Approximating Submodular Matroid-Constrained Partitioning

Kristóf Bérczi[a,b,1,2], Tamás Király[a,b,2], Daniel P. Szabo[b,2,*], Karthekeyan Chandrasekaran[c,3]

[a] *HUN-REN–ELTE Egerváry Research Group on Combinatorial Optimization,*
[b] *Department of Operations Research, Eötvös Loránd University, Budapest, Hungary*
[c] *Grainger College of Engineering, University of Illinois, Urbana-Champaign, USA*

Abstract

The submodular partitioning problem asks to minimize, over all partitions $\mathcal{P}$ of a ground set V, the sum of a given submodular function f over the parts of $\mathcal{P}$. The problem has seen considerable work in approximability, as it encompasses multiterminal cuts on graphs, k-cuts on hypergraphs, and elementary linear algebra problems such as matrix multiway partitioning. This research has been divided between the fixed terminal setting, where we are given a set of terminals that must be separated by $\mathcal{P}$, and the global setting, where the only constraint is the size of the partition. We investigate a generalization that unifies these two settings: minimum submodular matroid-constrained partition. In this problem, we are additionally given a matroid over the ground set and seek to find a partition $\mathcal{P}$ in which *there exists* some basis that is separated by $\mathcal{P}$. We explore the approximability of this problem and its variants for general, symmetric, and monotone submodular functions.

* Corresponding Author

Email addresses: `kristof.berczi@ttk.elte.hu` (Kristóf Bérczi), `tamas.kiraly@ttk.elte.hu` (Tamás Király), `dszabo2@wisc.edu` (Daniel P. Szabo), `karthe@illinois.edu` (Karthekeyan Chandrasekaran)

[1] The work was supported by the Lendület Programme of the Hungarian Academy of Sciences – grant number LP2021-1/2021 and by Dynasnet European Research Council Synergy project – grant number ERC-2018-SYG 810115.

[2] This research has been supported by the Ministry of Innovation and Technology of Hungary from the National Research, Development and Innovation Fund, financed under the ELTE TKP 2021-NKTA-62 and ADVANCED 150556 funding schemes.

[3] This research has been supported in part by NSF grant CCF-2402667.

Keywords: Approximation algorithms, Matroid constraints, Multiway cut, Submodular partitioning

1. Introduction

Many motivations, as well as techniques, for submodular optimization stem from graph partitioning problems, making them a natural starting point. Given an undirected graph $G = (V, E)$, edge weights $w \in \mathbb{R}^E_+$, and a set $T \subseteq V$ of k terminals, the *Multiway Cut* (MULTIWAY-CUT) problem asks for a partition $\mathcal{P} = (V_1, \ldots, V_k)$ of V minimizing $\sum_{i=1}^k d_w(V_i)$ such that each part V_i contains exactly one terminal from T, where $d_w(S)$ is the weight of edges with exactly one end-vertex in S. A well-known special case is $k = 2$, which corresponds to the polynomial-time solvable *Min-(s,t)-cut* problem. For $k \geq 3$, the problem is NP-hard [8] but admits approximation algorithms [15]. In contrast, the *k-cut* problem (k-CUT) takes as input only the graph $G = (V, E)$, edge weights $w \in \mathbb{R}^E_+$, and an integer k, and asks for a partition $\mathcal{P} = (V_1, \ldots, V_k)$ of V into k non-empty parts minimizing $\sum_{i=1}^k d_w(V_i)$. This problem is polynomial-time solvable for every fixed k, but NP-hard when k is part of the input. In this case, a $(2-2/k)$-approximation is possible [14], and this approximation is tight under the Small Set Expansion Hypothesis [11].

These problems have been generalized to arbitrary submodular functions. 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 all $A, B \subseteq V$. Additionally, it is *symmetric* if $f(A) = f(V \setminus A)$ for all $A \subseteq V$, and *monotone* if $f(A) \leq f(B)$ for all $A \subseteq B \subseteq V$. For ease of discussion, we assume throughout that $f(\emptyset) = 0$. In the *Submodular k-partition Problem* (SUB-K-P), we are given a submodular function $f : 2^V \to \mathbb{R}_{\geq 0}$ and an integer k, and the goal is to find a partition $\mathcal{P} = \{V_1, \ldots, V_k\}$ of V into k non-empty parts minimizing $\sum_{i=1}^k f(V_i)$. We recall that the cut function of a graph is symmetric and submodular; hence, if f is the cut function of an undirected graph, then the problem is identical to k-CUT. SUB-K-P does not admit a constant-factor approximation assuming the Exponential Time Hypothesis (ETH) [7], and the best known approximation factor is $O(k)$ [17, 12]. However, for certain families of submodular functions, it admits constant factor approximations. For symmetric submodular functions (SYMSUB-K-P), the problem admits a tight $(2 - 2/k)$-approximation [17, 4, 13], while for monotone submodular functions (MONSUB-K-P), the problem admits a tight 4/3-approximation [4, 13].

In the *Submodular Multiway Partition* problem (SUB-MP), we are given a submodular function $f\colon 2^V \to \mathbb{R}_+$ and k terminals $t_1, \dots, t_k$, and the goal is to find a partition $\mathcal{P} = (V_1, \dots, V_k)$ with $t_i \in V_i$ for every $i \in [k]$ minimizing $\sum_{i=1}^k f(V_i)$. If f is the cut function of an undirected graph, then the problem is identical to MULTIWAY-CUT. SUB-MP admits a 2-approximation [5]. For symmetric submodular functions (SYMSUB-MP), the problem admits a 3/2-approximation [5], while for monotone submodular functions (MONSUB-MP), the problem admits a 4/3-approximation. An important special case of MONSUB-MP is when f is a *graph coverage function* (GCOV-MP), that is, $f(S) \coloneqq \sum_{uv\in E\colon \{u,v\}\cap S\neq\emptyset} w(uv)$ for every $S \subseteq V$. We note that the objective function of GCOV-MP differs from the objective function of MULTIWAY-CUT by an additive factor of $|E|$. Yet, GCOV-MP differs substantially from MULTIWAY-CUT in terms of approximability: GCOV-MP admits a 1.125-approximation [2], while the best known approximation for MULTIWAY-CUT is 1.2965 [15].

Algorithmic techniques for global and fixed terminal partitioning problems differ significantly: the key difference is that fixed terminal versions allow for linear programming techniques. On the other hand, approximation algorithms for global partitioning problems have relied on global structural aspects of submodular functions, such as Gomory-Hu trees, greedy splitting, and cheapest singleton algorithms. Nevertheless, the approximation results are not always so different. For example, both MONSUB-K-P and MONSUB-MP admit a 4/3-approximation. Previous work has also generalized these two regimes in some way, such as in [1] and [6]. One of the motivations of this work is to present a general model for partitioning problems that unifies the global and fixed-terminal variants, and to investigate approximation algorithms for this general model. The choice of encoding the terminal choices as the bases of some matroid is not without precedent: the same was effective for the minimum Steiner tree problem [3].

