

We show that the ε-approximate algorithm can be turned into an exact algorithm in two important special cases. First of all, if the capacity functions are low-degree (say, at most 4) polynomials with *integer coefficients* in $[-U, U]$, then one can compute parametric min s,t-cut only $O(\text{polylog}\{n, U\})$ factors slower than best known max-flow algorithm for integer capacities (Gao et al., 2021; van den Brand et al., 2021; Goldberg & Rao, 1998). Second, we can solve the *discrete domain* case, i.e., when $\mathbb{D}$ has finite size ℓ with only $O(\text{polylog}\{n, \ell\})$ multiplicative overhead wrt. the respective maximum flow algorithm.

Moreover, since our reduction runs maximum flow computations only on *minors* of the input graph G, it also yields very efficient parametric min s,t-cut algorithms for planar graphs. In particular, since near-optimal strongly-polynomial s,t-max flow algorithms for planar graphs are known (Borradaile & Klein, 2009; Erickson, 2010), we obtain near-optimal algorithms for the integer polynomial capacity functions (as above) and discrete domains. What is perhaps more surprising, using our reduction we can even obtain a *strongly polynomial* exact parametric min s,t-cut algorithm for planar graphs with linear capacity functions with real coefficients. The algorithm runs in $\tilde{O}(n^{1.21875})$ time and constitutes the only known subquadratic strongly polynomial parametric min-s,t-cut algorithm.

The details of our parametric min s,t-cut algorithm and its applications are covered in Appendix C. It should be noted that the idea of using cuts contraction is not new and appeared previously in (Tarjan et al., 2006). Compared to (Tarjan et al., 2006), our reduction provably handles more general parameterized capacity functions.[4] As it does not operate on any auxiliary networks that may not preserve structural properties of G, but merely on minors of G, it proves much more robust in important special cases such as planar graphs. Finally, we believe that our reduction is also more natural and operates on the breakpoints of the cut function directly, whereas the reduction of (Tarjan et al., 2006) operates on so-called balanced flows.

4. Parametric Decomposable Submodular Minimization via Base Polytope Approximations

4.1. Algorithm Overview

In Algorithm 3 we give the description of our main routine.

4.2. Removing Assumptions

In Section 2 we assumed that for all i, $F_i(\emptyset) = F_i(V_i) = 0$ and $F_i(S) \geq 0$ for all S. These assumptions hold without loss of generality. In Section D we show how to preprocess the input such that these assumptions are valid.

4.3. From Parametric Minimum Cut to Cut Base Polytope Optimization

In this section, we will focus on the problem of minimizing a convex function over the base polytope of the s,t-cut function of a graph $G(V \cup \{s,t\}, E, c)$. We define the s,t-cut function $G : 2^V \to \mathbb{R}$ as $G(S) = c^+(S \cup \{s\})$ for $S \subseteq V$, and let $B(G)$ be the base polytope of G. Now, the parametric min s,t-cut problem can be written as

$$\min_{S \subseteq V} G(S) + \sum_{u \in S} \phi'_u(\lambda)\,, \tag{6}$$

where the capacity of an edge (u,t) at time λ is $c_{ut} + \phi'_u(\lambda)$ and ϕ is a function satisfying Assumption 2.6. In particular, our goal in this section is, given solutions to (6) for all λ, to solve the following dual problem:

