Combinatorica
https://doi.org/10.1007/s00493-023-00021-y

ORIGINAL PAPER

# Min–Max Partitioning of Hypergraphs and Symmetric Submodular Functions

**Karthekeyan Chandrasekaran**[1] · **Chandra Chekuri**[1]

Received: 8 February 2021 / Accepted: 25 October 2022


## Abstract

We consider the complexity of *minmax* partitioning of graphs, hypergraphs and (symmetric) submodular functions. Our main result is an algorithm for the problem of partitioning the ground set of a given *symmetric* submodular function $f : 2^V \to \mathbb{R}$ into $k$ non-empty parts $V_1, V_2, \ldots, V_k$ to minimize $\max_{i=1}^{k} f(V_i)$. Our algorithm runs in $n^{O(k^2)}T$ time, where $n = |V|$ and $T$ is the time to evaluate $f$ on a given set; hence, this yields a polynomial time algorithm for any fixed $k$ in the evaluation oracle model. As an immediate corollary, for any fixed $k$, there is a polynomial-time algorithm for the problem of partitioning a given hypergraph $H = (V, E)$ into $k$ non-empty parts to minimize the maximum capacity of the parts. The complexity of this problem, termed MINMAX- HYPERGRAPH- $k$- PART, was raised by Lawler in 1973 (Networks 3:275–285, 1973). In contrast to our positive result, the reduction in Chekuri and Li (Theory Comput 16(14):1–8, 2020) implies that when $k$ is part of the input, MINMAX-HYPERGRAPH- $k$- PART is hard to approximate to within an almost polynomial factor under the Exponential Time Hypothesis (ETH).

**Keywords** Hypergraphs · Submodular Functions · Partitioning

**Mathematics Subject Classification** 05C40 · 05C65 · 68R01 · 68W01

Supported in part by NSF Grant CCF-1907937. A preliminary version of this work appeared at ACM-SIAM Symposium on Discrete Algorithms (SODA), 2021.

✉ Karthekeyan Chandrasekaran
karthe@illinois.edu

Chandra Chekuri
chekuri@illinois.edu

1 University of Illinois, Urbana-Champaign, Urbana, IL, USA

Published online: 27 April 2023

## 1 Introduction

Partitioning problems in graphs and hypergraphs are extensively studied for their applications and theoretical value. In this work, we consider the *minmax* objective. A hypergraph $G = (V, E)$ consists of a finite vertex set $V$ and a collection of hyperedges where each hyperedge $e \in E$ is a subset of vertices, that is, $e \subseteq V$. If $|e| = 2$ for all $e \in E$, then the hypergraph is simply an undirected graph. The input to minmax hypergraph $k$-partitioning is a hypergraph $G = (V, E)$ with non-negative hyperedge weights $w : E \to \mathbb{R}_+$ and an integer $k$. The goal is to partition $V$ into *non-empty* sets $V_1, V_2, \ldots, V_k$ to minimize $\max_{i=1}^{k} w(\delta(V_i))$; here $\delta(V_i)$ is the set of hyperedges crossing[1] $V_i$ and $w(\delta(V_i)) = \sum_{e \in \delta(V_i)} w(e)$ is the total weight of the hyperedges in $\delta(V_i)$. We refer to this problem as Minmax- Hypergraph- $k$- Part. We refer to the special case when $G$ is a graph as Minmax- Graph- $k$- Part. Closely related to these problems are Graph- $k$- Cut, Hypergraph- $k$- Cut and Hypergraph- $k$- Part that we will discuss later. The complexity of Minmax- Hypergraph- $k$- Part was raised as early as 1973 in Lawler's work on hypergraph mincut [20], and has remained open. In this work, we show that Minmax- Hypergraph- $k$- Part has a polynomial-time algorithm for any fixed constant $k$.

**Theorem 1.1** Minmax- Hypergraph- $k$- Part *has a polynomial-time algorithm for any fixed $k$. In particular, there is an algorithm that runs in time $n^{O(k^2)}m$, where $n$ is the number of nodes and $m$ is the number of hyperedges.*

In contrast to the preceding positive result, when $k$ is part of the input, one can easily show that the reduction in [8] that proves conditional hardness of Hypergraph- $k$- Cut also applies to Minmax- Hypergraph- $k$- Part; this was observed in [6]. Consequently, under the Exponential Time Hypothesis (ETH) there is no $n^{1/(\log\log n)^c}$ -approximation for Minmax- Hypergraph- $k$- Part for some absolute constant $c$. Our algorithmic result in Theorem 1.1, of course, also applies to the special case of Minmax- Graph- $k$- Part. We will later point out that an alternative algorithm for Minmax- Graph- $k$- Part can be obtained from previous results on Graph- $k$- Cut while it is not the case for hypergraphs.

Several results on graphs and hypergraphs rely on *submodularity* of their cut function. We recall that a real-valued set function $f : 2^V \to \mathbb{R}$ is *submodular* if $f(A \cup B) + f(A \cap B) \leq f(A) + f(B)$ for all $A, B \subseteq V$ and is *symmetric* if $f(A) = f(V \backslash A)$ for all $A \subseteq V$. The cut function of a hypergraph is symmetric and submodular when the hyperedge weights are non-negative. Our algorithm for hypergraphs is a special case of our more general result on minmax partitioning of symmetric submodular functions. In this problem, the input is a finite ground set $V$, a symmetric submodular function $f$ (provided by an evaluation oracle[2]) and an integer $k$. The goal is to partition $V$ into $k$ non-empty parts $V_1, \ldots, V_k$ to minimize $\max_{i=1}^{k} f(V_i)$. We refer to this problem as Minmax- SymSubmod- k- Part and observe that Minmax- Hypergraph- $k$- Part is a special case. Minmax- Submod- $k$- Part refers to the problem when $f$ is submodular (but not necessarily symmetric).

[1] A hyperedge $e$ crosses $S \subseteq V$ if $e \cap S \neq \emptyset$ and $e \cap (V \backslash S) \neq \emptyset$.

[2] An evaluation oracle for a set function $f$ over a ground set $V$ returns the value of $f(S)$ given $S \subseteq V$.

MINMAX- SUBMOD- $k$- PART is NP-Hard even for $k = 2$.[3] However we show that MINMAX- SYMSUBMOD- $k$- PART is polynomial-time solvable for any fixed $k$.

**Theorem 1.2** MINMAX- SYMSUBMOD- $k$- PART *has a polynomial-time algorithm for any fixed k. In particular, there is an algorithm that runs in time* $n^{O(k^2)}T$, *where n is the size of the ground set and T is the time to evaluate the input function f on a given set.*

We note that Theorem 1.2 does not require the input function to be non-negative. This is not surprising since we can add a large positive constant to the function to make it non-negative without violating submodularity and symmetry and an optimum solution to the shifted function yields an optimum solution to the original function.

When $k$ is part of the input, MINMAX- SYMSUBMOD- $k$- PART inherits the hardness of approximation of MINMAX- HYPERGRAPH- $k$- PART that we already mentioned. One can also easily obtain a $2k$-approximation for MINMAX- SYMSUBMOD- $k$- PART when $f$ is non-negative.

## 1.1 Motivation and Related Problems

Given a real-valued set function $f : 2^V \to \mathbb{R}$ and a partition $V_1, \ldots, V_k$ of $V$, one can measure the quality of the partition in various natural ways. Two natural measures are $\max_{i=1}^k f(V_i)$ and $\sum_{i=1}^k f(V_i)$. Once a measure is defined, a corresponding optimization problem arises where one seeks to find a partition that minimizes the measure (we can also consider maximizing the measure but the focus of this paper is on minimizing the measure).

Minmax objective is particularly useful in load-balancing scenarios. Consider the classical MULTIPROCESSOR SCHEDULING problem of assigning $n$ jobs with given real-valued processing times $p_1, \ldots, p_n$ to $k$ machines to minimize the maximum load. This can be easily cast as a special case of MINMAX- SUBMOD- $k$- PART where the function $f$ is the modular function $p$ defined by $p(S) = \sum_{i \in S} p_i$; this special case is NP-Hard even for $k = 2$ via a reduction from 2-PARTITION. Motivated by such load balancing problems and considerations, several problems have been considered in algorithms literature. Svitkina and Tardos [28] introduced the minmax version of the multiway cut problem (MINMAX- MULTIWAY- CUT) motivated by applications in networking: the input is an edge-weighted graph $G = (V, E)$ and $k$ terminals $\{s_1, s_2, \ldots, s_k\} \subseteq V$, and the goal is to partition $V$ into $k$ parts $V_1, \ldots, V_k$ such that $s_i \in V_i$ for all $i \in [k]$ so as to minimize $\max_{i=1}^k w(\delta(V_i))$. They showed that it is NP-hard for $k = 4$ and gave a poly-logarithmic approximation for arbitrary $k$. This was subsequently improved by Bansal, Feige, Krauthgamer, Makarychev, Nagarajan, Naor, and Schwartz [1] who obtained an $O(\sqrt{\log n \log k})$ approximation. Bansal et

[3] MINMAX- SUBMOD- $k$- PART for $k = 2$ is weakly NP-hard by reduction from 2-PARTITION It is also strongly NP-hard: for matroid rank functions, the optimum value is strictly less than the rank of the ground set if and only if there exist two disjoint cocircuits in the matroid; Bernáth and Király have shown that verifying the existence of two disjoint cocircuits in a linear matroid specified by its matrix representation is NP-complete [3]. However, it is an interesting exercise to the reader to see that MINMAX- SYMSUBMOD- $k$- PART for $k = 2$ reduces to submodular function minimization and is hence, solvable in polynomial time.

al. also considered related problems where there are additional balance constraints on the number of vertices in each part; we refer the reader to their paper for more details. In addition, they showed a negative result that suggests that an approximation factor better than $k^{1-\epsilon}$ (without a dependence on $n$) is unlikely. We note that MINMAX-GRAPH- $k$- PART differs from MINMAX- MULTIWAY- CUT: no terminals are specified in the former problem.

Svitkina and Fleischer [27] considered SUBMOD- LOAD- BALANCING which is the restriction of MINMAX- SUBMOD- $k$- PART to *monotone*[4] submodular functions—monotonicity is natural in some applications. They showed that, when $k$ is part of the input, SUBMOD- LOAD- BALANCING is hard to approximate to within an $o(\sqrt{n/\log n})$-factor unless the algorithm makes exponential number of queries to the function evaluation oracle. They also describe an $O(\sqrt{n \log n})$ approximation. The approximability of the problem when $k$ is a fixed constant appears to be open.

