Counting and Enumerating Optimum Cut Sets for Hypergraph k-Partitioning Problems for Fixed k

Calvin Beideman
University of Illinois Urbana-Champaign, IL, USA

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

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

Abstract

We consider the problem of enumerating optimal solutions for two hypergraph k-partitioning problems – namely, HYPERGRAPH-k-CUT and MINMAX-HYPERGRAPH-k-PARTITION. The input in hypergraph k-partitioning problems is a hypergraph $G = (V, E)$ with positive hyperedge costs along with a fixed positive integer k. The goal is to find a partition of V into k non-empty parts $(V_1, V_2, \ldots, V_k)$ – known as a k-partition – so as to minimize an objective of interest.

1. If the objective of interest is the maximum cut value of the parts, then the problem is known as MINMAX-HYPERGRAPH-k-PARTITION. A subset of hyperedges is a MINMAX-k-CUT-SET if it is the subset of hyperedges crossing an optimum k-partition for MINMAX-HYPERGRAPH-k-PARTITION.
2. If the objective of interest is the total cost of hyperedges crossing the k-partition, then the problem is known as HYPERGRAPH-k-CUT. A subset of hyperedges is a MIN-k-CUT-SET if it is the subset of hyperedges crossing an optimum k-partition for HYPERGRAPH-k-CUT.

We give the first polynomial bound on the number of MINMAX-k-CUT-SETs and a polynomial-time algorithm to enumerate all of them in hypergraphs for every fixed k. Our technique is strong enough to also enable an $n^{O(k)}p$-time deterministic algorithm to enumerate all MIN-k-CUT-SETs in hypergraphs, thus improving on the previously known $n^{O(k^2)}p$-time deterministic algorithm, where n is the number of vertices and p is the size of the hypergraph. The correctness analysis of our enumeration approach relies on a structural result that is a strong and unifying generalization of known structural results for HYPERGRAPH-k-CUT and MINMAX-HYPERGRAPH-k-PARTITION. We believe that our structural result is likely to be of independent interest in the theory of hypergraphs (and graphs).

2012 ACM Subject Classification Theory of computation → Graph algorithms analysis

Keywords and phrases hypergraphs, k-partitioning, counting, enumeration

Digital Object Identifier 10.4230/LIPIcs.ICALP.2022.16

Category Track A: Algorithms, Complexity and Games

Related Version *Full Version*: https://arxiv.org/abs/2204.09178 [7]

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

1 Introduction

In hypergraph k-partitioning problems, the input consists of a hypergraph $G = (V, E)$ with positive hyperedge-costs $c : E \to \mathbb{R}_+$ and a fixed positive integer k (e.g., $k = 2, 3, 4, \ldots$). The goal is to find a partition of the vertex set into k *non-empty* parts $V_1, V_2, \ldots, V_k$ so as to minimize an objective of interest. There are several natural objectives of interest in hypergraph k-partitioning problems. In this work, we focus on two particular objectives: MINMAX-HYPERGRAPH-k-PARTITION and HYPERGRAPH-k-CUT:

© Calvin Beideman, Karthekeyan Chandrasekaran, and Weihang Wang;
licensed under Creative Commons License CC-BY 4.0
49th International Colloquium on Automata, Languages, and Programming (ICALP 2022).
Editors: Mikołaj Bojańczyk, Emanuela Merelli, and David P. Woodruff;
Article No. 16; pp. 16:1–16:18
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1. In Minmax-Hypergraph-k-Partition, the objective is to minimize the maximum cut value of the parts of the k-partition – i.e., minimize $\max_{i=1}^{k} c(\delta(V_i))$; here $\delta(V_i)$ is the set of hyperedges intersecting both V_i and $V \setminus V_i$ and $c(\delta(V_i)) = \sum_{e\in\delta(V_i)} c(e)$ is the total cost of hyperedges in $\delta(V_i)$.
2. In Hypergraph-k-Cut[1], the objective is to minimize the cost of hyperedges crossing the k-partition – i.e., minimize $c(\delta(V_1,\ldots,V_k))$; here $\delta(V_1,\ldots,V_k)$ is the set of hyperedges that intersect at least two sets in $\{V_1,\ldots,V_k\}$ and $c(\delta(V_1,\ldots,V_k)) = \sum_{e\in\delta(V_1,\ldots,V_k)} c(e)$ is the total cost of hyperedges in $\delta(V_1,\ldots,V_k)$.

If the input G is a graph, then we will refer to these problems as Minmax-Graph-k-Partition and Graph-k-Cut respectively. We note that the case of $k = 2$ corresponds to global minimum cut in both objectives. In this work, we focus on the problem of enumerating all optimum solutions to Minmax-Hypergraph-k-Partition and Hypergraph-k-Cut.

Motivations and Related Problems. We consider the problem of counting and enumerating optimum solutions for partitioning problems over hypergraphs for three reasons. Firstly, hyperedges provide more powerful modeling capabilities than edges and consequently, several problems in hypergraphs become non-trivial in comparison to graphs. Although hypergraphs and partitioning problems over hypergraphs (including Minmax-Hypergraph-k-Partition) were discussed as early as 1973 by Lawler [33], most of these problems still remain open. The powerful modeling capability of hyperedges has been useful in a variety of modern applications, which in turn, has led to a resurgence in the study of hypergraphs with recent works focusing on min-cuts, cut-sparsifiers, spectral-sparsifiers, etc. [6, 8, 12, 15, 17, 18, 20, 21, 28, 32, 38]. Our work adds to this rich and emerging theory of hypergraphs.

Secondly, hypergraph k-partitioning problems are special cases of submodular k-partitioning problems. In submodular k-partitioning problems, the input is a finite ground set V, a submodular function[2] $f : 2^V \to \mathbb{R}$ provided by an evaluation oracle[3] and a positive integer k (e.g., $k = 2, 3, 4, \ldots$). The goal is to partition the ground set V into k non-empty parts $V_1, V_2, \ldots, V_k$ so as to minimize an objective of interest. Two natural objectives are of interest: (1) In Minmax-Submod-k-Partition, the objective is to minimize $\max_{i=1}^{k} f(V_i)$ and (2) In Minsum-Submod-k-Partition, the objective is to minimize $\sum_{i=1}^{k} f(V_i)$. If the given submodular function is symmetric[4], then we denote the resulting problems as Minmax-SymSubmod-k-Partition and Minsum-SymSubmod-k-Partition respectively. Since the hypergraph cut function is symmetric submodular, it follows that Minmax-Hypergraph-k-Partition is a special case of Minmax-SymSubmod-k-Partition. Moreover, Hypergraph-k-Cut is a special case of Minsum-Submod-k-Partition (this reduction is slightly non-trivial with the submodular function in the reduction being asymmetric – e.g., see [36] for the reduction). Queyranne claimed, in 1999, a polynomial-time algorithm for Minsum-SymSubmod-k-Partition for every fixed k [37], however the claim was retracted subsequently (see [24]). The complexity status of submodular k-partitioning problems (for fixed $k \geq 4$) are open, so recent works have focused on hypergraph k-partitioning problems as a stepping stone towards submodular k-partitioning [8, 12, 13, 24, 36, 41, 42]. Our work contributes to this stepping stone by advancing the state of the art in hypergraph k-partitioning problems. We emphasize that the

[1] We emphasize that the objective of Hypergraph-k-Cut is not equivalent to minimizing $\sum_{i=1}^{k} c(\delta(V_i))$.
[2] A real-valued set function $f : 2^V \to \mathbb{R}$ is submodular if $f(A) + f(B) \geq f(A\cap B) + f(A\cup B)\ \forall\ A, B \subseteq V$.
[3] An evaluation oracle for a set function f over a ground set V returns the value of $f(S)$ given $S \subseteq V$.
[4] A real-valued set function $f : 2^V \to \mathbb{R}$ is symmetric if $f(A) = f(V \setminus A)\ \forall\ A \subseteq V$.

complexity status of two other variants of hypergraph k-partitioning problems which are also special cases of Minsum-Submod-k-Partition are still open (see [36, 41, 42] for these variants).

Thirdly, counting and enumeration of optimum solutions for *graph* k-partitioning problems are fundamental to graph theory and extremal combinatorics. They have found farther reaching applications than initially envisioned. We discuss some of the results and applications for $k = 2$ and $k > 2$ now. For $k = 2$ in connected graphs, it is well-known that the number of min-cuts and the number of α-approximate min-cuts are at most $\binom{n}{2}$ and $O(n^{2\alpha})$ respectively, and they can all be enumerated in polynomial time for constant α. These combinatorial results have been the crucial ingredients of several algorithmic and representation results in graphs. On the algorithmic front, these results enable fast randomized construction of graph skeletons which, in turn, plays a crucial role in fast algorithms to solve graph min-cut [29]. On the representation front, counting results form the backbone of cut sparsifiers which in turn have found applications in sketching and streaming [2–4, 32]. A polygon representation of the family of 6/5-approximate min-cuts in graphs was given by Benczur and Goemans in 1997 (see [9–11]) – this representation was used in the recent groundbreaking $(3/2 - \epsilon)$-approximation for metric TSP [31]. On the approximation front, in addition to the $(3/2 - \epsilon)$-approximation for metric TSP [31], counting results also led to the recent 1.5-approximation for path TSP [40]. For $k > 2$, we note that fast algorithms for Graph-k-Cut have been of interest since they help in generating cutting planes while solving TSP [5, 19]. A recent series of works aimed towards improving the bounds on the number of optimum solutions for Graph-k-Cut culminated in a drastic improvement in the run-time to solve Graph-k-Cut [25–27]. Given the status of counting and enumeration results for k-partitioning in graphs and their algorithmic and representation implications that were discovered subsequently, we believe that a similar understanding in hypergraphs could serve as an important ingredient in the algorithmic and representation theory of hypergraphs.

