Approximating Submodular k-Partition via Principal Partition Sequence

Karthekeyan Chandrasekaran
University of Illinois, Urbana-Champaign, IL, USA

Weihang Wang
University of Illinois, Urbana-Champaign, IL, USA

Abstract

In submodular k-partition, the input is a submodular function $f : 2^V \to \mathbb{R}_{\geq 0}$ (given by an evaluation oracle) along with a positive integer k and the goal is to find a partition of the ground set V into k non-empty parts $V_1, V_2, \ldots, V_k$ in order to minimize $\sum_{i=1}^{k} f(V_i)$. Narayanan, Roy, and Patkar [18] designed an algorithm for submodular k-partition based on the principal partition sequence and showed that the approximation factor of their algorithm is 2 for the special case of graph cut functions (which was subsequently rediscovered by Ravi and Sinha [22]). In this work, we study the approximation factor of their algorithm for three subfamilies of submodular functions – namely monotone, symmetric, and posimodular and show the following results:

1. The approximation factor of their algorithm for monotone submodular k-partition is 4/3. This result improves on the 2-factor that was known to be achievable for monotone submodular k-partition via other algorithms. Moreover, our upper bound of 4/3 matches the recently shown lower bound under polynomial number of function evaluation queries [23]. Our upper bound of 4/3 is also the first improvement beyond 2 for a certain graph partitioning problem that is a special case of monotone submodular k-partition.
2. The approximation factor of their algorithm for symmetric submodular k-partition is 2. This result generalizes their approximation factor analysis beyond graph cut functions.
3. The approximation factor of their algorithm for posimodular submodular k-partition is 2.

We also construct an example to show that the approximation factor of their algorithm for arbitrary submodular functions is $\Omega(n/k)$.

2012 ACM Subject Classification Theory of computation → Approximation algorithms analysis

Keywords and phrases Approximation algorithms

Digital Object Identifier 10.4230/LIPIcs.APPROX/RANDOM.2023.3

Category APPROX

Related Version *Full Version*: `https://arxiv.org/abs/2305.01069`

Funding *Karthekeyan Chandrasekaran*: Supported in part by NSF grants CCF 1814613 and 1907937
Weihang Wang: Supported in part by NSF grants CCF 1814613 and 1907937.

Acknowledgements Karthekeyan Chandrasekaran would like to thank Chandra Chekuri for asking about the approximation factor of the principal partition sequence based approach for symmetric submodular k-partition.

1 Introduction

A set function $f : 2^V \to \mathbb{R}$ is submodular if $f(A) + f(B) \geq f(A \cap B) + f(A \cup B)$ for every $A, B \subseteq V$. An evaluation oracle for a function $f : 2^V \to \mathbb{R}$ takes a subset $A \subseteq V$ as input and returns $f(A)$. We consider the *submodular k-partition* problem defined as follows: The input consists of a non-negative submodular function $f : 2^V \to \mathbb{R}_{\geq 0}$ on a finite ground set V via an evaluation oracle and an integer $k \geq 2$. The goal is to find a partition $V_1, V_2, \ldots, V_k$ of V into k non-empty parts in order to minimize $\sum_{i=1}^{k} f(V_i)$. Namely, the goal is to compute

$$\min\left\{\sum_{i\in[k]} f(V_i):\ V_1,V_2,\dots,V_k \text{ is a partition of } V,\ V_i\neq\emptyset\ \forall i\in[k]\right\}.$$

Throughout, we will assume that the input submodular function is non-negative and denote the size of the ground set V by n. If $k=2$, then the problem reduces to the classic submodular minimization problem. We emphasize that our focus is on submodular k-partitioning when k is part of input (see [4] for a discussion of the problem for fixed constant k). Submodular k-partition formulates several interesting partitioning problems and we will discuss some of the special cases below. For arbitrary submodular functions, the problem is NP-hard [10], does not admit a $(2-\epsilon)$-approximation assuming polynomial number of function evaluation queries [23], does not admit a $n^{1/(\log\log n)^c}$-approximation for every constant c assuming the Exponential Time Hypothesis [6], and the best approximation factor that is known is $O(k)$ [19, 26].

In this work, we will be interested in the submodular k-partition problem for subfamilies of submodular functions – namely monotone, symmetric, and posimodular submodular functions. A set function $f : 2^V \to \mathbb{R}$ is

1. monotone if $f(B)\geq f(A)$ for every $A\subseteq B\subseteq V$,
2. symmetric if $f(A)=f(V-A)$ for every $A\subseteq V$, and
3. posimodular if $f(A)+f(B)\geq f(A-B)+f(B-A)$ for every $A,B\subseteq V$.

If the input submodular function is monotone/symmetric/posimodular, then we call the associated submodular k-partition problem as monotone/symmetric/posimodular submodular k-partition. We note that monotone submodular functions and symmetric submodular functions are also posimodular[1]. Hence, posimodular submodular k-partition problem generalizes both monotone submodular k-partition and symmetric submodular k-partition problems. We now discuss the approximation status of symmetric/monotone/posimodular submodular k-partition and some of their well-known special cases (see Table 1 for a summary of approximation factors of symmetric/monotone/posimodular submodular k-partition achieved by different approaches).

Monotone submodular k-partition. Special cases of monotone submodular k-partition problem include matroid k-partition and coverage k-partition – the submodular functions of interest here are matroid rank functions and coverage functions respectively. Matroid k-partition captures several interesting problems: e.g., (1) partition the columns of a given matrix into k non-empty parts to minimize the sum of the dimension of the subspace spanned by the parts and (2) partition the edges of a given graph into k non-empty parts to *maximize* the sum of the number of connected components formed by the parts. Coverage k-partition also captures several interesting problems: e.g., (3) partition the vertices of a given graph into k non-empty parts $V_1,V_2,\dots,V_k$ in order to minimize $\sum_{i=1}^{k} f(V_i)$, where $f(P)$ is the number of edges incident to the vertex subset $P\subseteq V$. To gain a better understanding of the difficulty in solving/approximating monotone submodular k-partition, we encourage the reader to briefly think about the concrete special case of matrix column partitioning problem (i.e., problem (1) described above which is seemingly a linear algebra problem) before reading further.

[1] In fact, monotone functions are posimodular since $f(A)\geq f(A-B)$ and $f(B)\geq f(B-A)$ for every $A,B\subseteq V$. Symmetric submodular functions are posimodular: for every $A,B\subseteq V$, we have that $f(A)+f(B)=f(V-A)+f(B)\geq f((V-A)\cup B)+f((V-A)\cap B)=f(V-(A-B))+f(B-A)=f(A-B)+f(B-A)$.

Monotone submodular k-partition is NP-hard [23]. Moreover, it admits a simple (and fast) $(2-1/k)$-approximation algorithm that will be denoted henceforth as the *cheapest singleton partitioning algorithm*: return the partition $V_1 := \{v_1\}, V_2 := \{v_2\}, \ldots, V_{k-1} := \{v_{k-1}\}, V_k := V - \{v_1, \ldots, v_{k-1}\}$), where the n elements of the ground set are ordered as $v_1, \ldots, v_n$ such that $f(\{v_1\}) \le f(\{v_2\}) \le \ldots \le f(\{v_n\})$. Santiago [23] showed that this is a 2-approximation. His analysis can be extended to show that it is in fact a $(2-1/k)$-approximation[2] and this is the best possible approximation factor for this algorithm[3]. Alternatively, the greedy splitting algorithm presented in [26] achieves a $(2-2/k)$-approximation. On the inapproximability front, Santiago [23] showed that there does not exist an algorithm that makes polynomial number of function evaluation queries to obtain a $(4/3-\epsilon)$-approximation for every constant $\epsilon > 0$.

