Hypergraph k-cut for fixed k in deterministic polynomial time

Karthekeyan Chandrasekaran
University of Illinois, Urbana-Champaign
Urbana, USA
karthe@illinois.edu

Chandra Chekuri
University of Illinois, Urbana-Champaign
Urbana, USA
chekuri@illinois.edu

***Abstract*—We consider the HYPERGRAPH-k-CUT problem. The input consists of a hypergraph $G = (V, E)$ with non-negative hyperedge-costs $c : E \to \mathbb{R}_+$ and a positive integer k. The objective is to find a least-cost subset $F \subseteq E$ such that the number of connected components in $G - F$ is at least k. An alternative formulation of the objective is to find a partition of V into k non-empty sets $V_1, V_2, \ldots, V_k$ so as to minimize the cost of the hyperedges that cross the partition. GRAPH-k-CUT, the special case of HYPERGRAPH-k-CUT obtained by restricting to graph inputs, has received considerable attention. Several different approaches lead to a polynomial-time algorithm for GRAPH-k-CUT when k is fixed, starting with the work of Goldschmidt and Hochbaum (1988) [1], [2]. In contrast, it is only recently that a randomized polynomial time algorithm for HYPERGRAPH-k-CUT was developed [3] via a subtle generalization of Karger's random contraction approach for graphs. In this work, we develop the first deterministic polynomial time algorithm for HYPERGRAPH-k-CUT for all fixed k. We describe two algorithms both of which are based on a divide and conquer approach. The first algorithm is simpler and runs in $n^{O(k^2)}$ time while the second one runs in $n^{O(k)}$ time. Our proof relies on new structural results that allow for efficient recovery of the parts of an optimum k-partition by solving minimum (S, T)-terminal cuts. Our techniques give new insights even for GRAPH-k-CUT.**

***Keywords*-hypergraphs; partition; algorithms;**

I. INTRODUCTION

A hypergraph $G = (V, E)$ consists of a finite set V of vertices and a finite set E of hyperedges where each $e \in E$ is a subset of V. In this work, we consider the HYPERGRAPH-k-CUT problem, in particular when k is a fixed constant. The input to this problem consists of a hypergraph $G = (V, E)$ with non-negative hyperedge-costs $c : E \to \mathbb{R}_+$ and a positive integer k. The objective is to find a minimum-cost subset of hyperedges whose removal results in at least k connected components. An equivalent partitioning formulation turns out to be quite important. In this formulation, the objective is to find a partition of V into k non-empty sets $V_1, V_2, \ldots, V_k$ so as to minimize the cost of the hyperedges that cross the partition. A hyperedge $e \in E$ crosses a partition $(V_1, V_2, \ldots, V_k)$ if it has vertices in more than two parts, that is, there exist distinct $i, j \in [k]$ such that $e \cap V_i \neq \emptyset$ and $e \cap V_j \neq \emptyset$.

Cut and partitioning problems in graphs, hypergraphs, and related structures including submodular functions are extensively studied in algorithms and combinatorial optimization literature for their theoretical importance and numerous applications. HYPERGRAPH-k-CUT is a problem that is of inherent interest not only for its applications and simplicity but also because of its close connections to a special case, namely in graphs, and to a generalization, namely in submodular functions. For this reason the complexity of HYPERGRAPH-k-CUT has been an intriguing open problem for several years with some important recent progress. First we describe these closely related problems and some prior work on them.

GRAPH-k-CUT: This is a special case of HYPERGRAPH-k-CUT where the input is a graph instead of a hypergraph. When $k = 2$, GRAPH-k-CUT is the global minimum cut problem (GRAPH-MINCUT) which is a fundamental and well-known problem. It is easy to see that GRAPH-MINCUT can be solved in polynomial time via reduction to min s-t cuts but there is more structure in GRAPH-MINCUT, and this can be exploited to obtain faster deterministic and randomized algorithms [4]–[7]. The complexity of GRAPH-k-CUT for $k \geq 3$ has also been extensively investigated with substantial recent work. Goldschmidt and Hochbaum (1988) [1], [2] showed that GRAPH-k-CUT is NP-Hard when k is part of the input and that it is polynomial-time solvable when k is any fixed constant (this is not obvious even for $k = 3$). They used a divide-and-conquer approach for GRAPH-k-CUT which resulted in an algorithm with a running time of $n^{O(k^2)}$. We will describe the technical aspects of this approach in more detail later. This approach has been refined over several papers culminating in an algorithm of Kamidoi, Yoshida, and Nagamochi [8] that ran in $n^{(4+o(1))k}$ time. Two very different approaches also give polynomial-time algorithms for fixed k. The first approach is the random contraction approach of Karger that, via the improvement in Karger and Stein's work, led to a Monte Carlo randomized algorithm with a running time of $\tilde{O}(n^{2k-2})$; very recently Gupta, Lee, and Li [9] showed that the Karger-Stein algorithm in fact runs in $\hat{O}(n^k)$ time (where $\hat{O}(\cdot)$ hides $2^{O(\ln \ln n)^2}$); $n^{(1-o(1))k}$ appears to be lower bound on the run-time via a reduction from the problem of finding a maximum-weight clique of size k (see [10]). The second approach is the tree packing approach which was introduced

2575-8454/20/$31.00 ©2020 IEEE
DOI 10.1109/FOCS46700.2020.00080

by Karger for GRAPH-MINCUT. Thorup [11] showed that tree packings can also be used to obtain a polynomial-time algorithm for GRAPH-k-CUT. His algorithm runs in deterministic $n^{2k+O(1)}$ time; his approach was clarified in [12] via an LP relaxation and this also resulted in a slight improvement in the run-time and currently yields the fastest deterministic algorithm. We defer discussion of approximation algorithms for GRAPH-k-CUT when k is part of the input to the related work section.

Submodular Partition Problems: Graph and hypergraph cut functions are submodular and one can view GRAPH-k-CUT and HYPERGRAPH-k-CUT as special cases of a more general problem called SUBMODULAR-k-PARTITION (abbreviated to SUBMOD-k-PART) that we define now. We recall that a real-valued set function $f : 2^V \to \mathbb{R}$ is submodular iff $f(A \cap B) + f(A \cup B) \le f(A) + f(B)$ for all $A, B \subseteq V$. Zhao, Nagamochi, and Ibaraki [13] defined SUBMOD-k-PART as follows: given f specified via a value oracle and a positive integer k, the goal is to partition V into non-empty sets $V_1, V_2, \ldots, V_k$ so as to minimize $\sum_{i=1}^{k} f(V_i)$. A special case of SUBMOD-k-PART is SYM-SUBMOD-k-PART when f is symmetric (that is $f(A) = f(V \setminus A)$ for all $A \subseteq V$). It is not hard to see that GRAPH-k-CUT is a special case of SYM-SUBMOD-k-PART. However, HYPERGRAPH-k-CUT is not a special case of SYM-SUBMOD-k-PART even though the hypergraph cut function is itself symmetric;[1] as observed in [14], one can reduce HYPERGRAPH-k-CUT to SUBMOD-k-PART. SUBMOD-k-PART and SYM-SUBMOD-k-PART are very general problems. For $k = 2$, they can be solved in polynomial-time via submodular function minimization. It is a very interesting open problem to decide whether they admit polynomial-time algorithms for all fixed k. Okumoto, Fukunaga, and Nagamochi [14] showed that SUBMOD-k-PART is polynomial-time solvable for $k = 3$. They generalized the work of Xiao [15] who showed that HYPERGRAPH-k-CUT is polynomial-time solvable for $k = 3$. Queyranne claimed, in 1999, a polynomial-time algorithm for SYM-SUBMOD-k-PART when k is fixed [16], however the claim was retracted subsequently. This is reported in [17] where it is also shown that SYM-SUBMOD-k-PART has a polynomial-time algorithm for $k \le 4$.

Multiterminal variants: We also mention that GRAPH-k-CUT, HYPERGRAPH-k-CUT, and SUBMOD-k-PART have natural variants involving separating specified terminal vertices $s_1, s_2, \ldots, s_k$. These versions are NP-hard for $k \ge 3$. We discuss approximation algorithms for these problems in the related work section.

HYPERGRAPH-k-CUT and main result: The complexity of HYPERGRAPH-k-CUT for fixed k has been open since the work of Goldschmidt and Hochbaum for graphs (1988) [1]. For $k = 2$, this is the HYPERGRAPH-MINCUT problem and can be solved via reduction to min s-t cuts in directed graphs [18] or via other approaches that take advantage of the submodularity structure of the hypergraph cut function (see [19] and references therein). For $k \ge 3$ and bounded rank hypergraphs, Fukunaga [20] generalized Thorup's tree packing approach [11] to solve HYPERGRAPH-k-CUT for fixed k — the run-time depends exponentially in the rank (rank is the maximum cardinality of a hyperedge in the input hypergraph). It was also observed that Karger's random contraction approach for graphs easily extends to give a randomized algorithm for bounded rank hypergraphs. As we noted earlier, Xiao [15] obtained a polynomial-time algorithm for HYPERGRAPH-k-CUT when $k = 3$. In fairly recent work, Chandrasekaran, Xu, and Yu [3] obtained the first randomized polynomial-time algorithm for HYPERGRAPH-k-CUT for any fixed k; their Monte Carlo algorithm runs in $\tilde{O}(pn^{2k-1})$ time where $p = \sum_{e \in E} |e|$ is the representation size of the input hypergraph. Subsequently, Fox, Panigrahi, and Zhang [21] improved the randomized run-time to $\tilde{O}(mn^{2k-2})$, where m is the number of hyperedges in the input hypergraph. Both these randomized algorithms are based on random contraction of hyperedges and are inspired partly by earlier work in [22] for HYPERGRAPH-MINCUT.

