

Deterministic enumeration of all minimum cut-sets and k -cut-sets in hypergraphs for fixed k^*

Calvin Beideman Karthekeyan Chandrasekaran Weihang Wang

Abstract

We consider the problem of deterministically enumerating all minimum k -cut-sets in a given hypergraph for any fixed k . The input here is a hypergraph $G = (V, E)$ with non-negative hyper-edge costs. A subset $F \subseteq E$ of hyperedges is a k -cut-set if the number of connected components in $G - F$ is at least k and it is a minimum k -cut-set if it has the least cost among all k -cut-sets. For fixed k , we call the problem of finding a minimum k -cut-set as HYPERGRAPH- k -CUT and the problem of enumerating all minimum k -cut-sets as ENUM-HYPERGRAPH- k -CUT. The special cases of HYPERGRAPH- k -CUT and ENUM-HYPERGRAPH- k -CUT restricted to graph inputs are well-known to be solvable in (randomized as well as deterministic) polynomial time [17,25,28,38]. In contrast, it is only recently that polynomial-time algorithms for HYPERGRAPH- k -CUT were developed [2,3,12]. The randomized polynomial-time algorithm for HYPERGRAPH- k -CUT that was designed in 2018 [3] showed that the number of minimum k -cut-sets in a hypergraph is $O(n^{2k-2})$, where n is the number of vertices in the input hypergraph, and that they can all be enumerated in randomized polynomial time, thus resolving ENUM-HYPERGRAPH- k -CUT in randomized polynomial time. A deterministic polynomial-time algorithm for HYPERGRAPH- k -CUT was subsequently designed in 2020 [2], but it is not guaranteed to enumerate all minimum k -cut-sets. In this work, we give the first deterministic polynomial-time algorithm to solve ENUM-HYPERGRAPH- k -CUT (this is non-trivial even for $k = 2$). Our algorithm is based on new structural results that allow for efficient recovery of all minimum k -cut-sets by solving minimum (S, T) -terminal cuts. Our techniques give new structural insights even for enumerating all minimum cut-sets (i.e., minimum 2-cut-sets) in a given hypergraph.

*University of Illinois, Urbana-Champaign. Email: {calvinb2, karthe, weihang3}@illinois.edu. Supported in part by NSF grants CCF-1814613 and CCF-1907937.

1 Introduction

A hypergraph $G = (V, E)$ consists of a finite set V of vertices and a finite set E of hyperedges where each hyperedge $e \in E$ is a subset of V . We consider the problem of enumerating all optimum solutions to the HYPERGRAPH- k -CUT problem when k is a fixed constant. In HYPERGRAPH- k -CUT, the input consists of a hypergraph $G = (V, E)$ with non-negative hyperedge-costs $c : E \rightarrow \mathbb{R}_+$ 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. We will call a subset of hyperedges whose removal results in at least k connected components as a k -cut-set and a minimum-cost k -cut-set as a *minimum k -cut-set*; for $k = 2$, we will refer to a 2-cut-set as simply a *cut-set* and a minimum-cost cut-set as a *minimum cut-set*. The central problem of interest to this work is that of enumerating all minimum k -cut-sets in a given hypergraph with non-negative hyperedge-costs—we will denote this problem as ENUM-HYPERGRAPH- k -CUT. Throughout, we will consider k to be a fixed constant integer (e.g., $k = 2, 3, 4, \dots$). We will denote HYPERGRAPH- k -CUT and ENUM-HYPERGRAPH- k -CUT for graph inputs as GRAPH- k -CUT and ENUM-GRAPH- k -CUT respectively. We note that the case of $k = 2$ corresponds to global minimum cut which will be discussed shortly.

Partitioning formulation. There is a fundamental structural difference between HYPERGRAPH- k -CUT and GRAPH- k -CUT (even for $k = 2$), which is especially evident when attempting to enumerate all optimum solutions. In order to illustrate this difference, we discuss an equivalent partitioning formulation of HYPERGRAPH- k -CUT. In this equivalent formulation, the objective is to find a partition of the vertex set V into k non-empty sets V_1, V_2, \dots, V_k so as to minimize the cost of hyperedges that cross the partition. A hyperedge $e \in E$ is said to *cross a partition* V_1, V_2, \dots, V_k if it has vertices in at least two parts, that is, there exist distinct $i, j \in [k]$ such that $e \cap V_i \neq \emptyset$ and $e \cap V_j \neq \emptyset$. We will denote a partition of V into k non-empty parts as a k -partition and a 2-partition as a *cut*. The cost of a k -partition is the sum of the cost of hyperedges crossing the partition. A k -partition with minimum cost is said to be a *minimum k -partition*. We will denote the cost of a 2-partition as its cut value and a minimum 2-partition as a minimum cut.

By definition, the number of minimum k -cut-sets is at most the number of minimum k -partitions. Moreover, for a connected graph, the number of minimum k -partitions is $O(n^k)$, where n is the number of vertices (i.e., the number of minimum k -partitions is polynomial since k is a constant) [18, 22, 28]. However, for a connected¹ hypergraph, the number of minimum k -partitions could be exponential while the number of minimum k -cut-sets is only polynomial. For example, consider the *spanning-hyperedge-example*: this is the n -vertex hypergraph $G = (V, E)$ that consists of only one hyperedge e where $e = V$ with the cost of the hyperedge e being one. This hypergraph is connected and has only one minimum k -cut-set but $\Theta(k^n)$ minimum k -partitions (i.e., an exponential number of minimum k -partitions even for $k = 2$). Thus, if we are hoping for polynomial-time algorithms to enumerate all optimum solutions to HYPERGRAPH- k -CUT, then we cannot aim to enumerate all minimum k -partitions (in contrast to connected graphs). This is the reason for defining ENUM-HYPERGRAPH- k -CUT as the problem of enumerating all minimum k -cut-sets as opposed to enumerating all minimum k -partitions. For connected graphs, the two definitions are indeed equivalent.

GRAPH- k -CUT for $k = 2$ is the minimum cut problem in graphs which is well-known to be solvable in polynomial time. Although the minimum cut problem in graphs has been extensively studied, enumerating all minimum cut-sets in a graph in deterministic polynomial time is already non-trivial. Dinitz, Karzanov, and Lomonosov [11] constructed a compact representation for all minimum cuts in a connected graph (known as the *cactus representation*) which showed that the

¹A hypergraph is defined to be *connected* if the every cut has at least one hyperedge crossing it.

number of minimum cuts in a connected graph is at most $\binom{n}{2}$ and that they can all be enumerated in deterministic polynomial time. For $k \geq 3$, the number of minimum k -partitions in a connected graph is $O(n^k)$ —this bound is tight and is a consequence of a recent improved analysis of a random contraction algorithm to solve GRAPH- k -CUT [18, 22, 28]; the same random contraction algorithm can also be used to enumerate all minimum k -partitions in connected graphs in randomized polynomial time. Deterministic polynomial-time algorithms to enumerate all minimum k -partitions in connected graphs are also known. We discuss other techniques—both randomized and deterministic—for enumerating minimum cuts and minimum k -partitions in graphs in Section 1.2.

HYPERGRAPH- k -CUT is a natural generalization of GRAPH- k -CUT. HYPERGRAPH- k -CUT for $k = 2$ is the minimum cut problem in hypergraphs which is well-known to be solvable in polynomial time [29]. Once again, enumerating all minimum cut-sets in a hypergraph in deterministic polynomial-time is already non-trivial. There exists a compact representation of all minimum cut-sets in a hypergraph [6]—namely the *hypercactus representation*—which also implies that the number of minimum cut-sets in a hypergraph is at most $\binom{n}{2}$ and that they can all be enumerated in deterministic polynomial time. To the best of the authors’ knowledge, this is the only known technique for efficient deterministic enumeration of all minimum cut-sets in a hypergraph.

HYPERGRAPH- k -CUT is a special case of SUBMODULAR- k -PARTITION (e.g., see [2, 35, 41, 42]). Owing to this connection, the complexity of HYPERGRAPH- k -CUT for any fixed $k \geq 3$ has been an intriguing open question until recently. A randomized polynomial-time algorithm for HYPERGRAPH- k -CUT was designed in 2018 by Chandrasekaran, Xu, and Yu [3]. The analysis of this algorithm showed that the number of minimum k -cut-sets is $O(n^{2k-2})$, where n is the number of vertices in the input hypergraph (i.e., the number of minimum k -cut-sets is polynomial), and that they can all be enumerated in randomized polynomial time (also see [12]). Subsequently, Chandrasekaran and Chekuri designed a deterministic polynomial-time algorithm for HYPERGRAPH- k -CUT in 2020 [2]. However, their deterministic algorithm is guaranteed to identify only one minimum k -cut-set and not all. The next natural question is whether all minimum k -cut-sets can be enumerated in deterministic polynomial time—namely, can we solve ENUM-HYPERGRAPH- k -CUT in deterministic polynomial time?

As mentioned earlier, the only known technique for ENUM-HYPERGRAPH- k -CUT for $k = 2$ is via the hypercactus representation which does not seem to generalize to $k \geq 3$ (in fact, it is unclear if cactus representation generalizes to $k \geq 3$ even in graphs). Moreover, all deterministic techniques for ENUM-GRAPH- k -CUT address the problem of enumerating all minimum k -partitions in connected graphs—see Section 1.2; hence, all these techniques fail for ENUM-HYPERGRAPH- k -CUT (as seen from the spanning-hyperedge-example). For hypergraphs, we necessarily have to work with minimum k -cut-sets as opposed to minimum k -partitions. Working with minimum k -cut-sets as opposed to minimum k -partitions in the deterministic setting is a technical challenge that has not been undertaken in any of the previous works (even for graphs). We overcome this technical challenge in this work. We adapt Chandrasekaran and Chekuri’s deterministic approach for HYPERGRAPH- k -CUT and augment it with structural results for minimum k -cut-sets to prove our main result stated below.

Theorem 1.1. *There is a deterministic polynomial-time algorithm for ENUM-HYPERGRAPH- k -CUT for every fixed k .*

Although we chose to highlight the above algorithmic result in this introduction, we emphasize that the structural theorems that form the backbone of the algorithmic result are our main technical contributions (see Theorems 1.2 and 1.3). We discuss these structural theorems in the technical overview section. By tightening the proof technique of one of our structural theorems for $k = 2$,

we obtain an arguably elegant structural explanation for the number of minimum cut-sets in a hypergraph being at most $\binom{n}{2}$ —see Theorem 1.4. Theorem 1.4 leads to an alternative deterministic polynomial-time algorithm to enumerate all minimum cut-sets in a hypergraph (that is relatively simpler than computing a hypercactus representation). We believe that our structural theorems are likely to be of independent interest.

1.1 Technical overview and main structural results

We focus on the unit-cost variant of ENUM-HYPERGRAPH- k -CUT in the rest of this work for the sake of notational simplicity. 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.

A key algorithmic tool will be the use of terminal cuts. We need some notation. Let $G = (V, E)$ be a hypergraph. Throughout this work, n will denote the number of vertices 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, \dots, V_h) . For a non-empty proper subset U of vertices, we will use \bar{U} to denote $V \setminus U$, $\delta(U)$ to denote the set of hyperedges crossing the 2-partition (U, \bar{U}) , and $d(U) := |\delta(U)|$. We recall that $\delta(U) = \delta(\bar{U})$, so we will use $d(U)$ to denote the cost of the cut (U, \bar{U}) . More generally, given a partition $\mathcal{P} = (V_1, V_2, \dots, V_h)$, we denote the set of hyperedges crossing the partition by $\delta(V_1, V_2, \dots, V_h)$ (also by $\delta(\mathcal{P})$ for brevity) and the number of hyperedges crossing the partition by $\text{cost}(V_1, V_2, \dots, V_h) := |\delta(V_1, V_2, \dots, V_h)|$ (also by $\text{cost}(\mathcal{P})$ for brevity). Let S, T be disjoint non-empty subsets of vertices. A 2-partition (U, \bar{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 \bar{U} is known as the sink set. A minimum-cost (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 and *source maximal* 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; a similar result holds for source maximal minimum (S, T) -terminal cuts.