Symmetric submodular k-partition. Well-known special cases of symmetric submodular k-partition problem are graph k-cut and hypergraph k-partition – the submodular functions of interest here are the cut functions of an explicitly given graph and hypergraph respectively. Graph k-cut is NP-complete [10] and does not have a polynomial-time $(2-\epsilon)$-approximation for every constant $\epsilon > 0$ under the Small Set Expansion Hypothesis [12]. There are several known approaches to achieve a 2-approximation for graph k-cut – (i) greedy splitting approach [24], (ii) Gomory-Hu tree based approach [25], (iii) extreme sets based approach [13], (iv) principal partition sequence based approach [2, 18, 22], and (v) covering-LP based approach [7, 15, 21]. Greedy splitting, Gomory-Hu tree, and extreme sets based approaches lead to a $(2-2/k)$-approximation while the principal partition sequence and the covering-LP based approaches lead to a $(2-2/n)$-approximation for graph k-cut. The principal partition sequence and the covering-LP based approaches for graph k-cut have also been shown to be related to each other [7]. The principal partition sequence based approach is the main algorithm of interest to our work and we will discuss it in detail in Section 2.

For the more general problem of symmetric submodular k-partition, two of the approaches discussed in the previous paragraph for graph k-cut have been generalized to obtain 2-approximations – the greedy splitting approach [26] and the Gomory-Hu tree approach lead to a $(2-2/k)$-approximation. Analyzing the approximation factor of the principal partition sequence based approach for symmetric submodular k-partition was one of the driving motivations of our work. On the inapproximability front, Santiago [23] showed that there does not exist an algorithm that makes polynomial number of function evaluation queries to obtain a $(2-\epsilon)$-approximation for every constant $\epsilon > 0$.

[2] The cheapest singleton partitioning algorithm returns a solution whose cost is $f(V - V_k) + \sum_{i=1}^{k-1} f(\{v_i\}) \le f(V) + \sum_{i=1}^{k-1} f(\{v_i\}) \le f(V) + (1-1/k)\sum_{i=1}^{k} f(\{v_i\}) \le (2-1/k)\max\{f(V), \sum_{i=1}^{k} f(\{v_i\})\}$ while the cost of an optimum k-partition is at least $\max\{f(V), \sum_{i=1}^{k} f(\{v_i\})\}$. The lower bound on the cost of the optimum k-partition $V_1^*, \ldots, V_k^*$ is because $\sum_{i=1}^{k} f(V_i^*) \ge f(V)$ by non-negativity and submodularity and moreover, if the optimum partition is indexed such that $\min\{j \in [n] : v_j \in V_i^*\} \le \min\{j \in [n] : v_j \in V_{i+1}^*\}$ for all $i \in [k-1]$, then $f(V_i^*) \ge f(\{v_i\})$ by monotonicity and hence, $\sum_{i=1}^{k} f(V_i^*) \ge \sum_{i=1}^{k} f(\{v_i\})$.

[3] The best possible approximation factor for the cheapest singleton partitioning algorithm is $2-1/k$ as seen from this example: Let f be the rank function of a partition matroid on a k-partition $\{S_1, \ldots, S_k\}$ of the ground set S where $|S_i| \ge 2$ for all $i \in [k]$. Then, the algorithm may return $\{\{s_1\}, \{s_2\}, \ldots, \{s_{k-1}\}, S - \{s_1, \ldots, s_{k-1}\}\}$, where $s_i \in S_i$ for all $i \in [k-1]$ and this k-partition has objective value $2k-1$, whereas the partition $\{S_1, \ldots, S_k\}$ has objective value k.

Posimodular submodular k-partition. The only natural family of posimodular submodular functions that we are familiar with are symmetric submodular functions and monotone submodular functions as well as their positive linear combinations. As mentioned before, posimodular submodular k-partition is a unified generalization of symmetric submodular k-partition and monotone submodular k-partition. To the best of authors' knowledge, posimodular submodular k-partition has not been studied in the literature before and there are no specialized algorithms or approximation factor analysis of existing algorithms for posimodular submodular k-partition. A slight modification to the analysis of the greedy splitting algorithm presented in [26] shows that their algorithm achieves a $(3 - 2/k)$-approximation for posimodular submodular k-partition – we refrain from presenting this analysis in the interests of brevity. On the inapproximability front, since symmetric submodular functions are also posimodular submodular, the lower bound for symmetric submodular k-partition also holds for posimodular submodular k-partition, i.e., there does not exist an algorithm for posimodular submodular k-partition that makes polynomial number of function evaluation queries to obtain a $(2 - \epsilon)$-approximation for every constant $\epsilon > 0$.

1.1 Our Results

In this work, we investigate Narayanan, Roy, and Patkar's [18] principal partition sequence based algorithm for submodular k-partition. They showed that their algorithm achieves a 2-approximation for graph k-cut (which was subsequently rediscovered by Ravi and Sinha [22]). We show the following results:

1. Their algorithm achieves a 4/3-approximation for monotone submodular k-partition. This result improves on the 2-factor that is known to be achievable via two different algorithms: the cheapest singleton partitioning algorithm and the greedy splitting algorithm. Moreover, our upper bound of 4/3 matches the lower bound shown by Santiago [23]. We will discuss the significance of our upper bound result shortly.
2. Their algorithm achieves a 2-approximation for symmetric submodular k-partition. This factor matches the 2-factor that is known to be achievable via two other algorithms: the greedy splitting algorithm and the Gomory-Hu tree based algorithm, and also matches the lower bound [12, 23]. Our contribution here is generalizing the analysis of [18, 22] to beyond graph cut functions.
3. Their algorithm achieves a 2-approximation for posimodular submodular k-partition. This result improves on the 3-factor that is known to be achievable via the greedy splitting algorithm and matches the lower bound of 2 shown by Santiago [23].

See Table 1 for a comparison. Graph k-cut is the well-studied special case of symmetric submodular k-partition/posimodular submodular k-partition, so we include that as the last column in the table for comparison. Approximation factors in the row corresponding to principal partition sequence are the main results of this work. In the last row of the table, we include the known lower bounds on the approximation factor for comparison. The lower bound for graph cut function is assuming the Small Set Expansion Hypothesis [12] while the rest of the lower bounds are assuming polynomial number of function evaluation queries. Dashes in the table indicate that either the approach does not extend or there has been no analysis of the approximation factor of the approach for the subfamily.

We complement our upper bounds on the approximation factor of their algorithm with matching lower bound constructions for each subfamily of submodular functions. Our results show that the principal partition sequence based algorithm achieves the best possible approximation factor for broad subfamilies of submodular functions, thus illustrating the power and applicability of this algorithm. On the other hand, we show that the approximation factor of their algorithm for arbitrary submodular functions is $\Omega(n/k)$ via a lower bound construction. This construction shows that their principal partition sequence based algorithm cannot directly improve the approximation factor for submodular k-partition beyond the current best $O(k)$.

We briefly discuss the significance of our 4/3-approximation result for monotone submodular k-partition. Firstly, prior to our results, there were no known families of submodular functions for which the submodular k-partition problem could be approximated to a factor better than 2. Our result for monotone submodular functions breaks this 2-factor barrier for a broad family of submodular functions. Secondly, our result for monotone submodular k-partition leads to a new approximation result even for a graph partitioning problem that we describe now. For a graph $G = (V, E)$ with edge weights $w : E \to \mathbb{R}_+$, consider functions $d, f : 2^V \to \mathbb{R}_+$ defined by $d(S) := w(\delta(S))$ and $f(S) := w(E[S]) + w(\delta(S))$ for every $S \subseteq V$, where $\delta(S)$ denotes the set of edges with exactly one end-vertex in S, $E[S]$ denotes the set of edges with both end-vertices in S, and $w(F) := \sum_{e\in F} w(e)$ for every $F \subseteq E$. The function d is the cut function of the graph and is symmetric submodular. The function f is the coverage function of the graph and is monotone submodular. Submodular k-partition for the function d is known as graph k-cut and it is known that graph k-cut does not admit a $(2-\epsilon)$-approximation under the Small Set Expansion Hypothesis [12]. In contrast, our results show that coverage k-partition in graphs – i.e., submodular k-partition for the function f – admits a 4/3-approximation. We note that coverage k-partition in graphs is NP-hard [23] and its approximability is an intriguing open question.