The existence of a randomized algorithm for HYPERGRAPH-k-CUT raises the question of the existence of a deterministic algorithm. Random contraction based algorithms do not lend themselves naturally to derandomization. Perhaps, more pertinent is our interest in addressing the complexity of SUBMOD-k-PART. There is no natural random contraction approach for this more general problem. For GRAPH-k-CUT, two distinct approaches lead to deterministic algorithms and among these, the tree packing approach, like the random contraction approach, does not appear to apply to SUBMOD-k-PART. This leaves the divide and conquer approach initiated in the paper of Goldschmidt and Hochbaum [1], [2]. Is there a variant of this approach that works for HYPERGRAPH-k-CUT and SUBMOD-k-PART? We discovered certain structural properties of HYPERGRAPH-k-CUT (that do not hold for other submodular functions) to prove our main result stated below.

Theorem 1. *There is a deterministic polynomial-time algorithm for* HYPERGRAPH-k-CUT *for any fixed k.*

Our work raises the hope for a polynomial-time algorithm for SUBMOD-k-PART when k is fixed.

A. Technical overview and structural results

We focus on the unit-cost variant of the problem in the rest of this work for the sake of notational simplicity. Note that we allow multigraphs and hence this is without loss

[1] SYM-SUBMOD-k-PART when the input function f is the cut function of a hypergraph is known as HYPERGRAPH-k-PARTITION in the literature [13], [14]. We emphasize that the objective in HYPERGRAPH-k-PARTITION is different from the objective in HYPERGRAPH-k-CUT.

of generality. All our algorithms extend in a straightforward manner to arbitrary hyperedge costs. They rely only on minimum (s,t)-cut computations and hence, they are strongly polynomial.

A key algorithmic tool will be the use of terminal cuts. We need some notation. Let $G=(V,E)$ be a hypergraph. For a subset U of vertices, we will use $\overline{U}$ to denote $V\setminus U$, $\delta(U)$ to denote the set of hyperedges crossing U, and $d(U):=|\delta(U)|$ to denote the value of U. More generally, given a partition $(V_1,V_2,\ldots,V_h)$, we denote the number of hyperedges crossing the partition by $\mathrm{cost}(V_1,V_2,\ldots,V_h)$. Let S, T be disjoint 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 maximal* minimum (S,T)-terminal cuts and *source minimal* minimum (S,T)-terminal cuts. These cuts are unique and can be found in polynomial-time via standard maxflow algorithms. In fact, these definitions extend to general submodular functions. Given $f:2^V\rightarrow\mathbb{R}$ and disjoint sets $S,T\subseteq V$, we can define a minimum (S,T)-terminal cut for f as $\min_{U:S\subseteq U,T\subseteq\overline{U}}f(U)$. Uniqueness of source-maximal and source-minimal (S,T)-terminal cuts follow from submodularity and one can also find these in polynomial-time via submodular function minimization.

Our algorithm follows the divide-and-conquer approach that was first used by Goldschmidt and Hochbaum [1], [2] for GRAPH-k-CUT, and in a more general fashion by Kamidoi, Yoshida, and Nagamochi [8] to improve the running time for GRAPH-k-CUT. The goal in this approach is to identify one part of some fixed optimum k-partition $(V_1,V_2,\ldots,V_k)$, say V_1 without loss of generality, and then recursively find a $(k-1)$ partition of $V\setminus V_1$. How do we find such a part? Goldschmidt and Hochbaum proved a key structural lemma for GRAPH-k-CUT: Suppose $(V_1,V_2,\ldots,V_k)$ is an optimum k-partition such that V_1 is the part with the smallest cut value (i.e., $|\delta(V_1)|\le|\delta(V_i)|$ for all $i\in[k]$) and V_1 is maximal subject to this condition. Then, either $|V_1|\le k-2$ or there exist disjoint sets S,T such that $S\subseteq V_1,T\subseteq\overline{V_1}$ with $|S|\le k-1$ and $|T\cap V_j|=1$ for every $j\in\{2,\ldots,k\}$ so that the source maximal minimum (S,T)-terminal cut is $(V_1,\overline{V_1})$. One can guess/enumerate all small-sized (S,T)-pairs to find an $O(n^{2k-2})$-sized collection of sets containing V_1 and recursively find an optimum $(k-1)$-partition of $V\setminus U$ for each U in the collection. This leads to an $n^{O(k^2)}$-time algorithm for GRAPH-k-CUT.

Queyranne [16] claimed that a natural generalization of the preceding structural lemma holds in the more general setting of SYM-SUBMOD-k-PART. However, as reported in [17], the claimed proof was incorrect and it was only proved for $k=3,4$. More importantly, as also noted in [17], this structural lemma (even if true for arbitrary k) is not useful for SYM-SUBMOD-k-PART because one cannot recurse on $V\setminus V_1$; the function f restricted to $V\setminus V_1$ is no longer symmetric! The reader might now wonder how the approach works for GRAPH-k-CUT? Interestingly, GRAPH-k-CUT has the very nice property that the graph cut function restricted to $V\setminus V_1$ is still symmetric!

However, HYPERGRAPH-k-CUT, the problem of interest here, is *not* a special case of SYM-SUBMOD-k-PART. Nevertheless, we are able to prove a strong structural characterization. We state the structural characterization now. We consider the partition viewpoint of HYPERGRAPH-k-CUT. We will denote a k-partition by an ordered tuple. A k-partition is a minimum k-partition if it has the minimum number of crossing hyperedges among all possible k-partitions. Since there could be multiple minimum k-partitions, we will be interested in the k-partition $(V_1,\ldots,V_k)$ for which V_1 is maximal: formally, we define a minimum k-partition $(V_1,\ldots,V_k)$ to be a *maximal minimum k-partition* if there is no other minimum 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 2. *Let $G=(V,E)$ be a hypergraph and let $(V_1,\ldots,V_k)$ be a maximal minimum k-partition in G for an integer $k\ge 2$. Suppose $|V_1|\ge 2k-2$. Then, for every subset $T\subseteq\overline{V_1}$ such that T intersects V_j for every $j\in\{2,\ldots,k\}$, there exists a subset $S\subseteq V_1$ of size $2k-2$ such that $(V_1,\overline{V_1})$ is the source maximal minimum (S,T)-terminal cut.*

Some important remarks regarding the preceding theorem are in order. Firstly, this is surprising: for instance, if the optimum k-partition is unique, then the theorem allows us to find any part V_i of the optimum k-partition $(V_1,\ldots,V_k)$ by solving minimum (S,T)-terminal cuts for S and T of bounded sizes (by noting that the reordered k-partition $(V_i,V_1,\ldots,V_{i-1},V_{i+1},\ldots,V_k)$ is also a maximal minimum k-partition due to uniqueness and by applying Theorem 2 to this reordered k-partition). Such a result was not known even for graphs. Secondly, our structural theorem differs crucially from the structural lemma of Goldschmidt and Hochbaum [2] for GRAPH-k-CUT in that it does not rely on V_1 being the part with the smallest cut value. This also explains why we need S to be of size $2k-2$ instead of $k-1$: one can show that $2k-2$ is tight for our structural theorem if we want to identify an arbitrary part even when considering GRAPH-k-CUT. Thirdly, our structural theorem does not hold for general submodular functions. The theorem statement was partly inspired by experiments on small sized instances and the proof is partly inspired by a structural theorem in [8] for graphs.

Theorem 2 implies, relatively easily, an $n^{O(k^2)}$-time algorithm for HYPERGRAPH-k-CUT. We improve the running time to $n^{O(k)}$ using a similar but more involved structural result that allows us to recover the union of $k/2$ parts of an optimum k-partition. This high-level approach of recovering the union of $k/2$ parts of an optimum k-partition was developed in [8] for GRAPH-k-CUT. As we already mentioned in the preceding paragraph, a proof of a key structural lemma in [8] was an inspiration for our proofs though the precise statement of our structural theorem is different from the structural lemma of [8] and more subtle. We clarify this subtlety: the key structural lemma in [8] for graphs is that any 2-partition whose cut value is strictly smaller than half the optimum k-cut value can be recovered as a minimum (S,T)-terminal cut for S and T of sizes at most $k-1$. In contrast, our structural theorem (Theorem 2) states that V_1—whose cut value need not necessarily be smaller than half the optimum k-cut value—can be recovered as a minimum (S,T)-terminal cut for S and T of sizes at most $2k-2$. We emphasize that the factor 2 in the conclusion of our structural result (i.e., in the size of S) is not simply a consequence of weakening the hypothesis by a factor of 2 compared to that of [8].

Organization. In Section II, we formally describe and analyze the basic recursive algorithm that utilizes our main structural theorem (Theorem 2). We prove an important uncrossing property of the hypergraph cut function in Section III and use it to prove Theorem 2 in Section IV. Due to page limits, we defer the details of the faster $n^{O(k)}$-time algorithm based on divide-and-conquer to the full version of this work. All missing proofs also appear in the full version.