Our algorithm is inspired by the divide and conquer approach introduced by Goldschmidt and Hochbaum for GRAPH- k -CUT [17]. This approach was generalized by Kamidoi, Yoshida, and Nagamochi to solve ENUM-GRAPH- k -CUT [25] and by Chandrasekaran and Chekuri to solve HYPERGRAPH- k -CUT [2], both in deterministic polynomial time. The techniques of [17] and [25] are not applicable to ENUM-HYPERGRAPH- k -CUT since they are tailored to graphs and do not extend to hypergraphs. We describe the details of the divide and conquer approach for HYPERGRAPH- k -CUT due to Chandrasekaran and Chekuri [2]. The goal here is to identify one part of some fixed minimum k -partition (V_1, V_2, \dots, V_k) , say V_1 without loss of generality, and then recursively find a minimum $(k - 1)$ -partition in the subhypergraph $G[\bar{V}_1]$, where $G[\bar{V}_1]$ is the hypergraph obtained from G by discarding the vertices in V_1 and by discarding all hyperedges that intersect V_1 . Now, how does one find such a part V_1 ? Chandrasekaran and Chekuri proved a key structural theorem for this: Suppose (V_1, \dots, V_k) is a V_1 -*maximal* minimum k -partition—i.e., there is no other minimum k -partition (V'_1, \dots, V'_k) such that V_1 is a proper subset of V'_1 . Then, they showed that for every subset $T \subseteq \bar{V}_1$ such that $T \cap V_j \neq \emptyset$ for all $j \in \{2, \dots, k\}$, there exists a subset $S \subseteq V_1$ of size at most $2k - 2$ such that (V_1, \bar{V}_1) is the source maximal minimum (S, T) -terminal cut. A consequence of this structural theorem is that if we compute the collection \mathcal{C} consisting of the source side of the source maximal minimum (S, T) -terminal cut for all possible pairs (S, T) of disjoint subsets of vertices S and T with $|S| \leq 2k - 2$ and $|T| \leq k - 1$, then the set V_1 will be in this collection \mathcal{C} (by applying the

structural theorem to a set T of size $k - 1$ with $|T \cap V_j| = 1$ for all $j \in \{2, \dots, k\}$). Moreover, the size of the collection \mathcal{C} is only $O(n^{3k-3})$. Hence, recursing on $G[\bar{U}]$ for each set U in the collection \mathcal{C} will identify a minimum k -partition within a total run-time of $n^{O(k^2)}$ source maximal minimum (S, T) -terminal cut computations.

The limitation of the structural theorem of Chandrasekaran and Chekuri [2] is that it aims to recover a minimum k -partition and in particular, a V_1 -maximal minimum k -partition. For the purposes of enumerating all minimum k -cut-sets, this is insufficient as we have seen from the spanning-hyperedge-example. In particular, their structural theorem cannot be used to even enumerate all minimum cut-sets in a hypergraph. We prove two structural theorems that will help in enumerating minimum k -cut-sets. We describe these structural theorems now.

Our goal is to deterministically enumerate a polynomial-sized family \mathcal{F} of k -cut-sets such that \mathcal{F} contains all minimum k -cut-sets. Let F be an arbitrary minimum k -cut-set. Since F is a minimum k -cut-set, there exists a minimum k -partition (V_1, \dots, V_k) such that $F = \delta(V_1, \dots, V_k)$. We note that $d(V_1) \leq |F|$ by definition of the hypergraph cut function $d : 2^V \rightarrow \mathbb{R}$. We distinguish two cases:

Case 1. Suppose $d(V_1) < |F|$. In order to identify minimum k -cut-sets F that have this property, we show the following structural theorem.

Theorem 1.2. *Let $G = (V, E)$ be a hypergraph and let OPT_k be the value of a minimum k -cut-set in G for some integer $k \geq 2$. Suppose (U, \bar{U}) is a 2-partition of V with $d(U) < OPT_k$. Then, for every pair of vertices $s \in U$ and $t \in \bar{U}$, there exist subsets $S \subseteq U \setminus \{s\}$ and $T \subseteq \bar{U} \setminus \{t\}$ with $|S| \leq 2k - 3$ and $|T| \leq 2k - 3$ such that (U, \bar{U}) is the unique minimum $(S \cup \{s\}, T \cup \{t\})$ -terminal cut in G .*

The advantage of this structural theorem is that it allows for a recursive approach to enumerate a polynomial-sized family of minimum k -cut-sets containing F under the assumption that $d(V_1) < |F| = OPT_k$ (similar to the approach of Chandrasekaran and Chekuri).

The drawback of this structural theorem is that it only addresses the case of $d(V_1) < |F|$. It is possible that the minimum k -cut-set F satisfies $d(V_1) = |F|$. For example, consider the problem of enumerating all minimum cut-sets in a hypergraph (i.e., ENUM-HYPERGRAPH- k -CUT for $k = 2$)—Theorem 1.2 does not help in this case since there will be no cut (U, \bar{U}) with $d(U) < OPT_2$. This motivates the second case.

Case 2. Suppose $d(V_1) = |F|$. In this case, we need to enumerate a polynomial-sized family of k -cut-sets containing F , but we cannot hope to enumerate all minimum k -partitions (V'_1, \dots, V'_k) for which $F = \delta(V'_1, \dots, V'_k)$ (e.g., again consider the spanning-hyperedge-example for $k = 2$ for which the unique minimum cut-set F has $|F| = d(V_1)$ for exponentially many minimum cuts (V_1, V_2) and hence, we cannot hope to enumerate all minimum cuts). We observe that if $d(V_1) = |F|$, then the set F of hyperedges should be equal to the set of hyperedges crossing (V_1, \bar{V}_1) , i.e., $\delta(V_1) = F = \delta(V_1, \dots, V_k)$. We show the following structural theorem to exploit this observation.

Theorem 1.3. *Let $G = (V, E)$ be a hypergraph, $k \geq 2$ be an integer, and $\mathcal{P} = (V_1, \dots, V_k)$ be a minimum k -partition such that $\delta(V_1) = \delta(\mathcal{P})$. Then, for all subsets $T \subseteq \bar{V}_1$ such that $T \cap V_j \neq \emptyset$ for all $j \in \{2, 3, \dots, k\}$, there exists a subset $S \subseteq V_1$ with $|S| \leq 2k - 1$ such that the source minimal minimum (S, T) -terminal cut (A, \bar{A}) satisfies $\delta(A) = \delta(V_1)$ and $A \subseteq V_1$.*

We recall that for fixed disjoint subsets $S, T \subseteq V$, the source minimal minimum (S, T) -terminal cut is unique. We emphasize the main feature of Theorem 1.3: it aims to recover only the hyperedges crossing the cut (V_1, \bar{V}_1) but not the cut (V_1, \bar{V}_1) itself. It shows the existence of a small-sized witness which allows us to recover $\delta(V_1)$ —namely a pair (S, T) with $|S|, |T| = O(k)$ for which $\delta(V_1)$ is the

cut-set of the source minimal minimum (S, T) -terminal cut. In this sense, Theorem 1.3 addresses the drawback of Theorem 1.2.

Theorems 1.2 and 1.3 can be used to design a recursive algorithm that enumerates all minimum k -cut-sets in deterministic polynomial time (along the lines of the algorithm of Chandrasekaran and Chekuri described above). Here, we describe a more straightforward non-recursive deterministic polynomial-time algorithm. For each pair of subsets of vertices S, T of size at most $2k - 1$, we compute the source minimal minimum (S, T) -terminal cut $V_{S, T}$; if $G - \delta(V_{S, T})$ has at least k connected components, then we add $\delta(V_{S, T})$ to the candidate family \mathcal{F} ; otherwise, we add $V_{S, T}$ to the collection \mathcal{C} . Next, we consider all possible k -partitions (U_1, \dots, U_k) of the vertex set where all sets U_1, \dots, U_k are in the collection \mathcal{C} and add the set $\delta(U_1, \dots, U_k)$ of hyperedges to the family \mathcal{F} . We now sketch the argument to show that the family \mathcal{F} contains the (arbitrary) minimum k -cut-set F . Recall that there exists a minimum k -partition (V_1, \dots, V_k) such that F is the set of hyperedges crossing this k -partition, i.e., $F = \delta(V_1, \dots, V_k)$. We have two possibilities: (1) if $d(V_i) < |F|$ for every $i \in [k]$, then by Theorem 1.2, every set V_i is in the collection \mathcal{C} (by applying Theorem 1.2 to $(U = V_i, \bar{U} = \bar{V}_i)$ and arbitrary vertices $s \in V_i, t \in \bar{V}_i$), and hence $F \in \mathcal{F}$; (2) if $d(V_i) = |F|$ for some $i \in [k]$, then by Theorem 1.3, one of the sets $V_{S, T}$ has $\delta(V_{S, T}) = \delta(V_i) = F$ and hence, once again $F \in \mathcal{F}$. We can prune the family \mathcal{F} to return the subfamily of minimum k -cut-sets in it. The size of the collection \mathcal{C} is $O(n^{4k-2})$ and the size of the family \mathcal{F} is $O(n^{4k^2})$. The run-time is $O(n^{4k-2})T(n, p) + O(n^{4k^2})$, where $T(n, p)$ is the time complexity for computing the source minimal minimum (s, t) -terminal cut in a n -vertex hypergraph of size p .

Additional consequence of Theorem 1.3. Theorem 1.3 is the technical novelty of this work. We emphasize another structural consequence of Theorem 1.3 by using it to bound the number of minimum cut-sets in a hypergraph. Let t be an arbitrary vertex in the hypergraph $G = (V, E)$. Consider the sets

$$\begin{aligned} \mathcal{H} &:= \{U \subseteq V \setminus \{t\} : (U, \bar{U}) \text{ is a minimum cut in } G\} \text{ and} \\ \mathcal{M} &:= \{\delta(U) : U \in \mathcal{H}\}. \end{aligned}$$

We note that \mathcal{M} is the family of all minimum cut-sets in the hypergraph. By applying Theorem 1.3 for $k = 2$ and $T = \{t\}$, we obtain that for every set $U \in \mathcal{H}$, there exists a subset $S \subseteq U$ with $|S| \leq 3$ such that the source minimal minimum $(S, \{t\})$ -terminal cut (A, \bar{A}) satisfies $\delta(A) = \delta(U)$. Consequently, the size of the set \mathcal{M} is at most the number of possible ways to choose a non-empty subset $S \subseteq V \setminus \{t\}$ of size at most 3 which is $\binom{n-1}{1} + \binom{n-1}{2} + \binom{n-1}{3} = O(n^3)$, where $n := |V|$. Thus, we have concluded that the number of minimum cut-sets in a n -vertex hypergraph is $O(n^3)$.

We recall that the number of minimum cut-sets in a n -vertex hypergraph is known to be at most $\binom{n}{2}$ [6, 15]. So, the $O(n^3)$ upper bound on the number of minimum cut-sets that we obtained above based on Theorem 1.3 appears to be weak. We show the following strengthening of Theorem 1.3 for $k = 2$ to get the tighter bound.

Theorem 1.4. *Let $G = (V, E)$ be a hypergraph and $\mathcal{P} = (V_1, V_2)$ be a minimum cut. Then, for all non-empty subsets $T \subseteq V_2$, there exists a subset $S \subseteq V_1$ with $|S| \leq 2$ such that the source minimal minimum (S, T) -terminal cut (A, \bar{A}) satisfies $\delta(A) = \delta(V_1)$ and $A \subseteq V_1$.*

By applying Theorem 1.4 for $T = \{t\}$, we obtain that for every set $U \in \mathcal{H}$, there exists a subset $S \subseteq U$ with $|S| \leq 2$ such that the source minimal minimum $(S, \{t\})$ -terminal cut (A, \bar{A}) satisfies $\delta(A) = \delta(U)$. Hence, the size of the set \mathcal{M} is at most the number of possible ways to choose a non-empty subset $S \subseteq V \setminus \{t\}$ of size at most 2 which is $\binom{n-1}{1} + \binom{n-1}{2} = \binom{n}{2}$. Thus, we have obtained a structural explanation (based on Theorem 1.4) for the number of minimum cut-sets in

a hypergraph being at most $\binom{n}{2}$. Theorem 1.4 can also be used to enumerate all minimum cut-sets in a given hypergraph using $\binom{n}{2}$ source minimal minimum (S, T) -terminal cut computations.

Theorem 1.4 should be compared with a similar-looking structural theorem for graphs that was shown by Goemans and Ramakrishnan [16]. Goemans and Ramakrishnan showed that (Theorem 15 in [16]) if G is a connected graph, then for every set $U \in \mathcal{H}$, there exists a subset $S \subseteq V_1$ with $|S| \leq 2$ such that (U, \bar{U}) is the source minimal minimum $(S, \{t\})$ -terminal cut. This leads to a structural explanation for the number of minimum cuts in a connected graph being at most $\binom{n}{2}$. Our Theorem 1.4 can be seen as a counterpart of Goemans and Ramakrishnan’s result for hypergraphs, but it differs from their result in two aspects: (1) their result does not hold for hypergraphs—the number of minimum cuts in a connected hypergraph could be exponential as we have seen from the spanning-hyperedge-example and (2) the proof of their result is based on the *submodular triple inequality* which holds only for the graph cut function but fails for the hypergraph cut function. So, our Theorem 1.4 is more general as it handles minimum cut-sets in hypergraphs and moreover, needs a different proof technique compared to [16]. We mention that Goemans and Ramakrishnan’s result for connected graphs was our inspiration for Theorem 1.4, which in turn, was our starting point for Theorem 1.3.

Organization. We discuss special cases of ENUM-HYPERGRAPH- k -CUT that have been addressed in the literature in Section 1.2. In Section 1.3, we recall properties of the hypergraph cut function that will be useful to prove our structural theorems. This section contains a strengthening of a partition uncrossing theorem from [2] whose proof appears in Appendix A. In Section 2, we formally describe and analyze the deterministic polynomial-time algorithm for ENUM-HYPERGRAPH- k -CUT that utilizes our two structural theorems (Theorems 1.2 and 1.3). We prove Theorems 1.2 and 1.3 in Sections 3 and 4 respectively. We prove the strengthening of Theorem 1.3 for $k = 2$ —namely Theorem 1.4—in Section 5. We conclude with a few open problems in Section 6.