1.1. Results

We introduce a partitioning problem that unifies the global and the fixed terminal partitioning problems. The *Submodular Matroid-constrained Partitioning* problem (SUB-MCP) is defined as follows:

SUB-MCP

Input: A submodular function $f: 2^V \to \mathbb{R}_+$ given by a value oracle, and a rank-k matroid $\mathcal{M} = (V, \mathcal{B})$ given by an independence oracle, where $\mathcal{B}$ is the family of bases of $\mathcal{M}$.

Goal: Minimize $\sum_{i=1}^{k} f(V_i)$, where $V_1, V_2, \ldots, V_k$ is a partition of V such that there exists a base $B \in \mathcal{B}$ with $|B \cap V_i| = 1$ for every $i \in [k]$.

In other words, our aim is to find a k-partition that separates the elements of some basis of the matroid. SUB-MCP contains *Hypergraph k-cut* as a special case, so it does not admit a constant-factor approximation assuming the ETH. We study several algorithmic approaches for various familes of submodular functions. If the input function is symmetric or monotone submodular, then we denote the resulting problem as SYMSUB-MCP or MONSUB-MCP, respectively. A special case of SYMSUB-MCP is *Matroid-constrained Multiway Cut* (MC-MULTIWAY-CUT), formally defined as follows.

MC-MULTIWAY-CUT

Input: A graph $G = (V, E)$, edge weights $w \in \mathbb{R}_+^E$, and a rank-k matroid $\mathcal{M} = (V, \mathcal{B})$ given by an independence oracle, where $\mathcal{B}$ is the family of bases of $\mathcal{M}$.

Goal: Minimize $\sum_{i=1}^{k} d_w(V_i)$, where $V_1, V_2, \ldots, V_k$ is a partition of V such that there exists a base $B \in \mathcal{B}$ with $|B \cap V_i| = 1$ for every $i \in [k]$.

A special case of MONSUB-MCP is *Matroid-constrained Graph Coverage Partition* (GCOV-MCP), formally defined as follows.

GCOV-MCP

Input: A graph $G = (V, E)$, edge weights $w \in \mathbb{R}_+^E$, and a rank-k matroid $\mathcal{M} = (V, \mathcal{B})$ given by an independence oracle, where $\mathcal{B}$ is the family of bases of $\mathcal{M}$.

Goal: Minimize $\sum_{i=1}^{k} f(V_i)$, where $V_1, V_2, \ldots, V_k$ is a partition of V such that there exists a base $B \in \mathcal{B}$ with $|B \cap V_i| = 1$ for every $i \in [k]$ and $f(S) := \sum_{e \in E[S] \cup \delta(S)} w(e)$.

We now describe our results. See Table 1 for a summary of our results. The first problem that we study is SYMSUB-MCP. For this problem, we show that many of the algorithms for SYMSUB-K-P extend to SUB-MCP while achieving the same approximation guarantees. Firstly, we focus on the Gomory-Hu tree based algorithm for SymSub-k-P: this alorithm constructs a Gomory-Hu tree for the given symmetric submodular function and deletes the cheapest $k-1$ edges. It is well-known that this algorithm achieves a $(2-2/k)$-factor for SymSub-k-P. A natural generalization of this algorithm to SymSub-MCP is as follows (see Algorithm 1): construct a Gomory-Hu tree for the given symmetric submodular function and then greedily select a minimum-weight set of edges for which there exists a basis that is separated by the partition induced by the chosen edges. We show the following approximation guarantee for this generalization.

Theorem 1. *The Gomory-Hu tree algorithm given in Algorithm 1 achieves a $(2-2/k)$-approximation for* SymSub-MCP*.*

Secondly, we focus on the greedy splitting algorithm for SymSub-k-P. The greedy splitting algorithm was introduced by [17] and it proceeds in a greedy fashion as opposed to the the Gomory-Hu tree approach. It first finds the cheapest submodular cut $(S, V \setminus S)$ of V, and then iteratively finds, for a given intermediate partition $\mathcal{P}_i$, the next cheapest $\arg\min\{f(X) + f(W \setminus X) - f(W) : X \subset W, W \in \mathcal{P}_i\}$ split to increase the size of the partition by one. Zhao, Nagamochi, and Ibaraki showed that this algorithm achieves a $(2-2/k)$-factor for SymSub-k-P. We show that this algorithm generalizes to SymSub-MCP as well (see Algorithm 2) while achieving the same $(2-2/k)$-approximation factor.

Theorem 2. *The greedy splitting algorithm given in Algorithm 2 achieves a $(2-2/k)$-approximation for* SymSub-MCP*.*

A consequence of Theorems 1 and 2 is that we have a $(2-2/k)$-approximation for MC-Multiway-Cut.

The second problem that we study is MonSub-MCP. For the problem without matroid constraints, namely MonSub-k-P, there is a well-known cheapest singleton algorithm that is a fast $(2-1/k)$-approximation: the algorithm proceeds by choosing $k-1$ elements with the cheapest singleton cost, namely it picks an ordering $v_1, v_2, \ldots, v_n$ of the elements such that $f(v_1) \le \cdots \le f(v_{k-1}) \le f(v_j)$ for all $j \ge k$. It then returns the partition $(\{v_1\}, \ldots, \{v_{k-1}\}, \{v_k, \ldots, v_n\})$. To see why this yields a $(2-1/k)$-approximation, let $\mathcal{P}^* = \{V_1^*, \ldots, V_k^*\}$ be the optimal partition, ordered such that $f(V_1^*) \le \cdots \le f(V_k^*)$. By monotonicity, we have $\sum_{i=1}^{k-1} f(v_i) \le$

$\sum_{i=1}^{k-1} f(V_i^*) \le (1 - 1/k) \sum_{i=1}^{k} f(V_i^*)$. Moreover, monotonicity and submodularity imply that $f(\{v_k, \dots, v_n\}) \le f(V) \le \sum_{i=1}^{k} f(V_i^*)$. For the matroid-constrained setting, namely MonSub-MCP, we generalize the cheapest singleton algorithm as follows: assign the weight of each element v to be $f(v)$; next, find a minimum weight independent set of size $k-1$ in the matroid, and return the partition where these $k-1$ elements are singleton parts, and the last part consists of the remaining elements. The approximation factor of this algorithm for MonSub-MCP is at most $(2-1/k)$ by the same argument as that for MonSub-k-P.

Zhao, Nagamochi, and Ibaraki [17] showed that their greedy splitting algorithm achieves a $(2-2/k)$-factor for MonSub-k-P, thus improving on the cheapest singleton algorithm mildly. We generalize the greedy splitting algorithm (see Algorithm 2) and show that it achieves the same $(2-2/k)$-factor for MonSub-MCP.