$$\min_{w \in B(G)} \phi^*(-w)\,. \tag{7}$$

Algorithm 1 Parametric Decomposable Submodular Function Minimization

```
Input: $\epsilon$: error tolerance // Returns $\epsilon$-optimal solution of $\min_{w \in B(F)} \psi^*(-w)$
Set $w^{0,i} = 0$ for $i \in [r]$ // Initialize a feasible solution
Set $T = \alpha \log \frac{\psi^*(-w^0)+\psi(0)}{\varepsilon}$
for $t = 1 \ldots T$ do
    Set $G_i(V_i, E_i, c_i) = \textsc{GraphApprox}(w^{t-1,i})$, for $i \in [r]$, and construct $G(V,E,c)$ by combining the graphs $G_i$
    Set $\phi(x) := \psi(x) + \langle \sum_{i=1}^{r} w^{t-1,i}, x \rangle$
    Set $\widetilde{w} = \textsc{FindMinCuts}(G(V,E,c), \phi, \frac{1}{3L})$ // Find parametric min $s,t$-cuts
    Round all the entries of $\widetilde{w}$ to the nearest integer
    Decompose $\widetilde{w} = \sum_{i=1}^{r} \widetilde{w}^i$, with $\widetilde{w}_i \in B(G_i)$ using Lemma 4.3
    Set $w^{t,i} = w^{t-1,i} + \widetilde{w}^i$, for all $i \in [r]$
end for

return $\sum_{i=1}^{r} w^{T,i}$
```

Definition 4.1 (W-restricted function)**.** *A submodular function $F : 2^V \to \mathbb{R}_{\geq 0}$ is called W-restricted if $F(S) = F(S \cap W)$ for all $S \subseteq V$, where $W \subseteq V$.*

As the cut functions $G(S)$ that we will be concerned with will be decomposable, i.e. $G(S) = \sum_{i=1}^{r} G_i(S)$ for all $S \subseteq V$, we introduce the following notion of a *decomposition* of some $w \in B(G)$ into a sum of $w^i \in B(G_i)$.

Definition 4.2 (F-decomposition)**.** *Let $F : 2^V \to \mathbb{R}_{\geq 0}$ with $|V| = n$ be a submodular function that is decomposable, i.e. $F(S) = \sum_{i=1}^{r} F_i(S)$ for all $S \subseteq V$, where $F_i : 2^V \to \mathbb{R}_{\geq 0}$ are submodular functions. Then for any $w \in B(F)$ there exist vectors $w^1, \ldots, w^r \in \mathbb{R}^n$, where $w^i \in B(F_i)$ for all $i \in [r]$ and $\sum_{i=1}^{r} w^i = w$. We call the sequence of vectors $w^1, \ldots, w^r$ an $(F_1, \ldots, F_r)$-decomposition of w, or just an F-decomposition of w if the F_i's are clear from context.*

What follows is the main lemma of this section, whose full proof appears in Appendix E.1.

Lemma 4.3 (From parametric min-cut to cut base polytope

[4] (Tarjan et al., 2006) only claim a $O(\min(n, \log(nU)))$ multiplicative overhead when all the parametric capacities are *equal* to λ. Otherwise, it does not claim any bound, and when capacities are arbitrary linear functions their algorithm might pay $O(n)$ overhead in the worst case.

optimization)**.** *Consider a graph $G(V, E, c \geq 0)$ and a function $\phi(x) = \sum_{u \in V} \phi_u(x_u)$ that satisfies Assumption 2.6. Additionally, let $G(S) = c^+(S)$ for all $S \subseteq V$ be the cut function associated with the graph, and suppose it is decomposable as $G(S) = \sum_{i=1}^{r} G_i(S)$ where $G_i : 2^V \to \mathbb{Z}_{\geq 0}$ are V_i-restricted cut functions defined as $G_i(S) = c^{i+}(S)$ that correspond to graphs $G_i(V, E, c^i \geq 0)$, and $c = \sum_{i=1}^{r} c^i$.*

We define an extended vertex set $V' = V \cup \{s, t\}$ with edge set $E' = E \cup \bigcup_{u \in V} (s, u) \cup \bigcup_{u \in V} (u, t)$, the parametric capacity of an edge $(u, v) \in E'$ as

$$c_\lambda(u, v) = \begin{cases} \max\{0, \phi'_u(-\lambda)\} & \text{if } u \in V, v = t \\ \max\{0, -\phi'_u(-\lambda)\} & \text{if } u = s, v \in V \\ c_{uv} & \text{otherwise} \end{cases}$$

and let (Λ, τ) be a $\frac{1}{3L}$-approximate parametric min s, t-cut of $G'(V', E', c_\lambda)$.