1.2 Related work

In this section, we discuss known techniques for the enumeration problem in the special case of $k = 2$ and the special case of graphs along with challenges involved in adapting these techniques to hypergraphs for $k \geq 3$.

ENUM-GRAPH- k -CUT for $k = 2$. GRAPH- k -CUT for $k = 2$ is the global minimum cut problem (denoted GRAPH-MINCUT) which has been extensively studied. However, efficient deterministic enumeration of all minimum cut-sets in a given connected graph is already non-trivial. Dinitz, Karzanov, and Lomonosov [11] showed that the number of minimum cuts in a connected graph is at most $\binom{n}{2}$, where n is the number of vertices in the input graph, and they can all be enumerated in deterministic polynomial time. In particular, they designed a compact data structure, namely a cactus graph, to represent all minimum cuts in a connected graph. The upper bound of $\binom{n}{2}$ on the number of minimum cuts in a connected graph is tight as illustrated by the cycle-graph on n vertices. Using the seminal random contraction technique, Karger [26] showed a stronger result that the number of α -approximate minimum cuts in a connected graph is $O(n^{2\alpha})$ and they can all be enumerated in randomized polynomial time for constant α . Karger’s tree packing technique [27] also leads to a deterministic polynomial-time algorithm to enumerate all α -approximate minimum cuts in a connected graph for constant α . Nagamochi, Nishimura, and Ibaraki [33] tightened Karger’s bound for a particular value of α via the edge splitting operation: the number of $(4/3 - \epsilon)$ -approximate minimum cuts in a connected graph is at most $\binom{n}{2}$ for any $\epsilon > 0$. This fact was also shown by Goemans and Ramakrishnan [16] via a structural result (see discussion after Theorem 1.4 above). Henzinger and Williamson [24] extended Nagamochi, Nishimura, and Ibaraki’s edge splitting technique to show

that the number of $(3/2-\epsilon)$ -approximate minimum cuts in a connected graph is $O(n^2)$ for any $\epsilon > 0$. The results of Nagamochi, Nishimura, and Ibaraki, Goemans and Ramakrishnan, and Henzinger and Williamson are all constructive and deterministic (i.e., lead to deterministic polynomial-time algorithms to enumerate the respective approximate minimum cuts) and they bound the number of minimum cuts in a connected graph (as opposed to minimum cut-sets).

Polynomial-delay algorithms. An alternative line of work aims to enumerate *all* cuts in hypergraphs in non-decreasing order of cut value with polynomial time delay between outputs. Such algorithms are known as *polynomial-delay* algorithms in the literature. Polynomial-delay algorithms have been designed based on polynomial-time solvability of minimum (s, t) -terminal cut and using the Lawler-Murty schema [1, 23, 32, 39]. Since we know that the number of minimum cuts in a connected graph is polynomial, the existence of a polynomial-delay algorithm immediately implies a polynomial-time algorithm to solve ENUM-GRAPH- k -CUT for $k = 2$. This approach does not extend to ENUM-HYPERGRAPH- k -CUT for $k = 2$ since the number of minimum cuts in a hypergraph can be exponential (e.g., recall the spanning-hyperedge-example).

ENUM-HYPERGRAPH- k -CUT for $k = 2$. HYPERGRAPH- k -CUT for $k = 2$ is the global minimum cut problem (denoted HYPERGRAPH-MINCUT) which has also been extensively studied. We note that the number of minimum cuts in a connected hypergraph could be exponential (e.g., consider the spanning-hyperedge-example). But, how about the number of minimum cut-sets? The number of minimum cut-sets in a hypergraph is at most $\binom{n}{2}$ via decomposition theorems of Cunningham and Edmonds [9], Fujishige [13], and Cunningham [8] on submodular functions. Cheng [7] designed an explicit *hypercactus representation* for all minimum cut-sets in a hypergraph. Chekuri and Xu [6] designed a faster deterministic polynomial-time algorithm to obtain a hypercactus representation (along with all minimum cut-sets) of a given hypergraph. Ghaffari, Karger, and Panigrahi [15] (also see [3, 12]) introduced a random contraction technique to solve HYPERGRAPH-MINCUT which also implied that the number of minimum cut-sets in a hypergraph is at most $\binom{n}{2}$ and that they can all be enumerated in randomized polynomial time.

We mention that in contrast to graphs, the number of constant-approximate minimum cut-sets in a hypergraph can be exponential. In fact, the number of $(1 + \epsilon)$ -approximate minimum cut-sets in a connected hypergraph can be exponential² for any $\epsilon \in (0, 1)$. Moreover, the techniques of Nagamochi, Nishimura, and Ibaraki, Goemans and Ramakrishnan, and Henzinger and Williamson even when restricted to minimum cuts (as opposed to approximate minimum cuts) cannot extend to hypergraphs: This is because, their techniques are tailored to enumerate all minimum cuts in a connected graph as opposed to all minimum cut-sets; we have already seen that the spanning-hyperedge-example has exponential number of minimum cuts and hence, all of them cannot be enumerated in polynomial time.

Multiterminal variants for k -cut: We mention that GRAPH- k -CUT and HYPERGRAPH- k -CUT have natural variants involving separating specified terminal vertices s_1, s_2, \dots, s_k . These variants are NP-hard for $k \geq 3$ even in graphs and hence, these variants are not viable lines of attack for GRAPH- k -CUT and HYPERGRAPH- k -CUT. We refer the reader to [2] for a discussion of approximation algorithms for these variants.

ENUM-GRAPH- k -CUT. GRAPH- k -CUT for $k \geq 3$ has a rich literature with substantial recent work [5, 17–22, 25, 28, 30, 31, 36–38]. Goldschmidt and Hochbaum (1988) [17] initiated the study on GRAPH- k -CUT by showing that it is NP-hard when k is part of the input and that it is polynomial-time solvable when k is any fixed constant (polynomial-time solvability is not obvious even for $k = 3$).

²Consider the n -vertex hypergraph $G = (V, E)$ where E consists of all size-2 hyperedges each of cost $\delta = \epsilon \binom{n}{2} - (1 + \epsilon)(n - 1)^{-1}$ and a hyperedge $e = V$ of cost 1. The cost of a minimum cut is $\lambda := 1 + \delta(n - 1)$. The cost of every cut is at most $1 + \delta \binom{n}{2} \leq (1 + \epsilon)\lambda$.

Recall that we consider k to be a fixed constant throughout this work. Goldschmidt and Hochbaum introduced a divide-and-conquer approach for GRAPH- k -CUT which resulted in a deterministic polynomial-time algorithm. However, their result did not guarantee any bound on the number of minimum k -partitions or minimum k -cut-sets in connected graphs. Karger and Stein [28] gave a randomized polynomial-time algorithm for GRAPH- k -CUT via the random contraction technique. In addition, they showed that the number of minimum k -partitions in a connected graph is $O(n^{2k-2})$ and they can all be enumerated in randomized polynomial time. The bound on the number of minimum k -partitions in a connected graph has recently been improved to $O(n^k)$ [18, 22]. We mention that the upper bound of $O(n^k)$ on the number of minimum k -partitions in a connected graph is tight as illustrated by the cycle-graph on n vertices.

There are two known approaches to solve ENUM-GRAPH- k -CUT in deterministic polynomial time: (1) Thorup [38] showed that the tree packing approach can be used to obtain a polynomial-time algorithm for GRAPH- k -CUT; this approach also extends to solve ENUM-GRAPH- k -CUT (also see [5]). (2) Kamidoi, Yoshida, and Nagamochi [25] extended Goldschmidt and Hochbaum’s divide and conquer approach to solve ENUM-GRAPH- k -CUT.

HYPERGRAPH- k -CUT. The complexity of HYPERGRAPH- k -CUT was open since the work of Goldschmidt and Hochbaum for GRAPH- k -CUT (1988) [17] until recently. Although certain special cases of HYPERGRAPH- k -CUT were known to be solvable in polynomial time [14, 40], considerable progress on HYPERGRAPH- k -CUT happened only in the last 3 years. Chandrasekaran, Xu, and Yu (2018) [3] designed the first randomized polynomial-time algorithm for HYPERGRAPH- k -CUT; 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. Fox, Panigrahi, and Zhang [12] 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 [15] for HYPERGRAPH-MINCUT. These randomized algorithms also imply that the number of minimum k -cut-sets is $O(n^{2k-2})$ and that all of them can be enumerated in randomized polynomial time. Chandrasekaran and Chekuri (2020) [2] designed a deterministic polynomial-time algorithm for HYPERGRAPH- k -CUT via a divide and conquer approach. We emphasize that their algorithm finds a minimum k -partition and did not have the tools to find all minimum k -cut-sets.

A polynomial bound on the number of minimum k -cut-sets along with the existence of a randomized polynomial-time algorithm to enumerate all of them raises the possibility of a deterministic algorithm for ENUM-HYPERGRAPH- k -CUT. As we mentioned earlier, there are two deterministic approaches for ENUM-GRAPH- k -CUT—tree packing and divide-and-conquer. The tree packing approach does not seem to extend to hypergraphs (even for HYPERGRAPH-MINCUT). This leaves the divide-and-conquer approach. Notably, this approach also led to the first deterministic algorithm for HYPERGRAPH- k -CUT in the work of Chandrasekaran and Chekuri [2]. As mentioned earlier, we adapt Chandrasekaran and Chekuri’s divide-and-conquer approach and augment it with structural results for minimum k -cut-sets to prove our main result stated in Theorem 1.1.

1.3 Preliminaries

Let $G = (V, E)$ be a hypergraph. Throughout, we will follow the notation mentioned in the second paragraph of Section 1.1. We will repeatedly rely on the fact that the hypergraph cut function $d : 2^V \rightarrow \mathbb{R}_+$ is symmetric and submodular. We recall that a set function $f : 2^V \rightarrow \mathbb{R}$ is *symmetric* if $f(U) = f(\bar{U})$ for all $U \subseteq V$ 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 a partition uncrossing theorem that is a strengthening of a result from [2]. We state the strengthened version below. See Figure 1 for an illustration of the sets that appear

in the statement of Theorem 1.5. We emphasize that the second conclusion in the statement of Theorem 1.5 is the strengthening. The proof of the second conclusion is similar to the proof of the first conclusion which appears in [2]—we present a proof of both conclusions for the sake of completeness in Appendix A.

Theorem 1.5. *Let $G = (V, E)$ be a hypergraph, $k \geq 2$ be an integer and $\emptyset \neq R \subsetneq U \subsetneq V$. Let $S = \{u_1, \dots, u_p\} \subseteq U \setminus R$ 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, \dots, P_k) of V with $\overline{U} \subsetneq P_k$ such that*

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

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.*

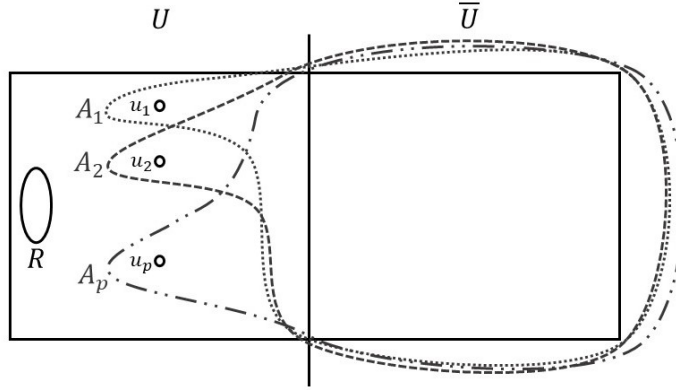


Figure 1: Illustration of the sets that appear in the statement of Theorem 1.5.

2 Enumeration Algorithm

We will use Theorems 1.2 and 1.3 to design a deterministic polynomial-time algorithm for ENUM-HYPERGRAPH- k -CUT in this section. We describe the formal algorithm in Figure 2. It enumerates $n^{O(k)}$ source minimal minimum (S, T) -terminal cuts and considers the cut-set crossing each cut in this collection. If the removal of the cut-set leads to at least k connected components, then it adds such a cut-set to the candidate family \mathcal{F} ; otherwise, it adds the source set of the cut into a candidate collection \mathcal{C} . Next, the algorithm considers all possible k -partitions that can be formed using the sets in the collection \mathcal{C} and adds the set of hyperedges crossing the k -partition to the family \mathcal{F} . Finally, it prunes the family \mathcal{F} to return all minimum k -cut-sets in it. The run-time guarantee and the cardinality of the family of k -cut-sets returned by the algorithm are given in Theorem 2.1. Theorem 1.1 follows from Theorem 2.1 by observing that the source minimal minimum (S, T) -terminal cut in a hypergraph can be computed in deterministic polynomial time—e.g., it can be computed in a n -vertex hypergraph of size p in $O(np)$ time [6].