The Enumeration Problem. There is a fundamental structural distinction between hypergraphs and graphs that becomes apparent while enumerating optimum solutions to k-partitioning problems. In connected graphs, the number of optimum k-partitions for Graph-k-Cut and for Minmax-Graph-k-Partition are $n^{O(k)}$ and $n^{O(k^2)}$ respectively and they can all be enumerated in polynomial time, where n is the number of vertices in the input graph [14, 16, 25, 27, 30, 39]. In contrast, a connected hypergraph could have exponentially many optimum k-partitions for both Minmax-Hypergraph-k-Partition and Hypergraph-k-Cut even for $k = 2$ – e.g., consider the hypergraph with a single hyperedge containing all vertices; we will denote this as the spanning-hyperedge-example. Hence, enumerating all optimum k-partitions for hypergraph k-partitioning problems in polynomial time is impossible. Instead, our goal in the enumeration problems is to enumerate *k-cut-sets* corresponding to optimum k-partitions. We will call a subset $F \subseteq E$ of hyperedges to be a k-cut-set if there exists a k-partition $(V_1, \ldots, V_k)$ such that $F = \delta(V_1, \ldots, V_k)$; we will call a 2-cut-set as a cut-set. In the enumeration problems that we will consider, the input consists of a hypergraph $G = (V, E)$ with positive hyperedge-costs $c : E \to \mathbb{R}_+$ and a fixed positive integer k (e.g., $k = 2, 3, 4, \ldots$).

1. For an optimum k-partition $(V_1, \ldots, V_k)$ for Minmax-Hypergraph-k-Partition in (G, c), we will denote $\delta(V_1, \ldots, V_k)$ as a minmax-k-cut-set. In Enum-MinMax-Hypergraph-k-Partition, the goal is to enumerate all minmax-k-cut-sets.
2. For an optimum k-partition $(V_1, \ldots, V_k)$ for Hypergraph-k-Cut in (G, c), we will denote $\delta(V_1, \ldots, V_k)$ as a min-k-cut-set. In Enum-Hypergraph-k-Cut, the goal is to enumerate all min-k-cut-sets.

We observe that in the spanning-hyperedge-example, although the number of optimum k-partitions for Minmax-Hypergraph-k-Partition (as well as Hypergraph-k-Cut) is exponential, the number of minmax-k-cut-sets (as well as min-k-cut-sets) is only one.

1.1 Results

In contrast to graphs, whose representation size is the number of edges, the representation size of a hypergraph $G = (V, E)$ is $p := \sum_{e \in E} |e|$. Throughout, our algorithmic discussion will focus on the case of fixed k (e.g., $k = 2, 3, 4, \ldots$).

There are no prior results regarding Enum-MinMax-Hypergraph-k-Partition in the literature. We recall the status of Minmax-Hypergraph-k-Partition. As mentioned earlier, Minmax-Hypergraph-k-Partition was discussed as early as 1973 by Lawler [33] with its complexity status being open until recently. We note that the objective here could be viewed as aiming to find a *fair* k-partition, i.e., a k-partition where no part pays too much in cut value. Motivated by this connection to fairness, Chandrasekaran and Chekuri (2021) [13] studied the more general problem of Minmax-SymSubmod-k-Partition. They gave the first (deterministic) polynomial-time algorithm to solve Minmax-SymSubmod-k-Partition and as a consequence, obtained the first polynomial-time algorithm to solve Minmax-Hypergraph-k-Partition. Their algorithm does not show any bound on the number of minmax-k-cut-sets since it solves the more general problem of Minmax-SymSubmod-k-Partition for which the number of optimum k-partitions can indeed be exponential (recall the spanning-hyperedge-example). Focusing on hypergraphs raises the question of whether all k-cut-sets corresponding to optimum solutions can be enumerated efficiently for every fixed k. We answer this question affirmatively by giving the first polynomial-time algorithm for Enum-MinMax-Hypergraph-k-Partition.

▶ **Theorem 1.** *There exists a deterministic algorithm to solve Enum-MinMax-Hypergraph-k-Partition that runs in time $O(kn^{4k^2-2k+1}p)$, where n is the number of vertices and p is the size of the input hypergraph. Moreover, the number of minmax-k-cut-sets in a n-vertex hypergraph is $O(n^{4k^2-2k})$.*

We emphasize that our result shows the first polynomial bound on the number of minmax-k-cut-sets in hypergraphs for every fixed k (in addition to a polynomial-time algorithm to enumerate all of them for every fixed k). Our upper bound of $n^{O(k^2)}$ on the number of minmax-k-cut-sets is tight – there exist n-vertex connected *graphs* for which the number of minmax-k-cut-sets is $n^{\Theta(k^2)}$.

Next, we briefly recall the status of Hypergraph-k-Cut and Enum-Hypergraph-k-Cut. Hypergraph-k-Cut was shown to be solvable in randomized polynomial time only recently [15, 20]; the randomized algorithms also showed that the number of min-k-cut-sets is $O(n^{2k-2})$ and they can all be enumerated in randomized polynomial time. A subsequent deterministic algorithm was designed to solve Hypergraph-k-Cut in time $n^{O(k)}p$ by Chandrasekaran and Chekuri [12]. Chandrasekaran and Chekuri's techniques were extended to design the first deterministic polynomial-time algorithm to solve Enum-Hypergraph-k-Cut in [8]. The algorithm for Enum-Hypergraph-k-Cut given in [8] runs in time $n^{O(k^2)}p$. We note that this run-time has a quadratic dependence on k in the exponent of n although the number of min-k-cut-sets has only linear dependence on k in the exponent of n (since it is $O(n^{2k-2})$). So, an open question that remained from [8] is whether one can obtain an $n^{O(k)}p$-time deterministic algorithm for Enum-Hypergraph-k-Cut. We resolve this question affirmatively.

▶ **Theorem 2.** *There exists a deterministic algorithm to solve* Enum-Hypergraph-k-Cut *that runs in time $O(n^{16k-25}p)$, where n is the number of vertices and p is the size of the input hypergraph.*

Our algorithms for both Enum-MinMax-Hypergraph-k-Partition and Enum-Hypergraph-k-Cut are based on a structural theorem that allows for efficient recovery of optimum k-cut-sets via minimum (s,t)-terminal cuts (see Theorem 4). Our structural theorem builds on structural theorems that have appeared in previous works on Minmax-Hypergraph-k-Partition and Hypergraph-k-Cut [8,12,13]. Our structural theorem may appear to be natural/incremental in comparison to ones that appeared in previous works, but formalizing the theorem and proving it is a significant part of our contribution. Moreover, our single structural theorem is strong enough to enable efficient algorithms for both Enum-Hypergraph-k-Cut as well as Enum-MinMax-Hypergraph-k-Partition in contrast to previously known structural theorems. In this sense, our structural theorem can be viewed as a strong and unifying generalization of structural theorems that have appeared in previous works. We believe that our structural theorem will be of independent interest in the theory of cuts and partitioning in hypergraphs (as well as graphs).

1.2 Technical overview and main structural result

We focus on the unit-cost variant of Enum-Hypergraph-k-Cut and Enum-MinMax-Hypergraph-k-Partition in the rest of this work for the sake of notational simplicity – i.e., the cost of every hyperedge is 1. Throughout, we will allow multigraphs and hence, this is without loss of generality. Our algorithms extend in a straightforward manner to arbitrary hyperedge costs. They rely only on minimum (s,t)-terminal cut computations and hence, they are strongly polynomial-time algorithms.

Notation and background. Let $G=(V,E)$ be a hypergraph. Throughout this work, n will denote the number of vertices in G, m will denote the number of hyperedges in G, and $p := \sum_{e\in E}|e|$ will denote the representation size of G. We will denote a partition of the vertex set into h non-empty parts by an ordered tuple $(V_1,\ldots,V_h)$ and call such an ordered tuple as an h-partition. For a partition $\mathcal{P}=(V_1,V_2,\ldots,V_h)$, we will say that a hyperedge e crosses the partition $\mathcal{P}$ if it intersects at least two parts of the partition. We will refer to a 2-partition as a cut. For a non-empty proper subset U of vertices, we will use $\overline{U}$ to denote $V\setminus U$, $\delta(U)$ to denote the set of hyperedges crossing the cut $(U,\overline{U})$, and $d(U) := |\delta(U)|$ to denote the cut value of U. We observe that $\delta(U)=\delta(\overline{U})$, so we will use $d(U)$ to denote the value of the cut $(U,\overline{U})$. More generally, given a partition $\mathcal{P}=(V_1,V_2,\ldots,V_h)$, we denote the set of hyperedges crossing the partition by $\delta(V_1,V_2,\ldots,V_h)$ (also by $\delta(\mathcal{P})$ for brevity) and the number of hyperedges crossing the partition by $|\delta(V_1,V_2,\ldots,V_h)|$. We will denote the optimum value of Minmax-Hypergraph-k-Partition and Hypergraph-k-Cut respectively by

$$OPT_{\text{minmax-}k\text{-partition}} := \min\left\{\max_{i\in[k]}|\delta(V_i)| : (V_1,\ldots,V_k) \text{ is a } k\text{-partition of } V\right\} \text{ and}$$
$$OPT_{k\text{-cut}} := \min\left\{|\delta(V_1,\ldots,V_k)| : (V_1,\ldots,V_k) \text{ is a } k\text{-partition of } V\right\}.$$

A key algorithmic tool will be the use of fixed-terminal cuts. Let S, T be disjoint non-empty subsets of vertices. A 2-partition $(U,\overline{U})$ is an (S,T)-terminal cut if $S\subseteq U\subseteq V\setminus T$. 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 *source minimal* minimum (S,T)-terminal cuts. For every pair of disjoint non-empty subsets S and T of vertices, there exists a unique source minimal minimum (S,T)-terminal cut and it can be found in deterministic polynomial time via standard maxflow algorithms. In particular, the source minimal minimum (S,T)-terminal cut can be found in time $O(np)$ [17].

Our technique to enumerate all MINMAX-k-CUT-SETS and all MIN-k-CUT-SETS will build on the approaches of Chandrasekaran and Chekuri for HYPERGRAPH-k-CUT and MINMAX-SYMSUBMOD-k-PARTITION [8,12,13]. We need the following structural theorem that was shown in [8].