There exists an algorithm that, given (Λ, τ), outputs $\widetilde{w}^* = \operatorname{argmin}_{w \in B(G)} \phi^*(-w)$ and a G-decomposition $\widetilde{w}^{*1}, \ldots, \widetilde{w}^{*r}$ of $\widetilde{w}^*$. The running time of this algorithm is $O\left(n + \sum_{i=1}^{r} |V_i|^2\right)$.

4.4. Dual Progress Analysis in One Step

The following lemma states that there is a step that improves the dual function value by at least an α fraction of the distance to the optimal value, where α is the approximation factor we lose when approximating a submodular function by a graph.

Lemma 4.4 (Dual progress in one step)**.** *Let $F : 2^V \to \mathbb{Z}_{\geq 0}$ be a submodular function that is separable, i.e. $F(S) = \sum_{i=1}^{r} F_i(S)$ for all $S \subseteq V$, where $F_i : 2^V \to \mathbb{Z}_{\geq 0}$ are V_i-restricted submodular functions with $F_i(\emptyset) = 0$. Additionally, let $\psi : \mathbb{R}^n \to \mathbb{R}$ be a function that satisfies Assumption 2.6, where $|V| = n$.*

Given a feasible dual solution $w \in B(F)$ and an F-decomposition $w^1, \ldots, w^r \in \mathbb{Z}^n$ of w, there is an algorithm that outputs a vector $w' \in \mathbb{Z}^n$, along with an $(F_1, \ldots, F_r)$-decomposition $w'^1, \ldots, w'^r \in \mathbb{Z}^n$ of w', such that

$$\psi^*(-w') - \psi^*(-w^*) \leq \left(1 - \frac{1}{\alpha}\right) (\psi^*(-w) - \psi^*(-w^*))$$

where $\alpha = \max_{i \in [r]} \{|V_i|^2/4\}$ and $w^* = \operatorname{argmin}_{w^* \in B(F)} \psi^*(-w^*)$ is the dual optimum. The running time of the algorithm is

$$\widetilde{O}\left(\sum_{i=1}^{r} |V_i|^2 \mathcal{O}_i + T_{\text{maxflow}}\left(n, n + \sum_{i=1}^{r} |V_i|^2\right)\right).$$

The full proof of Lemma 4.4 appears in Appendix E.2.

4.5. Main Theorem

Proof of Theorem 1.1. We repeatedly apply Lemma 4.4, and let $w^0, \ldots, w^T$ be the iterates after $T = \alpha \log \frac{\psi^*(-w^0) - \psi^*(-w^*)}{\zeta}$ iterations. We have

$$\begin{aligned} &\psi^*(-w^T) - \psi^*(-w^*) \\ &\leq \left(1 - \frac{1}{\alpha}\right) \left(\psi^*(-w^{T-1}) - \psi^*(-w^*)\right). \end{aligned}$$

Applying induction over T steps we obtain that

$$\begin{aligned} &\psi^*(-w^T) - \psi^*(-w^*) \\ &\leq \left(1 - \frac{1}{\alpha}\right)^T \left(\psi^*(-w^0) - \psi^*(-w^*)\right) \\ &\leq e^{-T/\alpha}(\psi^*(-w^0) - \psi^*(-w^*)) \leq \zeta. \end{aligned}$$

We have obtained a high precision solution to the objective (4). Finally, setting $\frac{1}{\zeta} = \text{poly}(\frac{L}{\sigma} n F_{\max}/\varepsilon) = (n + r)^{O(1)}/\varepsilon$ and applying Corollary B.4 to convert from this solution to the actual sets, we obtain the desired solution.