<p>Algorithm Enum-Cuts($G = (V, E), k$)</p> <p>Input: Hypergraph $G = (V, E)$ and an integer $k \geq 2$</p> <p>Output: Family of all minimum k-cut-sets in G</p> <p>Initialize $\mathcal{C} \leftarrow \emptyset, \mathcal{F} \leftarrow \emptyset$</p> <p>For each pair (S, T) such that $S, T \subseteq V$ with $S \cap T = \emptyset$ and $S , T \leq 2k - 1$</p> <p style="padding-left: 2em;">Compute the source minimal minimum (S, T)-terminal cut (U, \bar{U})</p> <p style="padding-left: 2em;">If $G - \delta(U)$ has at least k connected components</p> <p style="padding-left: 4em;">$\mathcal{F} \leftarrow \mathcal{F} \cup \{\delta(U)\}$</p> <p style="padding-left: 2em;">Else</p> <p style="padding-left: 4em;">$\mathcal{C} \leftarrow \mathcal{C} \cup \{U\}$</p> <p>For each k-partition (U_1, \dots, U_k) of V with $U_1, \dots, U_k \in \mathcal{C}$</p> <p style="padding-left: 2em;">$\mathcal{F} \leftarrow \mathcal{F} \cup \{\delta(U_1, \dots, U_k)\}$</p> <p>Among all k-cut-sets in the family \mathcal{F}, return the subfamily of cheapest ones</p>
--

Figure 2: Algorithm to enumerate hypergraph minimum k -cut-sets

Theorem 2.1. *Let $G = (V, E)$ be a n -vertex hypergraph of size p and let k be an integer. Then, Algorithm Enum-Cuts(G, k) in Figure 2 returns the family of all minimum k -cut-sets in G and it can be implemented to run in $O(n^{4k-2})T(n, p) + O(n^{4k^2-2k}p)$ time, where $T(n, p)$ denotes the time complexity for computing the source minimal minimum (s, t) -terminal cut in a n -vertex hypergraph of size p . Moreover, the cardinality of the family returned by the algorithm is $O(n^{2k(2k-1)})$.*

Proof. We begin by showing correctness. The last step of the algorithm considers only k -cut-sets in the family \mathcal{F} , so the algorithm returns a subfamily of k -cut-sets. We only have to show that every minimum k -cut-set is in the family \mathcal{F} ; this will also guarantee that every k -cut-set in the returned subfamily is indeed a minimum k -cut-set.

Let $F \subseteq E$ be a minimum k -cut-set in G and let (V_1, \dots, V_k) be a minimum k -partition such that $F = \delta(V_1, \dots, V_k)$. We will show that F is in the family \mathcal{F} . We know that $d(V_i) \leq OPT_k$ for every $i \in [k]$. We distinguish two cases:

1. Suppose $d(V_i) < OPT_k$ for every $i \in [k]$.

Consider an arbitrary part V_i where $i \in [k]$. By Theorem 1.2, there exist disjoint subsets $S, T \subseteq V$ with $|S|, |T| \leq 2k - 2$ such that (V_i, \bar{V}_i) is the unique minimum (S, T) -terminal cut. Hence, the set V_i is in the collection \mathcal{C} . Consequently, all parts V_1, \dots, V_k are in the collection \mathcal{C} . Hence, the set $F = \delta(V_1, \dots, V_k)$ is added to the family \mathcal{F} in the second for-loop.

2. Suppose there exists $i \in [k]$ such that $d(V_i) = OPT_k$.

In this case, we have $\delta(V_i) = F = \delta(V_1, \dots, V_k)$. By Theorem 1.3, there exist disjoint subsets $S, T \subseteq V$ with $|S|, |T| \leq 2k - 1$ such that the source minimal minimum (S, T) -terminal cut (A, \bar{A}) satisfies $\delta(A) = \delta(V_i) = F$. Therefore, the set F is added to the family \mathcal{F} in the first for-loop.

Thus, in both cases, we have shown that the set F is contained in the family \mathcal{F} . Since the algorithm returns the subfamily of hyperedge sets in \mathcal{F} that correspond to minimum k -cut-sets, the set F is in the family returned by the algorithm.

Next, we bound the run time and the number of minimum k -cut-sets returned by the algorithm. The first for-loop can be implemented using $O(n^{4k-2})$ source minimal minimum (s, t) -terminal cut computations. Moreover, the size of the collection \mathcal{C} is $O(n^{4k-2})$. The number of tuples $(U_1, \dots, U_k) \in \mathcal{C}^k$ is $O(n^{4k^2-2k})$. Verifying if a tuple (U_1, \dots, U_k) forms a k -partition takes $O(n)$ time. For a tuple which forms a k -partition, computing the hyperedges crossing that partition

takes $O(p)$ time. Thus, the second for-loop can be implemented to run in time $O(n^{4k^2-2k}p)$. The size of the family \mathcal{F} is $O(n^{4k^2-2k})$. Each k -cut-set in \mathcal{F} has representation size at most p . Hence, computing the size of each k -cut-set in \mathcal{F} and returning the cheapest ones can be implemented to run in time $O(n^{4k^2-2k}p)$. Thus, the overall run-time is $O(n^{4k-2})T(n, p) + O(n^{4k^2-2k}p)$. \square

3 Proof of Theorem 1.2

We prove Theorem 1.2 in this section. We will use the following theorem to prove Theorem 1.2.

Theorem 3.1. *Let $G = (V, E)$ be a hypergraph and let OPT_k be the value of a minimum k -cut-set in G for some integer $k \geq 2$. Suppose (U, \bar{U}) is a 2-partition of V with $d(U) < OPT_k$. Then, for every vertex $s \in U$, there exists a subset $S \subseteq U \setminus \{s\}$ with $|S| \leq 2k - 3$ such that (U, \bar{U}) is the unique minimum $(S \cup \{s\}, \bar{U})$ -terminal cut.*

Proof. Let $s \in U$. Consider the collection

$$\mathcal{C} := \{Q \subseteq V \setminus \{s\} : \bar{U} \subsetneq Q, d(Q) \leq d(U)\}.$$

Let S be an inclusion-wise minimal subset of $U \setminus \{s\}$ such that $S \cap Q \neq \emptyset$ for all $Q \in \mathcal{C}$, i.e., the set S is completely contained in $U \setminus \{s\}$ and is a minimal transversal of \mathcal{C} . Proposition 3.1 and Lemma 3.1 complete the proof of Theorem 3.1 for this choice of S . \square

Proposition 3.1. *The 2-partition (U, \bar{U}) is the unique minimum $(S \cup \{s\}, \bar{U})$ -terminal cut.*

Proof. For the sake of contradiction, suppose (Y, \bar{Y}) is a minimum $(S \cup \{s\}, \bar{U})$ -terminal cut with $Y \neq U$. This implies that $S \cup \{s\} \subseteq Y$ and $\bar{U} \subsetneq \bar{Y}$. Moreover, we have $d(\bar{Y}) \leq d(\bar{U})$ because (U, \bar{U}) is a $(S \cup \{s\}, \bar{U})$ -terminal cut. Consequently, the set \bar{Y} is in the collection \mathcal{C} . Since S is a transversal of the collection \mathcal{C} , we have that $S \cap \bar{Y} \neq \emptyset$. This contradicts the fact that S is contained in Y . \square

Lemma 3.1. *The size of the subset S is at most $2k - 3$.*

Proof. For the sake of contradiction, suppose $|S| \geq 2k - 2$. Our proof strategy is to show the existence of a k -partition with cost smaller than OPT_k , thus contradicting the definition of OPT_k . Let $S := \{u_1, u_2, \dots, u_p\}$ for some $p \geq 2k - 2$. For each $i \in [p]$, let (\bar{A}_i, A_i) be the source minimal minimum $((S \cup \{s\}) \setminus \{u_i\}, \bar{U})$ -terminal cut. The following claim will allow us to show that the cuts (\bar{A}_i, A_i) satisfy the hypothesis of Theorem 1.5.

Claim 3.1. *For every $i \in [p]$, we have $d(A_i) \leq d(U)$ and $u_i \in A_i$.*

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, (\bar{B}_i, B_i) is a $((S \cup \{s\}) \setminus \{u_i\}, \bar{U})$ -terminal cut. Therefore,

$$d(A_i) \leq d(B_i) \leq d(U).$$

We will show that A_i is in the collection \mathcal{C} . By definition, $A_i \subseteq V \setminus \{s\}$ and $\bar{U} \subseteq A_i$. If $A_i = \bar{U}$, then the above inequalities are equations implying that (B_i, \bar{B}_i) is a minimum $((S \cup \{s\}) \setminus \{u_i\}, \bar{U})$ -terminal cut, and consequently, (B_i, \bar{B}_i) contradicts source minimality of the minimum $((S \cup \{s\}) \setminus \{u_i\}, \bar{U})$ -terminal cut (A_i, \bar{A}_i) . Therefore, $\bar{U} \subsetneq A_i$. Hence, A_i is in the collection \mathcal{C} .

We recall that the set S is a transversal for the collection \mathcal{C} and moreover, none of the elements of $S \setminus \{u_i\}$ are in A_i by definition of A_i . Therefore, the vertex u_i must be in A_i . \square

Using Claim 3.1, we observe that the sets U , $R := \{s\}$, S , and the partitions $(\overline{A_i}, A_i)$ for $i \in [p]$ satisfy the conditions of Theorem 1.5. By the first conclusion of Theorem 1.5 and Claim 3.1, we obtain a k -partition (P_1, \dots, P_k) of V such that

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

The last inequality above is by the assumption in the theorem statement. Thus, we have obtained a k -partition whose cost is smaller than OPT_k , a contradiction. \square

Applying Theorem 3.1 to (\overline{U}, U) yields the following corollary.

Corollary 3.1. *Let $G = (V, E)$ be a hypergraph and let OPT_k be the value of a minimum k -cut-set in G for some integer $k \geq 2$. Suppose (U, \overline{U}) is a 2-partition of V with $d(U) < OPT_k$. Then, for every vertex $t \in \overline{U}$, there exists a subset $T \subseteq \overline{U} \setminus \{t\}$ with $|T| \leq 2k - 3$ such that (U, \overline{U}) is the unique minimum $(U, T \cup \{t\})$ -terminal cut.*

We now restate Theorem 1.2 and prove it using Theorem 3.1 and Corollary 3.1.

Theorem 1.2. *Let $G = (V, E)$ be a hypergraph and let OPT_k be the value of a minimum k -cut-set in G for some integer $k \geq 2$. Suppose (U, \overline{U}) is a 2-partition of V with $d(U) < OPT_k$. 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 .*

Proof. Let $s \in U$ and $t \in \overline{U}$. By Theorem 3.1, there exists a subset $S \subseteq U \setminus \{s\}$ such that $|S| \leq 2k - 3$ and (U, \overline{U}) is the unique minimum $(S \cup \{s\}, \overline{U})$ -terminal cut. By Corollary 3.1, there exists a subset $T \subseteq \overline{U}$ such that $|T| \leq 2k - 3$ and (U, \overline{U}) is the unique minimum $(U, T \cup \{t\})$ -terminal cut. We now show that (U, \overline{U}) is the unique minimum $(S \cup \{s\}, T \cup \{t\})$ -terminal cut.

We now show that (U, \overline{U}) is the unique minimum $(S \cup \{s\}, T \cup \{t\})$ -terminal cut. Let (Y, \overline{Y}) be a minimum $(S \cup \{s\}, T \cup \{t\})$ -terminal cut. Suppose $Y \neq U$. We have the following observations:

1. Since (U, \overline{U}) is a $(S \cup \{s\}, T \cup \{t\})$ -terminal cut, we have that $d(U) \geq d(Y)$.
2. Since $(U \cap Y, \overline{U \cap Y})$ is a $(S \cup \{s\}, \overline{U})$ -terminal cut, we have that $d(U \cap Y) \geq d(U)$.
3. Since $(U \cup Y, \overline{U \cup Y})$ is a $(U, T \cup \{t\})$ -terminal cut, we have that $d(U \cup Y) \geq d(U)$.

Moreover, since $Y \neq U$, we have that either $U \cap Y \neq U$ or $U \cup Y \neq U$. Since (U, \overline{U}) is the unique minimum $(S \cup \{s\}, \overline{U})$ -terminal cut and also the unique minimum $(U, T \cup \{t\})$ -terminal cut, it follows that either $d(U \cap Y) > d(U)$ or $d(U \cup Y) > d(U)$. These observations in conjunction with the submodularity of the hypergraph cut function imply that

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

a contradiction. Hence, $Y = U$. \square

4 Proof of Theorem 1.3

We prove Theorem 1.3 in this section. We begin with the following useful containment lemma. Variants of this containment lemma have appeared in the literature before under slightly different hypothesis (e.g., see [2, 10, 17, 35]).