Table 1 Approximation factors of symmetric/monotone/posimodular submodular k-partition using different approaches. Result in the first row marked with an asterisk follows by slight modifications to the known analysis of the approximation factor for symmetric submodular functions given in [26].

	Monotone Submodular Function	Symmetric Submodular Function	Posimodular Submodular Function	Graph Cut Function
Greedy splitting	$2-2/k$ [26]	$2-2/k$ [26]	$3-2/k$ [26]*	$2-2/k$ [24]
Extreme Sets	—	—	—	$2-2/k$ [13]
Gomory-Hu tree	—	$2-2/k$ [Folklore]	—	$2-2/k$ [25]
Covering-LP	—	—	—	$2-2/k$ [7, 15]
Cheapest Singleton Partitioning	$2-1/k$ [23]	—	—	—
Principal Partition Sequence	$4/3-4/(9n+3)$ (this work)	$2-2/n$ (this work)	$2-2/(n+1)$ (this work)	$2-2/n$ [18, 22]
Lower Bound	$4/3-\epsilon$ [23]	$2-\epsilon$ [23]	$2-\epsilon$ [23]	$\mathbf{2-\epsilon}$ [12]

Organization. We discuss the principal partition sequence based algorithm in Section 2. We analyze the approximation factor of the algorithm with matching lower bound constructions for monotone submodular k-partition in 3. Due to space limitations, we refer the reader to the full version of the work [5] for the analysis of the approximation factor of the constructionsalgorithm with matching lower bound constructions for symmetric submodular k-partition and posimodular submodular k-partition. We exhibit an instance of submodular k-partition where the algorithm achieves an approximation factor of $\Omega(n/k)$ in Section 4 and conclude with certain open questions in Section 5.

1.2 Related work

The principal partition sequence based algorithm for submodular k-partition was introduced by Narayanan, Roy, and Patkar [18]. We will formally define the principal partition sequence of a submodular function and describe their algorithm in Section 2. They analyzed the approximation factor of their algorithm for two variants of k-partitioning problems in hypergraphs. These two variants are not special cases of symmetric/monotone/posimodular submodular k-partition and are not of direct interest to our work. However, we describe these variants to highlight the versatility of the principal partition sequence based approach and also to shed light on the results of Narayanan, Roy, and Patkar's work which do not seem to be well-known in the literature. Given a hypergraph $H = (V, E)$, a hyperedge cost function $c : E \to \mathbb{R}_+$, and an integer k, the goal is to find a partition $\mathcal{P} := \{V_1, V_2, \ldots, V_k\}$ of V into k non-empty parts that minimizes an objective of interest:

1. If the objective is the sum of cost of hyperedges that intersect at least two parts of $\mathcal{P}$, then the problem is known as *hypergraph k-cut*.

2. If the objective is the sum of *cost of hyperedges relative to the partition* $\mathcal{P}$, where the cost of a hyperedge e relative to $\mathcal{P}$ is $c(e)(\ell - 1)$ with ℓ being the number of parts of $\mathcal{P}$ intersected by e, then the problem is known as *normalized coverage k-partition*[4].

Narayanan, Roy, and Patkar [18] showed that their principal partition sequence based algorithm achieves a $r(1 - 1/n)$-approximation for hypergraph k-cut, where r is the size of the largest hyperedge and n is the number of vertices in the input hypergraph, and achieves a $(2 - 2/n)$-approximation for normalized coverage k-partition. A consequence (of both of their results) is that the principal partition sequence based algorithm achieves a $(2 - 2/n)$-approximation for graph k-cut. Their principal partition sequence based algorithm for graph k-cut is equivalent to the Lagrangean relaxation approach suggested by Barahona [2]. The approximation factor of the principal partition sequence based algorithm for graph k-cut being at most 2 was rediscovered by Ravi and Sinha [22] and for hypergraph k-cut being at most r was rediscovered by Baïou and Barahona [1].

We mention that a slight modification to the analysis of the greedy splitting algorithm presented in [26] shows that the greedy splitting algorithm achieves a $(2-2/k)$-approximation for normalized coverage k-partition and hypergraph k-partition. We note that hypergraph k-partition is a special case of symmetric submodular k-partition and is different from hypergraph k-cut (for definition of hypergraph k-partition, see discussion of symmetric submodular k-partition at the beginning of the introduction). On the inapproximability front, it is known that hypergraph k-cut does not admit an approximation factor of $n^{1/(\log \log n)^c}$, where c is a constant, assuming the Exponential Time Hypothesis [6]. The best inapproximability result for the more general submodular k-partition problem (mentioned in the introduction) follows from this inapproximability for hypergraph k-cut.

[4] We introduce this nomenclature because the problem is equivalent to finding a partition $V_1, V_2, \ldots, V_k$ of the ground set V in order to minimize $\sum_{i=1}^{k} f(V_i) - f(V)$, where $f : 2^V \to \mathbb{R}_+$ is an explicitly given coverage function (every coverage function can be uniquely represented using a hypergraph [3]). We consider the subtraction of $f(V)$ as normalizing the objective since it is a trivial lower bound on the sum of the function values of the parts: $\sum_{i=1}^{k} f(V_i) \geq f(V)$ holds for every k-partition $V_1, V_2, \ldots, V_k$ since f is a coverage function.

2 Principal partition sequence based algorithm

In this section, we recall the principal partition sequence based algorithm for submodular k-partition designed by Narayanan, Roy, and Patkar [18]. We begin with some notation. Throughout this work, a *partition* of a set S is defined to be a collection of *nonempty* pairwise disjoint subsets of S whose union is S, and a *k-partition* of a set S is defined to be a partition of S with exactly k parts. For two distinct partitions $\mathcal{P}$ and $\mathcal{Q}$ of a set S, if every part of $\mathcal{Q}$ is completely contained in some part of $\mathcal{P}$ then we say that $\mathcal{Q}$ *refines* $\mathcal{P}$ (equivalently, $\mathcal{P}$ is a coarsening of $\mathcal{Q}$). For two distinct partitions $\mathcal{P}$ and $\mathcal{Q}$ of a set S, we will say that $\mathcal{Q}$ is obtained from $\mathcal{P}$ by refining only one part of $\mathcal{P}$ if there exists a part $P \in \mathcal{P}$ such that $P \notin \mathcal{Q}$ and every part $Q \in \mathcal{Q}$ satisfies either $Q \subsetneq P$ or $Q \in \mathcal{P}$ (i.e., either Q is a proper subset of the part P or Q is a part of the partition $\mathcal{P}$); we will denote such a part $P \in \mathcal{P}$ as the part refined by $\mathcal{Q}$.

Let $f : 2^V \to \mathbb{R}$ be a set function on ground set V. For a collection $\mathcal{P}$ of subsets of V, we write $f(\mathcal{P}) := \sum_{P\in\mathcal{P}} f(P)$. We will say that a partition $\mathcal{P} = \{P_1, \ldots, P_k\}$ is an optimal k-partition if $f(\mathcal{P}) \le f(\mathcal{Q})$ for every k-partition $\mathcal{Q}$ of V. We define the function $g_{f,\mathcal{P}} : \mathbb{R}_{\ge 0} \to \mathbb{R}$ for a partition $\mathcal{P}$ of the ground set V and the function $g_f : \mathbb{R}_{\ge 0} \to \mathbb{R}$ as follows:

$$\begin{aligned} g_{f,\mathcal{P}}(b) &:= f(\mathcal{P}) - b|\mathcal{P}| \text{ and} \\ g_f(b) &:= \min\{g_{f,\mathcal{P}}(b) : \mathcal{P} \text{ is a partition of } V\}. \end{aligned}$$

We drop the subscript f and instead write $g_{\mathcal{P}}$ and g respectively, if the function f is clear from context. By definition, the function g_f is piece-wise linear. It can be shown that g_f has at most $|V| - 1$ breakpoints. The next theorem shows that if the function $f : 2^V \to \mathbb{R}$ is submodular, then there exists a sequence of partitions achieving the g_f function values at the breakpoints that have a nice structure; moreover, the breakpoints and such a sequence of partitions can be computed in polynomial time given access to the evaluation oracle of the submodular function f. We emphasize that the theorem holds for arbitrary submodular functions (which may not be non-negative valued).