As Assumption 2.6 implies $\psi^*(-w^0) - \psi^*(-w^*) = (n + r)^{O(1)}$, the total running time is

$$\widetilde{O}\Bigg(\max_i |V_i|^2 \Bigg(\sum_{i=1}^{r} |V_i|^2 \mathcal{O}_i + T_{\text{maxflow}}\left(n, n + \sum_{i=1}^{r} |V_i|^2\right)\Bigg) \log \frac{1}{\epsilon}\Bigg).$$

□

Acknowledgements

KA was supported in part by the NSF grants CCF-1553428 and CNS-1815221. AK was supported by ERC Consolidator Grant 772346 TUgbOAT and by the Foundation for Polish Science (FNP) via the START programme. AM and PS were supported in part by ERC consolidator grant TUgbOAT no 772346 and NCN no 2020/37/B/ST6/04179. We are grateful to Michael B. Cohen for discussions on the possibility of using second order methods for submodular minimization, which motivated parts of this project. We thank Alina Ene for pointing us to the relevant material from (Bach, 2011).

References

Agarwal, P. K., Sharir, M., and Toledo, S. Applications of parametric searching in geometric optimization. *J. Algorithms*, 17(3):292–318, 1994. doi: 10.1006/jagm.1994.1038. URL https://doi.org/10.1006/jagm.1994.1038.

Arora, C., Banerjee, S., Kalra, P., and Maheshwari, S. Generic cuts: An efficient algorithm for optimal inference in higher order mrf-map. In *European Conference on Computer Vision*, pp. 17–30. Springer, 2012.

Bach, F. Learning with submodular functions: A convex optimization perspective. *arXiv preprint arXiv:1111.6453*, 2011.

Bern, M., Gilbert, J. R., Hendrickson, B., Nguyen, N., and Toledo, S. Support-graph preconditioners. *SIAM Journal on Matrix Analysis and Applications*, 27(4):930–951, 2006.

Borradaile, G. and Klein, P. N. An $O(n \log n)$ algorithm for maximum *st*-flow in a directed planar graph. *J. ACM*, 56(2):9:1–9:30, 2009. doi: 10.1145/1502793.1502798. URL https://doi.org/10.1145/1502793.1502798.

Borwein, J. and Lewis, A. S. *Convex analysis and nonlinear optimization: theory and examples*. Springer Science & Business Media, 2010.

Brand, J. v. d., Lee, Y. T., Liu, Y. P., Saranurak, T., Sidford, A., Song, Z., and Wang, D. Minimum cost flows, mdps, and ℓ_1-regression in nearly linear time for dense instances. *arXiv preprint arXiv:2101.05719*, 2021.

Chakrabarty, D., Lee, Y. T., Sidford, A., and Wong, S. C.-w. Subquadratic submodular function minimization. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pp. 1220–1231, 2017.

Chambolle, A. and Darbon, J. On total variation minimization and surface evolution using parametric maximum flows. *International journal of computer vision*, 84(3):288, 2009.

Dadush, D., Végh, L. A., and Zambelli, G. Geometric rescaling algorithms for submodular function minimization. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 832–848. SIAM, 2018.

Devanur, N. R., Dughmi, S., Schwartz, R., Sharma, A., and Singh, M. On the approximation of submodular functions. *arXiv preprint arXiv:1304.4948*, 2013.

Edmonds, J. Submodular functions, matroids, and certain polyhedra. In *Combinatorial Structures and Their Applications*, pp. 69–87. 1970.

Edmonds, J. Matroids and the greedy algorithm. *Mathematical programming*, 1(1):127–136, 1971.

Ene, A. and Nguyen, H. Random coordinate descent methods for minimizing decomposable submodular functions. In *International Conference on Machine Learning*, pp. 787–795. PMLR, 2015.

Ene, A., Nguyen, H. L., and Végh, L. A. Decomposable submodular function minimization: Discrete and continuous. In *NIPS*, 2017.

Erickson, J. Maximum flows and parametric shortest paths in planar graphs. In Charikar, M. (ed.), *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2010, Austin, Texas, USA, January 17-19, 2010*, pp. 794–804. SIAM, 2010. doi: 10.1137/1.9781611973075.65.

Fix, A., Joachims, T., Park, S. M., and Zabih, R. Structured learning of sum-of-submodular higher order energy functions. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3104–3111, 2013.

Fleischer, L. and Iwata, S. A push-relabel framework for submodular function minimization and applications to parametric optimization. *Discrete Applied Mathematics*, 131(2):311–322, 2003.