The minmax objective for submodular functions has also been investigated among other objectives in machine learning applications from an empirical perspective [30]. *Minsum objective:* The minmax objective has several important connections to the minsum objective in terms of motivation, problems, and techniques. The minsum objective in partition problems captures several well-known problems that we discuss now. The GRAPH- $k$- CUT problem is the following: given an undirected graph $G = (V, E)$ with edge weights $w : E \to \mathbb{R}_+$, remove a minimum weight subset of edges so that the resulting graph has at least $k$ connected components. One can also view this equivalently as a minsum partition problem where the goal is to partition $V$ into $k$ non-empty parts $V_1, \ldots, V_k$ to minimize $\sum_{i=1}^{k} w(\delta(V_i))$. There are two natural generalizations of GRAPH- $k$- CUT to hypergraphs based on these two viewpoints: (1) In the HYPERGRAPH- $k$- CUT problem, one seeks to find a minimum weight subset of hyperedges of a given hyperedge-weighted hypergraph whose deletion leads to at least $k$ non-empty connected components. (2) In the HYPERGRAPH- $k$- PART problem, one seeks to find a $k$-partition $V_1, \ldots, V_k$ of the vertex set of a given hypergraph $G = (V, E)$ with hyperedge-weights $w : E \to \mathbb{R}_+$ to minimize $\sum_{i=1}^{k} w(\delta(V_i))$. In contrast to graphs, HYPERGRAPH- $k$- CUT and HYPERGRAPH- $k$- PART are not equivalent. One can consider generalizations of minsum problems to submodular functions leading to the SUBMOD- $k$- PART problem: given a submodular function $f : 2^V \to \mathbb{R}$ over a ground set $V$ and an integer $k$, the goal is to partition $V$ into $k$ non-empty parts $V_1, \ldots, V_k$ to minimize $\sum_{i=1}^{k} f(V_i)$. SYM- SUBMOD- $k$- PART is the special case of SUBMOD- $k$- PART when the input function $f$ is symmetric submodular. One can easily see that HYPERGRAPH- $k$- PART is a special case of SYM- SUBMOD- $k$- PART while it takes a bit more work to see that HYPERGRAPH- $k$- CUT is a special case of SUBMOD- $k$- PART [23].

GRAPH- $k$- CUT has been extensively studied. It generalizes the global mincut problem and is non-trivial even when $k = 3$. GRAPH- $k$- CUT was shown to be polynomial-time solvable for any fixed $k$ by Goldschmidt and Hochbaum [12]. The same work also showed NP-Hardness when $k$ is part of the input. There have been several other algorithms including the random contraction approach of Karger and Stein [19], and the tree packing approach of Karger [18] and Thorup [29]. We refer the reader

[4] A set function $f : 2^V \to \mathbb{R}$ is *monotone* if $f(A) \leq f(B)$ for all $A \subseteq B \subseteq V$.

to [9, 14, 15] for several recent results and additional pointers on Graph- $k$- Cut. In contrast, the complexity of Hypergraph- $k$- Cut was open until fairly recently. A randomized polynomial-time algorithm for any fixed $k$, based on random contraction, was first described by Chandrasekaran, Xu, and Yu [5] which was subsequently improved by Fox, Panigrahi, and Zhang [10]. A deterministic algorithm based on a generalization of the Goldschmidt-Hochbaum approach was given very recently by the authors of the current paper [4]. When $k$ is part of the input, Graph- $k$- Cut admits a $2(1-1/k)$-approximation [25], and moreover conditional hardness results show that this is the best possible [21]. Chekuri and Li [8] show that Hypergraph- $k$- Cut is hard to approximate to within almost polynomial-factor under ETH.

The complexity of Submod- $k$- Part and Sym- Submod- $k$- Part when $k$ is fixed are important open problems. Polynomial time algorithms are known for Submod- $k$- Part for $k \leq 3$ [23] and for Sym- Submod- $k$- Part for $k \leq 4$ [13].

As far as we are aware, no prior results existed on the worst-case complexity of the minmax partition problems that we study in this work. The minmax objective is in general more complex to handle as shown by negative results and prior work.

*Minmax from Minsum objective:* There is a useful connection between the minsum and the minmax objectives that we describe now: an $\alpha$-approximation for Submod- $k$- Part implies an $\alpha k$ approximation for Minmax- Submod- $k$- Part when the underlying function $f$ is non-negative. We sketch this argument: Suppose there is an optimum $k$-partition for the minmax objective with value $B$. Then the sum-objective value of the same partition is at most $kB$. Thus, an $\alpha$-approximation for minsum yields a partition whose sum-objective value is at most $\alpha k B$ and this partition has max-objective value at most $\alpha k B$ (since $f$ is non-negative).

The above-mentioned connection also leads to an $n^{O(k^2)}$-time algorithm for Minmax- Graph- $k$- Part as follows: Suppose $\mathcal{P}$ is an optimum $k$-partition for Minmax- Graph- $k$- Part on the given graph with optimum minmax objective value being $B$. Then, the optimum minsum objective value is at least $B$. Moreover, $\mathcal{P}$ has sum-objective value at most $kB$. Thus, if we can enumerate all $k$-approximate solutions to the minsum objective, then one of them will have max-objective value at most $B$. In graphs, we can indeed enumerate all $\beta$-approximate minsum $k$-partitions in time $n^{O(\beta k)}$ [19]. So, we can get an optimum partition for Minmax- Graph- $k$- Part by choosing the best among the $k$-approximate optimum solutions to Graph- $k$- Cut, which would take $n^{O(k^2)}$-time. However, this approach does not extend to hypergraphs or general symmetric submodular functions since the problem of enumerating $k$-approximate optimum solutions to the minsum objective is not known to be solvable efficiently in these settings.

*Gomory–Hu tree and symmetric submodular functions.* An important structural property of symmetric submodular functions is that they admit a *Gomory–Hu tree* and it can be found efficiently [26]. A *Gomory–Hu tree* for a symmetric submodular function $f : 2^V \to \mathbb{R}$ is a tree $T = (V, E)$ such that for every tree-edge $st \in E$, the partition $(A, V \setminus A)$ is a minimum $(s, t)$-terminal cut (with respect to $f$), where $A$ is a component in $T - st$. In algorithmic applications, one often endows the tree $T$ with edge weights $w : E \to \mathbb{R}$ given by $w(st) = f(A)$, where $st \in E$ and $A$ is a component in $T - st$. The existence of Gomory–Hu trees provide a unified explanation for efficient solvability/approximability of certain partitioning problems for symmetric submodular functions (e.g., efficient solvability of $T$-odd cut, 2-approximation for SYM- SUBMOD- $k$- PART, etc.)—all these algorithms construct the Gomory–Hu tree with edge weights and solve the problem of interest on the resulting edge-weighted tree, which tends to be substantially simpler owing to the tree structure. However, we note that Gomory–Hu trees may carry no information for MINMAX- SYMSUBMOD- $k$- PART. E.g., consider MINMAX- GRAPH- $k$- PART for the complete graph $K_n$ on $n$ vertices: the optimal value is $(n-k+1)(k-1)$. The natural approach of finding a Gomory–Hu tree and solving the problem on the tree fails even for this example: the star graph on $n$ vertices with all edge weights being $n-1$ is a Gomory–Hu tree for $K_n$. The optimum value of MINMAX- GRAPH- $k$- PART for this tree is $(n-1)(k-1)$.

### 1.2 Technical Overview and Main Structural Result

Our algorithm for MINMAX- SYMSUBMOD- $k$- PART is inspired, at a high-level, by the work of Goldschmidt and Hochbaum on GRAPH- $k$- CUT [12]. Their approach has been subsequently refined and applied with additional ideas to several related problems [4, 11, 17, 22, 31] Our approach for MINMAX- SYMSUBMOD- $k$- PART also builds on the ideas of Goldschmidt and Hochbaum, so we briefly recall their ideas.

A key algorithmic tool in the approach of [12], as well as our approach here, is the use of terminal cuts. We need some notation. Let $f : 2^V \to \mathbb{R}$ be a symmetric submodular function over the ground set $V$. For subsets $A$ and $B$ of the ground set $V$, we will use $A - B$ to denote $A \setminus B$. For a subset $U$ of the ground set $V$, we use $\overline{U}$ to denote $V - U$. The value of a 2-partition $(U, \overline{U})$ is $f(U)$. Let $S, T$ be disjoint subsets of the ground set $V$. A 2-partition $(U, \overline{U})$ is an $(S, T)$-terminal cut if $S \subseteq U \subseteq V - T$. Here, the set $U$ is known as the source set and the set $\overline{U}$ is known as the sink set. A minimum valued $(S, T)$-terminal cut is known as a minimum $(S, T)$-terminal cut. Since there could be multiple minimum $(S, T)$-terminal cuts, we will be interested in the *source maximal* minimum $(S, T)$-terminal cut. There exists a unique source maximal minimum $(S, T)$-terminal cut and it can be found in polynomial-time if we are given evaluation access to the submodular function (by relying on submodular function minimization)—e.g., see [11].

The approach of Goldschmidt and Hochbaum [12] for GRAPH- $k$- CUT is the following (for unit-weights on the edges). For $S \subseteq V$, let $\delta(S)$ denote the set of edges crossing $S$ in the input graph. Suppose $(V_1, V_2, \ldots, V_k)$ is an optimum minsum $k$-partition such that $V_1$ is the cheapest part (that is, $|\delta(V_1)| \leq |\delta(V_i)|$ for every $i \in [k]$), and $V_1$ is maximal subject to this condition. They show that one can identify $V_1$ via the following key structural theorem: either $|V_1| \leq k-1$ or there *exist* disjoint vertex subsets $S, T \subseteq V$ with $|S| \leq k-2$, $|T| = k-1$ so that the source maximal minimum $(S, T)$-terminal cut is $(V_1, \overline{V_1})$. Thus, one can guess/enumerate all pairs $(S, T)$ of small sizes to find an $O(n^{2k})$-sized collection of sets containing $V_1$. This enables a simple recursive algorithm: For each set in the collection, we assume it is $V_1$ and recurse to find a cheapest $(k-1)$-partition in the graph $G[V \setminus V_1]$. This leads to an $n^{O(k^2)}$-time algorithm.