Lemma 4.1. *Let $G = (V, E)$ be a hypergraph, $k \geq 2$ be an integer, $\mathcal{P} = (V_1, \dots, V_k)$ be a minimum k -partition such that $\delta(\mathcal{P}) = \delta(V_1)$, and $S \subseteq V_1, T \subseteq \overline{V_1}$ such that $T \cap V_j \neq \emptyset$ for all $j \in \{2, 3, \dots, k\}$. Suppose that (U, \overline{U}) is the source minimal minimum (S, T) -terminal cut. Then, $U \subseteq V_1$ and (U, \overline{U}) is a minimum $(S, \overline{V_1})$ -terminal cut.*

Proof. We note that $S \subseteq U \cap V_1$, so $(U \cap V_1, \overline{U \cap V_1})$ is a (S, T) -terminal cut. Thus, we have

$$d(U \cap V_1) \geq d(U). \quad (1)$$

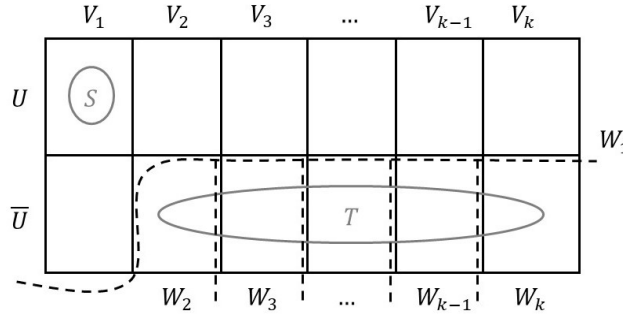


Figure 3: Uncrossing in the proof of Lemma 4.1.

Now consider $\mathcal{P}' = (W_1 := U \cup V_1, W_2 := V_2 \setminus U, \dots, W_k := V_k \setminus U)$ (see Figure 3). For each $i \in \{2, 3, \dots, k\}$, we have $\emptyset \neq T \cap V_i \subseteq V_i \setminus U$, so \mathcal{P}' is a k -partition. Since $\delta(\mathcal{P}) = \delta(V_1)$, every hyperedge which crosses \mathcal{P} must intersect V_1 . Consequently, every hyperedge which crosses \mathcal{P}' must intersect $U \cup V_1$. Therefore

$$d(U \cup V_1) = \text{cost}(\mathcal{P}') \geq \text{cost}(\mathcal{P}) = d(V_1). \quad (2)$$

By submodularity of the hypergraph cut function and inequalities (1) and (2), we have that

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

Therefore, inequality (1) is in fact an equation and hence, $(U \cap V_1, \overline{U \cap V_1})$ is a minimum (S, T) -terminal cut. If $U \setminus V_1 \neq \emptyset$, then $(U \cap V_1, \overline{U \cap V_1})$ contradicts source minimality of the minimum (S, T) -terminal cut (U, \overline{U}) . Hence, $U \setminus V_1 = \emptyset$ and consequently, $U \subseteq V_1$.

Since $U \subseteq V_1$, we have that (U, \overline{U}) is a $(S, \overline{V_1})$ -terminal cut. Furthermore, since $T \subseteq \overline{V_1}$, every $(S, \overline{V_1})$ -terminal cut is also a (S, T) -terminal cut. Therefore, every $(S, \overline{V_1})$ -terminal cut must have weight at least $d(U)$, and hence (U, \overline{U}) is a minimum $(S, \overline{V_1})$ -terminal cut. \square

We now restate and prove Theorem 1.3.

Theorem 1.3. *Let $G = (V, E)$ be a hypergraph, $k \geq 2$ be an integer, and $\mathcal{P} = (V_1, \dots, V_k)$ be a minimum k -partition such that $\delta(V_1) = \delta(\mathcal{P})$. Then, for all subsets $T \subseteq \overline{V_1}$ such that $T \cap V_j \neq \emptyset$ for all $j \in \{2, 3, \dots, k\}$, there exists a subset $S \subseteq V_1$ with $|S| \leq 2k - 1$ such that the source minimal minimum (S, T) -terminal cut (A, \overline{A}) satisfies $\delta(A) = \delta(V_1)$ and $A \subseteq V_1$.*

Proof. Let us fix an arbitrary $T \subseteq \overline{V_1}$ such that $T \cap V_j \neq \emptyset$ for all $j \in \{2, \dots, k\}$. For a subset $X \subseteq V_1$, we denote the source minimal minimum (X, T) -terminal cut by $(H_X, \overline{H_X})$. By Lemma 4.1, for all $X \subseteq V_1$ we have that $H_X \subseteq V_1$ and $d(H_X) \leq d(V_1)$. If $|V_1| \leq 2k - 1$, then choosing $S = V_1$ proves the theorem. So, we will assume henceforth that $|V_1| > 2k - 1$. We will show that there exists a subset $S \subseteq V_1$ with $|S| \leq 2k - 1$ such that the source minimal minimum (S, T) -terminal cut $(H_S, \overline{H_S})$ satisfies $\delta(H_S) = \delta(V_1)$. This suffices since we have that $H_S \subseteq V_1$ for all subsets $S \subseteq V_1$ (by Lemma 4.1).

For the sake of contradiction, suppose that for every $S \subseteq V_1$ with $|S| \leq 2k - 1$, the source minimal minimum (S, T) -terminal cut $(H_S, \overline{H_S})$ does not satisfy $\delta(H_S) = \delta(V_1)$. Our proof strategy is to obtain a cheaper k -partition than (V_1, \dots, V_k) , thereby contradicting the optimality of (V_1, \dots, V_k) .

Let $S \subseteq V_1$ be a set of size $2k - 1$ such that H_S is maximal—i.e., there does not exist $S' \subseteq V_1$ of size $2k - 1$ such that $H_{S'} \supsetneq H_S$. Let $S := \{u_1, u_2, \dots, u_{2k-1}\}$. By assumption, we have that $\delta(H_S) \neq \delta(V_1)$, but since $(V_1, \overline{V_1})$ is a (S, T) -terminal cut, we have that $d(H_S) \leq d(V_1)$. Therefore, $\delta(V_1) \setminus \delta(H_S)$ is non-empty. Let us fix a hyperedge $e \in \delta(V_1) \setminus \delta(H_S)$. Let $u_{2k} \in e \cap V_1$. Let $C := S \cup \{u_{2k}\} = \{u_1, \dots, u_{2k-1}, u_{2k}\}$. For notational convenience we will use $C - u_i$ to denote $C \setminus \{u_i\}$ and $C - u_i - u_j$ to denote $C \setminus \{u_i, u_j\}$ for all $i, j \in [2k]$. The choice of the hyperedge e is crucial to our proof—its properties will be used much later in our proof. We summarize the properties of the hyperedge e here.

Observation 4.1. *The hyperedge e has the following properties:*

1. $e \cap \overline{V_1} \neq \emptyset$,
2. $u_{2k} \in e$, and
3. $e \subseteq \overline{H_S}$.

Our strategy to arrive at a cheaper k -partition than (V_1, \dots, V_k) is to apply the second conclusion of Theorem 1.5. The next few claims will set us up to obtain sets that satisfy the hypothesis of Theorem 1.5.

Claim 4.1. *For every $i \in [2k]$, we have $u_i \notin H_{C-u_i}$.*

Proof. If $i = 2k$, then by Observation 4.1 we have $u_{2k} \in e$ and $e \subseteq \overline{H_S}$ so $u_{2k} \notin H_S = H_{C-u_{2k}}$. Suppose $i \in [2k - 1]$. Our proof will rely on the choice of S .

Suppose for contradiction that $u_i \in H_{C-u_i}$ for some $i \in [2k - 1]$. Then, we have that $S \subseteq H_{C-u_i}$, so $(H_{C-u_i} \cap H_S, \overline{H_{C-u_i} \cap H_S})$ is a (S, T) -terminal cut. Therefore,

$$d(H_{C-u_i} \cap H_S) \geq d(H_S). \quad (3)$$

Also, since $(H_{C-u_i} \cup H_S, \overline{H_{C-u_i} \cup H_S})$ is a $(C - u_i, T)$ -terminal cut, we have that

$$d(H_{C-u_i} \cup H_S) \geq d(H_{C-u_i}). \quad (4)$$

By submodularity of the hypergraph cut function and inequalities (3) and (4), we have that

$$d(H_S) + d(H_{C-u_i}) \geq d(H_{C-u_i} \cap H_S) + d(H_{C-u_i} \cup H_S) \geq d(H_S) + d(H_{C-u_i}).$$

Therefore, inequality (3) is an equation, and consequently, $(H_{C-u_i} \cap H_S, \overline{H_{C-u_i} \cap H_S})$ is a minimum (S, T) -terminal cut. If $H_{C-u_i} \cap H_S \subsetneq H_S$, then $(H_{C-u_i} \cap H_S, \overline{H_{C-u_i} \cap H_S})$ contradicts source minimality of the minimum (S, T) -terminal cut $(H_S, \overline{H_S})$. Therefore $H_{C-u_i} \cap H_S = H_S$ and hence, $H_S \subseteq H_{C-u_i}$. Also, the vertex u_{2k} is in $C - u_i$ but not in H_S and hence, $H_S \subsetneq H_{C-u_i}$. However, $|C - u_i| = 2k - 1$. Therefore, the set $C - u_i$ contradicts the choice of S . \square

The following claim will help in showing that $u_i, u_j \notin H_{C-u_i-u_j}$, which in turn, will be used to show that the hypothesis of Theorem 1.5 is satisfied by suitably chosen sets.

Claim 4.2. *For every $i, j \in [2k]$, we have $H_{C-u_i-u_j} \subseteq H_{C-u_i}$.*

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

$$d(H_{C-u_i-u_j} \cap H_{C-u_i}) \geq d(H_{C-u_i-u_j}). \quad (5)$$

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

$$d(H_{C-u_i-u_j} \cup H_{C-u_i}) \geq d(H_{C-u_i}). \quad (6)$$

By submodularity of the hypergraph cut function and inequalities (5) and (6), we have that

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

Therefore, inequality (5) is an equation, and consequently, $(H_{C-u_i-u_j} \cap H_{C-u_i}, \overline{H_{C-u_i-u_j} \cap H_{C-u_i}})$ is a minimum $(C - u_i - u_j, T)$ -terminal cut. If $H_{C-u_i-u_j} \setminus H_{C-u_i} \neq \emptyset$, then $(H_{C-u_i-u_j} \cap H_{C-u_i}, \overline{H_{C-u_i-u_j} \cap H_{C-u_i}})$ contradicts source minimality of the minimum $(C - u_i - u_j, T)$ -terminal cut $(H_{C-u_i-u_j}, \overline{H_{C-u_i-u_j}})$. Hence, $H_{C-u_i-u_j} \setminus H_{C-u_i} = \emptyset$ and consequently, $H_{C-u_i-u_j} \subseteq H_{C-u_i}$. \square

Claim 4.2 implies the following Corollary.

Corollary 4.1. *For every $i \in [2k]$, we have $u_i, u_j \notin H_{C-u_i-u_j}$.*

Proof. By Claim 4.1, we have that $u_i \notin H_{C-u_i}$ and $u_j \notin H_{C-u_j}$. Therefore, $u_i, u_j \notin H_{C-u_i} \cap H_{C-u_j}$. By Claim 4.2, $H_{C-u_i-u_j} \subseteq H_{C-u_i}$ and $H_{C-u_i-u_j} \subseteq H_{C-u_j}$. Therefore, $H_{C-u_i-u_j} \subseteq H_{C-u_i} \cap H_{C-u_j}$, and thus, $u_i, u_j \notin H_{C-u_i-u_j}$. \square

The next claim will help in controlling the cost of the k -partition that we will obtain by applying Theorem 1.5.

Claim 4.3. *For every $i, j \in [2k]$, we have $d(H_{C-u_i}) = d(V_1) = d(H_{C-u_i-u_j})$.*

Proof. Let $a, b \in [2k]$. Since $(V_1, \overline{V_1})$ is a $(C - u_a, T)$ -terminal cut, we have that $d(H_{C-u_a}) \leq d(V_1)$. Since $(H_{C-u_a}, \overline{H_{C-u_a}})$ is a $(C - u_a - u_b, T)$ -terminal cut, we have that $d(H_{C-u_a-u_b}) \leq d(H_{C-u_a}) \leq d(V_1)$. Thus, in order to prove the claim, it suffices to show that $d(H_{C-u_a-u_b}) \geq d(V_1)$.