▶ **Theorem 3** ([8])**.** *Let $G=(V,E)$ be a hypergraph and let $OPT_{k\text{-}cut}$ be the optimum value of HYPERGRAPH-k-CUT in G for some integer $k \geq 2$. Suppose $(U,\overline{U})$ is a 2-partition of V with $d(U) < OPT_{k\text{-}cut}$. Then, for every pair of vertices $s \in U$ and $t \in \overline{U}$, there exist subsets $S \subseteq U \setminus \{s\}$ and $T \subseteq \overline{U} \setminus \{t\}$ with $|S| \leq 2k-3$ and $|T| \leq 2k-3$ such that $(U,\overline{U})$ is the unique minimum $(S \cup \{s\}, T \cup \{t\})$-terminal cut in G.*

Enum-Hypergraph-*k*-Cut. We first focus on ENUM-HYPERGRAPH-k-CUT. We note that Theorem 3 will allow us to recover those parts V_i of an optimum k-partition $(V_1,\ldots,V_k)$ for which $d(V_i) < OPT_{k\text{-cut}}$. However, recall that our goal is *not* to recover all optimum k-partitions for HYPERGRAPH-k-CUT, but rather to recover all MIN-k-CUT-SETS (i.e., not to recover the parts of every optimum k-partition, but rather only to recover the k-cut-set of every optimum k-partition). The previous work [8] that designed an $n^{O(k^2)}p$-time deterministic enumeration algorithm achieved this by proving the following structural result: suppose $(V_1,\ldots,V_k)$ is an optimum k-partition for HYPERGRAPH-k-CUT for which $d(V_1) = OPT_{k\text{-cut}}$. Then, they showed that for every subset $T \subseteq \overline{V_1}$ satisfying $T \cap V_j \neq \emptyset$ for all $j \in \{2,\ldots,k\}$, there exists a subset $S \subseteq V_1$ with $|S| \leq 2k$ such that the source minimal minimum (S,T)-terminal cut $(A,\overline{A})$ satisfies $\delta(A) = \delta(V_1)$. This structural theorem in conjunction with Theorem 3 allows one to enumerate a candidate family $\mathcal{F}$ of $n^{O(k^2)}$ subsets of hyperedges such that every MIN-k-CUT-SET is present in the family. The drawback of their structural theorem is that it is driven towards recovering the cut-set $\delta(V_i)$ of every part V_i of every optimum k-partition $(V_1,\ldots,V_k)$. Hence, their algorithmic approach ends up with a run-time of $n^{O(k^2)}p$. In order to improve the run-time, we prove a stronger result: we show that for an *arbitrary* cut $(U,\overline{U})$ with cut value $OPT_{k\text{-cut}}$ (as opposed to only those sets V_i of an optimum k-partition $(V_1,\ldots,V_k)$), its cut-set $\delta(U)$ can be recovered as the cut-set of *any* minimum (S,T)-terminal cut for some S and T of small size. The following is the main structural theorem of this work.

▶ **Theorem 4.** *Let $G=(V,E)$ be a hypergraph and let $OPT_{k\text{-}cut}$ be the optimum value of HYPERGRAPH-k-CUT in G for some integer $k \geq 2$. Suppose $(U,\overline{U})$ is a 2-partition of V with $d(U) = OPT_{k\text{-}cut}$. Then, there exist sets $S \subseteq U$, $T \subseteq \overline{U}$ with $|S| \leq 2k-1$ and $|T| \leq 2k-1$ such that every minimum (S,T)-terminal cut $(A,\overline{A})$ satisfies $\delta(A) = \delta(U)$.*

We encourage the reader to compare and contrast Theorems 3 and 4. The former helps to recover cuts whose cut value is strictly smaller than $OPT_{k\text{-cut}}$ while the latter helps to recover *cut-sets* whose size is equal to $OPT_{k\text{-cut}}$. So, the latter theorem is weaker since it only recovers cut-sets, but we emphasize that this is the best possible that one can hope to do (as seen from the spanning-hyperedge-example). However, proving the latter theorem requires us to work with cut-sets (as opposed to cuts) which is a technical barrier to overcome. Indeed, our proof of Theorem 4 deviates significantly from the proof of Theorem 3 since we have to work with cut-sets. Our proof also deviates from the structural result in [8] that was mentioned in the paragraph above Theorem 4 since our result is stronger than their result – our result helps to recover the cut-set $\delta(U)$ of an arbitrary cut $(U, \overline{U})$ whose cut value is $d(U) = OPT_{k\text{-cut}}$ while their result helps only to recover the cut-set $\delta(V_i)$ of a part V_i of an optimum k-partition $(V_1, \ldots, V_k)$ for Hypergraph-k-Cut whose cut value is $d(V_i) = OPT_{k\text{-cut}}$; moreover, their proof technique crucially relies on a containment property with respect to the part V_i, whereas under the hypothesis of our structural theorem, the containment property fails with respect to the set U and consequently, our proof technique differs from theirs.

Theorems 3 and 4 lead to a deterministic $n^{O(k)}$-time algorithm to enumerate all min-k-cut-sets via a divide-and-conquer approach. We describe this algorithm now: For each pair (S, T) of disjoint subsets of vertices S and T with $|S|, |T| \leq 2k - 1$, compute the source minimal minimum (S, T)-terminal cut $(A, \overline{A})$; (i) if $G - \delta(A)$ has at least k connected components, then add $\delta(A)$ to the candidate family $\mathcal{F}$; (ii) otherwise, add the set A to a collection $\mathcal{C}$. We note that the sizes of the family $\mathcal{F}$ and the collection $\mathcal{C}$ are $O(n^{4k-2})$. Next, for each subset A in the collection $\mathcal{C}$, recursively enumerate all min-$k/2$-cut-sets in the subhypergraphs induced by A and $\overline{A}$ respectively[5] – denoted $G[A]$ and $G[\overline{A}]$ respectively – and add $\delta(A) \cup F_1 \cup F_2$ to the family $\mathcal{F}$ for each F_1 and F_2 being min-$k/2$-cut-set in $G[A]$ and $G[\overline{A}]$ respectively. Finally, return the subfamily of k-cut-sets from the family $\mathcal{F}$ that are of smallest size.

We sketch the correctness analysis of the above approach: let $F = \delta(V_1, \ldots, V_k)$ be a min-k-cut-set with $(V_1, \ldots, V_k)$ being an optimum k-partition for Hypergraph-k-Cut. We will show that the family $\mathcal{F}$ contains F. Let $U := \cup_{i=1}^{k/2} V_i$. We note that $\delta(U) \subseteq F$. We have two possibilities: (1) Say $d(U) = |F|$. Then, $d(U) = OPT_{k\text{-cut}}$. Consequently, by Theorem 4, the min-k-cut-set F will be added to the family $\mathcal{F}$ by step (i). (2) Say $d(U) < |F|$. Then, by Theorem 3, the set $U = \cup_{i=1}^{k/2} V_i$ will be added to the collection $\mathcal{C}$ by step (ii); moreover, $F_1 := F \cap E(G[U])$ and $F_2 := F \cap E(G[\overline{U}])$ are min-$k/2$-cut-sets in $G[U]$ and $G[\overline{U}]$ respectively and they would have been enumerated by recursion, and hence, the set $\delta(U) \cup F_1 \cup F_2 = F$ will be added to the family $\mathcal{F}$. The size of the family $\mathcal{F}$ can be shown to be $n^{O(k \log k)}$ and the run-time is $n^{O(k \log k)} p$. Using the known fact that the number of min-k-cut-sets in a n-vertex hypergraph is $O(n^{2k-2})$, we can improve the run-time analysis of this approach to $n^{O(k)} p$.

Enum-MinMax-Hypergraph-k-Partition. Next, we focus on Enum-MinMax-Hypergraph-k-Partition. There is a fundamental technical issue in enumerating minmax-k-cut-sets as opposed to min-k-cut-sets. We highlight this technical issue now. Suppose we find an optimum k-partition $(V_1, \ldots, V_k)$ for Minmax-Hypergraph-k-Partition (say via Chandrasekaran and Chekuri's algorithm [13]) and store only the minmax-k-cut-set $F = \delta(V_1, \ldots, V_k)$ but forget to store the partition $(V_1, \ldots, V_k)$; now, by knowing a minmax-k-cut-set F, can we recover *some* optimum k-partition for Minmax-Hypergraph-k-Partition (not necessarily $(V_1, \ldots, V_k)$)? Or by knowing a minmax-k-cut-set F, is it even possible to find the value $OPT_{\text{minmax-}k\text{-partition}}$ without solving Minmax-Hypergraph-k-Partition from scratch again – i.e., is there an advantage to knowing a minmax-k-cut-set in order to solve Minmax-Hypergraph-k-Partition? We are not aware of such an advantage. This is in stark contrast to Hypergraph-k-Cut where knowing a min-k-cut-set enables a linear-time solution to Hypergraph-k-Cut[6].

[5] Subhypergraph $G[A]$ has vertex set A and contains all hyperedges of G which are entirely contained within A.

[6] Suppose we know a min-k-cut-set F. Then consider the connected components $Q_1, \ldots, Q_t$ in $G - F$ and create a partition $(P_1, \ldots, P_k)$ by taking $P_i = Q_i$ for every $i \in [k-1]$ and $P_k = \cup_{j=k}^{t} Q_j$; such a k-partition $(P_1, \ldots, P_k)$ will be an optimum k-partition for Hypergraph-k-Cut.

Why is this issue significant while solving Enum-MinMax-Hypergraph-k-Partition? We recall that in our approach for Enum-Hypergraph-k-Cut, the algorithm computed a polynomial-sized family $\mathcal{F}$ containing all min-k-cut-sets and returned the ones with smallest size – the smallest size ones will exactly be min-k-cut-sets. It is unclear if a similar approach could work for enumerating minmax-k-cut-sets: suppose we do have an algorithm to enumerate a polynomial-sized family $\mathcal{F}$ containing all minmax-k-cut-sets; now, in order to return all minmax-k-cut-sets (which is a subfamily of $\mathcal{F}$), note that we need to identify them among the ones in the family $\mathcal{F}$ – i.e., we need to verify if a given subset $F \in \mathcal{F}$ of hyperedges is a minmax-k-cut-set; this verification problem is closely related to the question mentioned in the previous paragraph. We do not know how to address this verification problem directly. So, our algorithmic approach for Enum-MinMax-Hypergraph-k-Partition has to overcome this technical issue.