B. Other related work

Our main focus is on HYPERGRAPH-k-CUT and GRAPH-k-CUT when k is fixed. As we mentioned already, GRAPH-k-CUT is NP-Hard when k is part of the input [1]. A $2(1-1/k)$ approximation is known for GRAPH-k-CUT [23]; several other approaches also give a 2-approximation (see [12], [24] and references therein). Manurangsi [25] showed that there is no polynomial-time $(2-\epsilon)$-approximation for any constant $\epsilon > 0$ assuming the *Small Set Expansion Hypothesis* [26]. In contrast, HYPERGRAPH-k-CUT was recently shown [27] to be at least as hard as the *densest k-subgraph* problem. Combined with results in [28], this shows that HYPERGRAPH-k-CUT is unlikely to have a sub-polynomial factor approximation ratio and illustrates that HYPERGRAPH-k-CUT differs significantly from GRAPH-k-CUT when k is part of the input.

As we mentioned earlier, terminal versions of SUBMOD-k-PART and its special cases such as Multiway-Cut in graphs have been extensively studied. The most general version here is the following: given a submodular function $f : 2^V \to \mathbb{R}$ (by value oracle) and terminals $\{s_1, s_2, \ldots, s_k\} \subset V$ the goal is to find a partition $(V_1, \ldots, V_k)$ to minimize $\sum_i f(V_i)$ subject to the constraint that $s_i \in V_i$ for $1 \le i \le k$. These problems are NP-Hard even for $k = 3$ and the main focus has been on approximation algorithms. We refer the reader to [13], [29]–[31] for further references. We mention that for non-negative f and fixed k, the best approximation algorithms for SUBMOD-k-PART and SYM-SUBMOD-k-PART are via the terminal versions; a $(1.5-1/k)$ for SYM-SUBMOD-k-PART and a $2(1-1/k)$-approximation for SUBMOD-k-PART [30], [31].

Fixed parameter tractability of GRAPH-k-CUT has also been investigated. It is known that GRAPH-k-CUT is $W[1]$-hard (and hence not likely to be FPT) parameterized by k [32] while it is FPT when parameterized by k and the solution size [33]. We observed, via a simple reduction from a result of Marx on vertex separators [34], that HYPERGRAPH-k-CUT is $W[1]$ hard even when parameterized by k *and* the solution size. This also demonstrates that HYPERGRAPH-k-CUT differs in complexity from GRAPH-k-CUT.

Another problem closely related to HYPERGRAPH-k-CUT is the HYPERGRAPH-k-PARTITION problem. The input to HYPERGRAPH-k-PARTITION is a hypergraph $G = (V, E)$ and a positive integer k and the goal is to partition V into k non-empty sets $V_1, \ldots, V_k$ but the objective is to minimize $\sum_{i=1}^{k} |\delta_G(V_i)|$; this means that a hyperedge e that crosses $h \ge 2$ parts pays h instead of only once (as is the case in HYPERGRAPH-k-CUT). HYPERGRAPH-k-PARTITION is a special case of SYM-SUBMOD-k-PART and its complexity status for fixed $k \ge 5$ is open. HYPERGRAPH-k-PARTITION in constant rank hypergraphs is solvable in polynomial-time by relying on the fact that the number of constant-approximate minimum k-cuts in a constant rank hypergraph is polynomial.

II. RECURSIVE ALGORITHM

Theorem 2 allows us to design a recursive algorithm for hypergraph k-cut that we describe now. For a hypergraph $G = (V, E)$ and for a subset U of vertices, let $G[U]$ denote the hypergraph obtained from G by discarding the vertices in $\overline{U}$ and by discarding all hyperedges $e \in E$ that intersect $\overline{U}$. We describe the formal algorithm in Figure 1. It follows the high-level outline given in the technical overview. It enumerates $n^{O(k)}$ minimum (S,T)-terminal cuts, one of which is guaranteed to identify one part of an optimum k-partition, and then recursively finds an optimum $(k-1)$-partition after removing the found part. The run-time guarantee is given in Theorem 3.

Theorem 3. *Let $G = (V, E)$ be a n-vertex hypergraph of size p and let k be an integer. Then, algorithm CUT(G, k) returns a partition corresponding to a minimum k-cut in G and it can be implemented to run in $n^{O(k^2)}T(n, p)$ time, where $T(n, p)$ denotes the time complexity for computing the source maximal minimum (s,t)-terminal cut in a n-vertex hypergraph of size p.*

Proof: We first show the correctness of the algorithm. All candidates considered by the algorithm correspond to a k-partition, so we only have to show that the algorithm returns a k-partition corresponding to a minimum k-cut. We

Algorithm CUT(G, k)
Input: Hypergraph $G = (V, E)$ and an integer $k \geq 1$
Output: A k-partition corresponding to a minimum k-cut in G
If $k = 1$
    Return V
else
    Initialize $\mathcal{C} \leftarrow \{U \subset V : |U| \leq 2k-3\}$ and $\mathcal{R} \leftarrow \emptyset$
    For every disjoint $S, T \subset V$ with $|S| = 2k-2$ and $|T| = k-1$
        Compute the source maximal minimum (S, T)-terminal cut $(U, \overline{U})$
        $\mathcal{C} \leftarrow \mathcal{C} \cup \{U\}$
    For each $U \in \mathcal{C}$
        $\mathcal{P}_{\overline{U}} :=$ CUT$(G[\overline{U}], k-1)$
        $\mathcal{P} :=$ Partition of V obtained by concatenating U with $\mathcal{P}_{\overline{U}}$
        $\mathcal{R} \leftarrow \mathcal{R} \cup \{\mathcal{P}\}$
    Among all k-partitions in $\mathcal{R}$, pick the one with minimum cost and return it

Figure 1. Algorithm to compute minimum k-cut in hypergraphs.

show this by induction on k. The base case of $k = 1$ is trivial. We show the induction step. Assume that $k \geq 2$. Let $(V_1, \ldots, V_k)$ be a maximal minimum k-partition with cost OPT_k. By Theorem 2, the 2-partition $(V_1, \overline{V_1})$ is in $\mathcal{C}$. By induction hypothesis, the algorithm will return a minimum $(k-1)$-partition $(Q_1, \ldots, Q_{k-1})$ of $G[\overline{V_1}]$. Hence,

$$\text{cost}_{G[\overline{V_1}]}(Q_1, \ldots, Q_{k-1}) \leq \text{cost}_{G[\overline{V_1}]}(V_2, \ldots, V_k).$$

Therefore, the cost of the k-partition $(V_1, Q_1, \ldots, Q_{k-1})$ is

$$\begin{aligned} &d(V_1) + \text{cost}_{G[\overline{V_1}]}(Q_1, \ldots, Q_{k-1}) \\ &\quad \leq d(V_1) + \text{cost}_{G[\overline{V_1}]}(V_2, \ldots, V_k) \\ &\quad = OPT_k. \end{aligned}$$

Moreover, the k-partition $(V_1, Q_1, \ldots, Q_{k-1})$ is in $\mathcal{R}$. Hence, the algorithm returns a k-partition with cost at most OPT_k.

Next, we bound the run-time of the algorithm. Let $N(k, n)$ denote the number of source maximal minimum (s, t)-terminal cut computations executed by the algorithm CUT(G, k) on a n-vertex hypergraph G. We note that $|\mathcal{R}| = |\mathcal{C}| = O(n^{3k-3})$. Therefore,

$$\begin{aligned} &N(k, n) \leq O(n^{3k-3})(1 + N(k-1, n)) \text{ and} \\ &N(1, n) = O(1). \end{aligned}$$

Hence, $N(k, n) = O(n^{3k(k-1)/2})$. The total run-time is dominated by the time to implement these minimum (s, t)-terminal cuts and hence it is $O(n^{3k(k-1)/2})T(n, p)$. ∎

III. Uncrossing properties of the hypergraph cut function

In this section, we show the following uncrossing theorem which will be useful to prove the main structural theorem. See Figure 2 for an illustration of the sets that appear in the statement of Theorem 4. The motivation for the statement of this uncrossing theorem will be clearer in the proof of Theorem 2. The reader may want to skip the rather long and technical proof of the uncrossing theorem in the first reading and come back to it after seeing its use in the proof of Theorem 2.

Theorem 4. *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, there exists a k-partition $(P_1, \ldots, P_k)$ of V with $\overline{U} \subsetneq P_k$ such that*

$$\textit{cost}(P_1, \ldots, P_k) \leq \frac{1}{2} \min\{d(A_i) + d(A_j) : i, j \in [p], i \neq j\}.$$

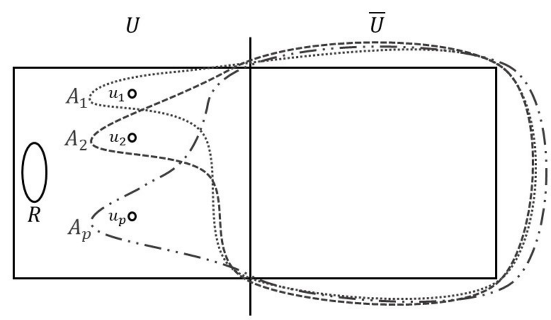

Figure 2. Illustration of the sets that appear in Theorem 4 and Lemma 2.

