Mathematical Programming
https://doi.org/10.1007/s10107-023-02013-8

FULL LENGTH PAPER

Series A

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

Calvin Beideman[1] · Karthekeyan Chandrasekaran[1] · Weihang Wang[1]

Received: 4 May 2023 / Accepted: 1 August 2023
© Springer-Verlag GmbH Germany, part of Springer Nature and Mathematical Optimization Society 2023

Abstract

We consider the problem of deterministically enumerating all minimum k-cut-sets in a given hypergraph for fixed constant k. The input here is a hypergraph $G = (V, E)$ with non-negative hyperedge 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 (Goldschmidt and Hochbaum in Math Oper Res 19(1):24–37, 1994; Karger and Stein in J ACM 43(4):601–640, 1996; Kamidoi et al. in SIAM J Comput 36(5):1329–1341, 2007; Thorup, in: Proceedings of the 40th annual ACM symposium on theory of computing, STOC, 2008). In contrast, it is only recently that polynomial-time algorithms for HYPERGRAPH- k- CUT were developed (Chandrasekaran et al. in Math Program 186:85–113, 2019; Fox et al., in: Proceedings of the 30th annual ACM-SIAM symposium on discrete algorithms, SODA, 2019; Chandrasekaran and Chekuri in Math Oper Res 47:3380–3399, 2022). The randomized polynomial-time algorithm for HYPERGRAPH- k- CUT that was designed in 2018 (Chandrasekaran et al. 2019) 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,

Supported in part by NSF Grants CCF-1814613 and CCF-1907937. A preliminary version of this work appeared at ACM-SIAM Symposium on Discrete Algorithms (SODA), 2022. This version contains complete proof of the structural theorem for min-cut-sets.

✉ Karthekeyan Chandrasekaran
karthe@illinois.edu

Calvin Beideman
calvinb2@illinois.edu

Weihang Wang
weihang3@illinois.edu

[1] University of Illinois Urbana-Champaign, Champaign, USA

Published online: 12 September 2023

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 (Chandrasekaran and Chekuri 2022), 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 minimum cut-sets (i.e., minimum 2-cut-sets) in hypergraphs—we give a new proof showing that the number of minimum cut-sets in a n-vertex hypergraph is at most $\binom{n}{2}$ and they can all be enumerated in deterministic polynomial time for a given hypergraph.

Mathematics Subject Classification 90C27 · 05C65 · 05C85

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 $c : E \to \mathbb{R}_+$ and a positive integer k. The objective is to find a minimum-cost subset of hyperedges whose removal results in at least k connected components. 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., $k = 2, 3, 4, ...$). 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 $V_1, V_2, \ldots, V_k$ so as to minimize the cost of hyperedges that cross the partition. A hyperedge $e \in E$ is said to *cross a partition* $V_1, V_2, \ldots, V_k$ 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 $O(n^k)$ for constant k, where n is the number of vertices (i.e., the number of minimum k-partitions is polynomial since k is a constant) [19, 23, 29]. However, for a connected[1] 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 [12] constructed a compact representation for all minimum cuts in a connected graph (known as the *cactus representation*) which showed that the 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 constant $k \geq 3$, 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 [19, 23, 29]; 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 Sect. 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 [30]. Once again, enumerating all minimum cut-sets in a hypergraph in deterministic polynomial-time is already non-trivial. We encourage the reader to pause and think briefly about possible approaches to enumerate all minimum cut-sets in a hypergraph before reading further. There exists a compact representation of all minimum cut-sets in a hypergraph [7]—namely the *hypercactus representation*—which can be used to show 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 poly-

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

nomial 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 [4, 36, 43, 44]). Owing to this connection, the complexity of HYPERGRAPH- k- CUT for 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 [5]. 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 [13]). Subsequently, Chandrasekaran and Chekuri designed a deterministic polynomial-time algorithm for HYPERGRAPH- k- CUT in 2020 [4]. 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 Sect. 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 in the theory of hypergraphs.