The proof of the key structural theorem in [12] is non-trivial and relies heavily on properties of the cut function of graphs. Queyranne [24] claimed that a natural generalization of the preceding structural theorem holds in the more general setting of symmetric submodular functions, namely for the problem of SYM- SUBMOD- $k$- PART which generalizes GRAPH- $k$- CUT. However, as reported in [13], the claimed proof was incorrect and it was only proved for $k = 3, 4$. A starting point for our work here is a proof of (a mild relaxation of) the claim of Queyranne for all $k$—see Theorem 4.1. Theorem 4.1 is not directly relevant to the present work, but it appears to be a promising direction towards obtaining an efficient algorithm for SYM- SUBMOD- $k$- PART, hence we devote Sect. 4 to state and discuss its implications. Our proof of his claim relies only on submodularity (and symmetry) and hence, it gives a conceptually clean proof of the original algorithmic approach of [12] for GRAPH- $k$- CUT. Unfortunately, as noted in [13], this structural theorem does not lead to an algorithm for SYM- SUBMOD- $k$- PART because one cannot recurse on $V \setminus V_1$; the function $f$ restricted to $V \setminus V_1$ may no longer be symmetric! However, the approach works for GRAPH- $k$- CUT fortuitously because in graphs, we can afford to work with the cut function of the subgraph $G[V \setminus V_1]$ as opposed to the original cut function restricted to $V \setminus V_1$.

Recall that we are actually interested in the minmax objective and we wish to handle the general setting of MINMAX- SYMSUBMOD- $k$- PART. For this objective we prove a structural theorem that is similar in spirit to that of the minsum objective but technically somewhat different. We state this structural theorem now.

We need some notation. We will denote a $k$-partition by an ordered tuple—it will be important to view it as an ordered tuple rather than a collection of $k$ disjoint sets whose union is $V$. Given a $k$-partition $(V_1, V_2, \ldots, V_k)$ of $V$, we denote

$$\text{cost}_f(V_1, V_2, \ldots, V_k) := \max\{f(V_i) : i \in [k]\}.$$

A $k$-partition is a *minmax $k$-partition with respect to $f$* if it has the least cost among all possible $k$-partitions. We will drop the subscript $f$ from the cost notation and avoid repeating the phrase “with respect to $f$” when the function $f$ of interest is clear from context (the subscript and the phrasing will be needed primarily in Sect. 3). We will be interested in minmax $k$-partitions $(V_1, \ldots, V_k)$ for which $V_1$ is maximal: formally, we define a minmax $k$-partition $(V_1, \ldots, V_k)$ to be a *$V_1$-maximal minmax $k$-partition* if there is no other minmax $k$-partition $(V_1', \ldots, V_k')$ such that $V_1$ is strictly contained in $V_1'$. The following is our main structural result.

**Theorem 1.3** *Let $f : 2^V \to \mathbb{R}$ be a symmetric submodular function and let $k \geq 2$ be an integer. Let $(V_1, \ldots, V_k)$ be a $V_1$-maximal minmax $k$-partition with respect to $f$.*

*Then, for every subset $T \subseteq \overline{V_1}$ such that $T \cap V_j \neq \emptyset$ for every $j \in \{2, \ldots, k\}$, there exists a subset $S \subseteq V_1$ of size at most $k-1$ such that $(V_1, \overline{V_1})$ is the source maximal minimum $(S, T)$-terminal cut.*

We emphasize a key feature of our structural result: it does not require $V_1$ to be a cheapest part among all parts of the optimum $k$-partition (in contrast to the structural result of Goldschmidt-Hochbaum for the minsum objective in graphs). Informally speaking, our structural result says that under maximality of $V_1$, there exist disjoint sets $S, T \subseteq V$ with $|S|, |T| \leq k-1$ such that $(V_1, \overline{V_1})$ is the source-maximal minimum

$(S, T)$-terminal cut. Thus, we can compute a collection of $n^{O(k)}$ candidate sets such that the collection contains $V_1$.

The key feature of our structural theorem immediately implies that the problem can be solved in $(n^{O(k^2)} + n^{O(k)}T)$-time if there exists a *unique* minmax $k$-partition, say $V_1, \ldots, V_k$ (i.e., if the only ordered tuple of minmax $k$-partitions are permutations of the ordered tuple $(V_1, \ldots, V_k)$). For this, we note that the reordered $k$-partition $(A_1 := V_i, A_2 := V_1, \ldots, A_i := V_{i-1}, A_{i+1} := V_{i+1}, \ldots, A_k := V_k)$ is an $A_1$-maximal minmax $k$-partition due to uniqueness and hence, the theorem applies to this reordered $k$-partition. As a consequence, our candidate collection of $n^{O(k)}$ sets not only contains $V_1$, but also contains $V_2, \ldots, V_k$. Hence, we can iterate over all possible $k$-tuples of the sets in the collection to compute their cost (if they form a $k$-partition of $V$) and return the cheapest $k$-partition among all. We emphasize that this approach fails if the optimum $k$-partition is not unique. Moreover, minmax objective tends to have multiple optimum solutions in general. We next discuss the case in which the input function has multiple optimum solutions.

It may now appear that once we have the structural theorem, we can remove $V_1$ (by trying all possible candidate sets) and recurse to find a minmax $(k-1)$-partition of $f$ restricted to $V \backslash V_1$. As we remarked earlier, the function $f$ when restricted to $V \backslash V_1$ is not symmetric, so we will not be able to apply the structural theorem in the next step already! Moreover, the minmax objective is not conducive to the removal of a part of an optimum $k$-partition (e.g., consider what happens when $f$ is the graph cut function and we try to find $V_2$ after removing $V_1$ from the graph). We overcome these issues by *contracting* $V_1$ as opposed to removing it. Contracting $V_1$ to a singleton allows us to continue working with a symmetric submodular function. Now, our goal is to find a second *non-singleton* part in the optimum $k$-partition to make progress. We show that this is indeed possible. In order to do this we crucially rely on two aspects: (1) the fact that we are working with the minmax objective, and (2) the key feature of our main structural result that only relies on the maximality of $V_1$. The same approach extends inductively to enable us to find all parts of the optimum $k$-partition (see Sect. 3 for the complete algorithm and details).

As we saw, minmax submodular $k$-partition is NP-Hard even for $k = 2$ for simple asymmetric functions (e.g., modular functions). Symmetry of the submodular function is a crucial ingredient in the proof of our structural result. Symmetric submodular functions are also *posi-modular*, i.e., they satisfy

$$f(A) + f(B) \geq f(A - B) + f(B - A) \; \forall \; A, B \subseteq V.$$

Posimodularity allows for certain uncrossing properties that have been exploited in past work [1, 7, 28] explicitly or implicitly. Another important ingredient in the proof of our structural result is a strengthening of an uncrossing lemma underlying a containment property from [23]—we use symmetry to strengthen their uncrossing lemma for the minmax objective (see Lemma 2.1). Our proof of Lemma 2.1 gives an alternate simpler proof of their uncrossing lemma for symmetric submodular functions.

**Organization.** We prove our main structural theorem in Sect. 2 and design the algorithm in Sect. 3. We extend our main structural theorem to the minsum objective in

Sect. 4 and conclude with some approaches for the minsum objective for symmetric submodular functions in Sect. 5.

## 2 Main Structural Theorem

In this section, we prove Theorem 1.3. The proof consists of two high-level steps:

1. In the first step, we show that for *every* $S \subseteq V_1$ and *every* $T \subseteq \overline{V_1}$ such that $T$ intersects $V_j$ for every $j \in \{2, \ldots, k\}$, a minimum $(S, T)$ terminal cut $(U, \overline{U})$ satisfies the property that $U \subseteq V_1$. This containment property is captured in Lemma 2.1.
2. In the second step, we show that for every $T$ as above, there exists a *small* set $S \subseteq V_1$ such that the source maximal minimum $(S, T)$-terminal cut will be $(V_1, \overline{V_1})$. We show that $|S| \leq k-1$ suffices. The proof of this relies on an uncrossing property that is captured in Theorem 2.1.

### 2.1 Containment Property

We show the containment property in this section. The uncrossing underlying the proof of the containment property is inspired by the one due to Okumoto, Fukunaga, and Nagamochi for the minsum objective (Theorem 5 of [23]). Our proof exploits the symmetry of the input function to apply their idea for the minmax objective.

**Lemma 2.1** *Let $f : 2^V \to \mathbb{R}$ be a symmetric submodular function, $k \geq 2$ be an integer, $(V_1, \ldots, V_k)$ be a $V_1$-maximal minmax $k$-partition with respect to $f$, and $S \subseteq V_1$, $T \subseteq \overline{V_1}$ such that $T \cap V_j \neq \emptyset$ for every $j \in \{2, \ldots, k\}$. Suppose $(U, \overline{U})$ is a minimum $(S, T)$-terminal cut. Then, $U \subseteq V_1$.*

***Proof*** For the sake of contradiction, suppose $U \setminus V_1 \neq \emptyset$. Consider $W_1 := V_1 \cup U$ and $W_j := V_j - U$ for every $j \in \{2, \ldots, k\}$ (see Fig. 1).

Since $W_1 \supseteq S \neq \emptyset$ and $W_j \supseteq T \cap V_j \neq \emptyset$ for all $j \in \{2, \ldots, k\}$, we have that $(W_1, \ldots, W_k)$ is a $k$-partition. Claim 2.1 shows that the cost of this $k$-partition is at most that of $(V_1, \ldots, V_k)$. Hence, $(W_1, \ldots, W_k)$ is a minmax $k$-partition. Moreover, $W_1$ is a strict superset of $V_1$ as $U \setminus V_1 \neq \emptyset$ and hence, $(W_1, \ldots, W_k)$ contradicts $V_1$-maximality of the minmax $k$-partition $(V_1, \ldots, V_k)$. □

**Claim 2.1** *For every $i \in [k]$, we have $f(W_i) \leq f(V_i)$.*

***Proof*** We distinguish two cases. Suppose $i = 1$. Then, $f(V_1 \cap U) \geq f(U)$ since $(V_1 \cap U, \overline{V_1 \cap U})$ is a $(S, T)$-terminal cut while $(U, \overline{U})$ is a minimum $(S, T)$-terminal cut. Hence, we have that

$$\begin{aligned} f(V_1) + f(U) &\geq f(V_1 \cup U) + f(V_1 \cap U) \qquad \text{(by submodularity)} \\ &\geq f(V_1 \cup U) + f(U). \end{aligned}$$

Consequently, $f(V_1) \geq f(V_1 \cup U) = f(W_1)$.