The rest of the section is devoted to the proof of Theorem 4. We begin with some background on the hypergraph cut function. Let $G = (V, E)$ be a hypergraph. For a subset A of vertices, we recall that $d(A)$ denotes the number of hyperedges that intersect both A and $\overline{A}$. The function $d : 2^V \to \mathbb{R}_+$ is known as the hypergraph cut function. The hypergraph cut function is symmetric, i.e.,

$$d(A) = d(\overline{A}) \text{ for all } A \subseteq V,$$

and submodular, i.e.,

$$d(A) + d(B) \geq d(A \cap B) + d(A \cup B) \text{ for all subsets } A, B \subseteq V.$$

For our purposes, it will help to count the hyperedges more accurately than employ the submodularity inequality. We define some notation that will help in more accurate counting. Let $(Y_1, \ldots, Y_p, W, Z)$ be a partition of V. We recall that $\mathrm{cost}(Y_1, \ldots, Y_p, W, Z)$ denotes the number of hyperedges that cross the partition. We note that when considering these hyperedges it is convenient to visualize each part of the partition as a single vertex obtained by contracting the part. We define the following quantities:

1) Let $\mathrm{cost}(W, Z) = |\{e \mid e \subseteq W \cup Z, e \cap W \neq \emptyset, e \cap Z \neq \emptyset\}|$ be the number of hyperedges contained in $W \cup Z$ that intersect both W and Z.
2) Let $\alpha(Y_1, \ldots, Y_p, W, Z)$ be the number of hyperedges that intersect Z and at least two of the sets in $\{Y_1, \ldots, Y_p, W\}$.
3) Let $\beta(Y_1, \ldots, Y_p, Z)$ be the number of hyperedges that are disjoint from Z but intersect at least two of the sets in $\{Y_1, \ldots, Y_p\}$.

For a partition $(Y_1, \ldots, Y_p, W, Z)$, we will be interested in the sum of $\mathrm{cost}(Y_1, \ldots, Y_p, W, Z)$ with the three quantities defined above which we denote as $\sigma(Y_1, \ldots, Y_p, W, Z)$, i.e.,

$$\begin{aligned}\sigma(Y_1, \ldots, Y_p, W, Z) := \mathrm{cost}(Y_1, \ldots, Y_p, W, Z) &+ \mathrm{cost}(W, Z) \\ &+ \alpha(Y_1, \ldots, Y_p, W, Z) \\ &+ \beta(Y_1, \ldots, Y_p, Z).\end{aligned}$$

We note that $\sigma(Y_1, \ldots, Y_p, W, Z)$ counts every hyperedge that crosses the partition twice except for those hyperedges that intersect exactly one of the sets in $\{Y_1, \ldots, Y_p\}$ and exactly one of the sets in $\{W, Z\}$ which are counted exactly once (see Figure 3).

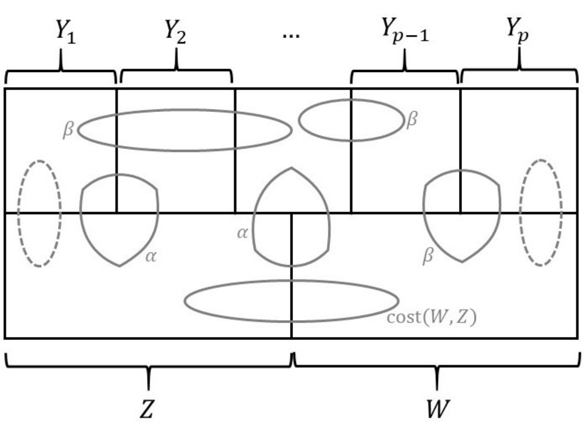

Figure 3. Hyperedges counted by $\sigma(Y_1, \ldots, Y_p, W, Z)$: The dashed hyperedges are counted only by $\mathrm{cost}(Y_1, \ldots, Y_p, W, Z)$. The rest of the hyperedges are counted twice in $\sigma(Y_1, \ldots, Y_p, W, Z)$: once by the term $\mathrm{cost}(Y_1, \ldots, Y_p, W, Z)$ and once more by the indicated term.

The motivation behind considering the function $\sigma(Y_1, \ldots, Y_p, W, Z)$ comes from Proposition 1. We emphasize that the interpretation for $\sigma(Y_1, \ldots, Y_p, W, Z)$ given in the proposition holds only for $p = 2$.

Proposition 1. *Let (Y_1, Y_2, W, Z) be a partition of V and let $A_1 := Y_1 \cup W$ and $A_2 := Y_2 \cup W$. Then,*

$$d(A_1) + d(A_2) = \sigma(Y_1, Y_2, W, Z).$$

The next lemma will help in obtaining a $(p+3)$-partition from a $(p+2)$-partition while controlling the increase in σ-value. This will be used in a subsequent inductive argument. See Figure 4 for an illustration of the sets appearing in the statement of the lemma. Our proof of Lemma 1 is through case analysis. Currently we do not know how to prove this lemma without a somewhat laborious case analysis. We remark that this is partly due to the fact that hyperedges can have different cardinalities as well as due to the fact that we cannot rely only on submodularity of the hypergraph cut function.

Lemma 1. *Let $G = (V, E)$ be a hypergraph and let $(X_1, \ldots, X_p, W_0, Z_0)$ be a partition for some integer $p \geq 1$. Let $Q \subset V$ be a set such that*

$$\begin{aligned} Y_i &:= X_i - Q \neq \emptyset \ \forall\ i \in [p], \\ Y_{p+1} &:= Q \cap Z_0 \neq \emptyset, \\ Z &:= Z_0 - Q \neq \emptyset, \ \textit{and} \\ W &:= W_0 \cup (Q \setminus Z_0) \neq \emptyset. \end{aligned}$$

Then, $(Y_1, \ldots, Y_p, Y_{p+1}, W, Z)$ is a partition of V such that

$$\begin{aligned}\sigma(Y_1, \ldots, Y_p, Y_{p+1}, W, Z) \leq \sigma(X_1, \ldots, X_p, W_0, Z_0) \\ + d(Q) - d(W_0 \cap Q).\end{aligned}$$

Proof: By definition $(Y_1, \ldots, Y_p, Y_{p+1}, W, Z)$ is a partition of V.

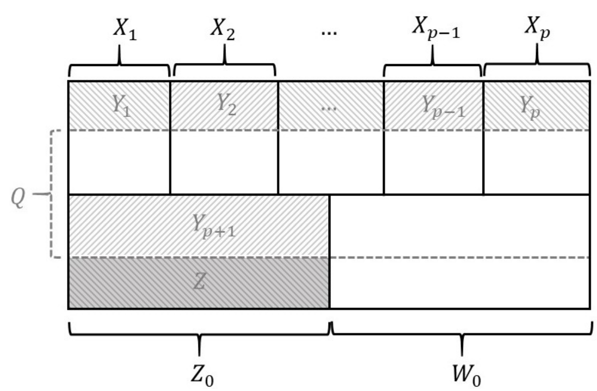

Figure 4. Sets appearing in Lemma 1. The unshaded portion corresponds to W.

We rewrite the required inequality in the following form as it becomes convenient to prove:

$$\begin{aligned}&\sigma(X_1, \ldots, X_p, W_0, Z_0) - \sigma(Y_1, \ldots, Y_p, Y_{p+1}, W, Z) \\ &\geq d(W_0 \cap Q) - d(Q). \qquad (1)\end{aligned}$$

For a hyperedge $e \in E$, let $\lambda_e^0 \in \{0, 1, 2\}$ and $\lambda_e^1 \in \{0, 1, 2\}$ be the number of times that e is counted by

$\sigma(X_1,\ldots,X_p,W_0,Z_0)$ and $\sigma(Y_1,\ldots,Y_p,Y_{p+1},W,Z)$ respectively, and let $\lambda_e^Q \in \{0,1\}$ and $\lambda_e^{W_0\cap Q} \in \{0,1\}$ be the number of times that e is counted by $d(Q)$ and $d(W_0 \cap Q)$ respectively.

Let $\ell_e := \lambda_e^0 - \lambda_e^1$ and $r_e := \lambda_e^{W_0\cap Q} - \lambda_e^Q$. Thus, ℓ_e and r_e denote the number of times the hyperedge e is counted in the LHS and RHS of inequality (1) respectively and moreover $\ell_e \in \{0,\pm1,\pm2\}$ and $r_e \in \{0,\pm1\}$ for every hyperedge $e \in E$. Let

$$\begin{aligned}\text{Positives}(\ell) &:= \sum_{e\in E:\ell_e\geq 1} \ell_e,\\ \text{Negatives}(\ell) &:= \sum_{e\in E:\ell_e\leq -1} \ell_e,\\ \text{Positives}(r) &:= \sum_{e\in E:r_e=1} r_e, \text{ and}\\ \text{Negatives}(r) &:= \sum_{e\in E:r_e=-1} r_e.\end{aligned}$$

Claims 1 and 2 complete the proof of the lemma. ■

Claim 1.

$$\textit{Positives}(\ell) \geq \textit{Positives}(r).$$

Proof: Let e be a hyperedge such that $r_e = 1$. Then, e is counted by $d(W_0 \cap Q)$ but not $d(Q)$. This means that $e \subseteq Q$, $e \cap (W_0 \cap Q) \neq \emptyset$, and $e \cap (Q \setminus W_0) \neq \emptyset$. Thus, e intersects $W_0 \cap Q$ and at least one of the sets in $\{X_1 \cap Q,\ldots,X_p \cap Q, Z_0 \cap Q\}$. It suffices to show that $\ell_e \geq 1$. We consider different cases for e below and show that $\ell_e \geq 1$ in all cases.