▶ **Theorem 1** ([16, 18]). *Let $f : 2^V \to \mathbb{R}$ be a submodular function on a ground set V. Then, there exists a sequence $\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_r$ of partitions of V and values $b_1, b_2, \ldots, b_{r-1}$ such that*

1. *$\mathcal{P}_1 = \{V\}$ and $\mathcal{P}_r = \{\{v\} : v \in V\}$,*
2. *For each $j \in [r-1]$, the partition $\mathcal{P}_{j+1}$ is obtained from $\mathcal{P}_j$ by refining only one part of $\mathcal{P}_j$,*
3. *$b_1 < b_2 < \ldots < b_{r-1}$,*
4. *$g(b_j) = g_{\mathcal{P}_j}(b_j) = g_{\mathcal{P}_{j+1}}(b_j)$ for each $j \in [r-1]$ and*
5. *$g(b) = g_{\mathcal{P}_1}(b)$ for all $b \in (-\infty, b_1]$,*
   $g(b) = g_{\mathcal{P}_{j+1}}(b)$ for all $b \in [b_j, b_{j+1}]$ for each $j \in [r-2]$, and
   $g(b) = g_{\mathcal{P}_r}(b)$ for all $b \in [b_{r-1}, \infty)$.

Moreover, such a sequence $\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_r$ of partitions of V and values $b_1, b_2, \ldots, b_{r-1}$ can be computed in polynomial time given access to the evaluation oracle of the submodular function f.

For a submodular function $f : 2^V \to \mathbb{R}$, we will denote a sequence of partitions $\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_r$ and the sequence of values $b_1, b_2, \ldots, b_{r-1}$ satisfying the conditions given in Theorem 1 as a *principal partition sequence* and the *critical value sequence* of f, respectively. We note that this definition differs from those in [16, 18] owing to the reversed indexing order and the imposition of condition 2 – we note that the proofs given in those papers also show that condition 2 holds (also see [22]). The principal partition sequence of submodular functions is known in the literature as *principal lattice of partitions* of submodular functions since there exists a lattice structure associated with the sequence of partitions. We choose to call it as principal partition sequence in this work since the sequence suffices for our purpose. For more on principal lattice of partitions of submodular functions and their computation, we refer the reader to [2, 8, 9, 11, 14, 16–18, 20].

We now discuss the principal partition sequence based algorithm for submodular k-partition that was proposed by Narayanan, Roy, and Patkar [18]. This algorithm computes a principal partition sequence satisfying all conditions in Theorem 1. If the sequence contains a partition that has exactly k parts, then the algorithm returns this k-partition. Otherwise, the algorithm returns a k-partition obtained by refining the partition in the sequence that has the largest number of parts that is less than k. The refinement is based on the partition in the sequence that has the fewest number of parts that is more than k. The formal description of the refinement is given in Algorithm 1. Since the sequence $\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_r$ satisfying the conditions of Theorem 1 can be computed in polynomial time, Algorithm 1 can indeed be implemented to run in polynomial time. By design, the algorithm returns a k-partition. The remainder of this work will focus on analyzing the approximation factor of the algorithm.

Algorithm 1 Principal partition sequence based algorithm for submodular k-partition.

```
Input: A submodular function $f : 2^V \to \mathbb{R}$ given by evaluation oracles and an integer $k \geq 2$.
Output: A $k$-partition $\mathcal{P}$ of $V$.
Use Theorem 1 to compute a principal partition sequence $\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_r$ of the submodular function $f$ satisfying all conditions stated in that theorem.
if $|\mathcal{P}_j| = k$ for some $j \in [r]$ then
  Return $\mathcal{P} := \mathcal{P}_j$.
end if
Let $i \in \{2, 3, \ldots, r\}$ so that $|\mathcal{P}_{i-1}| < k < |\mathcal{P}_i|$.
Let $S \in \mathcal{P}_{i-1}$ be the part refined by $\mathcal{P}_i$ and $\mathcal{P}'$ be the parts of $\mathcal{P}_i$ contained in $S$.
Let $\mathcal{P}' = \{B_1, \ldots, B_{|\mathcal{P}'|}\}$ such that $f(B_1) \leq f(B_2) \leq \ldots \leq f(B_{|\mathcal{P}'|})$.
Return $\mathcal{P} := (\mathcal{P}_{i-1} \backslash \{S\}) \cup \left\{B_1, B_2, \ldots, B_{k-|\mathcal{P}_{i-1}|}\right\} \cup \left\{\bigcup_{j=k-|\mathcal{P}_{i-1}|+1}^{|\mathcal{P}'|} B_j\right\}$.
```

To construct examples that exhibit tight lower bound on the approximation factor of Algorithm 1, we will need the following proposition that identifies a special case under which the principal partition is unique and consists only of two partitions – namely, the partition into singletons and the partition that consists of only one part. We refer the reader to the full version of this work for the proof [5].

▶ **Proposition 2.** *Let $f : 2^V \to \mathbb{R}_{\geq 0}$ be a non-negative submodular function. Suppose that for every partition $\mathcal{P} \neq \mathcal{Q}, \{V\}$ where $\mathcal{Q} := \{\{v\} : v \in V\}$, the function f satisfies*

$$\frac{f(\mathcal{P}) - f(V)}{|\mathcal{P}| - 1} > \frac{f(\mathcal{Q}) - f(V)}{|V| - 1}.$$

Then, the principal partition sequence of f is $\{V\}, \mathcal{Q}$.

3 Approximation factor analysis

In this section, we analyze the approximation factor of Algorithm 1 for various subfamilies of submodular functions. We state and prove certain lemmas that will be useful for all submodular functions (Lemmas 3 and 4). We analyze the approximation factor of Algorithm 1 for monotone submodular functions in Section 3.1. We refer the reader to the full version of this work [5] for the analysis of the approximation factor of Algorithm 1 for symmetric submodular functions and for posimodular submodular functions.

Our first lemma identifies a special case in which Algorithm 1 returns an optimum k-partition. This special case was also identified by Narayanan, Roy, and Patkar. We note that the following lemma holds for arbitrary submodular functions (which may not be non-negative).

▶ **Lemma 3** ([18]). *Let $k \geq 2$ be an integer, $f : 2^V \to \mathbb{R}$ be a submodular function on a ground set V, and $\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_r$ be a principal partition sequence of the submodular function f satisfying the conditions of Theorem 1. If there exists $j \in [r]$ such that $|\mathcal{P}_j| = k$, then $\mathcal{P}_j$ is an optimal k-partition.*

Proof. Let $\mathcal{P}^*$ be a k-partition of V that minimizes $f(\mathcal{P}^*)$ and let b_j be the value where $g(b_j) = g_{\mathcal{P}_j}(b_j)$. Then,

$$f(\mathcal{P}^*) - b_j \cdot k = g_{\mathcal{P}^*}(b_j) \geq g(b_j) = g_{\mathcal{P}_j}(b_j) = f(\mathcal{P}_j) - b_j|\mathcal{P}_j| = f(\mathcal{P}_j) - b_j \cdot k,$$

and hence, $f(\mathcal{P}^*) \geq f(\mathcal{P}_j)$. Therefore, $\mathcal{P}_j$ is indeed an optimal k-partition. ◀

In order to address the case where there is no $j \in [r]$ such that $|\mathcal{P}_j| = k$, we need the following lemma that shows two lower bounds on the optimum value. The first lower bound in the lemma holds for arbitrary submodular functions (which may not be non-negative) while the second lower bound holds for non-negative submodular functions.