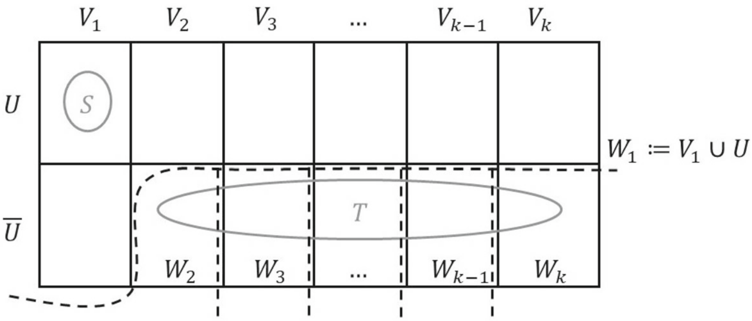


**Fig. 1** Uncrossing in the proof of Lemma 2.1

Next, suppose $i \in \{2, \ldots, k\}$. Then, $f(U - V_i) \geq f(U)$ since $(U - V_i, \overline{U - V_i})$ is a $(S, T)$-terminal cut while $(U, \overline{U})$ is a minimum $(S, T)$-terminal cut. Hence, we have that

$$\begin{aligned} f(V_i) + f(U) &\geq f(V_i - U) + f(U - V_i) \qquad \text{(by posimodularity)} \\ &\geq f(V_i - U) + f(U). \end{aligned}$$

Consequently, $f(V_i) \geq f(V_i - U) = f(W_i)$. □

### 2.2 Uncrossing Theorem

Our next ingredient is an uncrossing theorem to obtain a cheap $k$-partition.

**Theorem 2.1** *Let $f : 2^V \to \mathbb{R}$ be a symmetric submodular function, $k \geq 2$ be an integer, and $\emptyset \neq U \subsetneq V$. Let $C = \{u_1, \ldots, u_k\} \subseteq U$. Let $(\overline{A_i}, A_i)$ be a minimum $(C \backslash \{u_i\}, \overline{U})$-terminal cut for every $i \in [k]$. Suppose that $u_i \in A_i \setminus (\cup_{j \in [k] \backslash \{i\}} A_j)$ for every $i \in [k]$ and $f(A_1) \leq f(A_2) \leq \cdots \leq f(A_k)$. Then, there exists a $k$-partition $(P_1, \ldots, P_k)$ of $V$ such that*

$$f(P_i) \leq f(A_i)\ \forall\ i \in [k].$$

*In particular $cost_f(P_1, \ldots, P_k) \leq \max\{f(A_i) : i \in [k]\}$.*

***Proof*** See Fig. 2 for an illustration of the sets that appear in the statement of the theorem.

We begin with the following uncrossing claim showing that there exists a cheap $(k-1)$-partition of $\cup_{i=1}^{k-1} A_i$ (cheap in the sense that the function value of every part is small). Its proof relies on posimodularity and it has appeared implicitly (for the graph cut function) in previous works [1, 28].

**Claim 2.2** *There exist subsets $P_1, \ldots, P_{k-1}$ of $V$ such that*

(i) *$P_i$ and $P_j$ are disjoint for every distinct $i, j \in [k-1]$,*

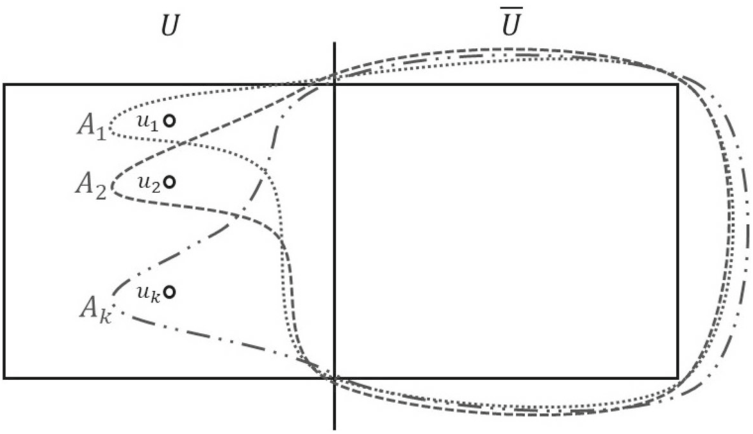


**Fig. 2** Illustration of the sets that appear in Theorem 2.1

```
Procedure
  Initialize P_i ← A_i for every i ∈ [k − 1]
  While there exist distinct i, j ∈ [k − 1] such that P_i ∩ P_j ≠ ∅
      If f(P_i − P_j) ≤ f(P_i)
          P_i ← P_i − P_j
      Else
          P_j ← P_j − P_i
```

**Fig. 3** Procedure for the proof of Claim 2.2

(ii) $u_i \in P_i$ *for every* $i \in [k-1]$,
(iii) $\cup_{i=1}^{k-1} P_i = \cup_{i=1}^{k-1} A_i$,
(iv) $f(P_i) \leq f(A_i)$ *for every* $i \in [k-1]$.

***Proof*** We use the procedure given in Fig. 3 to obtain subsets $P_1, \ldots, P_{k-1}$ of $V$ with the desired properties.

The procedure terminates in finite number of iterations since the steps in the while loop make progress towards ensuring that the sets $P_1, \ldots, P_{k-1}$ are mutually disjoint. Property (i) is achieved due to the termination condition. Properties (ii) and (iii) are maintained as invariants throughout the procedure (recall that $u_i \in A_i$ but $u_i \notin A_j$ for all distinct $i, j \in [k-1]$). Property (iv) is also maintained as an invariant throughout the procedure by posi-modularity: recall that by posimodularity, for every two sets $X$ and $Y$ we have that $f(X-Y) + f(Y-X) \leq f(X) + f(Y)$ and hence, either $f(X-Y) \leq f(X)$ or $f(Y-X) \leq f(Y)$ should hold. □

Let $P_k := V - \cup_{i=1}^{k-1} P_i$. By property (i) of Claim 2.2, the sets $P_i$ and $P_j$ are disjoint for every distinct $i, j \in [k]$. By properties (ii) and (iii) of Claim 2.2, we have that $u_i \in P_i$ for every $i \in [k]$. Hence, $(P_1, \ldots, P_k)$ form a $k$-partition of the ground set $V$. By property (iv) of Claim 2.2, we have that $f(P_i) \leq f(A_i)$ for every $i \in [k-1]$. It remains to show that $f(P_k) \leq f(A_k)$.

By property (iii) of Claim 2.2, we have that $P_k = \overline{\cup_{i=1}^{k-1} P_i} = \overline{\cup_{i=1}^{k-1} A_i} = \cap_{i=1}^{k-1}\overline{A_i}$. Let $B_i := \overline{A_i}$ for every $i \in [k]$.

Claim 2.3 below shows that the function value of $P_k$ is at most that of $A_k$ by relying on submodularity and the hypothesis that $(\overline{A_i}, A_i)$ is a minimum $(C\setminus\{u_i\}, \overline{U})$-terminal cut for every $i \in [k]$, thus completing the proof.

We note that Claim 2.3 has been used in previous works in different contexts [11]. □

**Claim 2.3** *We have that*

$$f\left(\cap_{i\in[k-1]} B_i\right) \le f(B_k).$$

***Proof*** Suppose not. Choose maximal $J \subseteq [k-1]$ such that $f(\cap_{j\in J} B_j) \le f(B_k)$. We note that $J \ne \emptyset$ since $f(B_j) \le f(B_k)$ for every $j \in [k-1]$. By assumption, $J \subsetneq [k-1]$ (otherwise we are done). Let $i \in [k-1]\setminus J$ and $R := \cap_{j\in J} B_j$. We note that $f(R \cup B_i) \ge f(B_i)$ since $(R \cup B_i, \overline{R \cup B_i})$ is a $(C \setminus \{u_i\}, \overline{U})$-terminal cut while $(B_i, \overline{B_i})$ is a minimum $(C\setminus\{u_i\}, \overline{U})$-terminal cut. Then,

$$\begin{aligned} f(B_k) + f(B_i) &\ge f(R) + f(B_i) && \text{(By choice of } J\text{)}\\ &\ge f(R \cup B_i) + f(R \cap B_i) && \text{(By submodularity)}\\ &\ge f(B_i) + f(R \cap B_i). \end{aligned}$$

Therefore, $f(R \cap B_i) \le f(B_k)$ and hence, the set $J \cup \{i\}$ contradicts the choice of $J$. □

### 2.3 Proof of Theorem 1.3

We now restate and prove Theorem 1.3.

**Theorem 1.3** *Let $f : 2^V \to \mathbb{R}$ be a symmetric submodular function and let $k \ge 2$ be an integer. Let $(V_1, \ldots, V_k)$ be a $V_1$-maximal minmax $k$-partition with respect to $f$.*

*Then, for every subset $T \subseteq \overline{V_1}$ such that $T \cap V_j \ne \emptyset$ for every $j \in \{2, \ldots, k\}$, there exists a subset $S \subseteq V_1$ of size at most $k-1$ such that $(V_1, \overline{V_1})$ is the source maximal minimum $(S, T)$-terminal cut.*

***Proof*** We emphasize that $V_1$-maximality of the minmax $k$-partition $(V_1, \ldots, V_k)$ with respect to $f$ will be used in this proof only to guarantee the applicability of Lemma 2.1. Suppose $|V_1| \le k-1$. Then, we consider $S = V_1$. We have that $|S| \le k-1$. Since $(V_1, \ldots, V_k)$ is a $V_1$-maximal minmax $k$-partition with respect to $f$ and $S \subseteq V_1$, we can apply Lemma 2.1. By this lemma,

we have that $(V_1, \overline{V_1})$ is the source maximal minimum $(S, T)$-terminal cut for every $T \subseteq \overline{V_1}$ such that $T \cap V_j \ne \emptyset$ for every $j \in \{2, \ldots, k\}$, thus proving the theorem. We consider the case of $|V_1| \ge k$ in the rest of the proof.