Our ingredient to overcome this technical issue is to enumerate *representatives* for minmax-k-cut-sets. For a k-partition $(V_1, \ldots, V_k)$ and disjoint subsets $U_1, \ldots, U_k \subseteq V$, we will call the k-tuple $(U_1, \ldots, U_k)$ to be a *k-cut-set representative* of $(V_1, \ldots, V_k)$ if $U_i \subseteq V_i$ and $\delta(U_i) = \delta(V_i)$ for all $i \in [k]$. We note that a fixed k-partition $(V_1, \ldots, V_k)$ could have several k-cut-set representatives and a fixed k-tuple $(U_1, \ldots, U_k)$ could be the k-cut-set representative of several k-partitions. Yet, it is possible to efficiently verify if a given k-tuple $(U_1, \ldots, U_k)$ is a k-cut-set representative. Moreover, knowing a k-cut-set representative $(U_1, \ldots, U_k)$ of a k-partition $(V_1, V_2, \ldots, V_k)$ allows one to recover the k-cut-set $F := \delta(V_1, \ldots, V_k)$ since $F = \cup_{i=1}^{k} \delta(U_i)$. Thus, in order to enumerate all minmax-k-cut-sets, it suffices to enumerate k-cut-set representatives of *all* optimum k-partitions for Minmax-Hypergraph-k-Partition. At this point, the astute reader may wonder if there exists a polynomial-sized family of k-cut-set representatives of all optimum k-partitions for Minmax-Hypergraph-k-Partition given that the number of optimum k-partitions for Minmax-Hypergraph-k-Partition could be exponential. For example, is there a polynomial-sized family of k-cut-set representatives of all optimum k-partitions for Minmax-Hypergraph-k-Partition in the spanning-hyperedge-example? Indeed, in the spanning-hyperedge-example, even though the number of optimum k-partitions for Minmax-Hypergraph-k-Partition is exponential, there exists a $(k!\binom{n}{k})$-sized family of k-cut-set representatives of all optimum k-partitions: consider the family $\{(\{v_1\}, \ldots, \{v_k\}) : v_1, \ldots, v_k \in V, v_i \neq v_j \ \forall \text{ distinct } i, j \in [k]\}$.

It turns out that Theorems 3 and 4 are strong enough to enable efficient enumeration of k-cut-set representatives of all optimum k-partitions for Minmax-Hypergraph-k-Partition. We describe the algorithm to achieve this: For each pair (S, T) of disjoint subsets of vertices with $|S|, |T| \leq 2k-1$, compute the source minimal minimum (S, T)-terminal cut $(U, \overline{U})$ and add U to a candidate collection $\mathcal{C}$. We note that the size of the collection $\mathcal{C}$ is $O(n^{4k-2})$. Next, for each k-tuple $(U_1, \ldots, U_k) \in \mathcal{C}^k$, verify if $(U_1, \ldots, U_k)$ is a k-cut-set representative and if so, then add the k-tuple to the candidate family $\mathcal{D}$. Finally, return $\arg\min\{\max_{i=1}^{k} d(U_i) : (U_1, \ldots, U_k) \in \mathcal{D}\}$, i.e., prune and return the subfamily of k-cut-set representatives $(U_1, \ldots, U_k)$ from the family $\mathcal{D}$ that have minimum $\max_{i=1}^{k} d(U_i)$.

We note that the size of the family $\mathcal{D}$ is $n^{O(k^2)}$ and consequently, the run-time is $n^{O(k^2)}p$. We sketch the correctness analysis of the above approach: let $(V_1, \ldots, V_k)$ be an optimum k-partition for Minmax-Hypergraph-k-Partition. We will show that the family $\mathcal{D}$ contains a k-cut-set representative of $(V_1, \ldots, V_k)$. By noting that $OPT_{\text{minmax-}k\text{-partition}} \leq OPT_{k\text{-cut}}$ and by Theorems 3 and 4, for every $i \in [k]$, we have a set U_i in the collection $\mathcal{C}$ with $U_i \subseteq V_i$ and $\delta(U_i) = \delta(V_i)$. Hence, the k-tuple $(U_1, \ldots, U_k) \in \mathcal{C}^k$ is a k-cut-set representative and it will be added to the family $\mathcal{D}$. The final pruning step will not remove $(U_1, \ldots, U_k)$ from the family $\mathcal{D}$ and hence, it will be in the subfamily returned by the algorithm.

Significance of our technique. As mentioned earlier, our techniques build on the structural theorems that appeared in previous works [8, 12, 13]. The main technical novelty of our contribution lies in Theorem 4 which can be viewed as the culmination of structural theorems developed in those previous works. We also emphasize that using minimum (s,t)-terminal cuts to solve global partitioning problems is not a new technique per se (e.g., minimum (s,t)-terminal cut is the first and most natural approach to solve global minimum cut). This technique of using minimum (s,t)-terminal cuts to solve global partitioning problems has a rich variety of applications in combinatorial optimization: e.g., (1) it was used to design the first efficient algorithm for Graph-k-Cut for fixed k [23], (2) it was used to design efficient algorithms for certain constrained submodular minimization problems [22, 35], and (3) more recently, it was used to design fast algorithms for global minimum cut in graphs as well as to obtain fast Gomory-Hu trees in unweighted graphs [1, 34]. The applicability of this technique relies on identifying and proving appropriate structural results. Our Theorem 4 is such a structural result. The merit of the structural result lies in its ability to solve two different enumeration problems in hypergraph k-partitioning which was not possible via structural theorems that were developed before. Moreover, it leads to the first polynomial bound on the number of minmax-k-cut-sets in hypergraphs for every fixed k.

Organization. In Section 1.3, we recall properties of the hypergraph cut function. In Section 2, we prove a special case of Theorem 4. In Section 3, we use this special case to prove Theorem 4. We refer the reader to the full version [7] for a discussion of related work, our algorithms to prove Theorems 1 and 2, and a lower bound example. We conclude with some open questions in Section 4.

1.3 Preliminaries

Let $G = (V, E)$ be a hypergraph. Throughout, we will follow the notation mentioned in the second paragraph of Section 1.2. For disjoint $A, B \subseteq V$, we define $E(A, B) := \{e \in E : e \subseteq A \cup B, e \cap A \neq \emptyset, e \cap B \neq \emptyset\}$, and $E[A] := \{e \in E : e \subseteq A\}$. We will repeatedly rely on the fact that the hypergraph cut function $d : 2^V \to \mathbb{R}_+$ is symmetric and submodular. We recall that a set function $f : 2^V \to \mathbb{R}$ is *symmetric* if $f(U) = f(\overline{U})$ for all subsets $U \subseteq V$ and is *submodular* if $f(A) + f(B) \geq f(A \cap B) + f(A \cup B)$ for all subsets $A, B \subseteq V$.

We will need the following partition uncrossing theorem that was proved in previous works on Hypergraph-k-Cut and Enum-Hypergraph-k-Cut (see Figure 1 for an illustration of the sets that appear in the statement of Theorem 5):

▶ **Theorem 5** ([8, 12]). *Let $G = (V, E)$ be a hypergraph, $k \geq 2$ be an integer and $\emptyset \neq R \subsetneq U \subsetneq V$. Let $S = \{u_1, \ldots, u_p\} \subseteq U \setminus R$ for $p \geq 2k - 2$. Let $(\overline{A_i}, A_i)$ be a minimum $((S \cup R) \setminus \{u_i\}, \overline{U})$-terminal cut. Suppose that $u_i \in A_i \setminus (\cup_{j \in [p] \setminus \{i\}} A_j)$ for every $i \in [p]$. Then, the following two hold:*

1. *There exists a k-partition $(P_1, \ldots, P_k)$ of V with $\overline{U} \subsetneq P_k$ such that*
$$|\delta(P_1, \ldots, P_k)| \leq \frac{1}{2} \min\{d(A_i) + d(A_j) : i, j \in [p], i \neq j\}.$$
2. *Moreover, if there exists a hyperedge $e \in E$ such that e intersects $W := \cup_{1 \leq i < j \leq p}(A_i \cap A_j)$, e intersects $Z := \cap_{i \in [p]} \overline{A_i}$, and e is contained in $W \cup Z$, then the inequality in the previous conclusion is strict.*

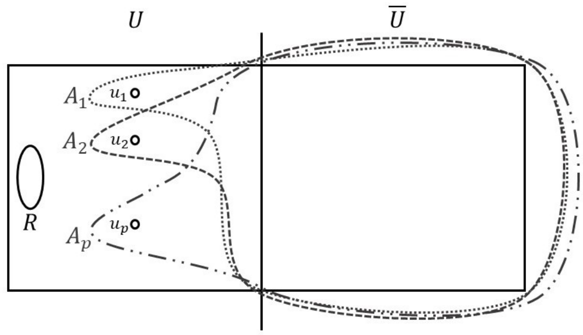

Figure 1 Illustration of the sets that appear in the statement of Theorem 5.

2 A special case of Theorem 4

The following is the main theorem of this section. Theorem 6 implies Theorem 4 in the special case where the 2-partition $(U, \overline{U})$ of interest to Theorem 4 is such that $|\overline{U}| \leq 2k-1$.