▶ **Lemma 4.** *Let $k \geq 2$ be an integer, $f : 2^V \to \mathbb{R}$ be a submodular function on a ground set V, $\mathcal{P}^*$ be a k-partition of V that minimizes $f(\mathcal{P}^*)$, and $\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_r$ be a principal partition sequence of the submodular function f satisfying the conditions of Theorem 1. Suppose $|\mathcal{P}_j| \neq k$ for all $j \in [r]$. Let $\mathcal{P}_{i-1}, \mathcal{P}_i$ be the partitions such that $|\mathcal{P}_{i-1}| < k < |\mathcal{P}_i|$. Then,*

(i) *$f(\mathcal{P}^*) \geq \frac{|\mathcal{P}_i|-k}{|\mathcal{P}_i|-|\mathcal{P}_{i-1}|} f(\mathcal{P}_{i-1}) + \frac{k-|\mathcal{P}_{i-1}|}{|\mathcal{P}_i|-|\mathcal{P}_{i-1}|} f(\mathcal{P}_i)$ and*

(ii) *$f(\mathcal{P}^*) \geq f(\mathcal{P}_{i-1})$ if f is non-negative.*

Proof. We prove the two lower bounds below.

(i) Let b_{i-1} be the value such that $g_{\mathcal{P}_{i-1}}(b_{i-1}) = g_{\mathcal{P}_i}(b_{i-1})$. Then, we have

$$f(\mathcal{P}_{i-1}) - b_{i-1}|\mathcal{P}_{i-1}| = f(\mathcal{P}_i) - b_{i-1}|\mathcal{P}_i| \implies b_{i-1} = \frac{f(\mathcal{P}_i) - f(\mathcal{P}_{i-1})}{|\mathcal{P}_i| - |\mathcal{P}_{i-1}|}. \tag{1}$$

By condition 4 of Theorem 1, we also have

$$\begin{aligned} f(\mathcal{P}^*) - b_{i-1} \cdot k = g_{\mathcal{P}^*}(b_{i-1}) \geq g(b_{i-1}) = g_{\mathcal{P}_i}(b_{i-1}) = f(\mathcal{P}_i) - b_{i-1}|\mathcal{P}_i| \\ \implies f(\mathcal{P}^*) \geq f(\mathcal{P}_i) + b_{i-1}(k - |\mathcal{P}_i|). \end{aligned} \tag{2}$$

Combining (1) and (2), we get

$$\begin{aligned} f(\mathcal{P}^*) &\geq f(\mathcal{P}_i) + \frac{f(\mathcal{P}_i) - f(\mathcal{P}_{i-1})}{|P_i| - |P_{i-1}|}(k - |\mathcal{P}_i|) \\ &= \frac{|\mathcal{P}_i| - k}{|\mathcal{P}_i| - |\mathcal{P}_{i-1}|} f(\mathcal{P}_{i-1}) + \frac{k - |\mathcal{P}_{i-1}|}{|\mathcal{P}_i| - |\mathcal{P}_{i-1}|} f(\mathcal{P}_i). \end{aligned}$$

(ii) Let $P_1^*, \ldots, P_k^*$ be the parts of $\mathcal{P}^*$ (in arbitrary order). Let $k' := |\mathcal{P}_{i-1}|$. We know that $k' < k$. Consider the k'-partition $\mathcal{Q}$ obtained as $Q_1 := P_1^*, Q_2 := P_2^*, \ldots, Q_{k'-1} := P_{k'-1}^*, Q_{k'} := \cup_{j=k'}^{k} P_j^*$. Then,

$$f(\mathcal{P}^*) = \sum_{i=1}^{k} f(P_i^*) \geq \left(\sum_{i=1}^{k'-1} f(P_i^*)\right) + f(\cup_{j=k'}^{k} P_j^*) = \sum_{i=1}^{k'} f(Q_i) = f(\mathcal{Q}).$$

The inequality above is due to submodularity and non-negativity. The partition $\mathcal{Q}$ is a k'-partition. By Lemma 3, the partition $\mathcal{P}_{i-1}$ is an optimal k'-partition. Hence,

$$f(\mathcal{Q}) \geq f(\mathcal{P}_{i-1}).$$

The above two inequalities together imply that $f(\mathcal{P}^*) \geq f(\mathcal{P}_{i-1})$. ◀

3.1 Monotone submodular functions

In this section, we bound the approximation factor of Algorithm 1 for monotone submodular k-partitioning. The following is the main theorem of this section.

▶ **Theorem 5.** *The approximation factor of Algorithm 1 for non-negative monotone submodular k-partitioning is $\frac{4}{3} - \frac{4}{9n+3}$, where n is the size of the ground set.*

The asymptotic approximation factor of 4/3 achieved by Algorithm 1 is the best possible for non-negative monotone submodular k-partition: for every constant $\epsilon > 0$, there does not exist an algorithm that achieves a $(4/3 - \epsilon)$-approximation using polynomial number of function evaluation queries [23]. We will also exhibit examples to show the tightness of the approximation factor for Algorithm 1 after proving the theorem. The proof of Theorem 5 follows from Lemma 3 and Lemma 6 shown below.

▶ **Lemma 6.** *Let $k \geq 2$ be an integer, $f : 2^V \to \mathbb{R}_{\geq 0}$ be a non-negative monotone submodular function on a ground set V of size n, $\mathcal{P}^*$ be a k-partition of V that minimizes $f(\mathcal{P}^*)$, and $\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_r$ be a principal partition sequence of the submodular function f satisfying the conditions of Theorem 1. Suppose $|\mathcal{P}_j| \neq k$ for all $j \in [r]$. Then, the partition $\mathcal{P}$ returned by Algorithm 1 satisfies*

$$f(\mathcal{P}) \leq \left(\frac{4}{3} - \frac{4}{9n+3}\right) f(\mathcal{P}^*).$$

Proof. Let $\mathcal{P}_{i-1}, \mathcal{P}_i$ be the partitions such that $|\mathcal{P}_{i-1}| < k < |\mathcal{P}_i|$. Let S and $\mathcal{P}' = \{B_1, B_2, \ldots, B_{|\mathcal{P}'|}\}$ be as in Algorithm 1.

Firstly, since $\cup_{j=k-|\mathcal{P}_{i-1}|+1}^{|\mathcal{P}'|} B_j \subseteq S$ and f is monotone, we have that

$$f\left(\cup_{j=k-|\mathcal{P}_{i-1}|+1}^{|\mathcal{P}'|} B_j\right) \leq f(S).$$

Secondly, by our choice of $B_1, B_2, \ldots, B_{k-|\mathcal{P}_{i-1}|}$, we know that

$$\sum_{j=1}^{k-|\mathcal{P}_{i-1}|} f(B_j) \leq \frac{k - |\mathcal{P}_{i-1}|}{|\mathcal{P}'|} f(\mathcal{P}').$$

Hence,

$$\begin{aligned}
f(\mathcal{P}) &= f(\mathcal{P}_{i-1}) - f(S) + \sum_{j=1}^{k-|\mathcal{P}_{i-1}|} f(B_j) + f\left(\bigcup_{j=k-|\mathcal{P}_{i-1}|+1}^{|\mathcal{P}'|} B_j\right)\\
&\leq f(\mathcal{P}_{i-1}) + \sum_{j=1}^{k-|\mathcal{P}_{i-1}|} f(B_j)\\
&\leq f(\mathcal{P}_{i-1}) + \frac{k-|\mathcal{P}_{i-1}|}{|\mathcal{P}'|} f(\mathcal{P}')\\
&= f(\mathcal{P}_{i-1}) + \frac{k-|\mathcal{P}_{i-1}|}{|\mathcal{P}_i|-|\mathcal{P}_{i-1}|+1}(f(\mathcal{P}_i)+f(S)-f(\mathcal{P}_{i-1})),
\end{aligned}$$

where the last equality follows from the fact that $f(\mathcal{P}_{i-1}) - f(S) + f(\mathcal{P}') = f(\mathcal{P}_i)$ and $|\mathcal{P}'| = |\mathcal{P}_i| - |\mathcal{P}_{i-1}| + 1$. Rearranging, we have that