Ford, L. R. and Fulkerson, D. R. *Flows in Networks*. Princeton University Press, 1962. ISBN 9780691625393.

Fujishige, S. Lexicographically optimal base of a polymatroid with respect to a weight vector. *Mathematics of Operations Research*, 5(2):186–196, 1980.

Gallo, G., Grigoriadis, M. D., and Tarjan, R. E. A fast parametric maximum flow algorithm and applications. *SIAM J. Comput.*, 18(1):30–55, 1989. doi: 10.1137/0218003. URL https://doi.org/10.1137/0218003.

Gao, Y., Liu, Y. P., and Peng, R. Fully dynamic electrical flows: Sparse maxflow faster than goldberg-rao. *CoRR*, abs/2101.07233, 2021. URL https://arxiv.org/abs/2101.07233.

Goldberg, A. V. and Rao, S. Beyond the flow decomposition barrier. *Journal of the ACM (JACM)*, 45(5):783–797, 1998.

Goldberg, A. V. and Tarjan, R. E. A new approach to the maximum-flow problem. *Journal of the ACM (JACM)*, 35(4):921–940, 1988.

Granot, F., McCormick, S. T., Queyranne, M., and Tardella, F. Structural and algorithmic properties for parametric minimum cuts. *Math. Program.*, 135(1-2):337–367, 2012. doi: 10.1007/s10107-011-0463-1.

Grötschel, M., Lovász, L., and Schrijver, A. The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, 1(2):169–197, 1981.

Gusfield, D. and Martel, C. U. A fast algorithm for the generalized parametric minimum cut problem and applications. *Algorithmica*, 7(5&6):499–519, 1992. doi: 10.1007/BF01758775.

Iwata, S. A faster scaling algorithm for minimizing submodular functions. *SIAM Journal on Computing*, 32(4): 833–840, 2003.

Iwata, S., Fleischer, L., and Fujishige, S. A combinatorial strongly polynomial algorithm for minimizing submodular functions. *Journal of the ACM (JACM)*, 48(4): 761–777, 2001.

Jegelka, S., Bach, F., and Sra, S. Reflection methods for user-friendly submodular optimization. *arXiv preprint arXiv:1311.4296*, 2013.

Jiang, H., Lee, Y. T., Song, Z., and Wong, S. C.-w. An improved cutting plane method for convex optimization, convex-concave games, and its applications. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pp. 944–953, 2020.

Kakade, S. M., Shalev-Shwartz, S., and Tewari, A. Regularization techniques for learning with matrices. *The Journal of Machine Learning Research*, 13(1):1865–1890, 2012.

Karczmarz, A. and Sankowski, P. A deterministic parallel apsp algorithm and its applications. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 255–272. doi: 10.1137/1.9781611976465.17. URL https://epubs.siam.org/doi/abs/10.1137/1.9781611976465.17.

Kathuria, T., Liu, Y. P., and Sidford, A. Unit capacity maxflow in almost $o(m^{4/3})$ time. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 119–130. IEEE, 2020.

Kohli, P., Torr, P. H., et al. Robust higher order potentials for enforcing label consistency. *International Journal of Computer Vision*, 82(3):302–324, 2009.

Kolmogorov, V. Minimizing a sum of submodular functions. *Discrete Applied Mathematics*, 160(15):2246–2258, 2012.

Koutis, I., Miller, G. L., and Tolliver, D. Combinatorial preconditioners and multilevel solvers for problems in computer vision and image processing. In *International Symposium on Visual Computing*, pp. 1067–1078. Springer, 2009.

Kumar, K., Bach, F., and Pock, T. Fast decomposable submodular function minimization using constrained total variation. *arXiv preprint arXiv:1905.11327*, 2019.

Lee, Y. T., Sidford, A., and Wong, S. C.-w. A faster cutting plane method and its implications for combinatorial and convex optimization. In *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*, pp. 1049–1065. IEEE, 2015.

Li, P. and Milenkovic, O. Revisiting decomposable submodular function minimization with incidence relations. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pp. 2242–2252, 2018.