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 $(V_1, \ldots, V_h)$. For a non-empty proper subset U of vertices, we will use $\overline{U}$ to denote $V \setminus U$, $\delta(U)$ to denote the set of hyperedges crossing the 2-partition $(U, \overline{U})$, and $d(U) := |\delta(U)|$. We recall that $\delta(U) = \delta(\overline{U})$, so we will use $d(U)$ to denote the cost of the cut $(U, \overline{U})$. More generally, given a partition $\mathcal{P} = (V_1, V_2, \ldots, V_h)$, we denote the set of hyperedges crossing the partition by $\delta(V_1, V_2, \ldots, V_h)$ (also by $\delta(\mathcal{P})$ for brevity) and the number of hyperedges crossing the partition by $\text{cost}(V_1, V_2, \ldots, V_h) := |\delta(V_1, V_2, \ldots, V_h)|$ (also by $\text{cost}(\mathcal{P})$ for brevity). Let S, T be disjoint non-empty subsets of vertices. A 2-partition $(U, \overline{U})$ is an (S, T)-terminal cut if $S \subseteq U \subseteq V \setminus T$. Here, the set U is known as the source set and the set $\overline{U}$ is known as the sink set. A minimum-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: a minimum (S, T)-terminal cut $(U, \overline{U})$ is a source minimal minimum (S, T)-terminal cut if there does not exist a minimum (S, T)-terminal cut $(U', \overline{U'})$ such that $U' \subsetneq U$ and is a source maximal minimum (S, T)-terminal cut if there does not exist a minimum (S, T)-terminal cut $(U', \overline{U'})$ such that $U' \supsetneq U$. 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 [18]. This approach was generalized by Kamidoi, Yoshida, and Nagamochi to solve ENUM- GRAPH- k- CUT [26] and by Chandrasekaran and Chekuri to solve HYPERGRAPH- k- CUT [4], both in deterministic polynomial time. The techniques of [18] and [26] 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 [4]. The goal here is to identify one part of some fixed minimum k-partiton $(V_1, V_2, \ldots, V_k)$, say V_1 without loss of generality, and then recursively find a minimum $(k-1)$-partition in the subhypergraph $G[\overline{V_1}]$, where $G[\overline{V_1}]$ 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 $(V_1, \ldots, V_k)$ is a V_1-*maximal* minimum k-partition—i.e.,

there is no other minimum k-partition $(V_1', \ldots, V_k')$ such that V_1 is a proper subset of V_1'. Then, they showed that for every subset $T \subseteq \overline{V_1}$ such that $T \cap V_j \neq \emptyset$ for all $j \in \{2, \ldots, k\}$, there exists a subset $S \subseteq V_1$ of size at most $2k-2$ such that $(V_1, \overline{V_1})$ 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 $j \in \{2, \ldots, k\}$). Moreover, the size of the collection $\mathcal{C}$ is only $O(n^{3k-3})$. Hence, recursing on $G[\overline{U}]$ 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 [4] 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 $(V_1, \ldots, V_k)$ such that $F = \delta(V_1, \ldots, V_k)$. We note that $d(V_1) \leq |F|$ by definition of the hypergraph cut function $d : 2^V \to \mathbb{R}$. 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 $(U, \overline{U})$ 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 $S \subseteq U \backslash \{s\}$ and $T \subseteq \overline{U} \backslash \{t\}$ 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.*

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 $(U, \overline{U})$ with $d(U) < \text{OPT}_2$. 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 $(V'_1, \ldots, V'_k)$ for which $F = \delta(V'_1, \ldots, V'_k)$ (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 $(V_1, \overline{V_1})$, i.e., $\delta(V_1) = F = \delta(V_1, \ldots, V_k)$. 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 $\mathcal{P} = (V_1, \ldots, V_k)$ 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 $j \in \{2, 3, \ldots, k\}$, 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 $A \subseteq V_1$.*

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 $(V_1, \overline{V_1})$ but not the cut $(V_1, \overline{V_1})$ 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 $(U_1, \ldots, U_k)$ of the vertex set where all sets $U_1, \ldots, U_k$ are in the collection $\mathcal{C}$ and add the set $\delta(U_1, \ldots, U_k)$ 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 $(V_1, \ldots, V_k)$ such that F is the set of hyperedges crossing this k-partition, i.e., $F = \delta(V_1, \ldots, V_k)$. 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 $(U = V_i, \overline{U} = \overline{V_i})$ and arbitrary vertices $s \in V_i, t \in \overline{V_i}$), 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

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

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 $(A, \overline{A})$ 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 $\binom{n}{2}$ [7, 16]. 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 $(A, \overline{A})$ 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 $(A, \overline{A})$ 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 [17]. Goemans and Ramakrishnan showed that (Theorem 15 in [17]) 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 $(U, \overline{U})$ 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 [17]. 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.

Table 1 Summary of known techniques to solve enumeration problems in graphs and hypergraphs in polynomial time

Techniques	Graph Min-Cut	Hypergraph Min-Cut	Graph-k-cut	Hypergraph-k-cut
Random contraction	[27, 29]	[5, 13, 16]	[19, 22, 23, 29]	[5, 13]
Cactus representation	[12]	[7–9]	?	?
Min (S, T)-cuts	[17]	Our work (Theorem 1.4)	[26, 41]	Our work (Theorems 1.2, 1.3)
Edge splitting	[25, 34]	?	?	?
Tree packing	[28]	?	[6, 39]	?