Suppose for contradiction that $d(H_{C-u_a-u_b}) < d(V_1)$. Let $\ell \in [2k] \setminus \{a, b\}$ be an arbitrary element (which exists since $k \geq 2$). Let $R := \{u_\ell\}$, $U := V_1$, $S' := C - u_a - u_\ell$, and $A_i := \overline{H_{C-u_a-u_i}}$ for every $i \in [2k] \setminus \{a, \ell\}$. We note that $|S'| = 2k - 2$. By Lemma 4.1, we have that $(\overline{A_i}, A_i)$ is a minimum $(C - u_a - u_i, \overline{V_1})$ -terminal cut for every $i \in [2k] \setminus \{a, \ell\}$. Moreover, by Corollary 4.1, we have that $u_i \in A_i \setminus (\cup_{j \in [2k] \setminus \{a, i, \ell\}} A_j)$ for every $i \in [2k] \setminus \{a, \ell\}$. Hence, the sets U , R , and S' , and the cuts $(\overline{A_i}, A_i)$ for $i \in [2k] \setminus \{a, \ell\}$ satisfy the conditions of Theorem 1.5. Therefore, by the first conclusion of Theorem 1.5, there exists a k -partition \mathcal{P}' with

$$\text{cost}(\mathcal{P}') \leq \frac{1}{2} \min\{d(H_{C-u_a-u_i}) + d(H_{C-u_a-u_j}) : i, j \in [2k] \setminus \{a, \ell\}\}.$$

By assumption, $d(H_{C-u_a-u_b}) < d(V_1)$ and $b \in [2k] \setminus \{a, \ell\}$, so $\min\{d(H_{C-u_a-u_i}) : i \in [2k] \setminus \{a, \ell\}\} < d(V_1)$. Since $(V_1, \overline{V_1})$ is a $(C - u_a - u_i, T)$ -terminal cut, we have that $d(H_{C-u_a-u_i}) \leq d(V_1)$ for every $i \in [2k] \setminus \{a, \ell\}$. Therefore,

$$\frac{1}{2} \min\{d(H_{C-u_a-u_i}) + d(H_{C-u_a-u_j}) : i, j \in [2k] \setminus \{a, \ell\}\} < d(V_1) = \text{cost}(\mathcal{P}).$$

Thus, we have that $\text{cost}(\mathcal{P}') < \text{cost}(\mathcal{P})$, which is a contradiction, since \mathcal{P} is a minimum k -partition. \square

The next two claims will help in arguing properties about the hyperedge e which will allow us to use the second conclusion of Theorem 1.5. In particular, we will need Claim 4.5. The following claim will help in proving Claim 4.5.

Claim 4.4. *For every $i, j \in [2k]$, we have*

$$d(H_{C-u_i} \cap H_{C-u_j}) = d(V_1) = d(H_{C-u_i} \cup H_{C-u_j}).$$

Proof. Since $(H_{C-u_i} \cap H_{C-u_j}, \overline{H_{C-u_i} \cap H_{C-u_j}})$ is a $(C - u_i - u_j, T)$ -terminal cut, we have that $d(H_{C-u_i} \cap H_{C-u_j}) \geq d(H_{C-u_i-u_j})$. By Claim 4.3, we have that $d(H_{C-u_i-u_j}) = d(V_1) = d(H_{C-u_i})$. Therefore,

$$d(H_{C-u_i} \cap H_{C-u_j}) \geq d(H_{C-u_i}). \quad (7)$$

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

$$d(H_{C-u_i} \cup H_{C-u_j}) \geq d(H_{C-u_j}). \quad (8)$$

By submodularity of the hypergraph cut function and inequalities (7) and (8), we have that

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

Therefore, inequalities (7) and (8) are equations. Thus, by Claim 4.3, we have that

$$d(H_{C-u_i} \cap H_{C-u_j}) = d(H_{C-u_i}) = d(V_1),$$

and

$$d(H_{C-u_i} \cup H_{C-u_j}) = d(H_{C-u_j}) = d(V_1).$$

\square

Claim 4.5. *For every $i, j, \ell \in [2k]$ with $i \neq j$, we have $H_{C-u_\ell} \subseteq H_{C-u_i} \cup H_{C-u_j}$.*

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

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

$$d(H_{C-u_\ell} \cap (H_{C-u_i} \cup H_{C-u_j})) = d(H_{C-u_\ell} \setminus Q) \geq d(H_{C-u_\ell}). \quad (9)$$

We also have that $(H_{C-u_\ell} \cup (H_{C-u_i} \cup H_{C-u_j}), \overline{H_{C-u_\ell} \cup (H_{C-u_i} \cup H_{C-u_j})})$ is a $(C - u_i, T)$ -terminal cut. Therefore, $d(H_{C-u_\ell} \cup (H_{C-u_i} \cup H_{C-u_j})) \geq d(H_{C-u_i})$. By Claims 4.3 and 4.4, we have that $d(H_{C-u_i}) = d(V_1) = d(H_{C-u_i} \cup H_{C-u_j})$. Therefore,

$$d(H_{C-u_\ell} \cup (H_{C-u_i} \cup H_{C-u_j})) \geq d(H_{C-u_i} \cup H_{C-u_j}). \quad (10)$$

By submodularity of the hypergraph cut function and inequalities (9) and (10), we have that

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

Therefore, inequalities (9) and (10) are equations, so $(H_{C-u_\ell} \setminus Q, \overline{H_{C-u_\ell} \setminus Q})$ is a minimum $(C - u_\ell, T)$ -terminal cut. \square

Let $R := \{u_{2k}\}$, $U := V_1$, and let $(\overline{A_i}, A_i) := (H_{C-u_i}, \overline{H_{C-u_i}})$ for every $i \in [2k - 1]$. By Lemma 4.1, we have that $(\overline{A_i}, A_i)$ is a minimum $(C - u_i, \overline{V_1})$ -terminal cut for every $i \in [2k - 1]$. Moreover, by Claim 4.1, we have that $u_i \in A_i \setminus (\cup_{j \in [2k-1] \setminus \{i\}} A_j)$. Hence, the sets U , R , and S , and the cuts $(\overline{A_i}, A_i)$ for $i \in [2k - 1]$ satisfy the conditions of Theorem 1.5. We will use the second conclusion of Theorem 1.5. We now show that the hyperedge e that we fixed at the beginning of the proof satisfies the conditions mentioned in the second conclusion of Theorem 1.5. We will use Claim 4.5 to prove this. Let $W := \cup_{1 \leq i < j \leq 2k-1} (A_i \cap A_j)$ and $Z := \cap_{i \in [2k-1]} \overline{A_i}$ as in the statement of Theorem 1.5.

Claim 4.6. *The hyperedge e satisfies the following conditions:*

1. $e \cap W \neq \emptyset$,
2. $e \cap Z \neq \emptyset$, and
3. $e \subseteq W \cup Z$.

Proof. 1. By Lemma 4.1, for every $i \in [2k - 1]$ we have $\overline{V_1} \subseteq A_i$, and therefore $\overline{V_1} \subseteq W$. Thus, by Observation 4.1, we have that $\emptyset \neq e \cap \overline{V_1} \subseteq e \cap W$.

2. By definition, for every $i \in [2k - 1]$, we have $u_{2k} \in H_{C-u_i} = \overline{A_i}$, and therefore $u_{2k} \in Z$. Thus, by Observation 4.1, we have that $e \cap Z \neq \emptyset$.

3. For every $i \in [2k - 1]$, let $Y_i := A_i \setminus W$. We note that $(Y_1, \dots, Y_{2k-1}, W, Z)$ is a partition of V . Therefore, in order to show that $e \subseteq W \cup Z$, it suffices to show that $e \cap Y_i = \emptyset$ for every $i \in [2k - 1]$. By Observation 4.1, we know that $e \subseteq \overline{H_S}$. We will show that $\overline{H_S} \cap Y_i = \emptyset$ for every $i \in [2k - 1]$ which implies that $e \cap Y_i = \emptyset$ for every $i \in [2k - 1]$. Let us fix an index $i \in [2k - 1]$. We note that

$$\begin{aligned} Y_i &= A_i \setminus W = A_i \setminus \left(\bigcup_{1 \leq a < b \leq 2k-1} (A_a \cap A_b) \right) = A_i \setminus \left(\bigcup_{1 \leq a < b \leq 2k-1} ((A_a \cap A_b) \cap A_i) \right) \\ &= A_i \setminus \left(\bigcup_{j \in [2k-1] \setminus \{i\}} (A_j \cap A_i) \right) = A_i \setminus \left(\bigcup_{j \in [2k-1] \setminus \{i\}} A_j \right) = A_i \cap \overline{\left(\bigcup_{j \in [2k-1] \setminus \{i\}} A_j \right)} \\ &= A_i \cap \left(\bigcap_{j \in [2k-1] \setminus \{i\}} \overline{A_j} \right) = \left(\bigcap_{j \in [2k-1] \setminus \{i\}} \overline{A_j} \right) \setminus \overline{A_i}. \end{aligned}$$

Therefore,

$$\begin{aligned} Y_i \cap \overline{H_S} &= \left(\left(\bigcap_{j \in [2k-1] \setminus \{i\}} \overline{A_j} \right) \setminus \overline{A_i} \right) \cap \overline{H_S} = \left(\left(\bigcap_{j \in [2k-1] \setminus \{i\}} \overline{A_j} \right) \setminus \overline{A_i} \right) \setminus H_S \\ &= \left(\bigcap_{j \in [2k-1] \setminus \{i\}} H_{C-u_j} \right) \setminus (H_{C-u_i} \cup H_{C-u_{2k}}). \end{aligned}$$

By Claim 4.5, we have that $H_{C-u_j} \subseteq H_{C-u_i} \cup H_{C-u_{2k}}$ for every $j \in [2k-1] \setminus \{i\}$. Therefore, $H_{C-u_j} \setminus (H_{C-u_i} \cup H_{C-u_{2k}}) = \emptyset$ for every $j \in [2k-1] \setminus \{i\}$, and hence,

$$\left(\bigcap_{j \in [2k-1] \setminus \{i\}} H_{C-u_j} \right) \setminus (H_{C-u_i} \cup H_{C-u_{2k}}) = \emptyset.$$

Thus, we have $Y_i \cap \overline{H_S} = \emptyset$. □

By Claim 4.6, the hyperedge e satisfies the conditions of the second conclusion of Theorem 1.5. Therefore, by Theorem 1.5, there exists a k -partition \mathcal{P}' with

$$\begin{aligned} \text{cost}(\mathcal{P}') &< \frac{1}{2} \min\{d(A_i) + d(A_j) : i, j \in [2k-1], i \neq j\} \\ &= d(V_1) \quad (\text{By Claim 4.3}) \\ &= \text{cost}(\mathcal{P}). \quad (\text{By assumption of the theorem}) \end{aligned}$$

Thus, we have obtained a k -partition \mathcal{P}' with $\text{cost}(\mathcal{P}') < \text{cost}(\mathcal{P})$, which is a contradiction since \mathcal{P} is a minimum k -partition. □

5 Stronger Structural Theorem for $k = 2$

We prove the stronger version of Theorem 1.3 for $k = 2$ —namely Theorem 1.4—in this section. We were able to prove Theorem 1.4 via two more techniques that are different from the one presented in this section—one technique is via a novel three-cut-set-lemma while the second technique is via the *canonical decomposition of hypergraphs* [6, 8, 9, 13]. It is unclear how to generalize both these techniques to $k \geq 3$. Here, we present a proof of Theorem 1.4 that closely resembles the proof of Theorem 1.3. We will again use the containment lemma (Lemma 4.1) in our proof. We mention how we obtain the stronger statement relative to Theorem 1.3 in the proof below. We restate and prove Theorem 1.4 now.

Theorem 1.4. *Let $G = (V, E)$ be a hypergraph and $\mathcal{P} = (V_1, V_2)$ be a minimum cut. Then, for all non-empty subsets $T \subseteq V_2$, there exists a subset $S \subseteq V_1$ with $|S| \leq 2$ such that the source minimal minimum (S, T) -terminal cut (A, \overline{A}) satisfies $\delta(A) = \delta(V_1)$ and $A \subseteq V_1$.*

Proof. Let us fix an arbitrary non-empty subset $T \subseteq \overline{V_1} = V_2$. For a subset $X \subseteq V_1$, we denote the source minimal minimum (X, T) -terminal cut by $(H_X, \overline{H_X})$. By Lemma 4.1, for all $X \subseteq V_1$ we have that $H_X \subseteq V_1$. If $|V_1| \leq 2$, then choosing $S = V_1$ proves the theorem. So, we will assume henceforth that $|V_1| \geq 3$. We will show that there exists a subset $S \subseteq V_1$ with $|S| \leq 2$ such that the

source minimal minimum (S, T) -terminal cut $(H_S, \overline{H_S})$ satisfies $\delta(H_S) = \delta(V_1)$. This suffices since we have that $H_S \subseteq V_1$ for all subsets $S \subseteq V_1$ (by Lemma 4.1).

We begin with the following useful claim. We note that Claim 5.1 crucially relies on the fact that V_1 is a part of a minimum cut (i.e, it crucially relies on $k = 2$)—it does not hold if V_1 is a part of a minimum k -partition for $k \geq 3$.

Claim 5.1. *For every $X \subseteq V_1$, we have that $d(H_X) = d(V_1)$.*

Proof. Since $X \subseteq V_1$ and $T \subseteq V_2$, we have that (V_1, V_2) is a (X, T) -terminal cut. Since $(H_X, \overline{H_X})$ is a minimum (X, T) -terminal cut, we have that $d(H_X) \leq d(V_1)$. Since $(H_X, \overline{H_X})$ is a cut, and (V_1, V_2) is a minimum cut, we have that $d(H_X) \geq d(V_1)$. Thus, $d(H_X) = d(V_1)$. \square

For the sake of contradiction, suppose that for every $S \subseteq V_1$ with $|S| \leq 2$, the source minimal minimum (S, T) -terminal cut $(H_S, \overline{H_S})$ does not satisfy $\delta(H_S) = \delta(V_1)$. Our proof strategy is to obtain a cheaper cut than (V_1, V_2) , thereby contradicting the optimality of (V_1, V_2) .