For the sake of contradiction, suppose that the theorem is false for some subset $T \subseteq \overline{V_1}$ such that $T \cap V_j \ne \emptyset$ for all $j \in \{2, \ldots, k\}$. Our proof strategy is to obtain a cheaper $k$-partition than $(V_1, \ldots, V_k)$, thereby contradicting the optimality of $(V_1, \ldots, V_k)$. Let $OPT_k$ denote the cost of $(V_1, \ldots, V_k)$. For a subset $X \subseteq V_1$, let $(V_X, \overline{V_X})$ be the source maximal minimum $(X, T)$-terminal cut. By Lemma 2.1, we have that $V_X \subseteq V_1$ for all $X \subseteq V_1$.

Among all possible subsets of $V_1$ of size $k-1$, pick a subset $S$ such that $f(V_S)$ is maximum. Then, by Lemma 2.1 and assumption, we have that $V_S \subsetneq V_1$. By source maximality of the minimum $(S, T)$-terminal cut $(V_S, \overline{V_S})$, we have that $f(V_S) < f(V_1)$. Let $u_1, \ldots, u_{k-1}$ be the vertices in $S$. Since $V_S \subsetneq V_1$, there exists a vertex $u_k \in V_1 \backslash V_S$. Let $C := \{u_1, \ldots, u_k\} = S \cup \{u_k\}$. For $i \in [k]$, let $(B_i, \overline{B_i})$ be the source maximal minimum $(C - \{u_i\}, T)$-terminal cut. We note that $(B_k, \overline{B_k}) = (V_S, \overline{V_S})$ and the size of $C - \{u_i\}$ is $k-1$ for every $i \in [k]$. By Lemma 2.1 and assumption, we have that $B_i \subsetneq V_1$ for every $i \in [k]$. Hence, we have

$$f(B_i) \leq f(V_S) < f(V_1) \text{ and } B_i \subsetneq V_1 \text{ for every } i \in [k]. \tag{1}$$

The next claim will set us up to apply Theorem 2.1.

**Claim 2.4** *For every $i \in [k]$, we have that $u_i \in \overline{B_i}$.*

***Proof*** The claim holds for $i = k$ by choice of $u_k$. For the sake of contradiction, suppose $u_i \in B_i$ for some $i \in [k-1]$. Then, the 2-partition $(V_S \cap B_i, \overline{V_S \cap B_i})$ is a $(S, T)$-terminal cut while $(V_S, \overline{V_S})$ is a minimum $(S, T)$-terminal cut and hence

$$f(V_S \cap B_i) \geq f(V_S).$$

We also have that

$$f(V_S \cup B_i) \geq f(V_S)$$

since $(V_S \cup B_i, \overline{V_S \cup B_i})$ is a $(S, T)$-terminal cut while $(V_S, \overline{V_S})$ is a minimum $(S, T)$-terminal cut. Thus,

$$\begin{aligned} 2f(V_S) &\geq f(V_S) + f(B_i) && \text{(By choice of } S\text{)} \\ &\geq f(V_S \cup B_i) + f(V_S \cap B_i) && \text{(By submodularity)} \\ &\geq 2f(V_S). && \text{(By the inequalities above)} \end{aligned}$$

Therefore, all inequalities above should be equations and hence, $f(V_S \cup B_i) = f(V_S)$. Consequently, the 2-partition $(V_S \cup B_i, \overline{V_S \cup B_i})$ is a minimum $(S, T)$-terminal cut. However, this contradicts source maximality of the minimum $(S, T)$-terminal cut $(V_S, \overline{V_S})$ since $u_k \in B_i$ and $u_k \notin V_S$. □

We note that for every $i \in [k]$, the 2-partition $(B_i, \overline{B_i})$ is a minimum $(C - \{u_i\}, \overline{V_1})$-terminal cut since $\overline{V_1} \subseteq \overline{B_i}$.

We will now apply Theorem 2.1. We consider $U := V_1$ and $C = \{u_1, \ldots, u_k\} \subseteq U$. Let $(\overline{A_i}, A_i) := (B_i, \overline{B_i})$ for every $i \in [k]$. The 2-partition $(\overline{A_i}, A_i)$ is a minimum $(C \setminus \{u_i\}, \overline{U})$-terminal cut for every $i \in [k]$. By Claim 2.4, we have that $u_i \in A_i$ for every $i \in [k]$. Since $(B_j, \overline{B_j})$ is a $(C - \{u_j\}, T)$-terminal cut, we have that $u_i \notin \overline{B_j}$ for every distinct $i, j \in [k]$. Thus, $u_i \in A_i \backslash (\cup_{j \in [k] \backslash \{i\}} A_j)$ for every $i \in [k]$. We may reindex the elements in $C$ so that $f(A_1) \leq f(A_2) \leq \cdots \leq f(A_k)$. Therefore, the sets $U$, $C$, and the 2-partitions $(\overline{A_i}, A_i)$ for $i \in [k]$ satisfy the conditions of Theorem 2.1. By Theorem 2.1 and statement (1), we obtain a $k$-partition $(P_1, \ldots, P_k)$ of $V$ such that

$$\text{cost}(P_1, \ldots, P_k) \leq \max\{f(A_i) : i \in [k]\} = f(V_S) < f(V_1) \leq OPT_k.$$

Thus, we have obtained a $k$-partition whose cost is smaller than $OPT_k$, a contradiction. □

## 3 Algorithm

In this section, we design an algorithm to solve MINMAX- SYMSUBMOD- $k$- PART based on Theorem 1.3. Using Theorem 1.3, it is possible to efficiently enumerate $n^{2k-2}$ candidate subsets such that one of them is $V_1$, where $(V_1, \ldots, V_k)$ is a $V_1$-maximal minmax $k$-partition with respect to the input symmetric submodular function $f : 2^V \to \mathbb{R}$. However, after finding $V_1$, we cannot recurse on the function $f$ restricted to $\overline{V_1}$ to find a cheapest $(k-1)$-partition: the restricted function may not be symmetric. Instead, we will work with the function obtained by contracting $V_1$. We define the contraction operation now.

Let $f : 2^V \to R$ be a symmetric submodular function. Let $U$ be a subset of the ground set $V$. We define the contracted function $f/U$ as follows: the ground set is $V' := V - U + \{u\}$, where $u$ denotes the contracted element. The function $f/U : 2^{V'} \to \mathbb{R}$ is defined as:

$$(f/U)(A) := \begin{cases} f(A \cup U) & \text{if } u \in A \subseteq V', \\ f(A) & \text{if } u \notin A \subseteq V'. \end{cases}$$

We note that the function $f/U$ is symmetric and submodular. The following observation is easy but we give a proof for the sake of completeness.

**Observation 3.1** *If $(V_1, \ldots, V_k)$ is a minmax $k$-partition with respect to $f$, then $(V_2, \ldots, V_k, \{v_1\})$ is a minmax $k$-partition with respect to $f/V_1$ where $v_1$ is the contracted element.*

***Proof*** For notational simplicity, let $g$ denote the function $f/V_1$. Say $(P_1, \ldots, P_k)$ is a minmax $k$-partition with respect to $g$. For the sake of contradiction, suppose

$$\text{cost}_g(P_1, \ldots, P_k) < \text{cost}_g(V_2, \ldots, V_k, \{v_1\}).$$

We observe that

$$\text{cost}_g(V_2, \ldots, V_k, \{v_1\}) = \text{cost}_f(V_1, \ldots, V_k).$$

Without loss of generality, let $v_1 \in P_1$. Consider the $k$-partition $(P_1' := (P_1 \backslash \{v_1\}) \cup V_1, P_2, \ldots, P_k)$ of $V$. We have that

$$\text{cost}_f(P_1', P_2, \ldots, P_k) = \text{cost}_g(P_1, \ldots, P_k).$$

Hence, we have a $k$-partition $(P_1', P_2, \ldots, P_k)$ that is cheaper with respect to $f$ than $(V_1, \ldots, V_k)$, thus contradicting optimality of $(V_1, \ldots, V_k)$. □

Next, we define a crucial tie-breaking rule which will help us find the next part $V_2$ by working with the contracted function $f/V_1$, where $(V_1, \ldots, V_k)$ is a minmax $k$-partition with respect to $f$. A minmax $k$-partition $(V_1, \ldots, V_k)$ is a *lexicographically maximal minmax $k$-partition* if there is no other minmax $k$-partition $(U_1, \ldots, U_k)$ with an $i \in [k-1]$ such that $U_1 = V_1, \ldots, U_{i-1} = V_{i-1}$, but $U_i \supsetneq V_i$. We observe that a lexicographically maximal minmax $k$-partition always exists. Furthermore, if $(V_1, \ldots, V_k)$ is a lexicographically maximal minmax $k$-partition, then it is also a $V_1$-maximal minmax $k$-partition. The following lemma shows that contraction of $V_1$ preserves lexicographic maximality in a certain sense.

**Lemma 3.1** *Let $f : 2^V \to \mathbb{R}$ be a symmetric submodular function, $k \geq 2$ be a positive integer, and $(V_1, \ldots, V_k)$ be a lexicographically maximal minmax $k$-partition with respect to $f$. Let $f/V_1$ be the contracted function with $v_1$ being the contracted element. Then, $(V_2, \ldots, V_k, \{v_1\})$ is a lexicographically maximal minmax $k$-partition with respect to the contracted function $f/V_1$.*

***Proof*** For notational simplicity, let $g$ denote the function $f/V_1$. By Observation 3.1, the $k$-partition $(V_2, \ldots, V_k, \{v_1\})$ is a minmax $k$-partition with respect to $g$. We now show that $(V_2, \ldots, V_k, \{v_1\})$ is a lexicographically maximal minmax $k$-partition with respect to $g$. Suppose not. Then, there exists a minmax $k$-partition $(U_1, \ldots, U_k)$ with respect to $g$ and an index $i \in [k-1]$ such that $U_1 = V_2, U_2 = V_3, \ldots, U_{i-1} = V_i$, but $U_i \supsetneq V_{i+1}$. Without loss of generality, let $v_1 \in U_j$ for some $j \in \{i, \ldots, k\}$. We distinguish two cases.

**Case 1** Suppose $U_j \backslash \{v_1\} \neq \emptyset$. Then, the partition

$$(V_1 \cup (U_j \setminus \{v_1\}), U_1, \ldots, U_{j-1}, U_{j+1}, \ldots, U_k)$$

is a minmax $k$-partition with respect to $f$ that contradicts lexicographic maximality of $(V_1, \ldots, V_k)$ (in particular, it contradicts $V_1$-maximality).