▶ **Theorem 6.** *Let $G = (V, E)$ be a hypergraph and let $OPT_{k\text{-}cut}$ be the optimum value of* Hypergraph-k-Cut *in G for some integer $k \geq 2$. Suppose $(U, \overline{U})$ is a 2-partition of V with $d(U) = OPT_{k\text{-}cut}$. Then, there exists a set $S \subseteq U$ with $|S| \leq 2k-1$ such that every minimum $(S, \overline{U})$-terminal cut $(A, \overline{A})$ satisfies $\delta(A) = \delta(U)$.*

Proof. Consider the collection

$$\mathcal{C} := \{Q \subseteq V\colon\ \overline{U} \subsetneq Q,\ d(Q) \leq d(U),\ \text{and } \delta(Q) \neq \delta(U)\}.$$

Let S be an inclusion-wise minimal subset of U such that $S \cap Q \neq \emptyset$ for all $Q \in \mathcal{C}$, i.e., the set S is completely contained in U and is a minimal transversal of the collection $\mathcal{C}$. Proposition 7 and Lemma 8 complete the proof of Theorem 6 for this choice of S. ◀

▶ **Proposition 7.** *Every minimum $(S, \overline{U})$-terminal cut $(A, \overline{A})$ has $\delta(A) = \delta(U)$.*

Proof. Let $(A, \overline{A})$ be a minimum $(S, \overline{U})$-terminal cut. If $A = U$, then we are done, so we may assume that $A \neq U$. This implies that $S \subseteq A$ and $\overline{U} \subsetneq \overline{A}$. Since $(U, \overline{U})$ is a $(S, \overline{U})$-terminal cut, we have that $d(\overline{A}) = d(A) \leq d(U)$. Since S intersects every set in the collection $\mathcal{C}$, we have that $\overline{A} \notin \mathcal{C}$. Hence, $\delta(\overline{A}) = \delta(U)$, and by symmetry of cut-sets, $\delta(A) = \delta(U)$. ◀

▶ **Lemma 8.** *The size of the subset S is at most $2k-1$.*

Proof. For the sake of contradiction, suppose $|S| \geq 2k$. Our proof strategy is to show the existence of a k-partition with fewer crossing hyperedges than $OPT_{k\text{-cut}}$, thus contradicting the definition of $OPT_{k\text{-cut}}$. Let $S := \{u_1, u_2, \ldots, u_p\}$ for some $p \geq 2k$. For notational convenience, we will use $S - u_i$ to denote $S \setminus \{u_i\}$ and $S - u_i - u_j$ to denote $S \setminus \{u_i, u_j\}$. For a subset $X \subseteq U$, we denote the source minimal minimum $(X, \overline{U})$-terminal cut by $(H_X, \overline{H_X})$.

Our strategy to arrive at a k-partition with fewer crossing hyperedges than $OPT_{k\text{-cut}}$ is to apply the second conclusion of Theorem 5. The next few claims will set us up to obtain sets that satisfy the hypothesis of Theorem 5.

▷ Claim 9. For every $i \in [p]$, we have $\overline{H_{S-u_i}} \in \mathcal{C}$.

Proof. Let $i \in [p]$. Since S is a minimal transversal of the collection $\mathcal{C}$, there exists a set $B_i \in \mathcal{C}$ such that $B_i \cap S = \{u_i\}$. Hence, $(\overline{B_i}, B_i)$ is a $(S - u_i, \overline{U})$-terminal cut. Therefore,

$$d(\overline{H_{S-u_i}}) \leq d(B_i) \leq d(U).$$

Since $(H_{S-u_i}, \overline{H_{S-u_i}})$ is a $(S - u_i, \overline{U})$-terminal cut, we have that $\overline{U} \subseteq \overline{H_{S-u_i}}$. If $d(\overline{H_{S-u_i}}) < d(U)$, then $\delta(\overline{H_{S-u_i}}) \neq \delta(U)$ and $\overline{U} \subsetneq \overline{H_{S-u_i}}$, and consequently, $\overline{H_{S-u_i}} \in \mathcal{C}$. So, we will assume henceforth that $d(\overline{H_{S-u_i}}) = d(U)$.

Since $(H_{S-u_i} \cap \overline{B_i}, \overline{H_{S-u_i} \cap \overline{B_i}})$ is a $(S - u_i, \overline{U})$-terminal cut, we have that

$$d(H_{S-u_i} \cap \overline{B_i}) \geq d(H_{S-u_i}).$$

Since $(H_{S-u_i} \cup \overline{B_i}, \overline{H_{S-u_i} \cup \overline{B_i}})$ is a $(S - u_i, \overline{U})$-terminal cut, we have that

$$d(H_{S-u_i} \cup \overline{B_i}) \geq d(H_{S-u_i}).$$

Therefore, by submodularity of the hypergraph cut function, we have that

$$2d(U) \geq d(H_{S-u_i}) + d(B_i) \geq d(H_{S-u_i} \cap \overline{B_i}) + d(H_{S-u_i} \cup \overline{B_i}) \geq 2d(H_{S-u_i}) = 2d(U). \quad (1)$$

Therefore, all inequalities above should be equations. In particular, we have that $d(H_{S-u_i} \cap \overline{B_i}) = d(U) = d(B_i) = d(H_{S-u_i})$ and hence, $(H_{S-u_i} \cap \overline{B_i}, \overline{H_{S-u_i} \cap \overline{B_i}})$ is a minimum $(S - u_i, \overline{U})$-terminal cut. Since $(H_{S-u_i}, \overline{H_{S-u_i}})$ is a source minimal minimum $(S - u_i, \overline{U})$-terminal cut, we must have $H_{S-u_i} \cap \overline{B_i} = H_{S-u_i}$, and thus $H_{S-u_i} \subseteq \overline{B_i}$. Therefore, $B_i \subseteq \overline{H_{S-u_i}}$. Since $B_i \in \mathcal{C}$, we have $\delta(B_i) \neq \delta(U)$. However, $d(B_i) = d(U)$. Therefore $\delta(U) \setminus \delta(B_i) \neq \emptyset$. Let $e \in \delta(U) \setminus \delta(B_i)$. Since $e \in \delta(\overline{U})$, but $e \notin \delta(B_i)$, and $\overline{U} \subseteq B_i$, we have that $e \subseteq B_i$, and thus $e \subseteq \overline{H_{S-u_i}}$. Thus, we conclude that $\delta(U) \setminus \delta(\overline{H_{S-u_i}}) \neq \emptyset$, and so $\delta(\overline{H_S - u_i}) \neq \delta(U)$. This also implies that $\overline{U} \subsetneq \overline{H_{S-u_i}}$. Thus, $\overline{H_{S-u_i}} \in \mathcal{C}$. ◁

Claim 9 implies the following Corollary.

▶ **Corollary 10.** *For every $i \in [p]$, we have $u_i \in \overline{H_{S-u_i}}$.*

Proof. By definition, $S - u_i \subseteq H_{S-u_i}$, so $S \cap \overline{H_{S-u_i}} \subseteq \{u_i\}$. By Claim 9 we have that $\overline{H_{S-u_i}} \in \mathcal{C}$. Since S is a transversal of the collection $\mathcal{C}$, we have that $S \cap \overline{H_{S-u_i}} \neq \emptyset$. So, the vertex u_i must be in $\overline{H_{S-u_i}}$. ◀

Having obtained Corollary 10, the next few claims (Claims 11, 13, 14, and 15) are similar to the claims appearing in the proof of a structural theorem that appeared in [8]. Since the hypothesis of the structural theorem that we are proving here is different from theirs, we present the complete proofs of these claims here. The way in which we use the claims will also be different from [8].

The following claim will help in showing that $u_i, u_j \notin H_{S-u_i-u_j}$, which in turn, will be used to show that the hypothesis of Theorem 5 is satisfied by suitably chosen sets.

▷ Claim 11. For every $i, j \in [p]$, we have $H_{S-u_i-u_j} \subseteq H_{S-u_i}$.

Proof. We may assume that $i \neq j$. We note that $(H_{S-u_i-u_j} \cap H_{S-u_i}, \overline{H_{S-u_i-u_j} \cap H_{S-u_i}})$ is a $(S - u_i - u_j, \overline{U})$-terminal cut. Therefore,

$$d(H_{S-u_i-u_j} \cap H_{S-u_i}) \geq d(H_{S-u_i-u_j}). \quad (2)$$

Also, $(H_{S-u_i-u_j} \cup H_{S-u_i}, \overline{H_{S-u_i-u_j} \cup H_{S-u_i}})$ is a $(S - u_i, \overline{U})$-terminal cut. Therefore,

$$d(H_{S-u_i-u_j} \cup H_{S-u_i}) \geq d(H_{S-u_i}). \quad (3)$$

By submodularity of the hypergraph cut function and inequalities (2) and (3), we have that

$$\begin{aligned} d(H_{S-u_i-u_j}) + d(H_{S-u_i}) &\geq d(H_{S-u_i-u_j} \cap H_{S-u_i}) + d(H_{S-u_i-u_j} \cup H_{S-u_i}) \\ &\geq d(H_{S-u_i-u_j}) + d(H_{S-u_i}). \end{aligned}$$

Therefore, inequality (2) is an equation, and consequently, $(H_{S-u_i-u_j} \cap H_{S-u_i}, \overline{H_{S-u_i-u_j} \cap H_{S-u_i}})$ is a minimum $(S-u_i-u_j, \overline{U})$-terminal cut. If $H_{S-u_i-u_j} \setminus H_{S-u_i} \neq \emptyset$, then $(H_{S-u_i-u_j} \cap H_{S-u_i}, \overline{H_{S-u_i-u_j} \cap H_{S-u_i}})$ contradicts source minimality of the minimum $(S-u_i-u_j, \overline{U})$-terminal cut $(H_{S-u_i-u_j}, \overline{H_{S-u_i-u_j}})$. Hence, $H_{S-u_i-u_j} \setminus H_{S-u_i} = \emptyset$ and consequently, $H_{S-u_i-u_j} \subseteq H_{S-u_i}$. ◁

Claim 11 implies the following Corollary.

▶ **Corollary 12.** *For every $i, j \in [p]$, we have $u_i, u_j \notin H_{S-u_i-u_j}$.*

Proof. By Corollary 10, we have that $u_i \notin H_{S-u_i}$. Therefore, $u_i, u_j \notin H_{S-u_i} \cap H_{S-u_j}$. By Claim 11, $H_{S-u_i-u_j} \subseteq H_{S-u_i}$ and $H_{S-u_i-u_j} \subseteq H_{S-u_j}$. Therefore, $H_{S-u_i-u_j} \subseteq H_{S-u_i} \cap H_{S-u_j}$, and thus, $u_i, u_j \notin H_{S-u_i-u_j}$. ◀

The next claim will help in controlling the cost of the k-partition that we will obtain by applying Theorem 5.