The random contraction technique leads to a randomized algorithm while techniques listed in the rest of the rows lead to deterministic algorithms. Entries with a question mark indicate that we do not *yet* know if the technique leads to a polynomial time algorithm for the corresponding enumeration problem

We conclude this section with a table summarizing known techniques for solving enumeration problems in graphs and hypergraphs in polynomial time (see Table 1). We refer the reader to Sect. 1.2 for a more elaborate discussion of related work.

Organization We discuss special cases of ENUM- HYPERGRAPH- k- CUT that have been addressed in the literature in Sect. 1.2. In Sect. 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 [4] whose proof appears in "Appendix A". In Sect. 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 Sects. 3 and 4 respectively. We prove the strengthening of Theorem 1.3 for $k = 2$—namely Theorem 1.4—in Sect. 5. We give an alternative proof of Theorem 1.4 based on a novel three-cut-set lemma in Sect. 6. We conclude with a few open problems in Sect. 7.

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 [12] 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 [27] 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 [28] 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 [34] 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 [17] via a structural result (see discussion after Theorem 1.4 above). Henzinger and Williamson [25] 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, 24, 33, 40]. 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 [10], Fujishige [14], and Cunningham [9] on submodular functions. Cheng [8] designed an explicit *hypercactus representation* for all minimum cut-sets in a hypergraph. Chekuri and Xu [7] 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 [16] (also see [5, 13]) 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 exponential[2] for any

[2] Consider 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 $1 + \delta\binom{n}{2} \leq (1 + \epsilon)\lambda$.

$\epsilon \in (0, 1)$. 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 $s_1, s_2, \ldots, s_k$. 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 [4] 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 [6, 18–23, 26, 29, 31, 32, 37–39]. Goldschmidt and Hochbaum [18] 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 a fixed constant (polynomial-time solvability is not obvious even for $k = 3$). 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 [29] 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 $O(n^k)$ [19, 23]. 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 [39] 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 [6]). (2) Kamidoi, Yoshida, and Nagamochi [26] extended Goldschmidt and Hochbaum's divide and conquer approach to solve ENUM- GRAPH- k- CUT (also see [41]).

HYPERGRAPH-k-CUT. The complexity of HYPERGRAPH- k- CUT was open since the work of Goldschmidt and Hochbaum for GRAPH- k- CUT [18] until recently. Although certain special cases of HYPERGRAPH- k- CUT were known to be solvable in polynomial time [15, 42], considerable progress on HYPERGRAPH- k- CUT happened only in the last 3 years. Chandrasekaran et al. [5] 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 [13] 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 [16] 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 [4] 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 [4]. 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 Sect. 1.1. We will repeatedly rely on the fact that the hypergraph cut function $d : 2^V \to \mathbb{R}_+$ is symmetric and submodular. We recall that a set function $f : 2^V \to \mathbb{R}$ is *symmetric* if $f(U) = f(\overline{U})$ for all $U \subseteq V$ 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 [4]. We state the strengthened version below. See Fig. 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 [4]—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 $S = \{u_1, \ldots, u_p\} \subseteq U \backslash R$ for $p \geq 2k - 2$. Let $(\overline{A_i}, A_i)$ be a minimum $((S \cup R) \backslash \{u_i\}, \overline{U})$-terminal cut. Suppose that $u_i \in A_i \backslash (\cup_{j \in [p] \backslash \{i\}} A_j)$ for every $i \in [p]$. Then, the following two hold:*

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

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

2. *Moreover, if there exists a hyperedge $e \in E$ such that e intersects $W := \cup_{1 \leq i < j \leq p} (A_i \cap A_j)$, e intersects $Z := \cap_{i \in [p]} \overline{A_i}$, and e is contained in $W \cup Z$, then the inequality in the previous conclusion is strict.*