1) Suppose e intersects $Z_0 \cap Q$.
   a) Suppose e is disjoint from $X_1 \cap Q,\ldots,X_p \cap Q$. Then, $\lambda_e^0 = 2$ since e is counted by both cost$(X_1,\ldots,X_p,W_0,Z_0)$ and by cost(W_0,Z_0). However, $\lambda_e^1 = 1$ since e is counted only by cost$(Y_1,\ldots,Y_{p+1},W,Z)$. Hence, $\ell_e = \lambda_e^0 - \lambda_e^1 \geq 1$.
   b) Suppose e intersects at least one of the sets in $\{X_1 \cap Q,\ldots,X_p \cap Q\}$. Then, $\lambda_e^0 = 2$ since e is counted by both cost$(X_1,\ldots,X_p,W_0,Z_0)$ and by $\alpha(X_1,\ldots,X_p,W_0,Z_0)$. However, $\lambda_e^1 = 1$ since e is counted only by cost$(Y_1,\ldots,Y_{p+1},W,Z)$. Hence, $\ell_e = \lambda_e^0 - \lambda_e^1 \geq 1$.
2) Suppose e is disjoint from $Z_0 \cap Q$. Then e has to intersect at least one of the sets in $\{X_1 \cap Q,\ldots,X_p \cap Q\}$.
   a) Suppose e intersects exactly one of the sets in $\{X_1 \cap Q,\ldots,X_p \cap Q\}$. Then, $\lambda_e^0 = 1$ since e is counted only by cost$(X_1,\ldots,X_p,W_0,Z_0)$. However, $\lambda_e^1 = 0$ since e does not cross the partition $(Y_1,\ldots,Y_{p+1},W,Z)$. Hence, $\ell_e = \lambda_e^0 - \lambda_e^1 \geq 1$.
   b) Suppose e intersects at least two of the sets in $\{X_1 \cap Q,\ldots,X_p \cap Q\}$. Then, $\lambda_e^0 = 2$ since e is counted by both cost$(X_1,\ldots,X_p,W_0,Z_0)$ and by $\beta(X_1,\ldots,X_p,Z_0)$. However, $\lambda_e^1 = 0$ since e does not cross the partition $(Y_1,\ldots,Y_{p+1},W,Z)$. Hence, $\ell_e = \lambda_e^0 - \lambda_e^1 = 2 \geq 1$.

■

Claim 2.

$$\textit{Negatives}(\ell) \geq \textit{Negatives}(r).$$

Proof: Let e be a hyperedge such that $\ell_e \leq -1$, i.e., $\lambda_e^1 \geq \lambda_e^0 + 1$. Then $\lambda_e^1 \geq 1$ and hence, e crosses the partition $(Y_1,\ldots,Y_{p+1},W,Z)$. It suffices to show that $r_e \leq \ell_e$, i.e., $\lambda_e^Q \geq \lambda_e^{W_0\cap Q} + \lambda_e^1 - \lambda_e^0$. We consider different cases for e below and for each case, we show that either $\lambda_e^Q \geq \lambda_e^{W_0\cap Q} + \lambda_e^1 - \lambda_e^0$ or the case is impossible.

1) Suppose e is disjoint from Z. Then, e intersects at least one of the sets in $\{Y_1,\ldots,Y_{p+1}\}$ since e crosses the partition $(Y_1,\ldots,Y_{p+1},W,Z)$.
   a) Suppose e intersects exactly one of the sets in $\{Y_1,\ldots,Y_{p+1}\}$, say Y_i for some $i \in [p+1]$. Then, e intersects W and consequently, $\lambda_e^1 = 1$ since e is counted only by cost$(Y_1,\ldots,Y_{p+1},W,Z)$. Since $1 = \lambda_e^1 \geq \lambda_e^0 + 1$, it follows that $\lambda_e^0 = 0$. This implies that e does not cross the partition $(X_1,\ldots,X_p,W_0,Z_0)$. Therefore, $i \in [p]$ and $e \subseteq X_i$ with e intersecting $X_i \cap Q$ and $Y_i = X_i \setminus Q$. Consequently, $\lambda_e^Q = 1$ and $\lambda_e^{W_0\cap Q} = 0$. Hence $\lambda_e^Q \geq \lambda_e^{W_0\cap Q} + \lambda_e^1 - \lambda_e^0$.
   b) Suppose e intersects at least two of the sets in $\{Y_1,\ldots,Y_{p+1}\}$. Then, $\lambda_e^1 = 2$ since e is counted by both cost$(Y_1,\ldots,Y_{p+1},W,Z)$ as well as $\beta(Y_1,\ldots,Y_{p+1},Z)$.
      i) Suppose e intersects at least two of the sets in $\{Y_1,\ldots,Y_p\}$. If e intersects Z_0, then $\lambda_e^0 = 2$ since e is counted by both cost$(X_1,\ldots,X_p,W_0,Z_0)$ and $\alpha(X_1,\ldots,X_p,W_0,Z_0)$. If e is disjoint from Z_0, then again $\lambda_e^0 = 2$ since e is counted by both cost$(X_1,\ldots,X_p,W_0,Z_0)$ and $\beta(X_1,\ldots,X_p,W_0,Z_0)$. In both cases, we have $2 = \lambda_e^1 \geq \lambda_e^0 + 1 = 3$, a contradiction.
      ii) Suppose e intersects Y_{p+1} and exactly one of the sets in $\{Y_1,\ldots,Y_p\}$, say Y_i for some $i \in [p]$. Then, $\lambda_e^0 \geq 1$ since e crosses the partition $(X_1,\ldots,X_p,W_0,Z_0)$. Since $2 = \lambda_e^1 \geq \lambda_e^0 + 1$, it follows that $\lambda_e^0 = 1$. This implies that none of cost(W_0,Z_0), $\alpha(X_1,\ldots,X_p,W_0,Z_0)$, and $\beta(X_1,\ldots,X_p,Z_0)$ count e. Therefore, e is disjoint from W and e intersects $Y_{p+1} = Z_0 \cap Q$ and $Y_i = X_i \setminus Q$. Thus, e is counted

by $d(Q)$ but not $d(W_0 \cap Q)$. Consequently, $\lambda_e^Q = 1$ and $\lambda_e^{W_0 \cap Q} = 0$. Hence, $\lambda_e^Q \geq \lambda_e^{W_0 \cap Q} + \lambda_e^1 - \lambda_e^0$.

2) Suppose e intersects Z. Then, e intersects at least one of the sets in $\{Y_1, \ldots, Y_{p+1}, W\}$ since e crosses the partition $(Y_1, \ldots, Y_{p+1}, W, Z)$.

   a) Suppose e intersects exactly one of the sets in $\{Y_1, \ldots, Y_{p+1}, W\}$. Then, $\lambda_e^1 = 1$ since e is counted only by cost$(Y_1, \ldots, Y_{p+1}, W)$.

      i) Suppose e is disjoint from W. Then, e intersects exactly one of the sets in $\{Y_1, \ldots, Y_{p+1}\}$. Since $1 = \lambda_e^1 \geq \lambda_e^0 + 1$, we have that $\lambda_e^0 = 0$. This implies that e does not cross the partition $(X_1, \ldots, X_p, W_0, Z_0)$. Hence, e can only intersect Y_{p+1}. Thus, $e \subseteq Z_0 = Z \cup Y_{p+1}$ with e intersecting $Z = Z_0 \setminus Q$ and $Y_{p+1} = Z_0 \cap Q$. Thus, e is counted by $d(Q)$ but not $d(W_0 \cap Q)$. Consequently, $\lambda_e^Q = 1$ and $\lambda_e^{W_0 \cap Q} = 0$. Hence, $\lambda_e^Q \geq \lambda_e^{W_0 \cap Q} + \lambda_e^1 - \lambda_e^0$.

      ii) Suppose e intersects W. Then, e has to cross the partition $(X_1, \ldots, X_p, W_0, Z_0)$ and therefore, $\lambda_e^0 \geq 1$. Thus, $1 = \lambda_e^1 \geq \lambda_e^0 + 1 = 2$, a contradiction.

   b) Suppose e intersects at least two of the sets in $\{Y_1, \ldots, Y_{p+1}, W\}$. Then, $\lambda_e^1 = 2$ since e is counted by both cost$(Y_1, \ldots, Y_{p+1}, W, Z)$ and $\alpha(Y_1, \ldots, Y_{p+1}, W, Z)$.

      i) Suppose e intersects at least two of the sets in $\{Y_1, \ldots, Y_p\}$. Then $\lambda_e^0 = 2$ since e is counted by cost$(X_1, \ldots, X_p, W_0, Z_0)$ as well as $\alpha(X_1, \ldots, X_p, W_0, Z_0)$. Thus, $2 = \lambda_e^1 \geq \lambda_e^0 + 1 = 3$, a contradiction.

      ii) Suppose e intersects exactly one of the sets in $\{Y_1, \ldots, Y_p\}$, say Y_i for some $i \in [p]$, and e intersects Y_{p+1} but is disjoint from W. Then, $\lambda_e^0 \geq 1$ since e crosses the partition $(X_1, \ldots, X_p, W_0, Z_0)$. Since $2 = \lambda_e^1 \geq \lambda_e^0 + 1$, it follows that $\lambda_e^0 = 1$. This implies that none of cost(W_0, Z_0), $\alpha(X_1, \ldots, X_p, W_0, Z_0)$, and $\beta(X_1, \ldots, X_p, Z_0)$ count e and hence, e is contained in $Y_i \cup Z_0 \subseteq X_i \cup Z_0$ with e intersecting $Y_{p+1} = Z_0 \cap Q$ and $Y_i = X_i \setminus Q$. Thus, e is counted by $d(Q)$ but not $d(W_0 \cap Q)$. Consequently, $\lambda_e^Q = 1$ and $\lambda_e^{W_0 \cap Q} = 0$. Hence, $\lambda_e^Q \geq \lambda^{W_0 \cap Q} + \lambda_e^1 - \lambda_e^0$.