Theorem 3. *The greedy splitting algorithm given in Algorithm 2 achieves a $(2-2/k)$-approximation for* MonSub-MCP.

We additionally show that this analysis is tight for the greedy splitting algorithm, even without the matroid constraints, i.e., there exist instances of MonSub-k-P for which the approximation factor of greedy splitting is at least $(2-2/k)$ – see Lemma 16.

Another consequence of the applicability of the greedy splitting algorithm is that we can extend the linear approximation guarantee for Sub-k-P from [17] to general Sub-MCP.

Theorem 4. *The greedy splitting algorithm in Algorithm 2 achieves a $(k-1)$-approximation for* Sub-MCP.

Finally, we consider the special case of graph coverage functions, namely GCov-MCP. We note that the graph coverage function is monotone, but the objective of GCov-MCP differs from the objective of MC-Multiway-Cut, since $f(\mathcal{P}) = d_w(\mathcal{P}) + |E|$ for every partition $\mathcal{P}$. For GCov-MCP, we can improve upon the result for monotone functions to achieve a factor that is identical to the current best factor for the global case. The proof is short and simple, so we include it here.

Theorem 5. *There exists a $4/3$-approximation for* GCov-MCP.

Proof: We recall that we have a $(2-2/k)$-approximation for SymSub-MCP by Theorems 1 and 2. Let $\mathcal{P}$ be the partition returned by such an algorithm with the input function being the cut function d_w. We return the same partition $\mathcal{P}$ for the coverage function.

We now show that the above algorithm achieves an approximation factor of $4/3$. Let $\mathcal{P}^*$ be an optimal partition for the graph coverage function f. We note that $\mathcal{P}^*$ is also an optimal partition for the cut function d_w. We have that $d_w(\mathcal{P}) \leq (2-2/k)d_w(\mathcal{P}^*) \leq 2d_w(\mathcal{P}^*)$. Let $c = \frac{d_w(\mathcal{P}^*)}{|E|}$. We derive two bounds on $f(\mathcal{P})/f(\mathcal{P}^*)$. On the one hand, we have that

$$\frac{f(\mathcal{P})}{f(\mathcal{P}^*)} \leq \frac{2d_w(\mathcal{P}^*)+|E|}{d_w(\mathcal{P}^*)+|E|} = \frac{2d_w(\mathcal{P}^*)+d_w(\mathcal{P}^*)/c}{d_w(\mathcal{P}^*)+d_w(\mathcal{P}^*)/c} = \frac{2c+1}{c+1}.$$

On the other hand, we have that $d_w(\mathcal{P}^*)+|E| = (c+1)|E|$, $f(\mathcal{P}) \leq 2|E|$, and hence,

$$\frac{f(\mathcal{P})}{f(\mathcal{P}^*)} \leq \frac{2|E|}{(c+1)|E|} = \frac{2}{c+1}.$$

Thus $\frac{f(\mathcal{P})}{f(\mathcal{P}^*)} \leq \max_{0\leq c\leq 1} \min\left\{\frac{2c+1}{c+1}, \frac{2}{c+1}\right\} = 4/3$, with the maximum attained at $c = 1/2$. $\square$

We summarize our results in Table 1 below.

	Matroid-Constrained	Global	Fixed Terminals
General	$O(k)$ (Thm. 4)	$O(k)$ [12, 17]	2 [5]
Symmetric	$2-2/k$ (Thms. 1, 2)	$2-2/k$ [17]	3/2 [5]
Monotone	$2-2/k$ (Thm. 3)	4/3 [4]	4/3 [2]
Graph Coverage	4/3 (Thm. 5)	4/3 [4]	9/8 [2]

Table 1: Comparison of our approximation results (first column) with the current best for global and fixed-terminal settings across different classes of submodular functions.

2. Preliminaries

Notation. We denote by $\mathbb{R}_+$ and $\mathbb{Z}_+$ the sets of nonnegative reals and integers, respectively. For a positive integer k, we use $[k] \coloneqq \{1,\ldots,k\}$. For a set function $f : 2^V \to \mathbb{R}_+$ and a partition $\mathcal{P}$ of V, we use the notation $f(\mathcal{P}) \coloneqq \sum_{X\in\mathcal{P}} f(X)$. For a graph $G=(V,E)$ and $X \subseteq V$, let $\delta_G(X)$ denote the set of edges leaving X, that is, $\delta_G(X) \coloneqq \{uv \in E\colon |\{u,v\} \cap X| = 1\}$. The cut function of G is then defined as $d_G(X) \coloneqq |\delta_G(X)|$ for $X \subseteq V$. We omit the subscript G when the graph is clear from the context. If edge weights $w \in \mathbb{R}^E_+$ are given, we use $d_w(X) \coloneqq \sum_{e\in\delta(X)} w(e)$. In directed graphs, we use δ^{in}, d^{in}, and δ^{out}, d^{out} for incoming and outgoing edges, respectively. For a graph $G = (V, E)$ and a partition $\mathcal{P} = \{V_1, \dots, V_k\}$ of V, $\delta_G(\mathcal{P}) \coloneqq \cup_{i=1}^k \delta_G(V_i)$ is called the *boundary* of $\mathcal{P}$. If edge weights $w \in \mathbb{R}^E_+$ are also given, we use $d_w(\mathcal{P}) = w(\delta_G(\mathcal{P}))$.

For a subset $F \subseteq E$, we denote the graph obtained by deleting the edges in F by $G - F$. When inserting (deleting) a single element e from a set X, we often use $X + e$ $(X - e)$ for $X \cup \{e\}$ $(X \setminus \{e\})$.

Matroids. A *matroid* $\mathcal{M} = (V, \mathcal{I})$ is defined by its *ground set* V and its *family of independent sets* $\mathcal{I} \subseteq 2^V$ that satisfies the *independence axioms*: (I1) $\emptyset \in \mathcal{I}$, (I2) $X \subseteq Y,\ Y \in \mathcal{I} \Rightarrow X \in \mathcal{I}$, and (I3) $X, Y \in \mathcal{I},\ |X| < |Y| \Rightarrow \exists e \in Y - X\ s.t.\ X + e \in \mathcal{I}$. The subsets of S not in $\mathcal{I}$ are called *dependent*. The *rank* $r(X)$ of a set X is the maximum size of an independent set in X. The *rank* of the matroid is $r(V)$. The maximal independent subsets of V are called *bases* and their family is usually denoted by $\mathcal{B}$. The matroid is uniquely defined by its rank function or by its family of bases, so we also use the notation $\mathcal{M} = (V, r)$ or $\mathcal{M} = (V, \mathcal{B})$. An inclusionwise minimal non-independent set is a *circuit*.