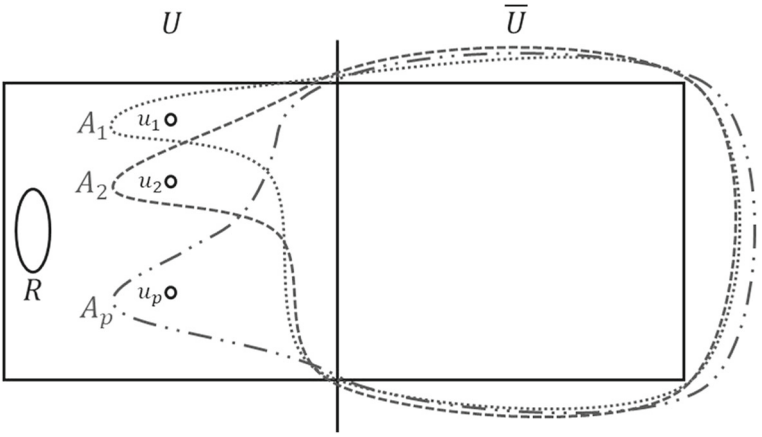

Fig. 1 Illustration of the sets that appear in the statement of Theorem 1.5

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

Fig. 2 Algorithm to enumerate hypergraph minimum k-cut-sets

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 Fig. 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 [7].

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 Fig. 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 $(V_1, \ldots, V_k)$ be a minimum k-partition such that $F = \delta(V_1, \ldots, V_k)$. 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 $(V_i, \overline{V_i})$ is the unique minimum (S, T)-terminal cut. Hence, the set V_i is in the collection $\mathcal{C}$. Consequently, all parts $V_1, \ldots, V_k$ are in the collection $\mathcal{C}$. Hence, the set $F = \delta(V_1, \ldots, V_k)$ 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 $\delta(V_i) = F = \delta(V_1, \ldots, V_k)$. 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 $(A, \overline{A})$ 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 $(U_1, \ldots, U_k) \in \mathcal{C}^k$ is $O(n^{4k^2-2k})$. Verifying if a tuple $(U_1, \ldots, U_k)$ 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 $O(n^{4k-2})T(n, p) + O(n^{4k^2-2k}p)$. □

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 $(U, \overline{U})$ is a 2-partition of V with $d(U) < OPT_k$. Then, for every vertex $s \in U$, there exists a subset $S \subseteq U \backslash \{s\}$ with $|S| \leq 2k - 3$ such that $(U, \overline{U})$ is the unique minimum $(S \cup \{s\}, \overline{U})$-terminal cut.*

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

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

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 $U \backslash \{s\}$ 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. □

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

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

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 $S := \{u_1, u_2, \ldots, u_p\}$ for some $p \geq 2k - 2$. For each $i \in [p]$, let $(\overline{A_i}, A_i)$ be the source minimal minimum $((S \cup \{s\}) \backslash \{u_i\}, \overline{U})$-terminal cut. The following claim will allow us to show that the cuts $(\overline{A_i}, A_i)$ 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, $(\overline{B_i}, B_i)$ is a $((S \cup \{s\}) \backslash \{u_i\}, \overline{U})$-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 $\overline{U} \subseteq A_i$. If $A_i = \overline{U}$, then the above inequalities are equations implying that $(B_i, \overline{B_i})$ is a minimum $((S \cup \{s\}) \backslash \{u_i\}, \overline{U})$-terminal cut, and consequently, $(B_i, \overline{B_i})$ contradicts source minimality of the minimum $((S \cup \{s\}) \backslash \{u_i\}, \overline{U})$-terminal cut $(A_i, \overline{A_i})$. Therefore, $\overline{U} \subsetneq A_i$. 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 $S \backslash \{u_i\}$ are in A_i by definition of A_i. Therefore, the vertex u_i must be in A_i. □

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 $(P_1, \ldots, P_k)$ of V such that

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

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. □

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 $T \subseteq \overline{U} \backslash \{t\}$ 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 $S \subseteq U \backslash \{s\}$ and $T \subseteq \overline{U} \backslash \{t\}$ 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 $S \subseteq U \backslash \{s\}$ 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. 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, $Y = U$. □

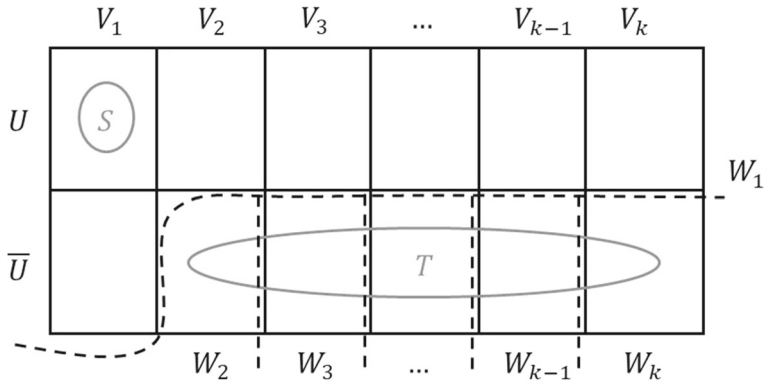

Fig. 3 Uncrossing in the proof of Lemma 4.1

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 [4, 11, 18, 36]).