Lin, H. and Bilmes, J. Optimal selection of limited vocabulary speech corpora. In *Twelfth Annual Conference of the International Speech Communication Association*, 2011.

Liu, Y. P. and Sidford, A. Faster energy maximization for faster maximum flow. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pp. 803–814, 2020.

Lovász, L. Submodular functions and convexity. In *Mathematical programming the state of the art*, pp. 235–257. Springer, 1983.

Madry, A. Navigating central path with electrical flows: From flows to matchings, and back. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pp. 253–262. IEEE, 2013.

Madry, A. Computing maximum flow with augmenting electrical flows. In *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 593–602. IEEE, 2016.

Mahler, K. An inequality for the discriminant of a polynomial. *Michigan Math. J.*, 11(3):257–262, 09 1964. doi: 10.1307/mmj/1028999140. URL https://doi.org/10.1307/mmj/1028999140.

McCormick, S. T. Fast algorithms for parametric scheduling come from extensions to parametric maximum flow. *Oper. Res.*, 47(5):744–756, 1999. doi: 10.1287/opre.47.5.744.

Megiddo, N. Applying parallel computation algorithms in the design of serial algorithms. *J. ACM*, 30(4):852–865, 1983. doi: 10.1145/2157.322410. URL https://doi.org/10.1145/2157.322410.

Narasimhan, M. and Bilmes, J. A. Local search for balanced submodular clusterings. In *IJCAI*, pp. 981–986, 2007.

Nishihara, R., Jegelka, S., and Jordan, M. I. On the convergence rate of decomposable submodular function minimization. *Advances in Neural Information Processing Systems*, 27:640–648, 2014.

Orlin, J. B. A faster strongly polynomial time algorithm for submodular function minimization. *Mathematical Programming*, 118(2):237–251, 2009.

Rockafellar, R. T. *Convex analysis*, volume 36. Princeton University Press, 1970.

Schrijver, A. A combinatorial algorithm minimizing submodular functions in strongly polynomial time. *Journal of Combinatorial Theory, Series B*, 80(2):346–355, 2000.

Shalev-Shwartz, S. and Singer, Y. Convex repeated games and fenchel duality. In *NIPS*, volume 6, pp. 1265–1272, 2006.

Shanu, I., Arora, C., and Singla, P. Min norm point algorithm for higher order mrf-map inference. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5365–5374, 2016.

Spielman, D. A. and Teng, S.-H. Smoothed analysis of algorithms: Why the simplex algorithm usually takes polynomial time. *Journal of the ACM (JACM)*, 51(3): 385–463, 2004.

Stobbe, P. and Krause, A. Efficient minimization of decomposable submodular functions. *arXiv preprint arXiv:1010.5511*, 2010.

Tarjan, R. E., Ward, J., Zhang, B., Zhou, Y., and Mao, J. Balancing applied to maximum network flow problems. In *Algorithms - ESA 2006, 14th Annual European Symposium, Zurich, Switzerland, September 11-13, 2006, Proceedings*, pp. 612–623, 2006. doi: 10.1007/11841036_55. URL https://doi.org/10.1007/11841036_55.

Toledo, S. and Avron, H. Combinatorial preconditioners. *Combinatorial Scientific Computing*, pp. 69–93, 2010.

van den Brand, J., Lee, Y. T., Liu, Y. P., Saranurak, T., Sidford, A., Song, Z., and Wang, D. Minimum cost flows, mdps, and ℓ_1-regression in nearly linear time for dense instances. *CoRR*, abs/2101.05719, 2021. URL https://arxiv.org/abs/2101.05719.

Veldt, N., Benson, A. R., and Kleinberg, J. Minimizing localized ratio cut objectives in hypergraphs. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1708–1718, 2020.

Vicente, S., Kolmogorov, V., and Rother, C. Joint optimization of segmentation and appearance models. In *2009 IEEE 12th International Conference on Computer Vision*, pp. 755–762. IEEE, 2009.

Yap, C.-K. et al. *Fundamental problems of algorithmic algebra*, volume 49. Oxford University Press Oxford, 2000.