▷ Claim 13. For every $i, j \in [p]$, we have $d(H_{S-u_i}) = d(U) = d(H_{S-u_i-u_j})$.

Proof. Let $a, b \in [p]$. We will show that $d(H_{S-u_a}) = d(U) = d(H_{S-u_a-u_b})$. Since $(U, \overline{U})$ is a $(S-u_a, \overline{U})$-terminal cut, we have that $d(H_{S-u_a}) \leq d(U)$. Since $(H_{S-u_a}, \overline{H_{S-u_a}})$ is a $(S-u_a-u_b, \overline{U})$-terminal cut, we have that $d(H_{S-u_a-u_b}) \leq d(H_{S-u_a}) \leq d(U)$. Thus, in order to prove the claim, it suffices to show that $d(H_{S-u_a-u_b}) \geq d(U)$.

Suppose for contradiction that $d(H_{S-u_a-u_b}) < d(U)$. Let $\ell \in [p] \setminus \{a, b\}$ be an arbitrary element (which exists since we have assumed that $p \geq 2k$ and $k \geq 2$). Let $R := \{u_\ell\}$, $S' := S - u_a - u_\ell$, and $A_i := \overline{H_{S-u_a-u_i}}$ for every $i \in [p] \setminus \{a, \ell\}$. We note that $|S'| = p - 2 \geq 2k - 2$. By definition, $(\overline{A_i}, A_i)$ is a minimum $(S-u_a-u_i, \overline{U})$-terminal cut for every $i \in [p] \setminus \{a, \ell\}$. Moreover, by Corollary 12, we have that $u_i \in A_i \setminus (\cup_{j \in [p] \setminus \{a,i,\ell\}} A_j)$ for every $i \in [p] \setminus \{a, \ell\}$. Hence, the sets U, R, and S', and the cuts $(\overline{A_i}, A_i)$ for $i \in [p] \setminus \{a, \ell\}$ satisfy the conditions of Theorem 5. Therefore, by the first conclusion of Theorem 5, there exists a k-partition $\mathcal{P}'$ with

$$|\delta(\mathcal{P}')| \leq \frac{1}{2} \min\{d(H_{S-u_a-u_i}) + d(H_{S-u_a-u_j}) \colon i, j \in [p] \setminus \{a, \ell\}\}.$$

By assumption, $d(H_{S-u_a-u_b}) < d(U)$ and $b \in [p] \setminus \{a, \ell\}$, so $\min\{d(H_{S-u_a-u_i}) \colon i \in [p] \setminus \{a, \ell\}\} < d(U)$. Since $(U, \overline{U})$ is a $(S-u_a-u_i, \overline{U})$-terminal cut, we have that $d(H_{S-u_a-u_i}) \leq d(U)$ for every $i \in [p] \setminus \{a, \ell\}$. Therefore,

$$\frac{1}{2} \min\{d(H_{S-u_a-u_i}) + d(H_{S-u_a-u_j}) \colon i, j \in [p] \setminus \{a, \ell\}\} < d(U) = OPT_{k\text{-cut}}.$$

Thus, we have that $|\delta(\mathcal{P}')| < OPT_{k\text{-cut}}$, which is a contradiction. ◁

The next two claims will help in arguing the existence of a hyperedge satisfying the conditions of the second conclusion of Theorem 5. In particular, we will need Claim 15. The following claim will help in proving Claim 15.

▷ Claim 14. For every $i, j \in [p]$, we have

$$d(H_{S-u_i} \cap H_{S-u_j}) = d(U) = d(H_{S-u_i} \cup H_{S-u_j}).$$

Proof. Since $(H_{S-u_i} \cap H_{S-u_j}, \overline{H_{S-u_i} \cap H_{S-u_j}})$ is a $(S - u_i - u_j, \overline{U})$-terminal cut, we have that $d(H_{S-u_i} \cap H_{S-u_j}) \geq d(H_{S-u_i-u_j})$. By Claim 13, we have that $d(H_{S-u_i-u_j}) = d(U) = d(H_{S-u_i})$. Therefore,

$$d(H_{S-u_i} \cap H_{S-u_j}) \geq d(H_{S-u_i}). \tag{4}$$

Since $(H_{S-u_i} \cup H_{S-u_j}, \overline{H_{S-u_i} \cup H_{S-u_j}})$ is a $(S - u_j, \overline{U})$-terminal cut, we have that

$$d(H_{S-u_i} \cup H_{S-u_j}) \geq d(H_{S-u_j}). \tag{5}$$

By submodularity of the hypergraph cut function and inequalities (4) and (5), we have that

$$d(H_{S-u_i}) + d(H_{S-u_j}) \geq d(H_{S-u_i} \cap H_{S-u_j}) + d(H_{S-u_i} \cup H_{S-u_j}) \geq d(H_{S-u_i}) + d(H_{S-u_j}).$$

Therefore, inequalities (4) and (5) are equations. Thus, by Claim 13, we have that

$$d(H_{S-u_i} \cap H_{S-u_j}) = d(H_{S-u_i}) = d(U),$$

and

$$d(H_{S-u_i} \cup H_{S-u_j}) = d(H_{S-u_j}) = d(U). \qquad \triangleleft$$

▷ Claim 15. For every $i, j, \ell \in [p]$ with $i \neq j$, we have $H_{S-u_\ell} \subseteq H_{S-u_i} \cup H_{S-u_j}$.

Proof. If $\ell = i$ or $\ell = j$ the claim is immediate. Thus, we assume that $\ell \notin \{i, j\}$. Let $Q := H_{S-u_\ell} \setminus (H_{S-u_i} \cup H_{S-u_j})$. We need to show that $Q = \emptyset$. We will show that $(H_{S-u_\ell} \setminus Q, \overline{H_{S-u_\ell} \setminus Q})$ is a minimum $(S - u_\ell, \overline{U})$-terminal cut. Consequently, Q must be empty (otherwise, $H_{S-u_\ell} \setminus Q \subsetneq H_{S-u_\ell}$ and hence, $(H_{S-u_\ell} \setminus Q, \overline{H_{S-u_\ell} \setminus Q})$ contradicts source minimality of the minimum $(S - u_\ell, \overline{U})$-terminal cut $(H_{S-u_\ell}, \overline{H_{S-u_\ell}})$).

We now show that $(H_{S-u_\ell} \setminus Q, \overline{H_{S-u_\ell} \setminus Q})$ is a minimum $(S - u_\ell, \overline{U})$-terminal cut. Since $H_{S-u_\ell} \setminus Q = H_{S-u_\ell} \cap (H_{S-u_i} \cup H_{S-u_j})$, we have that $S - u_i - u_j - u_\ell \subseteq H_{S-u_\ell} \setminus Q$. We also know that u_i and u_j are contained in both H_{S-u_ℓ} and $H_{S-u_i} \cup H_{S-u_j}$. Therefore, $S - u_\ell \subseteq H_{S-u_\ell} \setminus Q$. Thus, $(H_{S-u_\ell} \setminus Q, \overline{H_{S-u_\ell} \setminus Q})$ is a $(S - u_\ell, \overline{U})$-terminal cut. Therefore,

$$d(H_{S-u_\ell} \cap (H_{S-u_i} \cup H_{S-u_j})) = d(H_{S-u_\ell} \setminus Q) \geq d(H_{S-u_\ell}). \tag{6}$$

We also have that $(H_{S-u_\ell} \cup (H_{S-u_i} \cup H_{S-u_j}), \overline{H_{S-u_\ell} \cup (H_{S-u_i} \cup H_{S-u_j})})$ is a $(S - u_i, \overline{U})$-terminal cut. Therefore, $d(H_{S-u_\ell} \cup (H_{S-u_i} \cup H_{S-u_j})) \geq d(H_{S-u_i})$. By Claims 13 and 14, we have that $d(H_{S-u_i}) = d(V_1) = d(H_{S-u_i} \cup H_{S-u_j})$. Therefore,

$$d(H_{S-u_\ell} \cup (H_{S-u_i} \cup H_{S-u_j})) \geq d(H_{S-u_i} \cup H_{S-u_j}). \tag{7}$$

By submodularity of the hypergraph cut function and inequalities (6) and (7), we have that

$$\begin{aligned} d(H_{S-u_\ell}) + d(H_{S-u_i} \cup H_{S-u_j}) &\geq d(H_{S-u_\ell} \cap (H_{S-u_i} \cup H_{S-u_j})) + d(H_{S-u_\ell} \cup (H_{S-u_i} \cup H_{S-u_j})) \\ &\geq d(H_{S-u_\ell}) + d(H_{S-u_i} \cup H_{S-u_j}). \end{aligned}$$

Therefore, inequalities (6) and (7) are equations, so $(H_{S-u_\ell} \setminus Q, \overline{H_{S-u_\ell} \setminus Q})$ is a minimum $(S - u_\ell, \overline{U})$-terminal cut. ◁

Let $R := \{u_p\}$, $S' := S - u_p$, and $(\overline{A_i}, A_i) := (H_{S-u_i}, \overline{H_{S-u_i}})$ for every $i \in [p-1]$. By definition, $(\overline{A_i}, A_i)$ is a minimum $(S-u_i, \overline{U})$-terminal cut for every $i \in [p-1]$. Moreover, by Corollary 10, we have that $u_i \in A_i \setminus (\cup_{j\in[p-1]\setminus\{i\}} A_j)$. Hence, the sets U, R, and S', and the cuts $(\overline{A_i}, A_i)$ for $i \in [p-1]$ satisfy the conditions of Theorem 5. We will use the second conclusion of Theorem 5. We now show that there exists a hyperedge satisfying the conditions mentioned in the second conclusion of Theorem 5. We will use Claim 16 below to prove this. Let $W := \cup_{1\le i<j\le p-1}(A_i \cap A_j)$ and $Z := \cap_{i\in[p-1]}\overline{A_i}$ as in the statement of Theorem 5.