**Case 2** Suppose $U_j = \{v_1\}$. Since $|U_i| \geq |V_{i+1}| + 1 \geq 2$ while $|U_j| = 1$, it follows that $j \neq i$. We recall that $j \geq i$ and consequently, $j \geq i + 1$. Then, the partition

$$(V_1, V_2 = U_1, \ldots, V_i = U_{i-1}, U_i, U_{i+1}, \ldots, U_{j-1}, U_{j+1}, \ldots, U_k)$$

is a minmax $k$-partition with respect to $f$ that contradicts lexicographic maximality of $(V_1, \ldots, V_k)$ (in particular, it contradicts $V_{i+1}$-maximality as $U_i \supsetneq V_{i+1}$). □

```
Generate-Candidates (f : 2^V → ℝ, k, Q)
  Input: A function f : 2^V → ℝ, an integer k ≥ 2, and a subset Q of V
  Output: A collection 𝓡 ⊆ 2^V
  Initialize 𝓡 ← ∅
  For every disjoint S, T ⊂ V with |S| ≤ k − 1, T ⊇ Q and |T| = k − 1
      Compute the source maximal minimum (S, T)-terminal cut (U, Ū)
      𝓡 ← 𝓡 ∪ {U}
  Return 𝓡
```

**Fig. 4** Procedure to generate candidates for $V_1$

Lemma 3.1 suggests that if we know the part $V_1$ of a lexicographically maximal minmax $k$-partition $(V_1, \ldots, V_k)$ with respect to $f$, then using Theorem 1.3 for the function $f/V_1$, we can efficiently enumerate $n^{2k-2}$ candidate subsets such that one of them is $V_2$. We will now use this idea inductively to recover all parts of a lexicographically maximal minmax $k$-partition.

We state a corollary of Lemma 3.1 which will be helpful in the inductive algorithmic approach. The corollary follows from Lemma 3.1 by induction on $i$ and by the fact that contraction operation preserves symmetry and submodularity.

**Corollary 3.1** *Let $f : 2^V \to \mathbb{R}$ be a symmetric submodular function, $k \geq 2$ be a positive integer, and $(V_1, \ldots, V_k)$ be a lexicographically maximal minmax $k$-partition with respect to $f$. For $i \in [k]$, let $g = f/V_1/V_2/\ldots/V_{i-1}$ be the contracted function with $v_j$ being the contracted element corresponding to $V_j$ for all $j \in [i-1]$. Then, $(V_i, \ldots, V_k, \{v_1\}, \ldots, \{v_{i-1}\})$ is a lexicographically maximal minmax $k$-partition with respect to the function $g$.*

We begin with the procedure in Fig. 4 that returns a collection of candidate sets such that one of them is $V_1$, where $(V_1, \ldots, V_k)$ is a $V_1$-maximal minmax $k$-partition with respect to $f$. We summarize the guarantees of this procedure in Corollary 3.2. The corollary follows immediately from Theorem 1.3.

**Corollary 3.2** *Let $f : 2^V \to \mathbb{R}$ be a symmetric submodular function on a $n$-element ground set $V$ and let $k \geq 2$ be an integer. Let $(V_1, \ldots, V_k)$ be a $V_1$-maximal minmax $k$-partition with respect to $f$. Suppose $Q \subseteq V \setminus V_1$ such that $|Q \cap V_j| \leq 1$ for every $j \in \{2, \ldots, k\}$. Then,*

(i) *$V_1$ is in the collection $\mathcal{R}$ returned by Generate-Candidates$(f, k, Q)$, and*
(ii) *the size of the collection $\mathcal{R}$ that is returned by Generate-Candidates$(f, k, Q)$ is $O(n^{2k-2})$.*

*Moreover, Generate-Candidates procedure can be implemented to run in time $n^{2k-2}T(n)$, where $T(n)$ is the time for computing source maximal minimum $(S, T)$-terminal cuts for a symmetric submodular function on $n$ elements.*

We now describe our algorithm to find a minmax $k$-partition in Fig. 5 and summarize its guarantees in Theorem 3.1.

**Theorem 3.1** *Let $f : 2^V \to \mathbb{R}$ be a symmetric submodular function on a $n$-element ground set $V$ and let $k \geq 2$ be an integer. Then, Algorithm Partition$(f, k)$ given in*

```
Algorithm Partition (f : 2^V → ℝ, k)
  Input: Symmetric submodular function f : 2^V → ℝ, an integer k ≥ 2
  Output: A minmax k-partition with respect to f
  Initialize C_1 ← Generate-Candidates (f, k, Q ← ∅)
  For i = 2, ..., k − 1
      C_i ← ∅
      For each (i − 1)-partition (P_1, ..., P_{i−1}) ∈ C_{i−1}
          g ← f/P_1/P_2/.../P_{i−1} where p_1, ..., p_{i−1} are the contracted elements
          Q ← {p_1, ..., p_{i−1}}
          R_i ← Generate-Candidates (g, k, Q)
          C_i ← C_i ∪ {(P_1, ..., P_{i−1}, U) : U ∈ R_i}
  C_k ← {(P_1, ..., P_{k−1}, V \ ∪_{i=1}^{k−1} P_i) : (P_1, ..., P_{k−1}) ∈ C_{k−1}}
  Among all k-partitions in C_k, pick the one with minimum cost and return it
```

**Fig. 5** Algorithm to compute minimum $k$-partition for a symmetric submodular function

*Fig. 5 returns a minmax $k$-partition with respect to $f$ and it can be implemented to run in $n^{O(k^2)}T(n)$ time, where $T(n)$ denotes the time complexity for computing the source maximal minimum $(S, T)$-terminal cut for a submodular function defined over a ground set of size $n$.*

***Proof*** We first prove the run-time bound. For every $i \in [k-1]$, we have that $|\mathcal{C}_i| = |\mathcal{C}_{i-1}||\mathcal{R}_i| = \prod_{j=1}^{i} |\mathcal{R}_i| = O(n^{2i(k-1)})$ and the time to compute $\mathcal{C}_i$ is $O(n^{2i(k-1)})T(n)$ using Corollary 3.2. Hence, $|\mathcal{C}_k| = |\mathcal{C}_{k-1}| = O(n^{2(k-1)^2})$ and the total run-time is $O(n^{2(k-1)^2})T(n)$.

Next, we prove correctness. Let $(V_1, \ldots, V_k)$ be a lexicographically maximal minmax $k$-partition. We will show that $(V_1, \ldots, V_i) \in \mathcal{C}_i$ for every $i \in [k-1]$ by induction on $i$. The base case of $i = 1$ follows immediately from Corollary 3.2. For the induction step, let $i \geq 2$. By induction hypothesis, we have that $(V_1, \ldots, V_{i-1}) \in \mathcal{C}_{i-1}$. Consider $g = f/V_1/V_2/\ldots/V_{i-1}$ where $v_1, \ldots, v_{i-1}$ are the contracted elements. Let $A$ be the ground set of $g$. Contraction operation preserves symmetry and submodularity. So, $g$ is a symmetric submodular function. By Corollary 3.1, the $k$-partition given by

$$(A_1 := V_i, A_2 := V_{i+1}, \ldots, A_{k-i+1} := V_k, A_{k-i+2} := \{v_1\}, \ldots, A_k := \{v_{i-1}\})$$

is a lexicographically maximal minmax $k$-partition with respect to $g$. Thus, $(A_1, \ldots, A_k)$ is a $A_1$-maximal minmax $k$-partition with respect to $g$. Moreover, the set $Q = \{v_1, \ldots, v_{i-1}\}$ is a subset of $A \backslash A_1$ such that $|Q \cap A_j| = 1$ for every $j \in \{2, \ldots, k\}$ for which $Q \cap A_j \neq \emptyset$. Hence, by Corollary 3.2, the set $A_1$ is in the collection $\mathcal{R}_i$. Thus, $V_i \in \mathcal{R}_i$ and consequently, $(V_1, \ldots, V_{i-1}, V_i) \in \mathcal{C}_i$. □

We note that source maximal minimum $(S, T)$-terminal cut for a submodular function $f : 2^V \to \mathbb{R}$ can be computed in time $n^{O(1)}T$, where $n = |V|$ and $T$ is the time per evaluation (e.g., see [11]). Thus, Theorem 1.2 follows from Theorem 3.1. Moreover, the evaluation oracle for the hypergraph cut function can be implemented in $T = m$ time, where $m$ is the number of hyperedges in the input hypergraph. Thus, Theorem 1.1 follows from Theorem 1.2.

## 4 Minsum Symmetric Submodular *k*-Partition

The ideas underlying the proof of our main structural theorem—i.e., Theorem 1.3—can also be adapted to obtain a similar structural theorem for the minsum objective, namely SYM- SUBMOD- $k$- PART. Such a structural theorem helps in reducing SYM-SUBMOD- $k$- PART to SUBMOD- $(k-1)$- PART. We devote this section to discuss these extensions.

We begin with some notation for the minsum objective. Given a $k$-partition $(V_1, V_2, \ldots, V_k)$ of the ground set $V$ of a symmetric submodular function $f : 2^V \to \mathbb{R}$, the sum-objective value is $\sum_{i=1}^{k} f(V_i)$. A $k$-partition is a minsum $k$-partition if it has the least sum-objective value among all possible $k$-partitions. We will be interested in minsum $k$-partitions $(V_1, \ldots, V_k)$ for which $V_1$ is maximal: formally, we define a minsum $k$-partition $(V_1, \ldots, V_k)$ to be a $V_1$-maximal minsum $k$-partition if there is no other minsum $k$-partition $(V_1', \ldots, V_k')$ such that $V_1$ is strictly contained in $V_1'$. We emphasize that there always exists a $V_1$-maximal minsum $k$-partition such that $V_1$ is the cheapest part (i.e., $f(V_1) \le f(V_i)$ for every $i \in [k]$). With this notation, we have the following result.