Given a matroid $\mathcal{M} = (V, \mathcal{I})$, its *k-truncation* is the matroid $(V, \mathcal{I}_k)$ with $\mathcal{I}_k = \{X \in \mathcal{I} \mid |X| \le k\}$. We denote the k-truncation of $\mathcal{M}$ by $(\mathcal{M})_k$. For a set $Z \subseteq V$, the *contraction of* Z results in a matroid $\mathcal{M}/Z = (V \setminus Z, \mathcal{I}')$ in which a set $X \subseteq V \setminus Z$ is independent if and only if $X \cup I$ is independent for any maximum independent set of $\mathcal{M}$ in Z. Let $D = (V, A)$ be a directed graph, $V_1, V_2 \subseteq V$, and $M_1 = (V_1, \mathcal{I}_1)$ be a matroid. We say that a set $Y \subseteq V_2$ is *linked* to another set $X \subseteq V_1$ if $|X| = |Y|$ and there exists $|X|$ pairwise vertex-disjoint directed paths from X to Y. Then the matroid $M_2 = (V_2, \mathcal{I}_2)$ defined by $\mathcal{I}_2 \coloneqq \{I_2 \subseteq V_2\colon I_2$ is linked to I_1 for some $I_1 \in \mathcal{I}_1\}$ is called the *matroid of* M_1 *induced by* D *on* V_2.

A *laminar matroid* is a matroid $\mathcal{M} = (V, \mathcal{I})$ where $\mathcal{I} = \{F \subseteq V\colon |F \cap V_i| \le g_i$ for each $i \in [q]\}$ for some laminar family $V_1, \dots, V_q$ of subsets of V and upper bounds $g_1, \dots, g_q \in \mathbb{Z}_+$. A rank-$r$ matroid $\mathcal{M} = (V, \mathcal{I})$ is *paving* if every set of size at most $r - 1$ is independent, or equivalently, if every circuit has size at least r. Such matroids have a nice characterization using hypergraphs [10, 16].

Proposition 6. *For a non-negative integer r, a ground set V of size at least r, and a (possibly empty) family $\mathcal{H} = \{H_1, \ldots, H_q\}$ of proper subsets of V such that $|H_i \cap H_j| \leq r - 2$ for $1 \leq i < j \leq q$, the family $\mathcal{B}_{\mathcal{H}} = \{X \subseteq V \mid |X| = r,\ X \not\subseteq H_i$ for $i = 1, \ldots, q\}$ forms the set of bases of a paving matroid. Furthermore, every paving matroid can be obtained in this way.*

As is common in matroid algorithms, we assume that matroids are given by an independence oracle, and we measure the running time in terms of the number of oracle calls and other elementary operations. For simplicity, we refer to an algorithm as "polynomial time" if its number of oracle calls and elementary operations is polynomial in the size of the ground set.

3. Symmetric Submodular Functions

We prove Theorem 1 in this section. The proof builds on a method outlined in [1]. First, we show that MC-Multiway-Cut is solvable on trees by a type of greedy algorithm. Then we use this on the Gomory-Hu tree of a symmetric submodular function to obtain an approximation algorithm for SymSub-MCP.

3.1. MC-Multiway-Cut on Trees

As a first step, we show that MC-Multiway-Cut is solvable in polynomial time on trees. Interestingly, in this case the problem reduces to finding a minimum weight basis of a matroid. Recall that in this setting, we are given a tree $G = (V, E)$ together with edge weights $w \in \mathbb{R}^E_+$ and a matroid $\mathcal{M} = (V, \mathcal{I})$. Let us define the following family of subsets of E:

$$\begin{aligned} \mathcal{I}' = \{X' \subseteq E \colon & \text{ the components } V_1, \ldots, V_{|X'|+1} \text{ of } G - X' \\ & \text{have a transversal independent in } \mathcal{M}\}. \end{aligned} \tag{1}$$

In other words, an edge set X' is in $\mathcal{I}'$ if there exists $v_i \in V_i$ for every connected component V_i of $G - X'$ such that $\{v_1, \ldots, v_{|X'|+1}\}$ forms an independent set of $\mathcal{M}$.

Lemma 7. *If $\mathcal{M}$ has rank at least 1, then $\mathcal{I}'$ satisfies the independence axioms.*

PROOF: We verify the axioms one by one. Axiom (I1) clearly holds, since the rank of $\mathcal{M}$ is at least one, and any non-loop $v \in V$ is a transversal.

To see (I2), let $Y' \in \mathcal{I}'$ and $X' \subseteq Y'$. Let $\mathcal{V}$ and $\mathcal{U}$ be the families of connected components of $G-Y'$ and $G-X'$, respectively. Since $X' \subseteq Y'$, the partition $\mathcal{U}$ forms a coarsening of $\mathcal{V}$. By the definition of $\mathcal{I}'$, $Y' \in \mathcal{I}'$ implies that there exists a transversal Y of $\mathcal{V}$ that is independent in $\mathcal{M}$. Since $\mathcal{U}$ is a coarsening of $\mathcal{V}$, a properly chosen subset X of Y will form a transversal of $\mathcal{U}$, thus showing that $X' \in \mathcal{I}'$.

Finally, we verify (I3). Let $X', Y' \in \mathcal{I}'$ with $|X'| < |Y'|$. Let $X, Y \in \mathcal{I}$ be the corresponding independent sets in $\mathcal{M}$ chosen such that $|X \cap Y|$ is as large as possible. By axiom (I3) for $\mathcal{I}$, there exists $u \in Y \setminus X$ such that $X + u \in \mathcal{I}$. Let C be the component of $G - X'$ containing u, and v be the element of X in C. If $v \notin Y$, then $X - v + u \in \mathcal{I}$ is still a transversal of the conenected somponents of $G - X'$, contradicting the choice of X and Y. Thus $v \in Y$. As $u, v \in Y$, there exists an edge $e \in Y'$ that lies on the path between u and v in G. Then $X' + e$ is in $\mathcal{I}'$ as desired, as it has $X + u$ as an independent set of representatives. $\square$

Given the independence oracle access to $\mathcal{M}$, we can construct an independence oracle to the matroid $\mathcal{M}' = (E, \mathcal{I}')$, as deciding whether there is an independent set of representatives for a given partition of V reduces to the intersection of a partition matroid and $\mathcal{M}$.