▷ Claim 16. There exists a hyperedge $e \in E$ such that $e \cap W \neq \emptyset$, $e \cap Z \neq \emptyset$, and $e \subseteq W \cup Z$.

Proof. We note that $S \subseteq (S-u_i) \cup (S-u_j) \subseteq H_{S-u_i} \cup H_{S-u_j}$ for every distinct $i, j \in [p-1]$. Therefore, $S \cap (A_i \cap A_j) = \emptyset$ for every distinct $i, j \in [p-1]$, and thus $S \cap W = \emptyset$. Since S is a transversal of the collection $\mathcal{C}$, it follows that the set W is not in the collection $\mathcal{C}$.

By definition, $\overline{U} \subseteq A_i$ for every $i \in [p-1]$, and thus $\overline{U} \subseteq W$. Since $W \notin \mathcal{C}$, either $d(W) > d(U)$ or $\delta(W) = \delta(U)$. By Claim 9, we have that $\overline{H_{S-u_p}} \in \mathcal{C}$, and thus, $d(\overline{H_{S-u_p}}) \le d(U)$ and $\delta(\overline{H_{S-u_p}}) \neq \delta(U)$. Consequently, $d(W) \ge d(\overline{H_{S-u_p}})$, and $\delta(W) \neq \delta(\overline{H_{S-u_p}})$, and thus, $\delta(W) \setminus \delta(\overline{H_{S-u_p}}) \neq \emptyset$. Let $e \in \delta(W) \setminus \delta(\overline{H_{S-u_p}})$. We will show that this choice of e achieves the desired properties.

For each $i \in [p]$, let $Y_i := \overline{H_{S-u_i}} \setminus W$. By Claim 15, for every $i, j, \ell \in [p]$ with $i \neq j$ we have that $H_{S-u_\ell} \subseteq H_{S-u_i} \cup H_{S-u_j}$. Therefore $\overline{H_{S-u_i}} \cap \overline{H_{S-u_j}} \subseteq \overline{H_{S-u_\ell}}$ for every such $i, j, \ell \in [p]$, and hence $W \subseteq \overline{H_{S-u_\ell}}$ for every $\ell \in [p]$. Thus, $W \subseteq \overline{H_{S-u_p}}$. Since $e \in \delta(W) \setminus \delta(\overline{H_{S-u_p}})$, we have that $e \subseteq W \cup Y_p$, $e \cap W \neq \emptyset$ and $e \cap Y_p \neq \emptyset$. Therefore, in order to show that e has the three desired properties as in the claim, it suffices to show that $Y_p \subseteq Z$. We prove this next.

By definition, $Y_p \cap W = \emptyset$. By Claim 15, for every $i \in [p-1]$, we have that $\overline{H_{S-u_p}} \cap \overline{H_{S-u_i}} \subseteq \overline{H_{S-u_1}}$ and $\overline{H_{S-u_p}} \cap \overline{H_{S-u_i}} \subseteq \overline{H_{S-u_2}}$, so $\overline{H_{S-u_p}} \cap \overline{H_{S-u_i}} \subseteq \overline{H_{S-u_1}} \cap \overline{H_{S-u_2}} \subseteq W$. Thus, for every $i \in [p-1]$, $Y_p \cap Y_i \subseteq \overline{H_{S-u_p}} \cap \overline{H_{S-u_i}} \subseteq W$, so since $Y_p \cap W = \emptyset$, we have that $Y_p \cap Y_i = \emptyset$ for every $i \in [p-1]$. Therefore,

$$Y_p \subseteq W \cup \overline{\left(\bigcup_{i=1}^{p-1} Y_i\right)} = \overline{\bigcup_{i=1}^{p-1} \overline{H_{S-u_i}}} = \bigcap_{i=1}^{p-1} H_{S-u_i} = Z. \qquad \triangleleft$$

By Claim 16, there is a hyperedge e satisfying the conditions of the second conclusion of Theorem 5. Therefore, by Theorem 5, there exists a k-partition $\mathcal{P}'$ with

$$\begin{aligned}
|\delta(\mathcal{P}')| &< \frac{1}{2}\min\{d(A_i) + d(A_j) : i, j \in [p-1], i \neq j\} \\
&= d(U) && \text{(By Claim 13)} \\
&= OPT_{k\text{-cut}}. && \text{(By assumption of the theorem)}
\end{aligned}$$

Thus, we have obtained a k-partition $\mathcal{P}'$ with $|\delta(\mathcal{P}')| < OPT_{k\text{-cut}}$, which is a contradiction. ◀

3 Proof of Theorem 4

We prove Theorem 4 in this section. Applying Theorem 6 to $(\overline{U}, U)$ yields the following corollary.

▶ **Corollary 17.** *Let $G = (V, E)$ be a hypergraph and let $OPT_{k\text{-cut}}$ be the optimum value of* Hypergraph-k-Cut *in G for some integer $k \ge 2$. Suppose $(U, \overline{U})$ is a 2-partition of V with $d(U) = OPT_{k\text{-cut}}$. Then, there exists a set $T \subseteq \overline{U}$ with $|T| \le 2k-1$ such that every minimum (U, T)-terminal cut $(A, \overline{A})$ satisfies $\delta(A) = \delta(U)$.*

We now restate Theorem 4 and prove it using Theorem 6 and Corollary 17.

▶ **Theorem 4.** *Let $G = (V, E)$ be a hypergraph and let $OPT_{k\text{-}cut}$ be the optimum value of* Hypergraph-k-Cut *in G for some integer $k \geq 2$. Suppose $(U, \overline{U})$ is a 2-partition of V with $d(U) = OPT_{k\text{-}cut}$. Then, there exist sets $S \subseteq U$, $T \subseteq \overline{U}$ with $|S| \leq 2k - 1$ and $|T| \leq 2k - 1$ such that every minimum (S, T)-terminal cut $(A, \overline{A})$ satisfies $\delta(A) = \delta(U)$.*

Proof. By Theorem 6, there exists a subset $S \subseteq U$ with $|S| \leq 2k - 1$ such that every minimum $(S, \overline{U})$-terminal cut $(A, \overline{A})$ has $\delta(A) = \delta(U)$. By Corollary 17, there exists a subset $T \subseteq \overline{U}$ with $|T| \leq 2k - 1$ such that every minimum (U, T)-terminal cut $(A, \overline{A})$ has $\delta(A) = \delta(U)$. We will show that every minimum (S, T)-terminal cut $(A, \overline{A})$ has $\delta(A) = \delta(U)$. We will need the following claim.

▷ Claim 18. Let $(Y, \overline{Y})$ be the source minimal minimum (S, T)-terminal cut. Then $\delta(Y) = \delta(U)$.

Proof. Since $(U, \overline{U})$ is a (S, T)-terminal cut, and $(Y, \overline{Y})$ is a minimum (S, T)-terminal cut, we have that

$$d(U) \geq d(Y).$$

Since $(U \cap Y, \overline{U \cap Y})$ is a $(S, \overline{U})$-terminal cut, we have that

$$d(U \cap Y) \geq d(U).$$

Since $(U \cup Y, \overline{U \cup Y})$ is a (U, T)-terminal cut, we have that

$$d(U \cup Y) \geq d(U).$$

Thus, by the submodularity of the hypergraph cut function we have that

$$2d(U) \geq d(U) + d(Y) \geq d(U \cap Y) + d(U \cup Y) \geq 2d(U).$$

Therefore, we have that $d(U \cap Y) = d(U)$, so $(U \cap Y, \overline{U \cap Y})$ is a minimum (S, T)-terminal cut. Since $(Y, \overline{Y})$ is the source minimal (S, T)-terminal cut, we have that $U \cap Y = Y$, and hence $Y \subseteq U$. Therefore, $(Y, \overline{Y})$ is a minimum $(S, \overline{U})$-terminal cut. By the choice of S, we have that $\delta(Y) = \delta(U)$. ◁

Applying Claim 18 to both sides of the partition $(U, \overline{U})$, we have that the source minimal minimum (S, T)-terminal cut $(Y, \overline{Y})$ has $\delta(Y) = \delta(U)$, and the source minimal minimum (T, S)-terminal cut $(Z, \overline{Z})$ has $\delta(Z) = \delta(U)$. Therefore, for every $e \in \delta(U)$, we have that $e \cap Y \neq \emptyset$ and $e \cap Z \neq \emptyset$.

Let $(A, \overline{A})$ be a minimum (S, T)-terminal cut. Since $(Y, \overline{Y})$ is the source minimal minimum (S, T)-terminal cut, we have that $Y \subseteq A$. Since $(Z, \overline{Z})$ is the source minimal minimum (T, S)-terminal cut, we have that $Z \subseteq \overline{A}$. Since every $e \in \delta(U)$ intersects both Y and Z, it follows that every $e \in \delta(U)$ intersects both A and $\overline{A}$, and hence, $\delta(U) \subseteq \delta(A)$. Since $(A, \overline{A})$ is a minimum (S, T)-terminal cut, $d(A) \leq d(U)$, and thus we have that $\delta(A) = \delta(U)$. ◀

4 Conclusion

We showed the first polynomial bound on the number of minmax-k-cut-sets in hypergraphs for every fixed k and gave a polynomial-time algorithm to enumerate all minmax-k-cut-sets as well as all min-k-cut-sets in hypergraphs for every fixed k. Our main contribution is a structural theorem that is the backbone of the correctness analysis of our enumeration algorithms. In order to enumerate MINMAX-k-CUT-SETs in hypergraphs, we introduced the notion of k-cut-set representatives and enumerated k-cut-set representatives of all optimum k-partitions for MINMAX-HYPERGRAPH-k-PARTITION. Our technique builds on known structural results for HYPERGRAPH-k-CUT and MINMAX-HYPERGRAPH-k-PARTITION [8, 12, 13].