      iii) Suppose e intersects exactly one of the sets in $\{Y_1, \ldots, Y_p\}$, say Y_i for some $i \in [p]$, and e intersects W but is disjoint from Y_{p+1}. Then, $\lambda_e^0 \geq 1$ since e crosses the partition $(X_1, \ldots, X_p, W_0, Z_0)$. Since $2 = \lambda_e^1 \geq \lambda_e^0 + 1$, it follows that $\lambda_e^0 = 1$. This implies that none of cost(W_0, Z_0), $\alpha(X_1, \ldots, X_p, W_0, Z_0)$, and $\beta(X_1, \ldots, X_p, Z_0)$ count e. Therefore, e is contained in $X_i \cup Z$ and e intersects $X_i \cap Q$ since e has to intersect W. Moreoever, e intersects $Y_i = X_i \setminus Q$. Thus, e is counted by $d(Q)$ but not $d(W_0 \cap Q)$. Consequently, $\lambda_e^Q = 1$ and $\lambda_e^{W_0 \cap Q} = 0$. Hence, $\lambda_e^Q \geq \lambda_e^{W_0 \cap Q} + \lambda_e^1 - \lambda_e^0$.

      iv) Suppose e is disjoint from $Y_1, \ldots, Y_p$ and intersects both Y_{p+1} and W.

         A) Suppose e intersects at least two of the sets in $\{X_1 \cap Q, \ldots, X_p \cap Q\}$. Then, $\lambda_e^0 = 2$ since e is counted by cost$(X_1, \ldots, X_p, W_0, Z_0)$ as well as $\alpha(X_1, \ldots, X_p, W_0, Z_0)$. Thus, $2 = \lambda_e^1 \geq \lambda_e^0 + 1 = 3$, a contradiction.

         B) Suppose e does not intersect $X_1 \cap Q, \ldots, X_p \cap Q$. Then, e intersects W_0 since e is counted by both cost$(Y_1, \ldots, Y_{p+1}, W, Z)$ and $\alpha(Y_1, \ldots, Y_{p+1}, W, Z)$ (recall that we are in case (b)). Moreoever, $e \subseteq W_0 \cup Z_0$. Therefore, $\lambda_e^0 = 2$ since e is counted by cost$(X_1, \ldots, X_p, W_0, Z_0)$ as well as cost(W_0, Z_0). Thus, $2 = \lambda_e^1 \geq \lambda_e^0 + 1 = 3$, a contradiction.

         C) Suppose e intersects exactly one of the sets in $\{X_1 \cap Q, \ldots, X_p \cap Q\}$, say $X_i \cap Q$ for some $i \in [p]$, and e intersects $W_0 \cap Q$. Then, $\lambda_e^0 = 2$ since e is counted by both cost$(X_1, \ldots, X_p, W_0, Z_0)$ and $\alpha(X_1, \ldots, X_p, W_0, Z_0)$. Thus, $2 = \lambda_e^1 \geq \lambda_e^0 + 1 = 3$, a contradiction.

         D) Suppose e intersects exactly one of the sets in $\{X_1 \cap Q, \ldots, X_p \cap Q\}$, say $X_i \cap Q$ for some $i \in [p]$, and e is disjoint from $W_0 \cap Q$. Then, $\lambda_e^0 \geq 1$ since e crosses the partition $(X_1, \ldots, X_p, W_0, Z_0)$. Since $2 = \lambda_e^1 \geq \lambda_e^0 + 1$, it follows that $\lambda_e^0 = 1$. This implies that none of cost(W_0, Z_0), $\alpha(X_1, \ldots, X_p, W_0, Z_0)$, and $\beta(X_1, \ldots, X_p, Z_0)$ count e. Therefore, e is contained in $(X_i \cap Q) \cup Z_0$ and e intersects $Y_{p+1} = Z_0 \cap Q$ and $Z = Z_0 \setminus Q$. Thus, e is counted by $d(Q)$ but not $d(W_0 \cap Q)$. Consequently, $\lambda_e^Q = 1$ and $\lambda_e^{W_0 \cap Q} = 0$. Hence, $\lambda_e^Q \geq \lambda_e^{W_0 \cap Q} + \lambda_e^1 - \lambda_e^0$.

■

The next lemma will help in uncrossing a collection of sets to obtain a partition with small σ-value. See Figure 2 for an illustration of the sets that appear in the statement of the lemma.

Lemma 2. *Let $G=(V,E)$ be a hypergraph and $\emptyset \neq R \subsetneq U \subsetneq V$. Let $S=\{u_1,\ldots,u_p\} \subseteq U\setminus R$ for $p\geq 2$. 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 $i\in[p]$. Let*

$$Z := \cap_{i=1}^{p}\overline{A_i},\ W := \cup_{1\leq i<j\leq p}(A_i\cap A_j),\ \text{and}$$
$$Y_i := A_i - W\ \forall i\in[p].$$

Then, $(Y_1,\ldots,Y_p,W,Z)$ is a $(p+2)$-partition of V with

$$\sigma(Y_1,\ldots,Y_p,W,Z)\leq \min\{d(A_i)+d(A_j): i,j\in[p], i\neq j\}.$$

Proof: For every $i\in[p]$, the set Y_i is non-empty since $u_i\in Y_i$. The set W is non-empty since $\overline{U}\subseteq W$. The set Z is non-empty since $R\subseteq Z$. By definition, the sets $Y_1,\ldots,Y_p,W,Z$ are all disjoint and their union contains all vertices. Hence, $(Y_1,\ldots,Y_p,W,Z)$ is a partition of V. Without loss of generality, let $d(A_1)\leq d(A_2)\leq\ldots\leq d(A_p)$. We bound the σ-value of the partition by induction on p.

The base case of $p=2$ follows from Proposition 1. We show the induction step. Suppose that the statement holds for $p=q$. We prove that it holds for $p=q+1$. Consider $R_0 := R\cup\{u_{q+1}\}$ and $S_0 := S\setminus\{u_{q+1}\}$. Then, $(\overline{A_i},A_i)$ is still a minimum $((S_0\cup R_0)\setminus\{u_i\},\overline{U})$-terminal cut for every $i\in[q]$ and moreover, $u_i\in A_i\setminus\cup_{j\in[q]\setminus\{i\}}A_j$ for every $i\in[q]$. By induction hypothesis, we get that for the sets

$$Z_0 := \cap_{i=1}^{q}\overline{A_i},\ W_0 := \cup_{1\leq i<j\leq q}(A_i\cap A_j),\ \text{and}$$
$$X_i := A_i - W\ \forall i\in[q],$$

we have

$$\sigma(X_1,\ldots,X_q,W_0,Z_0)\leq d(A_1)+d(A_2).$$

The partition $(X_1,\ldots,X_q,W_0,Z_0)$ and the set $Q := A_{q+1}$ satisfy the conditions of Lemma 1. By Lemma 1, we obtain that

$$\begin{aligned}\sigma(Y_1,\ldots,Y_q,Y_{q+1},W,Z)&\leq\sigma(X_1,\ldots,X_q,W_0,Z_0)\\&\quad+d(A_{q+1})-d(W_0\cap A_{q+1}).\end{aligned}$$

Since $(\overline{W_0\cap A_{q+1}},W_0\cap A_{q+1})$ is a feasible $((S\cup R)\setminus\{u_{q+1}\},\overline{U})$-terminal cut, we have that $d(A_{q+1})\leq d(W_0\cap A_{q+1})$. Hence,

$$\begin{aligned}\sigma(Y_1,\ldots,Y_q,Y_{q+1},W,Z)&\leq\sigma(X_1,\ldots,X_q,W_0,Z_0)\\&\leq d(A_1)+d(A_2).\end{aligned}$$

■

The next lemma will help in aggregating the parts of a $2k$-partition $\mathcal{P}$ to a k-partition $\mathcal{K}$ so that the cost of $\mathcal{K}$ is at most half the σ-value of $\mathcal{P}$.