Remark 8. *It is worth mentioning that $\mathcal{I}'$ can be identified with the independent set of a matroid of $\mathcal{M}$ induced by a directed graph. To see this, pick an arbitrary vertex $r \in V$, and orient the edges of G such that the resulting directed graph is an in-arborescence rooted at r. Then, subdivide every arc uv with a new vertex x_{uv}, resulting in arcs ux_{uv} and $x_{uv}v$. Finally, add a new vertex x to the graph together with the arc rx. Note that the vertex set of the directed graph thus obtained is $V \cup \{x_{uv} \colon uv \in E\} \cup \{x\}$. Now consider the matroid obtained by taking the matroid of $\mathcal{M}$ induced by the directed graph on $\{x_{uv} \colon uv \in E\} \cup \{x\}$, and contracting x in it. It is not difficult to check that $Y \subseteq \{x_{uv} \colon uv \in E\}$ is independent in this matroid if and only if the corresponding subset of edges is in $\mathcal{I}'$.*

Corollary 9. MC-MULTIWAY-CUT *is solvable in polynomial time on trees.*

PROOF: First, we observe that MC-MULTIWAY-CUT has an optimal solution $\mathcal{P} = \{V_1, \ldots, V_k\}$ where $G[V_i]$ is connected for every $i \in [k]$. To show this, we choose $\mathcal{P}$ so that $|\delta_G(\mathcal{P})|$ is smallest, and choose a basis B with $|B \cap V_i| = 1$ for all $i \in [k]$. Suppose for contradiction that some $G[V_i]$ is not connected, and let U be a component that does not contain $B \cap V_i$. Since G is connected, there exists an edge from U to some V_j. The partition $\mathcal{P}' = \mathcal{P} \setminus \{V_i, V_j\} \cup \{V_j \cup U, V_i \setminus U\}$ is also optimal and $|\delta_G(\mathcal{P}')| < |\delta_G(\mathcal{P})|$, a contradiction.

By the above observation and Lemma 7, the minimum weight matroid-constrained multiway cut on trees corresponds to the minimum weight basis of the matroid $\mathcal{M}' = (E, \mathcal{I}')$. Thus, MC-Multiway-Cut can be solved in polynomial time by using the greedy algorithm to find a minimum weight basis of $\mathcal{M}'$. $\square$

3.2. Symmetric Submodular Functions

Goemans and Ramakrishnan [9] were the first to note that symmetric submodular systems have a Gomory-Hu tree. Given a symmetric submodular function f over a ground set V, its Gomory-Hu tree is a tree $H = (V, F)$ with a weight function $w_H \in \mathbb{R}_+^F$ that encodes the minimum $s-t$ cuts for each pair s, t of vertices in the following sense: given a path $P_{s,t}$ between s and t, $\min_{e \in P_{s,t}} w_H(e) = \min_{s \in S \subseteq V-t} f(S)$. Furthermore, the two components of the tree obtained by removing the edge of minimum w_H-weight on the path give the two sides of a minimum $s-t$ cut in the submodular system f.

The greedy algorithm solves MC-Multiway-Cut in the special case when G is a tree. The classical $(2-2/k)$-approximation for Multiway-Cut uses 2-way cuts coming from the Gomory-Hu tree, and so does the $(2-2/k)$-approximation for k-Cut. We follow a similar approach to obtain a $(2-2/k)$-approximation for SymSub-MCP, presented in Algorithm 1. The algorithm can be interpreted as taking the minimum edges in the Gomory-Hu tree as long as they allow a valid system of representatives.

Theorem 10. *Algorithm 1 provides a $(2-2/k)$-approximation to* SymSub-MCP.

Proof: Let $\mathrm{OPT} = \{V_1^*, \ldots, V_k^*\}$ be an optimal partition, and let $t_i^* \in V_i^*$ for $i \in [k]$ be representatives such that $\{t_1^*, \ldots, t_k^*\} \in \mathcal{B}$. Without loss of generality, we may assume that V_k^* has the maximum f value among the partition classes. Let $H = (V, E)$ be the Gomory-Hu tree of f.

We transform OPT into a solution OPT_{GH} on H, losing at most a factor of $(2-2/k)$. We do this by repeatedly removing the minimum weight

Algorithm 1 Approximation algorithm for SymSub-MCP

Input: A symmetric, submodular function $f\colon 2^V \to \mathbb{R}_+$ and a rank-k matroid $\mathcal{M}$.

Output: A feasible partition $\mathcal{P}$.

Compute the Gomory-Hu tree $H = (V, E)$ of f and its weight function w_H.

Let $\mathcal{M}' = (E, \mathcal{I}')$ be the matroid defined as in (1) using $\mathcal{M}$ and H.

Set $C \leftarrow \emptyset$.

while $|C| < k - 1$ **do**

  $e \leftarrow \arg\min\{w_H(e) \colon e \notin C, C + e \in \mathcal{I}'\}$

  $C \leftarrow C + e$

end while

Return the connected components of $(V, E \setminus C)$.

edge in E that separates a pair among the representatives $t_1^*, \ldots, t_k^*$ that are in the same component of H. More precisely, we start with $H_0 = H$, and take the minimum-weight edge $e_1 \in E(H_0)$ separating some pair of representatives t_i^*, t_j^* that are in the same component of H_0. Define the edge $f_1 = (t_i^*, t_j^*)$. Then we construct $H_1 = H_0 - e_1$, and repeat this process to get a sequence of edges $e_1, \ldots, e_{k-1}$ and a tree of representative pairs $F = (\{t_1^*, \ldots, t_k^*\}, \{f_1, \ldots, f_{k-1}\})$.

Fix a vertex x arbitrarily. For an edge $e \in E$, let $U(e)$ denote the vertices of the connected component of $H - e$ containing x. Direct the edges of F away from t_k^*, and reorder the edges such that f^1 is the edge going into t_1^*, f^2 into t_2^*, and so on. Let e^i be the edge of the Gomory-Hu tree corresponding to f^i, i.e., the minimum weight edge of the path between the two endpoints of f^i. Then we have $f(V_i^*) \geq f(U(e^i))$ for every i, since $(V_i^*, V \setminus V_i^*)$ separates the two representatives in f^i as well, and $f(U(e^i))$ is the weight of the minimum submodular cut between these. Let OPT_{GH} be the connected components of H_{k-1}, $\mathrm{ALG} = \{V_1, \ldots, V_k\}$ be the partition found by Algorithm 1, ALG_{GH} be the corresponding edges in the Gomory-Hu tree H, and w_H be the weight function on H.

We first argue that $f(\mathrm{ALG}) \leq \sum_{i=1}^{k} d_{w_H}(V_i)$. This follows from basic properties of the Gomory-Hu tree of symmetric submodular systems, but needs some work. Let $C = \{g_1, \ldots, g_{k-1}\}$ be the cut found by Algorithm 1, which means $\sum_{i=1}^{k} d_{w_H}(V_i) = 2\sum_{i=1}^{k-1} w_H(g_i)$. As f is symmetric, we have $2w_H(g_i) = f(U(g_i)) + f(V \setminus U(g_i))$.