The technique underlying our enumeration algorithms is not necessarily novel – we simply rely on minimum (s, t)-terminal cuts. Using fixed-terminal cuts to address global partitioning problems is not a novel technique by itself – it is common knowledge that minimum (s, t)-terminal cuts can be used to solve global minimum cut. However, there are several problems where naive use of this technique fails to lead to efficient algorithms: e.g., multiway cut does not help in solving GRAPH-k-CUT since multiway cut is NP-hard. Adapting this technique for specific partitioning problems requires careful identification of structural properties. In fact, beautiful structural properties have been shown for a rich variety of partitioning problems in combinatorial optimization in order to exploit this technique: for example, it was used (1) to design the first efficient algorithm for GRAPH-k-CUT [23], (2) to solve certain constrained submodular minimization problems [22, 35], and (3) more recently, to design fast algorithms for global minimum cut in graphs and for Gomory-Hu tree in unweighted graphs [1, 34]. Our use of this technique also relies on identifying and proving a suitable structural property, namely Theorem 4. The advantage of our structural property is that it simultaneously enables enumeration of MIN-k-CUT-SETs as well as MINMAX-k-CUT-SETs in hypergraphs which was not possible via structural theorems that were developed before. Furthermore, it helps in showing the first polynomial bound on the number of MINMAX-k-CUT-SETs in hypergraphs for every fixed k.

We also emphasize a limitation of our technique. Although it helps in solving ENUM-HYPERGRAPH-k-CUT and ENUM-MINMAX-HYPERGRAPH-k-PARTITION, it does not help in solving a seemingly related hypergraph k-partitioning problem – namely, given a hypergraph $G = (V, E)$ and a fixed integer k, find a k-partition $(V_1, \ldots, V_k)$ of the vertex set that minimizes $\sum_{i=1}^{k} |\delta(V_i)|$. Natural variants of our structural theorem fail to hold for this objective. Resolving the complexity of this variant of the hypergraph k-partitioning problem for $k \geq 5$ remains open.

We mention an open question concerning HYPERGRAPH-k-CUT and the enumeration of MIN-k-CUT-SETs in hypergraphs for fixed k. We recall the status in graphs: the number of minimum k-partitions in a connected graph was known to be $O(n^{2k-2})$ via Karger-Stein's algorithm [30] and $\Omega(n^k)$ via the cycle example, where n is the number of vertices; recent works have improved on the upper bound to match the lower bound for fixed k – this improvement in upper bound also led to the best possible $O(n^k)$-time algorithm for GRAPH-k-CUT for fixed k [25–27]. For hypergraphs, the number of MIN-k-CUT-SETs is known to be $O(n^{2k-2})$ and $\Omega(n^k)$. Can we improve the upper/lower bound? Is it possible to design an algorithm for HYPERGRAPH-k-CUT that runs in time $O(n^k p)$?

References

1 A. Abboud, R. Krauthgamer, and O. Trabelsi. Subcubic Algorithms for Gomory–Hu Tree in Unweighted Graphs. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, STOC, pages 1725–1737, 2021.

2 K. Ahn and S. Guha. Graph sparsification in the semi-streaming model. In *Proceeings of the 36th International Colloquium on Automata, Languages and Programming: Part II*, ICALP, pages 328–338, 2009.

3 K. Ahn, S. Guha, and A. McGregor. Analyzing graph structure via linear measurements. In *Proceedings of the 23rd Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA, pages 459–467, 2012.

4 K. Ahn, S. Guha, and A. McGregor. Graph sketches: Sparsification, spanners, and subgraphs. In *Proceedings of the 31st Symposium on Principles of Database Systems*, PODS, pages 5–14, 2012.

5 D. Applegate, R. Bixby, V. Chvátal, and W. Cook. *The Traveling Salesman Problem: A Computational Study*. Princeton University Press, 2006.

6 N. Bansal, O. Svensson, and L. Trevisan. New notions and constructions of sparsification for graphs and hypergraphs. In *Proceedings of the 60th Annual IEEE Symposium on Foundations of Computer Science*, FOCS, pages 910–928, 2019.

7 C. Beideman, K. Chandrasekaran, and W. Wang. Counting and enumerating optimum cut sets for hypergraph k-partitioning problems for fixed k, 2022. `arXiv:2204.09178`.

8 C. Beideman, K. Chandrasekaran, and W. Wang. Deterministic enumeration of all minimum k-cut-sets in hypergraphs for fixed k. In *Proceedings of the 33rd Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA, 2022.

9 A. Benczur. A representation of cuts within 6/5 times the edge connectivity with applications. In *Proceedings of the 36th Annual IEEE Foundations of Computer Science*, FOCS, pages 92–102, 1995.

10 A. Benczur. *Cut structures and randomized algorithms in edge-connectivity problems*. PhD thesis, MIT, 1997.

11 A. Benczur and M. Goemans. Deformable polygon representation and near-mincuts. *Building Bridges: Between Mathematics and Computer Science, M. Groetschel and G.O.H. Katona, Eds., Bolyai Society Mathematical Studies*, 19:103–135, 2008.

12 K. Chandrasekaran and C. Chekuri. Hypergraph k-cut for fixed k in deterministic polynomial time. In *Proceedings of the 61st Annual Symposium on Foundations of Computer Science*, FOCS, pages 810–821, 2020.

13 K. Chandrasekaran and C. Chekuri. Min-max partitioning of hypergraphs and symmetric submodular functions. In *Proceedings of the 32nd annual ACM-SIAM Symposium on Discrete Algorithms*, SODA, pages 1026–1038, 2021.

14 K. Chandrasekaran and W. Wang. Fixed Parameter Approximation Scheme for Min-max k-cut. In *Integer Programming and Combinatorial Optimization*, IPCO, pages 354–367, 2021.

15 K. Chandrasekaran, C. Xu, and X. Yu. Hypergraph k-cut in randomized polynomial time. *Mathematical Programming (Preliminary version in SODA 2018)*, 186:85–113, 2019.

16 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.

17 C. Chekuri and C. Xu. Minimum cuts and sparsification in hypergraphs. *SIAM Journal on Computing*, 47(6):2118–2156, 2018.

18 Y. Chen, S. Khanna, and A. Nagda. Near-linear size hypergraph cut sparsifiers. In *Proceedings of the 61st Annual IEEE Symposium on Foundations of Computer Science*, FOCS, pages 61–72, 2020.

19 W. Cook, W. Cunningham, W. Pulleyblank, and A. Schrijver. *Combinatorial Optimization*. Wiley, 1997.

20 K. Fox, D. Panigrahi, and F. Zhang. Minimum cut and minimum k-cut in hypergraphs via branching contractions. In *Proceedings of the 30th Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA, pages 881–896, 2019.

21 M. Ghaffari, D. Karger, and D. Panigrahi. Random contractions and sampling for hypergraph and hedge connectivity. In *Proceedings of the 28th Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA, pages 1101–1114, 2017.

22 M. X. Goemans and V. S. Ramakrishnan. Minimizing submodular functions over families of sets. *Combinatorica*, 15:499–513, 1995.

23 O. Goldschmidt and D. Hochbaum. A Polynomial Algorithm for the k-cut Problem for Fixed k. *Mathematics of Operations Research (Preliminary version in FOCS 1988)*, 19(1):24–37, February 1994.

24 F. Guinez and M. Queyranne. The size-constrained submodular k-partition problem. Unpublished manuscript. Available at `https://docs.google.com/viewer?a=v&pid=sites&srcid=ZGVmYXVsdGRvbWFpbnxmbGF2aW9ndWluZXpob21lcGFnZXxneDo0NDVlMThkMDg4ZWRlOGI1`. See also `https://smartech.gatech.edu/bitstream/handle/1853/43309/Queyranne.pdf`, 2012.

25 A. Gupta, D. Harris, E. Lee, and J. Li. Optimal Bounds for the k-cut Problem. *arXiv preprint*, 2020. `arXiv:2005.08301`.

26 A. Gupta, E. Lee, and J. Li. The number of minimum k-cuts: improving the Karger-Stein bound. In *Proceedings of the 51st ACM Symposium on Theory of Computing*, STOC, pages 229–240, 2019.

27 A. Gupta, E. Lee, and J. Li. The Karger-Stein algorithm is optimal for k-cut. In *Proceedings of the 52nd Annual ACM Symposium on Theory of Computing*, STOC, pages 473–484, 2020.

28 M. Kapralov, R. Krauthgamer, J. Tardos, and Y. Yoshida. Towards tight bounds for spectral sparsification of hypergraphs. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, STOC, pages 598–611, 2021.

29 D. Karger. Minimum cuts in near-linear time. *Journal of the ACM*, 47(1):46–76, 2000.

30 D. Karger and C. Stein. A new approach to the minimum cut problem. *Journal of the ACM*, 43(4):601–640, July 1996.

31 A. Karlin, N. Klein, and S-O. Gharan. A (slightly) improved approximation algorithm for metric tsp. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, STOC, pages 32–45, 2021.

32 D. Kogan and R. Krauthgamer. Sketching cuts in graphs and hypergraphs. In *Proceedings of the 2015 Conference on Innovations in Theoretical Computer Science*, ITCS '15, pages 367–376, 2015.

33 E. Lawler. Cutsets and Partitions of Hypergraphs. *Networks*, 3:275–285, 1973.

34 J. Li and D. Panigrahi. Deterministic Min-cut in Poly-logarithmic Max-flows. In *Proceedings of the 61st Annual Symposium on Foundations of Computer Science*, FOCS, pages 85–92, 2020.

35 M. Nägele, B. Sudakov, and R. Zenklusen. Submodular minimization under congruency constraints. *Combinatorica*, 39:1351–1386, 2019.

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

37 M. Queyranne. On optimum size-constrained set partitions, 1999. Talk at Aussiois workshop on Combinatorial Optimization.

38 T. Soma and Y. Yoshida. Spectral sparsification of hypergraphs. In *Proceedings of the 30th Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA, pages 2570–2581, 2019.

39 M. Thorup. Minimum k-way Cuts via Deterministic Greedy Tree Packing. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing*, STOC, pages 159–166, 2008.

40 R. Zenklusen. A 1.5-Approximation for Path TSP. In *Proceedings of the 30th Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA, pages 1539–1549, 2019.

41 L. Zhao. *Approximation algorithms for partition and design problems in networks*. PhD thesis, Graduate School of Informatics, Kyoto University, Japan, 2002.

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