Lemma 3. *Let $G=(V,E)$ be a hypergraph, $k\geq 2$ be an integer, and $(Y_1,\ldots,Y_p,W,Z)$ be a partition of V for some integer $p\geq 2k-2$. Then, there exist distinct $i_1,\ldots,i_{k-1}\in[p]$ such that $2cost\left(Y_{i_1},\ldots,Y_{i_{k-1}},V\setminus(\cup_{j=1}^{k-1}Y_{i_j})\right)$ is at most*

$$\begin{aligned}&cost(Y_1,\ldots,Y_p,W,Z)+\alpha(Y_1,\ldots,Y_p,W,Z)\\&\quad+\beta(Y_1,\ldots,Y_p,Z).\end{aligned}$$

Proof: Suppose that the lemma is false. Pick a counterexample hypergraph $G=(V,E)$ such that $|V|+|E|$ is minimum. Hence, for every distinct $i_1,\ldots,i_{k-1}\in[p]$, we have $2\text{cost}\left(Y_{i_1},\ldots,Y_{i_{k-1}},V\setminus(\cup_{j=1}^{k-1}Y_{i_j})\right)$

$$\begin{aligned}&>\text{cost}(Y_1,\ldots,Y_p,W,Z)+\alpha(Y_1,\ldots,Y_p,W,Z)\\&\quad+\beta(Y_1,\ldots,Y_p,Z).\end{aligned}$$

Minimality of the counterexample implies that $|Y_i|=1$ for every $i\in[p]$ and $|W|=1=|Z|$ (otherwise, we can obtain a smaller counterexample by contracting the corresponding subset). If there exists a hyperedge $e\subseteq W\cup Z$ with e intersecting both W and Z, then discarding e would still preserve the counterexample property since e is not counted in LHS but is counted in RHS, hence no such hyperedge exists in G. For similar reasons, if there exists a hyperedge e that is double counted by RHS (see Figure 3), then discarding this hyperedge would still preserve the counterexample property. Minimality of the counterexample implies that no such hyperedge can exist. Consequently, all hyperedges present in the hypergraph G are in fact edges with one end-vertex in Y_i for some $i\in[p]$ and another end-vertex in W or Z. Thus,

$$RHS=\text{cost}(Y_1,\ldots,Y_p,W,Z)=\sum_{i=1}^{p}d(Y_i).$$

Without loss of generality, let $d(Y_1)\leq d(Y_2)\leq\ldots\leq d(Y_p)$. Then,

$$\begin{aligned}2\text{cost}\left(Y_1,\ldots,Y_{k-1},V\setminus(\cup_{j=1}^{k-1}Y_{i_j})\right)&=2\sum_{i=1}^{k-1}d(Y_i)\\&\leq\sum_{i=1}^{p}d(Y_i)=RHS.\end{aligned}$$

The inequality above is because $p\geq 2(k-1)$. Thus, G cannot be a counterexample. ■

We now restate and prove the main uncrossing theorem of this section.

Theorem 4. *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, there exists a k-partition $(P_1,\ldots,P_k)$ of V with $\overline{U}\subsetneq P_k$ such that*

$$cost(P_1,\ldots,P_k)\leq\frac{1}{2}\min\{d(A_i)+d(A_j): i,j\in[p], i\neq j\}.$$

Proof: By applying Lemma 2, we obtain a $(p+2)$-partition $(Y_1, \ldots, Y_p, W, Z)$ such that

$$\sigma(Y_1, \ldots, Y_p, W, Z) \leq \min\{d(A_i)+d(A_j) : i, j \in [p], i \neq j\}$$

and moreover, $\overline{U} \subseteq W$. We recall that $p \geq 2k-2$. Hence, by applying Lemma 3 to the $(p+2)$-partition $(Y_1, \ldots, Y_p, W, Z)$, we obtain a k-partition $(P_1, \ldots, P_k)$ of V such that $W \cup Z \subseteq P_k$ and

$$\begin{aligned}\text{cost}(P_1, \ldots, P_k) &\leq \frac{1}{2}\sigma(Y_1, \ldots, Y_p, W, Z) \\ &\leq \frac{1}{2}\min\{d(A_i)+d(A_j) : i, j \in [p], i \neq j\}.\end{aligned}$$

We note that $\overline{U}$ is strictly contained in P_k since $\overline{U} \cup Z \subseteq W \cup Z \subseteq P_k$ and Z is non-empty. ■

IV. Proof of Theorem 2

In this section, we prove Theorem 2. We start with a useful containment property captured by the next lemma.

Lemma 4. *Let $G = (V, E)$ be a hypergraph, $(V_1, \ldots, V_k)$ be a maximal minimum k-partition in G for an integer $k \geq 2$, 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 $U \setminus V_1 \neq \emptyset$. We will obtain another minimum k-partition that will contradict the maximality of V_1 in the minimum k-partition $(V_1, \ldots, V_k)$. We observe that

$$d(U) \leq d(U \cap V_1) \tag{2}$$

since $(U \cap V_1, \overline{U \cap V_1})$ is a (S,T)-terminal cut. We need the following claim:

Claim 3.

$$d(V_1) \leq d(U \cup V_1).$$

Proof: For the sake of contradiction, suppose $d(U \cup V_1) < d(V_1)$. Then, consider $W_1 := U \cup V_1$ and $W_j := V_j \setminus U$ for every $j \in \{2, \ldots, k\}$ (see Figure 5). We have $d(W_1) < d(V_1)$. Since $S \subseteq W_1$ and $T \cap W_j \neq \emptyset$ for every $j \in \{2, \ldots, k\}$, we have that $(W_1, \ldots, W_k)$ is a k-partition. We will show that $\text{cost}(W_1, \ldots, W_k)$ is strictly smaller than $\text{cost}(V_1, \ldots, V_k)$, thus contradicting the optimality of the k-partition $(V_1, \ldots, V_k)$.

We recall that for a subset A of vertices, the graph $G[A]$ is obtained from G by discarding the vertices in $\overline{A}$ and by discarding the hyperedges that intersect $\overline{A}$. With this notation, we can write

$$\text{cost}_G(W_1, \ldots, W_k) = d(W_1) + \text{cost}_{G[\overline{W_1}]}(W_2, \ldots, W_k) \text{ and}$$
$$\text{cost}_G(V_1, \ldots, V_k) = d(V_1) + \text{cost}_{G[\overline{V_1}]}(V_2, \ldots, V_k).$$

Moreover, every hyperedge that is disjoint from $W_1 = U \cup V_1$ but crosses the $(k-1)$-partition $(W_2 = V_2 \setminus U, \ldots, W_k = V_k \setminus U)$ is also disjoint from V_1 but crosses the $(k-1)$-partition $(V_2, \ldots, V_k)$. Hence, $\text{cost}_{G[\overline{W_1}]}(W_2, \ldots, W_k) \leq \text{cost}_{G[\overline{V_1}]}(V_2, \ldots, V_k)$. We also have $d(W_1) < d(V_1)$. Therefore,

$$\text{cost}(W_1, \ldots, W_k) < \text{cost}(V_1, \ldots, V_k),$$

a contradiction to optimality of the k-partition $(V_1, \ldots, V_k)$. ■

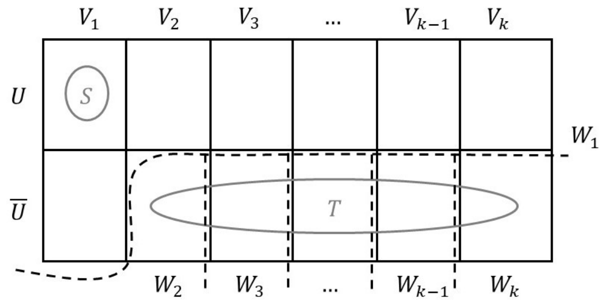

Figure 5. Uncrossing in the proof of Claim 3.

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

$$d(U) + d(V_1) \leq d(U \cap V_1) + d(U \cup V_1) \leq d(U) + d(V_1).$$

Therefore, the inequality in Claim 3 should in fact be an equation, i.e.,

$$d(V_1) = d(U \cup V_1).$$

Going through the proof of Claim 3 with this additional fact, we obtain that the k-partition $(U \cup V_1, V_2 \setminus U, \ldots, V_k \setminus U)$ has cost at most that of $(V_1, \ldots, V_k)$. Hence, the k-partition $(U \cup V_1, V_2 \setminus U, \ldots, V_k \setminus U)$ is also a minimum k-partition and it contradicts the maximality of V_1. ■

We now restate and prove Theorem 2.

Theorem 2. *Let $G = (V, E)$ be a hypergraph and let $(V_1, \ldots, V_k)$ be a maximal minimum k-partition in G for an integer $k \geq 2$. Suppose $|V_1| \geq 2k-2$. Then, for every subset $T \subseteq \overline{V_1}$ such that T intersects V_j for every $j \in \{2, \ldots, k\}$, there exists a subset $S \subseteq V_1$ of size $2k-2$ such that $(V_1, \overline{V_1})$ is the source maximal minimum (S,T)-terminal cut.*

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 $(V_1, \ldots, V_k)$. For a subset $X \subseteq V_1$, let $(V_X, \overline{V_X})$ be the source maximal minimum (X,T)-terminal cut.

Among all possible subsets of V_1 of size $2k-2$, pick a subset S such that $d(V_S)$ is maximum. By Lemma 4 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

$d(V_S) < d(V_1)$. Let $u_1, \ldots, u_{2k-2}$ be the vertices in S. Since $V_S \subsetneq V_1$, there exists a vertex $u_{2k-1} \in V_1 \setminus V_S$. Let $C := \{u_1, \ldots, u_{2k-1}\} = S \cup \{u_{2k-1}\}$. For $i \in [2k-1]$, let $(B_i, \overline{B_i})$ be the source maximal minimum $(C - \{u_i\}, T)$-terminal cut. We note that $(B_{2k-1}, \overline{B_{2k-1}}) = (V_S, \overline{V_S})$ and the size of $C - \{u_i\}$ is $2k-2$ for every $i \in [2k-1]$. By Lemma 4 and assumption, we have that $B_i \subsetneq V_1$ for every $i \in [2k-1]$. Hence, we have

$$d(B_i) \le d(V_S) < d(V_1) \text{ and } B_i \subsetneq V_1 \text{ for every } i \in [2k-1]. \tag{3}$$

The next claim will set us up to apply Theorem 4.