Let us direct each g_i away from x in the tree H, and let $\vec{g_i}$ denote the directed edge. We then have

$$\begin{aligned}
f(\mathrm{ALG}) = \sum_{i=1}^{k} f(V_i) &= \sum_{i=1}^{k} f\left(\left(\bigcap_{\vec{g_j}\in\delta^{out}(V_i)} U(g_j)\right)\bigcap\left(\bigcap_{\vec{g_j}\in\delta^{in}(V_i)} V\setminus U(g_j)\right)\right)\\
&\le \sum_{i=1}^{k}\left(\sum_{\vec{g_j}\in\delta^{out}(V_i)} f(U(g_j)) + \sum_{\vec{g_j}\in\delta^{in}(V_i)} f(V\setminus U(g_j))\right)\\
&= \sum_{i=1}^{k}\sum_{g_j\in\delta(V_i)} w_H(g_j) = \sum_{i=1}^{k} d_{w_H}(V_i),
\end{aligned}$$

where the inequality follows by submodularity. Then

$$\begin{aligned}
f(\mathrm{ALG}) &\le \sum_{i=1}^{k} d_{w_H}(V_i) \le 2\sum_{i=1}^{k-1} w_H(e^i) = 2\sum_{i=1}^{k-1} f(U(e^i)) \le 2\sum_{i=1}^{k-1} f(V_i^*)\\
&\le 2(1-1/k)\sum_{i=1}^{k} f(V_i^*) \le (2-2/k)f(\mathrm{OPT}),
\end{aligned}$$

where the second inequality holds because $\{V_1,\ldots,V_k\}$ is an optimal solution of MC-Multiway-Cut on H by Lemma 7, and the second to last inequality holds because of the assumption that V_k^* has maximum f value among the classes. This concludes the proof of the theorem. $\square$

3.3. Intersection of Two Matroids

Lemma 7 implies the solvability on trees of the following more general problem involving two matroids:

> Double MC-Multiway-Cut
> **Input:** A graph $G=(V,E)$, edge weights $w\in\mathbb{R}_+^E$, and rank-k matroids $\mathcal{M}_1=(V,\mathcal{B}_1)$ and $\mathcal{M}_2=(V,\mathcal{B}_2)$ given by independence oracles, where $\mathcal{B}_i$ is the family of bases of $\mathcal{M}_i$.
> **Goal:** Minimize $\sum_{i=1}^{k} d_w(V_i)$, where $V_1,V_2,\ldots,V_k$ is a partition of V such that there exists $B_1\in\mathcal{B}_1, B_2\in\mathcal{B}_2$ with $|B_j\cap V_i|=1$ for all $i\in[k], j\in[2]$.

Here, the goal is to find a minimum multiway cut where there is a choice of terminals forming a basis of $\mathcal{M}_1$, and *not necessarily the same* choice of terminals forming a basis of $\mathcal{M}_2$.

Theorem 11. Double MC-Multiway-Cut *can be solved in polynomial time if G is a tree.*

Proof: We can construct $\mathcal{M}_i'$ as in (1) using $\mathcal{M}_i$ $(i = 1, 2)$, and we can find a minimum-weight common spanning set C of $\mathcal{M}_1'$ and $\mathcal{M}_2'$ in polynomial time. Let $\{V_1, \dots, V_\ell\}$ denote the connected components of $E \setminus C$, and let $B_1 \in \mathcal{B}_1$ and $B_2 \in \mathcal{B}_2$ be bases such that $|B_j \cap V_i| \le 1$ for all $i \in [\ell], j \in [2]$. If $\ell > k$, then there must be classes V_i and V_j such that $|(V_i \cup V_j) \cap B_1| \le 1$ and $|(V_i \cup V_j) \cap B_2| \le 1$. Remove V_i and V_j from the partition, and add $V_i \cup V_j$. Repeat this operation until $\ell = k$, and let $\{V_1', \dots, V_k'\}$ be the obtained feasible partition. Then $\sum_{i=1}^k d_w(V_i') \le \sum_{i=1}^k d_w(V_i)$, so $\{V_1', \dots, V_k'\}$ is an optimal solution, because the boundary of any feasible partition is a common spanning set of $\mathcal{M}_1'$ and $\mathcal{M}_2'$. $\square$

We get a significantly more difficult problem if the representatives for the two matroids are required to be the same, that is, if we require the set of representatives to form a *common basis* of $\mathcal{M}_1$ and $\mathcal{M}_2$. The problem is defined as follows.

> Common MC-Multiway-Cut
>
> **Input:** A graph $G = (V, E)$, edge weights $w \in \mathbb{R}_+^E$, and rank-k matroids $\mathcal{M}_1 = (V, \mathcal{B}_1)$ and $\mathcal{M}_2 = (V, \mathcal{B}_2)$ given by independence oracles, where $\mathcal{B}_i$ is the family of bases of $\mathcal{M}_i$.
>
> **Goal:** Minimize $\sum_{i=1}^k d_w(V_i)$, where $V_1, V_2, \dots, V_k$ is a partition of V such that there exists $B \in \mathcal{B}_1 \cap \mathcal{B}_2$ with $|B \cap V_i| = 1 \ \forall i \in [k]$.

We show that Common MC-Multiway-Cut includes the problem of finding a common basis in the intersection of three partition matroids; this latter problem is known to be hard as it generalizes e.g. the 3-dimensional matching problem.

Theorem 12. *The problem of verifying whether three partition matroids have a common basis reduces to the problem of deciding if the optimum of a* Common MC-Multiway-Cut *instance is* 0*.*

PROOF: Let P_1, P_2, P_3 be partition matroids on a ground set S with partition classes $\mathcal{U} = \{U_1, \ldots, U_r\}$, $\mathcal{V} = \{V_1, \ldots, V_r\}$, and $\mathcal{W} = \{W_1, \ldots, W_r\}$, respectively. We encode P_3 into a tree of depth two by creating a root z, a depth one vertex v_i for each W_i, and leaf vertices for each element of $s \in S$. The edges are between v_i and z, and for each $s \in W_i$, we add an edge between s and v_i. We add the new vertices to the ground set of P_1 and P_2 to create M_1 and M_2: each v_i is added as a loop in both matroids, while z is added as a free element, i.e., it is added to every base. We assign weight 0 to the edges from the root z, and all other edges get unit weights.

The only possible solution with objective value 0 is the partition formed by $\{z\}$ and $W_i + v_i$ $(i \in [r])$. This is a feasible solution if and only if P_1 and P_2 have a common basis that intersects each W_i in exactly one element, that is, P_1, P_2, and P_3 have a common basis. $\square$

3.4. Paving Matroids

In this section, we show that SUB-MCP is equivalent to SUB-K-P when the matroid $\mathcal{M}$ is a paving matroid. The key observation is that if the matroid is a rank-k paving matroid, then *any* k-partition has a valid basis representative.