$$\begin{aligned}
f(\mathcal{P}) &\leq \frac{f(\mathcal{P}_{i-1})}{|\mathcal{P}_i|-|\mathcal{P}_{i-1}|+1}(|\mathcal{P}_i|-k+1) + \frac{f(\mathcal{P}_i)}{|\mathcal{P}_i|-|\mathcal{P}_{i-1}|+1}(k-|\mathcal{P}_{i-1}|)\\
&\qquad + \frac{f(S)}{|\mathcal{P}_i|-|\mathcal{P}_{i-1}|+1}(k-|\mathcal{P}_{i-1}|)\\
&= \left(\frac{|\mathcal{P}_i|-|\mathcal{P}_{i-1}|}{|\mathcal{P}_i|-|\mathcal{P}_{i-1}|+1}\right)\left(\frac{f(\mathcal{P}_{i-1})}{|\mathcal{P}_i|-|\mathcal{P}_{i-1}|}(|\mathcal{P}_i|-k+1) + \frac{f(\mathcal{P}_i)+f(S)}{|\mathcal{P}_i|-|\mathcal{P}_{i-1}|}(k-|\mathcal{P}_{i-1}|)\right)\\
&\leq \left(\frac{|\mathcal{P}_i|-|\mathcal{P}_{i-1}|}{|\mathcal{P}_i|-|\mathcal{P}_{i-1}|+1}\right)\left(f(\mathcal{P}^*) + \frac{f(\mathcal{P}_{i-1})}{|\mathcal{P}_i|-|\mathcal{P}_{i-1}|} + \frac{f(S)}{|\mathcal{P}_i|-|\mathcal{P}_{i-1}|}(k-|\mathcal{P}_{i-1}|)\right)\\
&\leq \left(\frac{|\mathcal{P}_i|-|\mathcal{P}_{i-1}|}{|\mathcal{P}_i|-|\mathcal{P}_{i-1}|+1}\right)\left(f(\mathcal{P}^*) + \frac{f(\mathcal{P}_{i-1})}{|\mathcal{P}_i|-|\mathcal{P}_{i-1}|}(1+k-|\mathcal{P}_{i-1}|)\right). \qquad (3)
\end{aligned}$$

where the second inequality above is by by Lemma 4(i) and the third inequality above is because $f(S) \leq f(\mathcal{P}_{i-1})$. Inequality (3) implies that

$$\begin{aligned}
f(\mathcal{P}) &\leq \left(\frac{|\mathcal{P}_i|-|\mathcal{P}_{i-1}|}{|\mathcal{P}_i|-|\mathcal{P}_{i-1}|+1}\right)\left(f(\mathcal{P}^*) + f(\mathcal{P}_{i-1}) + \frac{f(\mathcal{P}_{i-1})}{|\mathcal{P}_i|-|\mathcal{P}_{i-1}|}(1+k-|\mathcal{P}_i|)\right)\\
&\leq \left(1 - \frac{1}{|\mathcal{P}_i|-|\mathcal{P}_{i-1}|+1}\right)(f(\mathcal{P}^*)+f(\mathcal{P}_{i-1})) \quad \text{(since } k<|\mathcal{P}_i|\text{)}\\
&\leq 2f(\mathcal{P}^*),
\end{aligned}$$

where the last inequality is because $f(\mathcal{P}_{i-1}) \leq f(\mathcal{P}^*)$ by Lemma 4(ii). The above analysis shows that the approximation factor is at most 2. We tighten the analysis now. As a consequence of the above inequality, we may assume that $f(\mathcal{P}^*) \neq 0$ because if $f(\mathcal{P}^*) = 0$, then the returned k-partition $\mathcal{P}$ also satisfies $f(\mathcal{P}) = 0$ and thus, is optimal. Let $c := f(\mathcal{P}_{i-1})/f(\mathcal{P}^*)$. By Lemma 4(ii), we have that $f(\mathcal{P}_{i-1}) \leq f(\mathcal{P}^*)$ and hence, $c \in [0,1]$. For convenience, we define $A := k - |\mathcal{P}_{i-1}|$ and $B := |\mathcal{P}_i| - k$ and note that $A, B \geq 1$. Using this notation, we may rewrite inequality (3) as

$$\begin{aligned}
f(\mathcal{P}) &\leq \left(\frac{A+B}{A+B+1}\right)\left(f(\mathcal{P}^*) + \frac{1+A}{A+B}f(\mathcal{P}_{i-1})\right)\\
&= \left(\frac{A+B}{A+B+1}\right)\left(1 + \frac{1+A}{A+B}\cdot c\right)f(\mathcal{P}^*). \qquad (4)
\end{aligned}$$

By Lemma 4(i), we have

$$f(\mathcal{P}^*) \geq \left(\frac{B}{A+B}\right)f(\mathcal{P}_{i-1}) + \left(\frac{A}{A+B}\right)f(\mathcal{P}_i) = \left(\frac{B}{A+B}\right)cf(\mathcal{P}^*) + \left(\frac{A}{A+B}\right)f(\mathcal{P}_i).$$

Rearranging, we have

$$f(\mathcal{P}_i) \le \left(1 - \frac{B}{A+B} \cdot c\right)\left(\frac{A+B}{A}\right) f(\mathcal{P}^*) = \left(\frac{A+B}{A} - \frac{B}{A} \cdot c\right) f(\mathcal{P}^*).$$

Since $\mathcal{P}$ is obtained by coarsening $\mathcal{P}_i$, we have $f(\mathcal{P}) \le f(\mathcal{P}_i)$ by submodularity and non-negativity of f. This implies

$$f(\mathcal{P}) \le \left(\frac{A+B}{A} - \frac{B}{A} \cdot c\right) f(\mathcal{P}^*). \tag{5}$$

Combining inequalities (4) and (5), we have

$$\frac{f(\mathcal{P})}{f(\mathcal{P}^*)} \le \max_{c \in [0,1]} \min\left\{\left(\frac{A+B}{A+B+1}\right)\left(1 + \frac{1+A}{A+B} \cdot c\right), \frac{A+B}{A} - \frac{B}{A} \cdot c\right\}. \tag{6}$$

Thus, in order to upper bound the approximation factor, it suffices to upper bound the right hand side of inequality (6). Since $\left(\frac{A+B}{A+B+1}\right)\left(1 + \frac{1+A}{A+B} \cdot c\right)$ and $\frac{A+B}{A} - \frac{B}{A} \cdot c$ are both linear in c, with the former increasing and the latter decreasing as a function of c, the value

$$\max_{c \in \mathbb{R}} \min\left\{\left(\frac{A+B}{A+B+1}\right)\left(1 + \frac{1+A}{A+B} \cdot c\right), \frac{A+B}{A} - \frac{B}{A} \cdot c\right\}$$

is achieved when the two terms are equal. Setting $\left(\frac{A+B}{A+B+1}\right)\left(1 + \frac{1+A}{A+B} \cdot c^*\right) = \frac{A+B}{A} - \frac{B}{A} \cdot c^*$ and solving for c^*, we get

$$c^* = \frac{\frac{A+B}{A} - \frac{A+B}{A+B+1}}{\frac{1+A}{A+B+1} + \frac{B}{A}} = \frac{\frac{B}{A} + \frac{1}{A+B+1}}{\frac{1+A}{A+B+1} + \frac{B}{A}}.$$

Plugging $c = c^*$ into $\frac{A+B}{A} - \frac{B}{A} \cdot c$ yields

$$\begin{aligned}
\frac{f(\mathcal{P})}{f(\mathcal{P}^*)} &\le \max_{c \in [0,1]} \min\left\{\frac{A+B}{A+B+1}\left(1 + \frac{1+A}{A+B} \cdot c\right), \frac{A+B}{A} - \frac{B}{A} \cdot c\right\} \\
&\le \frac{A+B}{A} - \frac{B}{A} \cdot c^* = 1 + \frac{B}{A}(1 - c^*) = 1 + \frac{B}{A}\left(1 - \frac{\frac{B}{A} + \frac{1}{A+B+1}}{\frac{1+A}{A+B+1} + \frac{B}{A}}\right) \\
&= 1 + \frac{AB}{A + A^2 + AB + B^2 + B} \\
&\le 1 + \frac{AB}{3AB + A + B} \quad (\text{since } A^2 + B^2 \ge 2AB) \\
&= \frac{4}{3} - \frac{1}{3} \cdot \frac{A+B}{3AB/2 + 3AB/2 + A + B} \\
&\le \frac{4}{3} - \frac{1}{3} \cdot \frac{A+B}{3AB/2 + 3(A^2+B^2)/4 + A + B} \quad (\text{since } AB \le \frac{A^2+B^2}{2}) \\
&= \frac{4}{3} - \frac{1}{3} \cdot \frac{1}{3(A+B)/4 + 1} \\
&\le \frac{4}{3} - \frac{4}{9n+3}.
\end{aligned}$$