**Theorem 4.1** *Let $f : 2^V \to \mathbb{R}$ be a symmetric submodular function and let $k \ge 2$ be an integer. Let $(V_1, V_2, \ldots, V_k)$ be a $V_1$-maximal minsum k-partition such that $V_1$ is the cheapest part. Then, for every subset $T \subseteq \overline{V_1}$ such that $T \cap V_j \ne \emptyset$ for every $j \in \{2, \ldots, k\}$, there exists a subset $S \subseteq V_1$ of size at most $k-1$ such that $(V_1, \overline{V_1})$ is the source maximal minimum $(S, T)$-terminal cut.*

Our proof of Theorem 4.1 closely resembles the proof of Theorem 1.3 and is included in the appendix. We emphasize that Theorem 4.1 does not allow us to solve SYM- SUBMOD- $k$- PART. We explain the bottleneck here. Suppose that the optimum minsum $k$-partition is unique and let it be $V_1, \ldots, V_k$. Then, the theorem helps us find the part $V_i$ whose function value is the cheapest. Now, we have two options: either (i) consider the contracted function $f/V_i$ which is symmetric submodular and recurse or (ii) consider the function $f$ restricted to $V \setminus V_i$ which is submodular but may not be symmetric and solve SUBMOD- $(k-1)$- PART on this instance. We note that the contract and recurse approach does not work: the theorem applied to the contracted function $f/V_i$ will only help in finding $v_i$ but none of the remaining parts of the optimum $k$-partition. However, we can use the restriction approach to reduce SYM-SUBMOD- $k$- PART to SUBMOD- $(k-1)$- PART as shown in Corollary 4.1 below. It is known that SUBMOD- $k$- PART is polynomial time solvable for $k \le 4$ [16, 23]. Thus, Corollary 4.1 implies that SYM- SUBMOD- $k$- PART is polynomial time solvable for $k \le 5$.

**Corollary 4.1** *If there exists an algorithm for* SUBMOD- $(k-1)$- PART *that runs in time $R(n,k)$ for n element ground sets, then there exists an algorithm for* SYM- SUBMOD- $k$- PART *that runs in time $n^{2k-2} R(n,k) + n^{2k-2} T(n)$ for n element ground sets, where $T(n)$ is the time for computing source maximal minimum $(S,T)$-terminal cut for a submodular function defined over a ground set of size n.*

***Proof*** Let the symmetric submodular function $f : 2^V \to \mathbb{R}$ be the input instance of SYM- SUBMOD- $k$- PART. We describe the algorithm: For every pair of disjoint subsets

$S, T \subseteq V$ satisfying $1 \le |S|, |T| \le k-1$, compute a source maximal minimum $(S, T)$-terminal cut $(U, \overline{U})$ and add $U$ to the collection of candidates $\mathcal{C}$. Next, for each set $U \in \mathcal{C}$, solve SUBMOD- $(k-1)$- PART on the input instance $f : 2^{V-U} \to \mathbb{R}$, i.e., the function $f$ restricted to $V-U$, to obtain an optimum $(k-1)$-partition $(P_1^U, \ldots, P_{k-1}^U)$. Among all $k$-partitions $(U, P_1^U, \ldots, P_{k-1}^U)$, return the one with the cheapest sum-objective value. The running time of the algorithm is $n^{2k-2}R(n,k) + n^{2k-2}T(n)$.

Next, we prove correctness. Consider a $V_1$-maximal minsum $k$-partition $(V_1, V_2, \ldots, V_k)$ such that $V_1$ is the cheapest part. By Theorem 4.1, there exist subsets $S, T \subseteq V$ of size at most $k-1$ such that $(V_1, \overline{V_1})$ is the source maximal minimum $(S, T)$-terminal cut. Hence, $V_1 \in \mathcal{C}$. Consequently, SUBMOD- $(k-1)$- PART on the input instance $f : 2^{V-V_1} \to \mathbb{R}$, i.e., the function $f$ restricted to $V - V_1$, will return an optimum $(k-1)$-partition $(P_1^{V_1}, \ldots, P_{k-1}^{V_1})$ of $V - V_1$ with respect to the function $f$ restricted to $V - V_1$.

We claim that the $k$-partition $(V_1, P_1^{V_1}, \ldots, P_{k-1}^{V_1})$ is indeed an optimum $k$-partition of $V$ for the function $f$. For the sake of contradiction, suppose that $\sum_{i=1}^{k} f(V_i) < f(V_1) + \sum_{j=1}^{k-1} f(P_j^{V_1})$. Then, $\sum_{i=2}^{k} f(V_i) < \sum_{j=1}^{k-1} f(P_j^{V_1})$ and moreover, $V_2, \ldots, V_k$ is a partition of $V - V_1$. Thus, $V_2, \ldots, V_k$ contradict the optimality of the $(k-1)$-partition $(P_1^{V_1}, \ldots, P_{k-1}^{V_1})$ of $V - V_1$ with respect to the function $f$ restricted to $V - V_1$. $\square$

We further note that a form of the restriction approach allows us to solve GRAPH- $k$- CUT in $n^{O(k^2)}$ time: we can enumerate a collection of $n^{2k-2}$ candidate sets for $V_1$ and for each candidate set $U$ in the collection, we recurse on $G[V \setminus U]$ to find a minsum $(k-1)$-partition, concatenate it with $U$ to obtain a $k$-partition, and return the best of all $k$-partitions.

## 5 Conclusion

Given the general sense that it is harder to design algorithms/approximations for the minmax objective than the minsum objective, our result is somewhat surprising: we designed a polynomial-time algorithm for MINMAX- SYMSUBMOD- $k$- PART for all fixed $k$ while such a result is not yet known for SYM- SUBMOD- $k$- PART (or even for the special case of HYPERGRAPH- $k$- PART). As a special case, our algorithm resolves the complexity of MINMAX- HYPERGRAPH- $k$- PART for fixed $k$ which was posed by Lawler in 1973. Our key technical contribution is a structural theorem (Theorem 1.3) that enables efficient recovery of each part of an optimum minmax $k$-partition by solving minimum $(S, T)$-terminal cuts. We were able to adapt the ideas underlying the proof of the structural theorem to prove a claim for symmetric submodular functions under the minsum objective that helps solve SYM- SUBMOD- $k$- PART for $k \le 5$—see Theorem 4.1, Corollary 4.1, and [16, 23]. It would be interesting to see if Theorem 4.1 can be generalized/adapted to obtain a polynomial-time algorithm for SYM- SUBMOD- $k$- PART for fixed $k$. Subsequent to the publication of this work, Beideman, Chandrasekaran, and Wang [2] used the minimum $(S, T)$-terminal cuts approach to also enumerate all subsets of hyperedges that cross an optimum partition for a given instance of MINMAX- HYPERGRAPH- $k$- PART in polynomial time for fixed $k$.

**Acknowledgements** We thank Weihang Wang for helpful discussions during preliminary stages of this work and pointing to the failure of the Gomory–Hu tree based approach for the minmax objective. We also thank the reviewers for carefully reading and providing constructive feedback.

## Appendix A: Proof of Theorem 4.1

Our proof strategy for Theorem 4.1 is similar to the proof of Theorem 1.3: The first step shows a containment lemma similar to Lemma 2.1, but for the minsum objective. The second step will use the uncrossing result from Theorem 2.1. We begin with the containment lemma for the minsum objective.

**Lemma A.1** *Let $f : 2^V \to \mathbb{R}$ be a symmetric submodular function, $k \geq 2$ be an integer, $(V_1, \ldots, V_k)$ be a $V_1$-maximal minsum $k$-partition with respect to $f$, and $S \subseteq V_1$, $T \subseteq \overline{V_1}$ such that $T \cap V_j \neq \emptyset$ for every $j \in \{2, \ldots, k\}$. Suppose $(U, \overline{U})$ is a minimum $(S, T)$-terminal cut. Then, $U \subseteq V_1$.*

***Proof*** The proof strategy is identical to the proof of the containment lemma for the minmax objective (i.e., Lemma 2.1). For the sake of contradiction, suppose $U \setminus V_1 \neq \emptyset$. Consider $W_1 := V_1 \cup U$ and $W_j := V_j - U$ for every $j \in \{2, \ldots, k\}$ (see Fig. 1).

Since $W_1 \supseteq S \neq \emptyset$ and $W_j \supseteq T \cap V_j \neq \emptyset$ for all $j \in \{2, \ldots, k\}$, we have that $(W_1, \ldots, W_k)$ is a $k$-partition. We note that Claim 2.1 is still applicable for the minsum setting as well. As a consequence of this claim, the sum-objective value of the $k$-partition $(W_1, \ldots, W_k)$ is at most that of $(V_1, \ldots, V_k)$. Hence, $(W_1, \ldots, W_k)$ is a minsum $k$-partition. Moreover, $W_1$ is a strict superset of $V_1$ as $U \setminus V_1 \neq \emptyset$ and hence, $(W_1, \ldots, W_k)$ contradicts $V_1$-maximality of the minsum $k$-partition $(V_1, \ldots, V_k)$. □

We are now ready to prove Theorem 4.1.

**Theorem 4.1** *Let $f : 2^V \to \mathbb{R}$ be a symmetric submodular function and let $k \geq 2$ be an integer. Let $(V_1, V_2, \ldots, V_k)$ be a $V_1$-maximal minsum $k$-partition such that $V_1$ is the cheapest part. Then, for every subset $T \subseteq \overline{V_1}$ such that $T \cap V_j \neq \emptyset$ for every $j \in \{2, \ldots, k\}$, there exists a subset $S \subseteq V_1$ of size at most $k-1$ such that $(V_1, \overline{V_1})$ is the source maximal minimum $(S, T)$-terminal cut.*

***Proof*** The proof is almost identical to the proof of Theorem 1.3 except for the last but one sentence.

Suppose $|V_1| \leq k-1$. Then, we consider $S = V_1$. We have that $|S| \leq k-1$ and moreover, using Lemma A.1, we have that $(V_1, \overline{V_1})$ is the source maximal minimum $(S, T)$-terminal cut for every $T \subseteq \overline{V_1}$ such that $T \cap V_j \neq \emptyset$ for every $j \in \{2, \ldots, k\}$, thus proving the theorem. We consider the case of $|V_1| \geq k$ in the rest of the proof.

For the sake of contradiction, suppose that the theorem is false for some subset $T \subseteq \overline{V_1}$ such that $T \cap V_j \neq \emptyset$ for all $j \in \{2, \ldots, k\}$. Our proof strategy is to obtain a cheaper $k$-partition than $(V_1, \ldots, V_k)$, thereby contradicting the optimality of $(V_1, \ldots, V_k)$. Let $OPT_k$ denote the sum-objective value of $(V_1, \ldots, V_k)$. For a subset $X \subseteq V_1$, let $(V_X, \overline{V_X})$ be the source maximal minimum $(X, T)$-terminal cut. By Lemma A.1, we have that $V_X \subseteq V_1$ for all $X \subseteq V_1$.