Lemma 13. *Let $\mathcal{M} = (V, \mathcal{B})$ a rank-k paving matroid for some $k \geq 1$ and $\mathcal{P} = \{V_1, \ldots, V_k\}$ be a partition of V. Then there exists $B \in \mathcal{B}$ such that $|B \cap V_i| = 1$ for $i \in [k]$.*

PROOF: Let $\mathcal{H} = \{H_1, \ldots, H_q\}$ be the hypergraph representation of $\mathcal{M}$. Since every set of size at most $k-1$ is independent in $\mathcal{M}$, there is an independent set $X = \{v_1, \ldots, v_{k-1}\}$ such that $v_i \in V_i$ for $i \in [k-1]$. If $X + v$ is a basis for some $v \in V_k$, then we are done. Otherwise, for every $v \in V_k$ there exists a hyperedge $H_v \in \mathcal{H}$ such that $X + v \subseteq H_v$. As $|H_u \cap H_v| = |X| = k-1$ for any distinct $u, v \in V_k$, we get that $H_u = H_v$ by the properties of the hypergraph representation of the paving matroid. Let us denote this unique hyperedge by H; we thus have $V_k \subseteq H$. By repeating the same argument for each partition class in place of V_k, we get that $H = V$, contradicting the hyperedges being proper subsets of V. $\square$

4. Greedy Splitting for Sub-MCP

In the analysis of the greedy splitting algorithm by Zhao, Nagamochi and Ibaraki [17], they rely on a key lemma in their approximation guarantees for Sub-MP, SymSub-MP, MonSub-MP, Sub-k-P, SymSub-k-P, and MonSub-k-P. We show that this lemma holds even when an additional matroid constraint is imposed, and then follow their proof approaches to prove Theorems 2, 3, and 4; all aforementioned results of [17] follow from these three theorems as well. The modified greedy splitting algorithm is described as Algorithm 2.

Algorithm 2 Greedy Splitting Algorithm for Sub-MCP

Input: A submodular function $f\colon 2^V \to \mathbb{R}_+$ and a rank-k matroid $\mathcal{M} = (V, \mathcal{I})$.

Output: A feasible partition $\mathcal{P}$.

1: $\mathcal{P}_1 \leftarrow \{V\}$.
2: **for** $i = 1, \ldots, k-1$ **do**
3: Let $(X, W) \in \arg\min\{f(X) + f(W \setminus X) - f(W)\colon X \subseteq W \in \mathcal{P}_i$, there is $I \in \mathcal{I}$ s.t. I is a transversal of $(\mathcal{P}_i \setminus W) \cup \{X, W \setminus X\}\}$.
4: $\mathcal{P}_{i+1} \leftarrow (\mathcal{P}_i \setminus W) \cup \{X, W \setminus X\}$
5: **end for**
6: Return $\mathcal{P}_k$.

Remark 14. *Line 3 of Algorithm 2 can be executed in polynomial time by the following subroutine.*

1. *For every pair x, y of elements in W, we can decide if x, y can be extended to an independent set so that we pick a single element from each class of $\mathcal{P}_i \setminus W$.*
2. *Now that we know which pairs x, y are possible representatives, we can do the following: for each fixed pair x, y, find the best partition of W into X and $W \setminus X$ such that $x \in X$, $y \in W \setminus X$.*

Lemma 15 (Main Lemma in [17]). *Let $\mathcal{P}_i$ be the i^{th} partition returned by the greedy splitting procedure, and $\mathcal{P} = (V_1, \ldots, V_i)$ be any partition satisfying* *the matroid constraint. Then for any submodular f,*

$$f(\mathcal{P}_i) \leq \sum_{j=1}^{i-1} (f(V_j) + f(V \setminus V_j)) - (i-2)f(V)$$

PROOF: We proceed, as in [17], by induction on i. The base case $i = 1$ holds trivially. In the i^{th} step, we are given a partition $\mathcal{P} = (V_1, \ldots, V_i)$ along with some $I_i \in \mathcal{I}$ that is also independent in the partition matroid induced by $\mathcal{P}$. Let $\mathcal{P}_{i-1} = (U_1, \ldots, U_{i-1})$ be the partition created by the algorithm in the previous step, and $I_{i-1} \in \mathcal{I}$ be the representatives corresponding to $\mathcal{P}_{i-1}$. As $|I_i| \geq |I_{i-1}|$, there must be an element $v \in I_i \setminus I_{i-1}$ such that $I_{i-1} + v$ is independent in $\mathcal{M}$. The element v must be in some part of $\mathcal{P}_{i-1}$, say U_h, and some part of $\mathcal{P}$, say V_j. Let $u \in U_h$ be the element of I_{i-1} in U_h.

Case 1: Say $u \notin V_j \cap U_h$. Then $V_j \cap U_h, U_h \setminus V_j$ is a valid candidate split for Line 4 in Algorithm 2, so

$$f(\mathcal{P}_i) - f(\mathcal{P}_{i-1}) \leq f(V_j \cap U_h) + f(U_h \setminus V_j) - f(U_h).$$

Submodularity then implies

$$\begin{aligned} f(\mathcal{P}_i) - f(\mathcal{P}_{i-1}) &\leq f(V_j) + f(U_h \setminus V_j) - f(V_j \cup U_h) \\ &\leq f(V_j) + f(V \setminus V_j) - f(V). \end{aligned}$$

Case 2: For the other case, assume $u \in V_j \cap U_h$. Then $u \notin I_i$ since the element of I_i in V_j is v; furthermore, $I'_{i-1} = I_{i-1} + v - u$ is a valid representative for $\mathcal{P}_{i-1}$. We can repeat the same argument with I'_{i-1}. This process will eventually terminate, as $|I'_{i-1} \cap I_i| = |I_{i-1} \cap I_i| + 1$, hence after a finite number of steps either we are in Case 1, or $I_{i-1} \subseteq I_i$. In the latter case, u cannot be in V_j, as there is another $v \in V_j \cap I_i$, and we are again in Case 1.

Thus, there is some $V_j \in \mathcal{P}$, such that $f(\mathcal{P}_i) - f(\mathcal{P}_{i-1}) \leq f(V_j) + f(V \setminus V_j) - f(V)$. By the induction hypothesis, when applied to $\mathcal{P} \setminus V_j$, we get

$$f(\mathcal{P}_{i-1}) \leq \sum_{j=1}^{i-2} (f(V_j) + f(V \setminus V_j)) - (i-3)f(V).$$

Therefore,

$$\begin{aligned} f(\mathcal{P}_i) &\leq f(\mathcal{P}_{i-1}) + f(V_j) + f(V \setminus V_j) - f(V) \\ &\leq \sum_{j=1}^{i-1} (f(V_j) + f(V \setminus V_j)) - (i-2)f(V), \end{aligned}$$

concluding the proof of the lemma. $\square$

With Lemma 15 in hand, the results of [17] follow immediately. In order to make the paper self-contained, we repeat them here.