The last inequality above is because $A + B = |\mathcal{P}_i| - |\mathcal{P}_{i-1}| \le n - 1$. ◀

▶ Remark 7. The approximation factor of Algorithm 1 for non-negative monotone submodular functions is at least $4/3 - 4/(9n+3)$. We show this for $n = 3$ using the following example: Let $V = \{a, b, c\}$, $k = 2$, and $f : 2^V \to \mathbb{R}_{\geq 0}$ be defined by

$$f(\emptyset) = 0,\ f(\{a\}) = 1,\ f(\{b\}) = f(\{c\}) = 1 + \epsilon,$$
$$f(\{a, b\}) = f(\{a, c\}) = \frac{3}{2} + \epsilon,\ f(\{b, c\}) = f(V) = 2 + 2\epsilon.$$

Submodularity and monotonicity of f can be verified by considering all possible subsets. Moreover, the principal partition sequence of this instance is $\{V\}, \{\{a\}, \{b\}, \{c\}\}$. Thus, Algorithm 1 returns the 2-partition $\{\{a\}, \{b, c\}\}$, whose objective value is $3+2\epsilon$. An optimum 2-partition is $\{\{c\}, \{a, b\}\}$, whose objective value is $5/2 + \epsilon$. Thus, the approximation factor is $\frac{3+2\epsilon}{5/2+\epsilon} \to \frac{6}{5}$ as $\epsilon \to 0$. We note that for $n = 3$, the approximation factor guaranteed by Theorem 5 is $\frac{4}{3} - \frac{4}{9n+3} = \frac{6}{5}$.

We conclude the section by showing that there exist monotone submodular functions for which the approximation factor of Algorithm 1 is at least 4/3 asymptotically (i.e., as $n \to \infty$).

▶ **Lemma 8.** *For every odd positive integer $n \geq 5$, there exists a function $k = k(n)$ (i.e., k is a function of n) and an instance of non-negative monotone submodular k-partition over an n-element ground set such that the approximation factor of Algorithm 1 on that instance is arbitrarily close to $4/3 - 4/(3n+3)$.*

Proof. Let $n \geq 5$ be an arbitrary odd number, $k = \frac{n+1}{2}$, and let $V = \{v_1, \ldots, v_n\}$ be the ground set. Moreover, let $U := \{v_1, \ldots, v_{\frac{n-1}{2}}\}$ and $D := \{v_{\frac{n+1}{2}}, \ldots, v_n\}$ so that $V = U \uplus D$. Let $g : 2^U \to \mathbb{R}_{\geq 0}$ be a function over the ground set U defined by

$$g(S) = \begin{cases} \frac{1}{2} + \frac{1}{2} \cdot |S| & \text{if } \emptyset \neq S \subseteq U, \\ 0 & \text{if } S = \emptyset. \end{cases}$$

and $f : 2^V \to \mathbb{R}_{\geq 0}$ be defined by

$$f(S) := \min\left\{g(S \cap U) + (1+\epsilon)|S \cap D|,\ \frac{n+1}{2}\right\}\ \forall\ S \subseteq V,$$

where $\epsilon > 0$ is infinitesimally small. The function f satisfies $f(\emptyset) = 0$, $f(V) = \frac{n+1}{2}$, $f(U) = \frac{n+1}{4}$, $f(D) = \frac{n+1}{2}$, $f(\{v\}) = 1$ for each $v \in U$, and $f(\{v\}) = 1 + \epsilon$ for each $u \in D$. We will use $\mathcal{Q}$ to denote the partition of V into n singleton sets. The following claims show properties about the function f. We refer the reader to the full version of this work for the proof [5].

▷ Claim 9. The function f is submodular and monotone.

▷ Claim 10. For every partition $\mathcal{P} \neq \mathcal{Q}, \{V\}$, the function f satisfies

$$\frac{f(\mathcal{P}) - f(V)}{|\mathcal{P}| - 1} > \frac{f(\mathcal{Q}) - f(V)}{n - 1}.$$

By Proposition 2, Algorithm 1 returns the partition $\{\{v_1\}, \ldots, \{v_{\frac{n-1}{2}}\}, D\}$ (because $\{v_1\}, \ldots, \{v_{\frac{n-1}{2}}\}$ are the $k-1$ singleton sets that minimize the f values), whose objective is $\frac{n-1}{2} + \frac{n+1}{2} = n$. The partition $\{\{v_1, \ldots, v_{\frac{n+1}{2}}\}, \{v_{\frac{n+3}{2}}\}, \ldots, \{v_n\}\}$ has objective $\frac{n+1}{4} + (1 + \epsilon) + (1+\epsilon)\frac{n-1}{2} = \frac{3n+3}{4} + \frac{(n+1)\epsilon}{2}$. The approximation factor is at least

$$\frac{n}{\frac{3n+3}{4} + \frac{(n+1)\epsilon}{2}} \to \frac{4}{3} - \frac{4}{3n+3} \text{ (as } \epsilon \to 0).$$

This completes the proof of Lemma 8. ◀

4 Lower bound for arbitrary submodular functions

In this section, we present an instance of submodular k-partition where Algorithm 1 achieves an approximation factor of $\Omega(n/k)$. We emphasize that the submodular function in our instance is not symmetric/monotone/posimodular.

Let $V = \{v_0, v_1, \ldots, v_{n-1}\}$ be the ground set. We define a digraph $D = (V, E(D))$ and a hypergraph $H = (V, E(H))$ on the same vertex set V as follows (see Fig.1):

$$E(D) = \{v_0 v_i : i \in [n-1]\} \text{ and}$$
$$E(H) = \{\{v_1, v_2, \ldots, v_{n-1}\}\}.$$

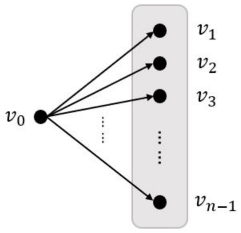

Figure 1 Example in Section 4. The arcs belong to the digraph D and the hyperedge $\{v_1, \ldots, v_{n-1}\}$ belongs to the hypergraph H.

For every subset $S \subseteq V$, we will use $d_D^{in}(S)$ to denote the number of arcs in D whose tails are in $\bar{S}$ and heads are in S. We will use $d_H(S)$ to denote the number of hyperedges in H that have at least one vertex in S and one vertex in $\bar{S}$. Next, we define a set function $f : 2^V \to \mathbb{R}_{\geq 0}$ by

$$f(S) := a \cdot d_D^{in}(S) + d_H(S) \quad \forall S \subseteq V,$$

where $a \gg 1$ is a large constant. We note that f is submodular because it is a positive linear combination of two submodular functions (and it is not monotone/symmetric/posimodular).

▷ Claim 11. The principal partition sequence of f is $\{V\}, \{\{v_i\} : i \in \{0, 1, \ldots, n-1\}\}$.