Among all possible subsets of $V_1$ of size $k-1$, pick a subset $S$ such that $f(V_S)$ is maximum. Then, by Lemma A.1 and assumption, we have that $V_S \subsetneq V_1$. By source maximality of the minimum $(S, T)$-terminal cut $(V_S, \overline{V_S})$, we have that $f(V_S) < f(V_1)$. Let $u_1, \ldots, u_{k-1}$ be the vertices in $S$. Since $V_S \subsetneq V_1$, there exists a vertex $u_k \in V_1 \backslash V_S$. Let $C := \{u_1, \ldots, u_k\} = S \cup \{u_k\}$. For $i \in [k]$, let $(B_i, \overline{B_i})$ be the source maximal minimum $(C - \{u_i\}, T)$-terminal cut. We note that $(B_k, \overline{B_k}) = (V_S, \overline{V_S})$ and the size of $C - \{u_i\}$ is $k - 1$ for every $i \in [k]$. By Lemma 2.1 and assumption, we have that $B_i \subsetneq V_1$ for every $i \in [k]$. Hence, we have

$$f(B_i) \leq f(V_S) < f(V_1) \text{ and } B_i \subsetneq V_1 \text{ for every } i \in [k]. \tag{2}$$

The next claim will set us up to apply Theorem 2.1.

**Claim A.1** *For every $i \in [k]$, we have that $u_i \in \overline{B_i}$.*

***Proof*** The claim holds for $i = k$ by choice of $u_k$. For the sake of contradiction, suppose $u_i \in B_i$ for some $i \in [k - 1]$. Then, the 2-partition $(V_S \cap B_i, \overline{V_S \cap B_i})$ is a $(S, T)$-terminal cut while $(V_S, \overline{V_S})$ is a minimum $(S, T)$-terminal cut and hence

$$f(V_S \cap B_i) \geq f(V_S).$$

We also have that

$$f(V_S \cup B_i) \geq f(V_S)$$

since $(V_S \cup B_i, \overline{V_S \cup B_i})$ is a $(S, T)$-terminal cut while $(V_S, \overline{V_S})$ is a minimum $(S, T)$-terminal cut. Thus,

$$\begin{aligned}
2f(V_S) &\geq f(V_S) + f(B_i) && \text{(By choice of } S\text{)}\\
&\geq f(V_S \cup B_i) + f(V_S \cap B_i) && \text{(By submodularity)}\\
&\geq 2f(V_S). && \text{(By the inequalities above)}
\end{aligned}$$

Therefore, all inequalities above should be equations and hence, $f(V_S \cup B_i) = f(V_S)$. Consequently, the 2-partition $(V_S \cup B_i, \overline{V_S \cup B_i})$ is a minimum $(S, T)$-terminal cut. However, this contradicts source maximality of the minimum $(S, T)$-terminal cut $(V_S, \overline{V_S})$ since $u_k \in B_i$ and $u_k \notin V_S$. □

We note that for every $i \in [k]$, the 2-partition $(B_i, \overline{B_i})$ is a minimum $(C - \{u_i\}, \overline{V_1})$-terminal cut since $\overline{V_1} \subseteq \overline{B_i}$.

We will now apply Theorem 2.1. We consider $U := V_1$ and $C = \{u_1, \ldots, u_k\} \subseteq U$. Let $(\overline{A_i}, A_i) := (B_i, \overline{B_i})$ for every $i \in [k]$. The 2-partition $(\overline{A_i}, A_i)$ is a minimum $(C \setminus \{u_i\}, \overline{U})$-terminal cut for every $i \in [k]$. By Claim A.1, we have that $u_i \in A_i$ for every $i \in [k]$. Since $(B_j, \overline{B_j})$ is a $(C - \{u_j\}, T)$-terminal cut, we have that $u_i \notin \overline{B_j}$ for every distinct $i, j \in [k]$. Thus, $u_i \in A_i \backslash (\cup_{j \in [k] \backslash \{i\}} A_j)$ for every $i \in [k]$. We may reindex the elements in $C$ so that $f(A_1) \leq f(A_2) \leq \ldots \leq f(A_k)$. Therefore, the sets $U$, $C$, and the 2-partitions $(\overline{A_i}, A_i)$ for $i \in [k]$ satisfy the conditions of Theorem 2.1.

By Theorem 2.1 and statement (2), we obtain a $k$-partition $(P_1, \ldots, P_k)$ of $V$ such that

$$\sum_{i=1}^{k} f(P_i) \leq k \max\{f(A_i) : i \in [k]\} = kf(V_S) < kf(V_1) \leq \sum_{i=1}^{k} f(V_i) = OPT_k.$$

Thus, we have obtained a $k$-partition whose sum-objective value is strictly smaller than $OPT_k$, a contradiction. $\square$

## References


1. Bansal, N., Feige, U., Krauthgamer, R., Makarychev, K., Nagarajan, V., Naor, J., Schwartz, R.: Min-max graph partitioning and small set expansion. SIAM J. Comput. **43**(2), 872–904 (2014)
2. Beideman, C., Chandrasekaran, K., Wang, W.: Counting and enumerating optimum cut sets for hypergraph $k$-partitioning problems for fixed $k$. In: International Colloquium on Automata, Languages and Programming, ICALP, pp. 16:1–16:18 (2022)
3. Bernáth, A., Király, Z.: On the tractability of some natural packing, covering and partitioning problems. Discret. Appl. Math. **180**, 25–35 (2015)
4. Chandrasekaran, K., Chekuri, C.: Hypergraph $k$-cut for fixed $k$ in deterministic polynomial time, Mathematics of Operations Research (2022), Prelim. version in FOCS (2020)
5. Chandrasekaran, K., Xu, C., Yu, X.: Hypergraph $k$-Cut in randomized polynomial time. Math. Program. **186**, 85–113 (2021)
6. Chandrasekaran, K., Zhang, X.: Private communication
7. Chekuri, C., Ene, A.: Approximation algorithms for submodular multiway partition. In: Proceedings of the 52nd IEEE Annual Symposium on Foundations of Computer Science, FOCS, pp. 807–816 (2011)
8. Chekuri, C., Li, S.: On the hardness of approximating the $k$-way hypergraph cut problem. Theory Comput. **16**(14), 1–8 (2020)
9. Chekuri, C., Quanrud, K., Xu, C.: LP relaxation and tree packing for minimum $k$-cut. SIAM J. Discret. Math. **34**(2), 1334–1353 (2020)
10. Fox, K., Panigrahi, D., Zhang, F.: Minimum cut and minimum $k$-cut in hypergraphs via branching contractions. In: Proceedings of the 30th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA, pp. 881–896 (2019)
11. Goemans, M.X., Ramakrishnan, V.S.: Minimizing submodular functions over families of sets. Combinatorica **15**, 499–513 (1995)
12. Goldschmidt, O., Hochbaum, D.: A polynomial algorithm for the $k$-cut problem for fixed $k$. Math. Oper. Res. **19**(1), 24–37 (1994)
13. Guinez, F., Queyranne, M.: The size-constrained submodular k-partition problem, Unpublished manuscript. https://docs.google.com/viewer?a=v&pid=sites&srcid=ZGVmYXVsdGRvbWFpbnxmbGF2aW9ndWluZXpob21lcGFnZXxneDo0NDVlMThkMDg4ZWRlOGI1. See also https://smartech.gatech.edu/bitstream/handle/1853/43309/Queyranne.pdf (2012)
14. Gupta, A., Harris, D., Lee, E., Li, J.: Optimal bounds for the $k$-cut problem. Preprint in arXiv arxiv:2005.08301 (2020)
15. Gupta, A., Lee, E., Li, J.: The Karger-Stein algorithm is optimal for $k$-cut. In: Proceedings of the 52nd annual ACM SIGACT symposium on theory of computing, STOC, pp. 473–484 (2020)
16. Hirayama, T., Liu, K., Makino, Y., Shi, K., Xu, C.: A polynomial time algorithm for finding a minimum 4-partition of a submodular function, (To appear) Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms, SODA (2023)
17. Kamidoi, Y., Yoshida, N., Nagamochi, H.: A deterministic algorithm for finding all minimum $k$-way cuts. SIAM J. Comput. **36**(5), 1329–1341 (2007)
18. Karger, D.: Minimum cuts in near-linear time. J. ACM **47**(1), 46–76 (2000)
19. Karger, D., Stein, C.: A new approach to the minimum cut problem. J. ACM **43**(4), 601–640 (1996)
20. Lawler, E.: Cutsets and partitions of hypergraphs. Networks **3**, 275–285 (1973)

21. Manurangsi, P.: Inapproximability of maximum biclique problems, minimum k-cut and densest at-least-k-subgraph from the small set expansion hypothesis. Algorithms **11**(1), 1–22 (2018)
22. Nägele, M., Sudakov, B., Zenklusen, R.: Submodular minimization under congruency constraints. Combinatorica **39**, 1351–1386 (2019)
23. Okumoto, K., Fukunaga, T., Nagamochi, H.: Divide-and-conquer algorithms for partitioning hypergraphs and submodular systems. Algorithmica **62**(3), 787–806 (2012)
24. Queyranne, M.: On optimum size-constrained set partitions, Talk at Aussois workshop on Combinatorial Optimization (1999)
25. Saran, H., Vazirani, V.: Finding k cuts within twice the optimal. SIAM J. Comput. **24**(1), 101–108 (1995)
26. Schrijver, A.: Combinatorial Optimization: Polyhedra and Efficiency, Algorithms and Combinatorics. Springer, New York (2003)
27. Svitkina, Z., Fleischer, L.: Submodular approximation: sampling-based algorithms and lower bounds. SIAM J. Comput. **40**(6), 1715–1737 (2011)
28. Svitkina, Z., Tardos, É.: Min-max multiway cut. In: Approximation, Randomization, and Combinatorial Optimization. APPROX, Algorithms and Techniques, pp. 207–218 (2004)
29. Thorup, M.: Minimum $k$-way cuts via deterministic greedy tree packing. In: Proceedings of the 40th Annual ACM Symposium on Theory of Computing, STOC, pp. 159–166 (2008)
30. Wei, K., Iyer, R., Wang, S., Bai, W., Bilmes, J.: Mixed robust/average submodular partitioning: fast algorithms, guarantees, and applications. Adv. Neural Inf. Process. Syst. **28**, 2233–2241 (2015)
31. Xiao, M.: An improved divide-and-conquer algorithm for finding all minimum k-way cuts. In: Proceedings of 19th International Symposium on Algorithms and Computation, ISAAC, pp. 208–219 (2008)