PROOF: [Proof of Theorem 4] Lemma 15 shows that for the optimal partition $\mathcal{P} = \{V_1, \ldots, V_k\}$, the output $\mathcal{P}_k$ of the Algorithm 2 satisfies

$$
\begin{aligned}
f(\mathcal{P}_k) &\leq \sum_{j=1}^{k-1}(f(V_j) + f(V \setminus V_j)) - (k-2)f(V) \\
&\leq \sum_{j=1}^{k-1}\sum_{i=1}^{k} f(V_i) - (k-2)f(V) \leq (k-1)f(\mathcal{P}).
\end{aligned}
$$

This concludes the proof. $\square$

PROOF: [Proof of Theorem 2] Let $\mathcal{P}^* = \{V_1^*, \ldots, V_k^*\}$ be the optimal partition for SYMSUB-MCP, ordered so that $V_1^* \leq V_2^* \leq \cdots \leq V_k^*$. Then by Lemma 15,

$$
f(\mathcal{P}_k) \leq \sum_{j=1}^{k-1}(f(V_j^*) + f(V \setminus V_j^*)) - (k-2)f(V),
$$

where $\mathcal{P}_k$ is the partition returned by Algorithm 2. Using the fact that f is symmetric and submodular, we have

$$
f(\mathcal{P}_k) \leq 2\sum_{j=1}^{k-1} f(V_j^*) - (k-2)f(V) \leq 2\sum_{j=1}^{k-1} f(V_j^*) \leq \left(2 - \frac{2}{k}\right) f(\mathcal{P}^*),
$$

concluding the proof of the theorem. $\square$

PROOF: [Proof of Theorem 3] Let $\mathcal{P}^* = \{V_1^*, \ldots, V_k^*\}$ be the optimal partition for MONSUB-MCP, ordered so that $V_1^* \leq V_2^* \leq \cdots \leq V_k^*$. Then, by Lemma 15,

$$
f(\mathcal{P}_k) \leq \sum_{j=1}^{k-1}(f(V_j^*) + f(V \setminus V_j^*)) - (k-2)f(V),
$$

where $\mathcal{P}_k$ is the partition returned by Algorithm 2. Using the fact that f is monotone and submodular, we have

$$f(\mathcal{P}_k) \leq \sum_{j=1}^{k-1} f(V_j^*) + f(V \setminus V_{k-1}^*) \leq \sum_{j=1}^{k-2} f(V_j^*) + f(V_{k-1}^*) + f(V \setminus V_{k-1}^*)$$
$$\leq \left(1 - \frac{2}{k}\right) f(\mathcal{P}^*) + f(V_{k-1}^*) + f(V \setminus V_{k-1}^*) \leq \left(2 - \frac{2}{k}\right) f(\mathcal{P}^*),$$

concluding the proof of the theorem. $\square$

Our analysis of the $(2 - 2/k)$-approximation for monotone submodular functions is tight for the greedy splitting algorithm, even without the matroid constraints, by the following lemma.

Lemma 16. *There is an instance of monotone submodular k-partition for which the greedy splitting algorithm achieves exactly $(2-2/k)$-approximation.*

PROOF: Let $\mathcal{P} = \{S_1, \ldots, S_k\}$ be a partition of the ground set V into subsets of size at least 2. Consider the laminar matroid $\mathcal{M}$ in which a set $X \subseteq V$ is independent if and only if $|V \cap S_i| \leq 1$ for $i \in [k]$ and $|X| \leq k-1$. Let f be the rank function of $\mathcal{M}$ – clearly, f is monotone and submodular.

For any initial split of the greedy splitting algorithm, we have $f(S) + f(V \setminus S) \geq k$. Therefore, the algorithm might choose S to be a singleton. In general, as the greedy splitting algorithm proceeds, in the i^{th} step the algorithm might split the class S_i into a singleton and the set of remaining elements. In such a scenario, the resulting partition has the form $\{s_1, \ldots, s_{k-1}, V \setminus \{s_1, \ldots, s_{k-1}\}\}$. The sum of the ranks of these sets is $2k-2$, while the optimal partition is $\mathcal{P}$ with a total f value of k. This shows that greedy splitting might lead to a multiplicative error of $2 - 2/k$. $\square$

References

[1] K. Bérczi, T. Király, and D. P. Szabo. Multiway Cuts with a Choice of Representatives. In *49th International Symposium on Mathematical Foundations of Computer Science (MFCS 2024)*, pages 25:1–25:17, 2024.

[2] R. Bi, K. Chandrasekaran, and S. Joshi. Monotone submodular multiway partition. In *Integer Programming and Combinatorial Optimization*, pages 72–85, 2025.

[3] G. Calinescu and A. Zelikovsky. The polymatroid steiner problems. *Journal of Combinatorial Optimization*, 9:281–294, 2005.

[4] K. Chandrasekaran and W. Wang. Approximating submodular k-partition via principal partition sequence. *SIAM Journal on Discrete Mathematics*, 38:3198–3219, 2024.

[5] C. Chekuri and A. Ene. Approximation algorithms for submodular multiway partition. In *2011 IEEE 52nd Annual Symposium on Foundations of Computer Science*, pages 807–816. IEEE, 2011.

[6] C. Chekuri, S. Guha, and J. Naor. The steiner k-cut problem. *SIAM Journal on Discrete Mathematics*, 20(1):261–271, 2006.

[7] C. Chekuri and S. Li. On the hardness of approximating the k-way hypergraph cut problem. *Theory of Computing*, 16(1), 2020.

[8] E. Dahlhaus, D. S. Johnson, C. H. Papadimitriou, P. D. Seymour, and M. Yannakakis. The complexity of multiterminal cuts. *SIAM Journal on Computing*, 23(4):864–894, 1994.

[9] M. X. Goemans and V. Ramakrishnan. Minimizing submodular functions over families of sets. *Combinatorica*, 15(4):499–513, 1995.

[10] J. Hartmanis. Lattice theory of generalized partitions. *Canadian Journal of Mathematics*, 11:97–106, 1959.

[11] P. Manurangsi. Inapproximability of maximum biclique problems, minimum k-cut and densest at-least-k-subgraph from the small set expansion hypothesis. *Algorithms*, 11(1):10, 2018.

[12] K. Okumoto, T. Fukunaga, and H. Nagamochi. Divide-and-conquer algorithms for partitioning hypergraphs and submodular systems. *Algorithmica*, 62(3):787–806, 2012.

[13] R. Santiago. New approximations and hardness results for submodular partitioning problems. In *Combinatorial Algorithms: 32nd International Workshop (IWOCA)*, pages 516–530, 2021.

[14] H. Saran and V. V. Vazirani. Finding k cuts within twice the optimal. *SIAM Journal on Computing*, 24(1):101–108, 1995.

[15] A. Sharma and J. Vondrák. Multiway cut, pairwise realizable distributions, and descending thresholds. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pages 724–733, 2014.

[16] D. J. A. Welsh. *Matroid Theory*. L. M. S. Monographs, No. 8. Academic Press, London-New York, 1976.

[17] L. Zhao, H. Nagamochi, and T. Ibaraki. Greedy splitting algorithms for approximating multiway partition problems. *Mathematical Programming*, 102:167–183, 2005.