Proof. For convenience, we will use $\mathcal{Q}$ to denote the partition of V into singletons. By Proposition 2 and the fact that $f(V) = 0$, it suffices to prove that for every partition $\mathcal{P}$ of V such that $\mathcal{P}$ is not $\mathcal{Q}$ or $\{V\}$, we have that

$$\frac{f(\mathcal{P})}{|\mathcal{P}| - 1} > \frac{f(\mathcal{Q})}{n-1}. \tag{7}$$

Let $P_0 \in \mathcal{P}$ be the part that contains v_0. Then, we have that $f(P_0) \geq 1$ if $P_0 \neq \{v_0\}$ and $f(P_0) = 0$ otherwise. For each part $P \in \mathcal{P}$ that does not contain v_0, we have that $f(P) \geq 1 + a$ if $|P| = 1$ and $f(P) \geq 2 + a$ if $|P| \geq 2$. Since $\mathcal{P} \neq \mathcal{Q}$, we have that either $P_0 \neq \{v_0\}$ or at least one of the parts $P \in \mathcal{P} \backslash \{P_0\}$ has size $|P| \geq 2$. Thus, $f(\mathcal{P}) = \sum_{P \in \mathcal{P}} f(P) \geq (1+a)(|\mathcal{P}|-1)+1$. Moreover, we have $f(\mathcal{Q}) = (1 + a)(n - 1)$ and hence

$$\frac{f(\mathcal{P})}{|\mathcal{P}| - 1} \geq \frac{(1+a)(|\mathcal{P}|-1)+1}{|\mathcal{P}|-1} = 1 + a + \frac{1}{|\mathcal{P}|-1} > 1 + a = \frac{(1+a)(n-1)}{n-1} = \frac{f(\mathcal{Q})}{n-1}.$$

This proves inequality (7). ◁

▷ Claim 12. The approximation factor of Algorithm 1 on input (f, k) is $\Omega(n/k)$.

Proof. We note that $f(\{v_0\}) = 0$ and $f(\{v_i\}) = 1 + a$ for all $i \in [n-1]$. By Claim 11, on input (f, k), Algorithm 1 returns a partition $\mathcal{P}$ consisting of the $k-1$ singleton parts that minimizes f among all singleton sets and the complement of the union of these $k-1$ singleton parts. Therefore, the returned partition $\mathcal{P}$ contains $\{v_0\}$ as a part and thus

$$f(\mathcal{P}) = \sum_{P \in \mathcal{P}: P \neq \{v_0\}} f(P) \geq f(V - \{v_0\}) = a(n-1).$$

The second inequality follows from submodularity of the function f. Consider the k-partition $\{\{v_1\}, \{v_2\}, \ldots, \{v_{k-1}\}, V - \{v_1, \ldots, v_{k-1}\}\}$, which has objective $(1+a)(k-1)+1$. This implies that the optimum k-partition $\mathcal{P}^*$ satisfies $f(\mathcal{P}^*) \leq (1+a)(k-1)+1$. Thus, the approximation factor of the solution returned by Algorithm 1 is

$$\frac{f(\mathcal{P})}{f(\mathcal{P}^*)} \geq \frac{a(n-1)}{(1+a)(k-1)+1} \to \frac{n-1}{k-1} \text{ as } a \to \infty. \qquad \triangleleft$$

5 Conclusion

The principal partition sequence of submodular functions was shown to exist by Narayanan [16]. The principal partition sequence of submodular functions is known in the literature as *principal lattice of partitions* of submodular functions since there exists a lattice structure associated with the sequence of partitions [2, 8, 9, 11, 14, 17, 20]. We chose to call it as principal partition sequence in this work since the sequence suffices for our purpose. Narayanan, Roy, and Patkar [18] used the principal partition sequence to design an algorithm for submodular k-partition. They analyzed the approximation factor of their algorithm for certain subfamilies of submodular functions that arise from hypergraphs. In this work, we investigated the approximation factor of their algorithm for three broad subfamilies of submodular functions – namely monotone, symmetric, and posimodular submodular functions. Our results show that the principal partition sequence based algorithm achieves the best possible asymptotic approximation factor for all these three subfamilies. A novelty of our contributions is the improvement in the approximability of monotone submodular k-partition from 2 to 4/3, thus matching the inapproximability threshold. It would be interesting to pin down the approximability of special cases of monotone submodular k-partition – e.g., matroid k-partition and coverage k-partition which are interesting by themselves since they capture several interesting partitioning problems.

References

1 M. Baïou and F. Barahona. Packing Hypertrees and the k-cut problem in Hypergraphs. In *International Conference on Learning and Intelligent Optimization*, LION, pages 521–534, 2023.

2 F. Barahona. On the k-cut problem. *Operations Research Letters*, 26(3):99–105, 2000.

3 D. Chakrabarty and Z. Huang. Testing coverage functions. In *Automata, Languages, and Programming*, ICALP, pages 170–181, 2012.

4 K. Chandrasekaran and C. Chekuri. Hypergraph k-cut for fixed k in deterministic polynomial time. *Mathematics of Operations Research*, 2022. Prelim. version in FOCS 2020: pages 810–821.

5 K. Chandrasekaran and W. Wang. Approximating submodular k-partition via principal partition sequence. arXiv, 2023. arXiv:2305.01069.

6 C. Chekuri and S. Li. On the hardness of approximating the k-way hypergraph cut problem. *Theory Comput.*, 16:1–8, 2020.

7 C. Chekuri, K. Quanrud, and C. Xu. LP Relaxation and Tree Packing for Minimum k-Cut. *SIAM Journal on Discrete Mathematics*, 34(2):1334–1353, 2020.

8 W. Cunningham. Optimal attack and reinforcement of a network. *J. ACM*, 32(3):549–561, July 1985.

9 M. P. Desai, H. Narayanan, and S. B. Patkar. The realization of finite state machines by decomposition and the principal lattice of partitions of a submodular function. *Discrete Appl. Math.*, 131:299–310, 2003.

10 O. Goldschmidt and D. Hochbaum. A Polynomial Algorithm for the k-cut Problem for Fixed k. *Mathematics of Operations Research*, 19(1):24–37, February 1994.

11 V. Kolmogorov. A Faster Algorithm for Computing the Principal Sequence of Partitions of a Graph. *Algorithmica*, pages 394–412, 2010.

12 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.

13 H. Nagamochi and Y. Kamidoi. Minimum cost subpartitions in graphs. *Information Processing Letters*, 102(2):79–84, 2007.

14 K. Nagano, Y. Kawahara, and S. Iwata. Minimum average cost clustering. In *Advances in Neural Information Processing Systems*, volume 23 of *NIPS*, 2010.

15 J. Naor and Y. Rabani. Tree packing and approximating k-cuts. In *Proceedings of the 12th Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA, pages 26–27, 2001.

16 H. Narayanan. The principal lattice of partitions of a submodular function. *Linear Algebra and its Applications*, 144:179–216, 1991.

17 H. Narayanan. Submodular functions and electrical networks (2nd edition), 1997. URL: `https://www.ee.iitb.ac.in/~hn/book/SubmodularFunction_2ed.pdf`.

18 H. Narayanan, S. Roy, and S. Patkar. Approximation algorithms for min-k-overlap problems using the principal lattice of partitions approach. *Journal of Algorithms*, 21(2):306–330, 1996.

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

20 S. B. Patkar and H. Narayanan. Improving graph partitions using submodular functions. *Discrete Appl. Math.*, 131:535–553, 2003.

21 K. Quanrud. Fast and Deterministic Approximations for k-Cut. In *Proceedings of Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, APPROX, pages 23:1–23:20, 2019.

22 R. Ravi and A. Sinha. Approximating k-cuts using network strength as a lagrangean relaxation. *European Journal of Operational Research*, 186(1):77–90, 2008.

23 R. Santiago. New approximations and hardness results for submodular partitioning problems. In *Proceedings of International Workshop on Combinatorial Algorithms*, IWOCA, pages 516–530, 2021.

24 H. Saran and V. Vazirani. Finding k Cuts within Twice the Optimal. *SIAM Journal on Computing*, 24(1):101–108, 1995.

25 V. Vazirani. *Approximation algorithms.* Springer, Berlin, Germany, 2003.

26 L. Zhao, H. Nagamochi, and T. Ibaraki. Greedy splitting algorithms for approximating multiway partition problems. *Mathematical Programming*, 102(1):167–183, 2005.