Let $S \subseteq V_1$ be a set of size 2 such that H_S is maximal—i.e., there does not exist $S' \subseteq V_1$ of size 2 such that $H_{S'} \supsetneq H_S$. In contrast to the proof of Theorem 1.3, where subsets S of size 3 had to be considered, here we only consider subsets S of size 2. We will see that this suffices to arrive at a contradiction. Let $S := \{u_1, u_2\}$. By assumption, we have that $\delta(H_S) \neq \delta(V_1)$, but by Claim 5.1, we have that $d(H_S) = d(V_1)$. Therefore, $\delta(V_1) \setminus \delta(H_S)$ is non-empty. Let $e \in \delta(V_1) \setminus \delta(H_S)$. Let $u_3 \in e \cap V_1$. Let $C := \{u_1, u_2, u_3\}$. For notational convenience we will use $C - u_i$ to denote $C \setminus \{u_i\}$ and $C - u_i - u_j$ to denote $C \setminus \{u_i, u_j\}$ for all $i, j \in [3]$. The choice of the hyperedge e is crucial to our proof—its properties will be used much later in our proof. We summarize the properties of the hyperedge e here.

Observation 5.1. *The hyperedge e has the following properties:*

1. $e \cap V_2 \neq \emptyset$
2. $u_3 \in e$, and
3. $e \subseteq \overline{H_S}$,

Our strategy to arrive at a cheaper cut than (V_1, V_2) is to apply the second conclusion of Theorem 1.5. The next few claims will set us up to obtain sets that satisfy the hypothesis of Theorem 1.5.

Claim 5.2. *For every $i \in [3]$, we have $u_i \notin H_{C-u_i}$.*

Proof. By Observation 5.1 we have $u_3 \in e$ and $e \subseteq \overline{H_S}$, so $u_3 \notin H_S = H_{C-u_3}$. Suppose $i \in [2]$. Our proof will rely on the choice of S .

Suppose for contradiction that $u_i \in H_{C-u_i}$ for some $i \in [2]$. Then we have that $S \subseteq H_{C-u_i}$, so $(H_{C-u_i} \cap H_S, \overline{H_{C-u_i} \cap H_S})$ is a (S, T) -terminal cut. Therefore,

$$d(H_{C-u_i} \cap H_S) \geq d(H_S). \quad (11)$$

Also, since $(H_{C-u_i} \cup H_S, \overline{H_{C-u_i} \cup H_S})$ is a $(C - u_i, T)$ -terminal cut, we have that

$$d(H_{C-u_i} \cup H_S) \geq d(H_{C-u_i}). \quad (12)$$

By submodularity of the hypergraph cut function and inequalities (11) and (12), we have that

$$d(H_S) + d(H_{C-u_i}) \geq d(H_{C-u_i} \cap H_S) + d(H_{C-u_i} \cup H_S) \geq d(H_S) + d(H_{C-u_i}).$$

Therefore, inequality (11) is an equation, and consequently, $(H_{C-u_i} \cap H_S, \overline{H_{C-u_i} \cap H_S})$ is a minimum (S, T) -terminal cut. If $H_{C-u_i} \cap H_S \subsetneq H_S$, then this contradicts the source minimality of the minimum (S, T) -terminal cut $(H_S, \overline{H_S})$. Therefore, $H_{C-u_i} \cap H_S = H_S$ and hence, $H_S \subseteq H_{C-u_i}$. Also, the vertex u_3 is in $C - u_i$ but not in H_S and hence, $H_S \subsetneq H_{C-u_i}$. However, $|C - u_i| = 2$. Therefore, the set $C - u_i$ contradicts the choice of S . \square

The next two claims will help in arguing properties about the hyperedge e which will allow us to use the second conclusion of Theorem 1.5. In particular, we will need Claim 5.4. The following claim will help in proving Claim 5.4. We note the similarity of Claims 5.3 and 5.4 in this proof to Claims 4.4 and 4.5 in the proof of Theorem 1.3. In order to prove Claims 4.4 and 4.5, we needed the size of C to be $2k$. Here, we are able to prove Claims 5.3 and 5.4 with the size of C being $2k - 1$ (for $k = 2$). We do this by exploiting Claim 5.1 shown earlier (which holds only for $k = 2$).

Claim 5.3. *For every $i, j \in [3]$, we have*

$$d(H_{C-u_i} \cap H_{C-u_j}) = d(V_1) = d(H_{C-u_i} \cup H_{C-u_j}).$$

Proof. Since $(H_{C-u_i} \cap H_{C-u_j}, \overline{H_{C-u_i} \cap H_{C-u_j}})$ is a $(C - u_i - u_j, T)$ -terminal cut, we have that $d(H_{C-u_i} \cap H_{C-u_j}) \geq d(H_{C-u_i-u_j})$. By Claim 5.1, we have that $d(H_{C-u_i-u_j}) = d(V_1) = d(H_{C-u_i})$. Therefore,

$$d(H_{C-u_i} \cap H_{C-u_j}) \geq d(H_{C-u_i}). \quad (13)$$

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

$$d(H_{C-u_i} \cup H_{C-u_j}) \geq d(H_{C-u_j}). \quad (14)$$

By submodularity of the hypergraph cut function and inequalities (13) and (14), we have that

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

Therefore, inequalities (7) and (8) are equations. Thus, by Claim 5.1 we have that

$$d(H_{C-u_i} \cap H_{C-u_j}) = d(H_{C-u_i}) = d(V_1),$$

and

$$d(H_{C-u_i} \cup H_{C-u_j}) = d(H_{C-u_j}) = d(V_1).$$

\square

The next claim follows from Claim 5.3 similar to the proof of Claim 4.5 from Claim 4.4 earlier. We include the proof for the sake of completeness.

Claim 5.4. *For every $i, j, \ell \in [3]$ with $i \neq j$, we have $H_{C-u_\ell} \subseteq H_{C-u_i} \cup H_{C-u_j}$.*

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

We now show that $(H_{C-u_\ell} \setminus Q, \overline{H_{C-u_\ell} \setminus Q})$ is a minimum $(C - u_\ell, T)$ -terminal cut. Since $H_{C-u_\ell} \setminus Q = H_{C-u_\ell} \cap (H_{C-u_i} \cup H_{C-u_j})$, we have that $C - u_i - u_j - u_\ell \subseteq H_{C-u_\ell} \setminus Q$. We also know

that u_i and u_j are contained in both H_{C-u_ℓ} and $H_{C-u_i} \cup H_{C-u_j}$. Therefore, $C - u_\ell \subseteq H_{C-u_\ell} \setminus Q$. Thus, $(H_{C-u_\ell} \setminus Q, \overline{H_{C-u_\ell} \setminus Q})$ is a $(C - u_\ell, T)$ -terminal cut. Therefore,

$$d(H_{C-u_\ell} \cap (H_{C-u_i} \cup H_{C-u_j})) = d(H_{C-u_\ell} \setminus Q) \geq d(H_{C-u_\ell}). \quad (15)$$

We also have that $(H_{C-u_\ell} \cup (H_{C-u_i} \cup H_{C-u_j}), \overline{H_{C-u_\ell} \cup (H_{C-u_i} \cup H_{C-u_j})})$ is a $(C - u_i, T)$ -terminal cut. Therefore, $d(H_{C-u_\ell} \cup (H_{C-u_i} \cup H_{C-u_j})) \geq d(H_{C-u_i})$. By Claims 5.1 and 5.3, we have that $d(H_{C-u_i}) = d(V_1) = d(H_{C-u_i} \cup H_{C-u_j})$. Therefore,

$$d(H_{C-u_\ell} \cup (H_{C-u_i} \cup H_{C-u_j})) \geq d(H_{C-u_i} \cup H_{C-u_j}). \quad (16)$$

By submodularity of the hypergraph cut function and inequalities (15) and (16), we have that

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

Therefore, inequalities (15) and (16) are equations, so $(H_{C-u_\ell} \setminus Q, \overline{H_{C-u_\ell} \setminus Q})$ is a minimum $(C - u_\ell, T)$ -terminal cut. \square

Let $R := \{u_3\}$, $U := V_1$, and for every $i \in [2]$, let $(\overline{A_i}, A_i) := (H_{C-u_i}, \overline{H_{C-u_i}})$. By Lemma 4.1, we have that $(\overline{A_i}, A_i)$ is a minimum $(C - u_i, V_2)$ -terminal cut for every $i \in [2]$. Moreover, by Claim 5.2, we have that $u_i \in A_i \setminus A_{3-i}$. Hence, the sets U , R , and S , and the cuts $(\overline{A_i}, A_i)$ for $i \in [2]$ satisfy the conditions of Theorem 1.5. We will use the second conclusion of Theorem 1.5. We now show that the hyperedge e that we fixed at the beginning of the proof satisfies the conditions mentioned in the second conclusion of Theorem 1.5. We will use Claim 5.4 to prove this. Let $W := A_1 \cap A_2$ and $Z := \overline{A_1} \cap \overline{A_2}$ as in the statement of Theorem 1.5.

Claim 5.5. *The hyperedge e satisfies the following conditions:*

1. $e \cap W \neq \emptyset$,
2. $e \cap Z \neq \emptyset$, and
3. $e \subseteq W \cup Z$.

Proof. 1. By Lemma 4.1, for every $i \in [2]$ we have $V_2 \subseteq A_i$, and therefore $V_2 \subseteq W$. Thus, by Observation 5.1, we have that $\emptyset \neq e \cap V_2 \subseteq e \cap W$.

2. By definition, for every $i \in [2]$, we have $u_3 \in H_{C-u_i} = \overline{A_i}$, and therefore $u_3 \in Z$. Thus, by Observation 5.1, we have that $e \cap Z \neq \emptyset$.

3. For every $i \in [2]$, let $Y_i := A_i \setminus W$. We note that (Y_1, Y_2, W, Z) is a partition of V . Therefore, in order to show that $e \subseteq W \cup Z$, it suffices to show that $e \cap Y_1 = e \cap Y_2 = \emptyset$. By Observation 5.1, we know that $e \subseteq \overline{H_S}$. We will show that $\overline{H_S} \cap Y_i = \emptyset$ for every $i \in [2]$ which implies that $e \cap Y_i = \emptyset$ for every $i \in [2]$. Let us fix a $i \in [2]$. We note that

$$Y_i = A_i \setminus W = A_i \setminus (A_1 \cap A_2) = A_i \setminus A_{3-i} = A_i \cap \overline{A_{3-i}} = \overline{A_{3-i}} \setminus \overline{A_i}.$$

Therefore,

$$Y_i \cap \overline{H_S} = (\overline{A_{3-i}} \setminus \overline{A_i}) \cap \overline{H_S} = (\overline{A_{3-i}} \setminus \overline{A_i}) \setminus H_S = H_{C-u_{3-i}} \setminus (H_{C-u_i} \cup H_{C-u_3}).$$

By Claim 5.4, we have that $H_{C-u_{3-i}} \subseteq H_{C-u_i} \cup H_{C-u_3}$. Therefore,

$$H_{C-u_{3-i}} \setminus (H_{C-u_i} \cup H_{C-u_3}) = \emptyset.$$

Thus, for every $i \in [2]$, we have $Y_i \cap \overline{H_S} = \emptyset$. □

By Claim 5.5, the hyperedge e satisfies the conditions of the second conclusion of Theorem 1.5. Therefore, by Theorem 1.5, there exists a cut (V'_1, V'_2) with

$$\text{cost}(V'_1, V'_2) = d(V'_1) < \frac{1}{2}(d(A_1) + d(A_2)) = d(V_1).$$

The last equality above is by Claim 5.1. Thus, we have obtained a cut (V'_1, V'_2) with $d(V'_1) < d(V_1)$, which is a contradiction since (V_1, V_2) is a minimum cut. □

6 Conclusion and Open Problems

Several works in the literature have approached global cut and partitioning problems via minimum (S, T) -terminal cuts (e.g., see [2, 4, 16, 17, 34]). Our work adds to this rich literature by showing that ENUM-HYPERGRAPH- k -CUT can be solved via minimum (S, T) -terminal cuts. As a special case, our approach leads to a more straightforward approach to enumerate all minimum cut-sets in a given hypergraph. We mention a couple of open questions raised by our work.

1. For a long time, the known upper bound on the number of minimum k -partitions in connected graphs was $O(n^{2k-2})$ [5, 28, 38] while the known lower bound was $\Omega(n^k)$ (cycle), where n is the number of vertices in the input graph. A recent result improved the upper bound to $O(n^k)$ which also resulted in a faster randomized algorithm to solve GRAPH- k -CUT [18, 22]. We currently know that the number of minimum k -cut-sets in hypergraphs is $O(n^{2k-2})$ and is $\Omega(n^k)$ (the lower bound comes from graphs). Can we improve the upper bound on the number of minimum k -cut-sets in hypergraphs to $O(n^k)$?
2. Can we improve the deterministic run-time to solve ENUM-HYPERGRAPH- k -CUT? Our deterministic algorithm for ENUM-HYPERGRAPH- k -CUT runs in time $n^{O(k^2)}p$, where p is the size of the input hypergraph. In particular, the run-time of our algorithm has a quadratic dependence on k in the exponent of n . In contrast, the number of optimum solutions is only $O(n^{2k-2})$ —i.e., linear dependence on k in the exponent of n . We note that ENUM-GRAPH- k -CUT as well as HYPERGRAPH- k -CUT can be solved deterministically in $n^{O(k)}p$ time [2, 5, 38].