Claim 4. *For every $i \in [2k-1]$, we have that $u_i \in \overline{B_i}$.*

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

$$d(V_S \cap B_i) \ge d(V_S).$$

We also have that

$$d(V_S \cup B_i) \ge d(V_S)$$

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

$$\begin{aligned} 2d(V_S) &\ge d(V_S) + d(B_i) && \text{(By choice of } S\text{)} \\ &\ge d(V_S \cup B_i) + d(V_S \cap B_i) && \text{(By submodularity)} \\ &\ge 2d(V_S). \end{aligned}$$

Therefore, $d(V_S) = d(V_S \cup B_i)$. Moreover, $B_i \setminus V_S$ is non-empty since the vertex $u_{2k-1} \in B_i \setminus V_S$. Hence, 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 $u_{2k-1} \in B_i$ and $u_{2k-1} \notin V_S$. ∎

We note that for every $i \in [2k-1]$, 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 4. We consider $U := V_1$, $R := \{u_{2k-1}\} \subseteq U$, $S = \{u_1, \ldots, u_{2k-2}\} \subseteq U \setminus R$. Let $p := 2k-2$ and let $(\overline{A_i}, A_i) := (B_i, \overline{B_i})$ for every $i \in [p]$. The 2-partition $(\overline{A_i}, A_i)$ is a minimum $((S \cup R) \setminus \{u_i\}, \overline{U})$-terminal cut for every $i \in [p]$. By Claim 4, we have that $u_i \in A_i$ for every $i \in [p]$. 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 $i, j \in [p]$. Thus, $u_i \in A_i \setminus (\cup_{j \in [p] \setminus \{i\}} A_j)$ for every $i \in [p]$. Therefore, the sets U, R, S and the 2-partitions $(\overline{A_i}, A_i)$ for $i \in [p]$ satisfy the conditions of Theorem 4. By Theorem 4, symmetry of the cut function, and statement (3), we obtain a k-partition $(P_1, \ldots, P_k)$ of V such that

$$\begin{aligned} \text{cost}(P_1, \ldots, P_k) &\le \frac{1}{2} \min\{d(A_i) + d(A_j) : i, j \in [p], i \ne j\} \\ &= \frac{1}{2} \min\{d(B_i) + d(B_j) : i, j \in [p], i \ne j\} \\ &< d(V_1) \le OPT_k. \end{aligned}$$

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

∎

Acknowledgment

We thank Kristóf Bérczi, Tamás Király, and Chao Xu for helpful discussions on alternative approaches for deterministic hypergraph k-cut during preliminary stages of this work. We also thank Sagemath (www.sagemath.org) and CoCalc (www.cocalc.com) for providing software platforms to conduct experiments that led us towards our main structural theorem.

References

[1] O. Goldschmidt and D. S. Hochbaum, "Polynomial algorithm for the k-cut problem," in *Proceedings of the 29th Annual Symposium on Foundations of Computer Science*, 1988, pp. 444–451.

[2] O. Goldschmidt and D. Hochbaum, "A Polynomial Algorithm for the k-cut Problem for Fixed k," *Mathematics of Operations Research*, vol. 19, no. 1, pp. 24–37, Feb 1994.

[3] K. Chandrasekaran, C. Xu, and X. Yu, "Hypergraph k-cut in randomized polynomial time," *Mathematical Programming (Preliminary version in SODA 2018)*, Nov 2019.

[4] H. Nagamochi and T. Ibaraki, "A linear-time algorithm for finding a sparse k-connected spanning subgraph of a k-connected graph," *Algorithmica*, vol. 7, no. 1-6, pp. 583–596, 1992.

[5] M. Stoer and F. Wagner, "A simple min-cut algorithm," *Journal of the ACM (JACM)*, vol. 44, no. 4, pp. 585–591, 1997.

[6] D. Karger and C. Stein, "A new approach to the minimum cut problem," *Journal of the ACM*, vol. 43, no. 4, pp. 601–640, Jul. 1996.

[7] D. Karger, "Minimum cuts in near-linear time," *Journal of the ACM*, vol. 47, no. 1, pp. 46–76, 2000.

[8] Y. Kamidoi, N. Yoshida, and H. Nagamochi, "A Deterministic Algorithm for Finding All Minimum k-Way Cuts," *SIAM Journal on Computing*, vol. 36, no. 5, pp. 1329–1341, 2007.

[9] A. Gupta, E. Lee, and J. Li, "The Karger-Stein Algorithm is Optimal for k-cut," Preprint in arXiv: 1911.09165, 2019.

[10] J. Li, "Faster minimum k-cut of a simple graph," in *Proceedings of the 60th Annual Symposium on Foundations of Computer Science*, ser. FOCS, 2019, pp. 1056–1077.

[11] M. Thorup, "Minimum k-way Cuts via Deterministic Greedy Tree Packing," in *Proceedings of the 40th Annual ACM Symposium on Theory of Computing*, ser. STOC, 2008, pp. 159–166.

[12] C. Chekuri, K. Quanrud, and C. Xu, "LP Relaxation and Tree Packing for Minimum k-cuts," in *2nd Symposium on Simplicity in Algorithms*, ser. SOSA, 2019, pp. 7:1–7:18.

[13] L. Zhao, H. Nagamochi, and T. Ibaraki, "Greedy splitting algorithms for approximating multiway partition problems," *Mathematical Programming*, vol. 102, no. 1, pp. 167–183, 2005.

[14] K. Okumoto, T. Fukunaga, and H. Nagamochi, "Divide-and-conquer algorithms for partitioning hypergraphs and submodular systems," *Algorithmica*, vol. 62, no. 3, pp. 787–806, 2012.

[15] M. Xiao, "An Improved Divide-and-Conquer Algorithm for Finding All Minimum k-Way Cuts," in *Proceedings of 19th International Symposium on Algorithms and Computation*, ser. ISAAC, 2008, pp. 208–219.

[16] M. Queyranne, "On optimum size-constrained set partitions," 1999, talk at Aussiois workshop on Combinatorial Optimization.

[17] F. Guinez and M. Queyranne, "The size-constrained submodular k-partition problem," 2012, unpublished manuscript. See also https://smartech.gatech.edu/bitstream/handle/1853/43309/Queyranne.pdf.

[18] E. Lawler, "Cutsets and Partitions of Hypergraphs," *Networks*, vol. 3, pp. 275–285, 1973.

[19] C. Chekuri and C. Xu, "Minimum cuts and sparsification in hypergraphs," *SIAM Journal on Computing*, vol. 47, no. 6, pp. 2118–2156, 2018.

[20] T. Fukunaga, "Computing minimum multiway cuts in hypergraphs," *Discrete Optimization*, vol. 10, no. 4, pp. 371–382, 2013.

[21] 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*, ser. SODA, 2019, pp. 881–896.

[22] 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*, ser. SODA, 2017, p. 1101–1114.

[23] H. Saran and V. Vazirani, "Finding k Cuts within Twice the Optimal," *SIAM Journal on Computing*, vol. 24, no. 1, pp. 101–108, 1995.

[24] K. Quanrud, "Fast and Deterministic Approximations for k-Cut," in *Proceedings of Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, ser. APPROX, 2019, pp. 23:1–23:20.

[25] P. Manurangsi, "Inapproximability of Maximum Biclique Problems, Minimum k-Cut and Densest At-Least-k-Subgraph from the Small Set Expansion Hypothesis," in *Proceedings of the 44th International Colloquium on Automata, Languages, and Programming*, ser. ICALP, 2017, pp. 79:1–79:14.

[26] P. Raghavendra and D. Steurer, "Graph Expansion and the Unique Games Conjecture," in *Proceedings of the 42nd Annual ACM Symposium on Theory of Computing*, ser. STOC, 2010, pp. 755–764.

[27] C. Chekuri and S. Li, "A note on the hardness of approximating the k-way Hypergraph Cut problem," Manuscript, http://chekuri.cs.illinois.edu/papers/hypergraph-kcut.pdf, 2015.

[28] P. Manurangsi, "Almost-polynomial Ratio ETH-hardness of Approximating Densest k-subgraph," in *Proceedings of the 49th Annual ACM Symposium on Theory of Computing*, ser. STOC, 2017, pp. 954–961.

[29] N. Buchbinder, R. Schwartz, and B. Weizman, "A simple algorithm for the multiway cut problem," *Operations Research Letters*, vol. 47, no. 6, pp. 587–593, 2019.

[30] C. Chekuri and A. Ene, "Approximation Algorithms for Submodular Multiway Partition," in *Proceedings of the 52nd IEEE Annual Symposium on Foundations of Computer Science*, ser. FOCS, 2011, pp. 807–816.

[31] A. Ene, J. Vondrák, and Y. Wu, "Local distribution and the symmetry gap: Approximability of multiway partitioning problems," in *Proceedings of the 24th annual ACM-SIAM symposium on Discrete algorithms*, ser. SODA, 2013, pp. 306–325.

[32] R. Downey, V. Estivill-Castro, M. Fellows, E. Prieto, and F. Rosamund, "Cutting up is hard to do: The parameterised complexity of k-cut and related problems," *Electronic Notes in Theoretical Computer Science*, vol. 78, pp. 209–222, 2003.

[33] K. Kawarabayashi and M. Thorup, "The minimum k-way cut of bounded size is fixed-parameter tractable," in *Proceedings of the 52nd Annual Symposium on Foundations of Computer Science*, ser. FOCS, 2011, pp. 160–169.

[34] D. Marx, "Parameterized graph separation problems," *Theoretical Computer Science*, vol. 351, no. 3, pp. 394–406, 2006.