Acknowledgements. We would like to thank Chandra Chekuri for feedback that helped improve the introductory section of this work. Karthik would like to thank Michel Goemans for clarifying the proof of Theorem 15 in [16] via email.

References

- [1] H. Aissi, A. Mahjoub, T. McCormick, and M. Queyranne, *Strongly polynomial bounds for multiobjective and parametric global minimum cuts in graphs and hypergraphs*, Mathematical Programming (Preliminary version in IPCO 2014) **154** (2015), no. 1-2, 3–28.

- [2] K. Chandrasekaran and C. Chekuri, *Hypergraph k -cut for fixed k in deterministic polynomial time*, Proceedings of the 61st Annual Symposium on Foundations of Computer Science (FOCS), 2020, pp. 810–821.
- [3] K. Chandrasekaran, C. Xu, and X. Yu, *Hypergraph k -cut in randomized polynomial time*, Mathematical Programming (Preliminary version in SODA 2018) **186** (2019), 85–113.
- [4] Karthekeyan Chandrasekaran and Chandra Chekuri, *Min-max partitioning of hypergraphs and symmetric submodular functions*, Proceedings of the 32nd annual ACM-SIAM Symposium on Discrete Algorithms, SODA, 2021, pp. 1026–1038.
- [5] C. Chekuri, K. Quanrud, and C. Xu, *LP relaxation and tree packing for minimum k -cut*, SIAM Journal on Discrete Mathematics **34** (2020), no. 2, 1334–1353.
- [6] C. Chekuri and C. Xu, *Minimum cuts and sparsification in hypergraphs*, SIAM Journal on Computing **47** (2018), no. 6, 2118–2156.
- [7] E. Cheng, *Edge-augmentation of hypergraphs*, Math Programming **84** (1999), 443–465.
- [8] W. Cunningham, *Decomposition of submodular functions*, Combinatorica **3** (1980), 53–68.
- [9] W. Cunningham and J. Edmonds, *A combinatorial decomposition theory*, Canad. J. Math **32** (1980), 734–765.
- [10] E. Dahlhaus, D. Johnson, C. Papadimitriou, P. Seymour, and M. Yannakakis, *The complexity of multiterminal cuts*, SIAM Journal on Computing **23** (1994), no. 4, 864–894.
- [11] E. A. Dinitz, A. V. Karzanov, and M. V. Lomonosov, *On the structure of a family of minimum weighted cuts in a graph*, Studies in Discrete Optimization (A. A. Fridman, ed.), Nauka Publishers, 1976.
- [12] K. Fox, D. Panigrahi, and F. Zhang, *Minimum cut and minimum k -cut in hypergraphs via branching contractions*, Proceedings of the 30th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA, 2019, pp. 881–896.
- [13] S. Fujishige, *Canonical decompositions of symmetric submodular functions*, Discrete Appl. Math. **5** (1983), 175–190.
- [14] T. Fukunaga, *Computing minimum multiway cuts in hypergraphs*, Discrete Optimization **10** (2013), no. 4, 371–382.
- [15] M. Ghaffari, D. Karger, and D. Panigrahi, *Random contractions and sampling for hypergraph and hedge connectivity*, Proceedings of the 28th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA, 2017, p. 1101–1114.
- [16] M. X. Goemans and V. S. Ramakrishnan, *Minimizing submodular functions over families of sets*, Combinatorica **15** (1995), 499–513.
- [17] 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** (1994), no. 1, 24–37.
- [18] A. Gupta, D. Harris, E. Lee, and J. Li, *Optimal Bounds for the k -cut Problem*, Preprint in arXiv: 2005.08301, 2020.

- [19] A. Gupta, E. Lee, and J. Li, *An FPT Algorithm Beating 2-Approximation for k -Cut*, Proceedings of the 29th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA, 2018, pp. 2821–2837.
- [20] ———, *Faster exact and approximate algorithms for k -cut*, Proceedings of the 59th IEEE annual Symposium on Foundations of Computer Science, FOCS, 2018, pp. 113–123.
- [21] ———, *The number of minimum k -cuts: improving the Karger-Stein bound*, Proceedings of the 51st ACM Symposium on Theory of Computing, STOC, 2019, pp. 229–240.
- [22] ———, *The Karger-Stein algorithm is optimal for k -cut*, Proceedings of the 52nd Annual ACM Symposium on Theory of Computing, STOC, 2020, pp. 473–484.
- [23] J. W. Hamacher, J.-C. Picard, and M. Queyranne, *Ranking the cuts and cut-sets of a network*, North Holland Mathematical Studies **95** (1984), 183–200.
- [24] M. Henzinger and D. Williamson, *On the number of small cuts in a graph*, Information Processing Letters **59** (1996), 41–44.
- [25] Y. Kamidoi, N. Yoshida, and H. Nagamochi, *A Deterministic Algorithm for Finding All Minimum k -Way Cuts*, SIAM Journal on Computing **36** (2007), no. 5, 1329–1341.
- [26] D. Karger, *Global min-cuts in RNC, and other ramifications of a simple min-cut algorithm*, Proceedings of the 4th annual ACM-SIAM Symposium on Discrete Algorithms, SODA, 1993, pp. 21–30.
- [27] ———, *Minimum cuts in near-linear time*, Journal of the ACM **47** (2000), no. 1, 46–76.
- [28] D. Karger and C. Stein, *A new approach to the minimum cut problem*, Journal of the ACM **43** (1996), no. 4, 601–640.
- [29] E. Lawler, *Cutsets and Partitions of Hypergraphs*, Networks **3** (1973), 275–285.
- [30] D. Lokshtanov, S. Saurabh, and V. Surianarayanan, *A Parameterized Approximation Scheme for MIN k -CUT*, Proceedings of the 61st IEEE annual Symposium on Foundations of Computer Science, FOCS, 2020, pp. 798–809.
- [31] P. Manurangsi, *Inapproximability of Maximum Biclique Problems, Minimum k -Cut and Densest At-Least- k -Subgraph from the Small Set Expansion Hypothesis*, Algorithms **11(1)** (2018), 10.
- [32] H. Nagamochi and T. Ibaraki, *Algorithmic Aspects of Graph Connectivity*, Cambridge University Press, Cambridge, 2008.
- [33] H. Nagamochi, K. Nishimura, and T. Ibaraki, *Computing all small cuts in an undirected network*, SIAM Journal on Discrete Mathematics **10** (1997), no. 3, 469–481.
- [34] M. Nägele, B. Sudakov, and R. Zenklusen, *Submodular minimization under congruency constraints*, Combinatorica **39** (2019), 1351–1386.
- [35] K. Okumoto, T. Fukunaga, and H. Nagamochi, *Divide-and-conquer algorithms for partitioning hypergraphs and submodular systems*, Algorithmica **62** (2012), no. 3, 787–806.
- [36] R. Ravi and A. Sinha, *Approximating k -cuts using network strength as a lagrangean relaxation*, European Journal of Operational Research **186** (2008), no. 1, 77–90.

- [37] H. Saran and V. Vazirani, *Finding k Cuts within Twice the Optimal*, SIAM Journal on Computing **24** (1995), no. 1, 101–108.
- [38] M. Thorup, *Minimum k -way Cuts via Deterministic Greedy Tree Packing*, Proceedings of the 40th Annual ACM Symposium on Theory of Computing, STOC, 2008, pp. 159–166.
- [39] V. Vazirani and M. Yannakakis, *Suboptimal cuts: Their enumeration, weight and number (extended abstract)*, Proceedings of the 19th International Colloquium on Automata, Languages and Programming, ICALP '92, 1992, pp. 366–377.
- [40] M. Xiao, *Finding minimum 3-way cuts in hypergraphs*, Information Processing Letters (Preliminary version in TAMC 2008) **110** (2010), no. 14, 554–558.
- [41] L. Zhao, *Approximation algorithms for partition and design problems in networks*, Ph.D. 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** (2005), no. 1, 167–183.

A Proof of Theorem 1.5

We prove Theorem 1.5 in this section. We will need certain partition uncrossing and partition aggregation results from [2] that rely on more careful counting of hyperedges than simply employing the submodularity inequality. We begin with some notation that will help in such careful counting—our notation will be identical to the notation in [2]. Let (Y_1, \dots, Y_p, W, Z) be a partition of V . We recall that $\text{cost}(Y_1, \dots, Y_p, W, Z)$ denotes the number of hyperedges that cross the partition. We define the following quantities:

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

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

$$\sigma(Y_1, \dots, Y_p, W, Z) := \text{cost}(Y_1, \dots, Y_p, W, Z) + \text{cost}(W, Z) + \alpha(Y_1, \dots, Y_p, W, Z) + \beta(Y_1, \dots, Y_p, Z).$$

The precise interpretation of the quantity $\sigma(Y_1, \dots, Y_p, W, Z)$ will not be important for our purposes—see [2] for the interpretation.

The following result from [2] shows that a collection of sets can be uncrossed to obtain a partition with small σ -value. We note the similarity of the hypothesis of Lemma A.1 with the hypothesis of Theorem 1.5 and once again, refer to Figure 1 for an illustration of the sets that appear in the statement of Lemma A.1.

Lemma A.1. [2] Let $G = (V, E)$ be a hypergraph and $\emptyset \neq R \subsetneq U \subsetneq V$. Let $S = \{u_1, \dots, u_p\} \subseteq U \setminus R$ 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

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

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

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

Moreover, if $p = 2$, then the above inequality is an equation.

The next lemma from [2] will help in aggregating the parts of a ℓ -partition \mathcal{P} where $\ell \geq 2k$ to a k -partition \mathcal{K} while controlling the cost of \mathcal{K} .

Lemma A.2. [2] Let $G = (V, E)$ be a hypergraph, $k \geq 2$ be an integer, and (Y_1, \dots, Y_p, W, Z) be a partition of V for some integer $p \geq 2k - 2$. Then, there exist distinct $i_1, \dots, i_{k-1} \in [p]$ such that

$$2\text{cost}(Y_{i_1}, \dots, Y_{i_{k-1}}, V \setminus (\cup_{j=1}^{k-1} Y_{i_j})) \leq \text{cost}(Y_1, \dots, Y_p, W, Z) + \alpha(Y_1, \dots, Y_p, W, Z) + \beta(Y_1, \dots, Y_p, Z).$$

We now restate and prove Theorem 1.5.

Theorem 1.5. Let $G = (V, E)$ be a hypergraph, $k \geq 2$ be an integer and $\emptyset \neq R \subsetneq U \subsetneq V$. Let $S = \{u_1, \dots, u_p\} \subseteq U \setminus R$ 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, \dots, P_k) of V with $\overline{U} \subsetneq P_k$ such that

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

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.

Proof. For the first conclusion of the theorem, we will use the same proof that appeared in [2]. We need the details of this proof to prove the second conclusion of the theorem.

We begin by proving the first conclusion. By applying Lemma A.1, we obtain a $(p+2)$ -partition (Y_1, \dots, Y_p, W, Z) such that

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

and moreover, $\overline{U} \subseteq W$, where $Y_i = A_i - W$ for all $i \in [p]$, $W = \cup_{1 \leq i < j \leq p} (A_i \cap A_j)$, and $Z = \cap_{i \in [p]} \overline{A}_i$. We recall that $p \geq 2k - 2$. Hence, by applying Lemma A.2 to the $(p+2)$ -partition (Y_1, \dots, Y_p, W, Z) , we obtain a k -partition (P_1, \dots, P_k) of V such that $W \cup Z \subseteq P_k$ and

$$\text{cost}(P_1, \dots, P_k) \leq \frac{1}{2} (\text{cost}(Y_1, \dots, Y_p, W, Z) + \alpha(Y_1, \dots, Y_p, W, Z) + \beta(Y_1, \dots, Y_p, Z)) \quad (17)$$

$$\leq \frac{1}{2} (\text{cost}(Y_1, \dots, Y_p, W, Z) + \text{cost}(W, Z) + \alpha(Y_1, \dots, Y_p, W, Z) + \beta(Y_1, \dots, Y_p, Z)) \quad (18)$$

$$= \frac{1}{2} \sigma(Y_1, \dots, Y_p, W, Z) \quad (19)$$

$$\leq \frac{1}{2} \min\{d(A_i) + d(A_j) : i, j \in [p], i \neq j\}. \quad (20)$$

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

We now prove the second conclusion of the theorem. If there exists a hyperedge $e \in E$ such that e intersects W , e intersects Z , and e is contained in $W \cup Z$, then $\text{cost}(W, Z) > 0$. Consequently, inequality (18) in the above sequence of inequalities should be strict. \square