Lemma 4.1 *Let $G = (V, E)$ be a hypergraph, $k \geq 2$ be an integer, $\mathcal{P} = (V_1, \ldots, V_k)$ 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 $j \in \{2, 3, \ldots, k\}$. 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

$$d(U \cap V_1) \geq d(U). \tag{1}$$

Now consider $\mathcal{P}' = (W_1 := U \cup V_1, W_2 := V_2 \backslash U, \ldots, W_k := V_k \backslash U)$ (see Fig. 3). For each $i \in \{2, 3, \ldots, k\}$, we have $\emptyset \neq T \cap V_i \subseteq V_i \backslash U$, 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

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

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, $U \backslash V_1 = \emptyset$ 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. □

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 $\mathcal{P} = (V_1, \ldots, V_k)$ 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 $j \in \{2, 3, \ldots, k\}$, 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 $j \in \{2, \ldots, k\}$. 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 $(V_1, \ldots, V_k)$, thereby contradicting the optimality of $(V_1, \ldots, V_k)$.

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 $S := \{u_1, u_2, \ldots, u_{2k-1}\}$. 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 $C := S \cup \{u_{2k}\} = \{u_1, \ldots, u_{2k-1}, u_{2k}\}$. 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 $(V_1, \ldots, V_k)$ 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,

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

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

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

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. □

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

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

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

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

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, $H_{C-u_i-u_j} \subseteq H_{C-u_i}$. □

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, $u_i, u_j \notin H_{C-u_i-u_j}$. □

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 $i \in [2k]\backslash\{a, \ell\}$. 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 $u_i \in A_i \backslash (\cup_{j \in [2k]\backslash\{a,i,\ell\}} A_j)$ for every $i \in [2k] \setminus \{a, \ell\}$. Hence, the sets U, R, and S', and the cuts $(\overline{A_i}, A_i)$ for $i \in [2k]\backslash\{a, \ell\}$ satisfy the conditions of Theorem 1.5. Therefore, by the first conclusion of Theorem 1.5, there exists a k-partition $\mathcal{P}'$ with

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

By assumption, $d(H_{C-u_a-u_b}) < d(V_1)$ and $b \in [2k]\backslash\{a, \ell\}$, so $\min\{d(H_{C-u_a-u_i}) \colon i \in [2k]\backslash\{a, \ell\}\} < d(V_1)$. 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 $i \in [2k]\backslash\{a, \ell\}$. Therefore,

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

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

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,

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

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

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

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

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

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

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

□

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,

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

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,

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

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

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

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. □

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 $(Y_1, \ldots, Y_{2k-1}, W, Z)$ 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,

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

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,

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

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

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

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* [7, 9, 10, 14]. 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 give an alternative proof of this theorem based on the three-cut-set-lemma in Sect. 6.

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 serves a similar role in our proof to that of Claim 4.3 in the proof of Theorem 1.3. Unlike Claim 4.3, Claim 5.1 places no restriction on the size of the sets involved. This is important for obtaining the stronger structural result of Theorem 1.4. □

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, $d(H_X) = d(V_1)$. □

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. We note that the following claim and its proof are identical to Claim 4.1 and its proof with $2k$ replaced by 3.

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,

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

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

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

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

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

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. □

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. 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 $2k$ in order to invoke Claim 4.3. Here, we are able to prove Claims 5.3 and 5.4 with the size of C being $2k - 1$ (for $k = 2$). This is because we can use Claim 5.1 (which holds only for $k = 2$) instead of Claim 4.3, and Claim 5.1 does not require the size of C to be $2k$.

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,

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

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

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

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

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

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

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

□

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,

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

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,

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

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. □

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,

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

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

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

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 Alternative Proof of Stronger Structural Theorem for $k = 2$

In this section we give an alternative proof of Theorem 1.4. This alternative proof was the first proof that we discovered for Theorem 1.4, however we are unable to generalize its proof technique to $k \geq 2$ (i.e., to prove Theorem 1.3). We present this alternative proof since we believe that it is based on a novel 3-cut-set lemma which may be of independent interest. Moreover, it is self-contained and does not rely on Theorem 1.5 (as opposed to the proof presented in Sect. 5).

Notation. Let $G = (V, E)$ be a hypergraph and let $R, S, T, U \subseteq V$ be subsets of vertices. We define

$$E[S] := \{e \in E : e \subseteq S\},$$
$$E(S, T) := \{e \in E : e \subseteq S \cup T \text{ and } e \cap S, e \cap T \neq \emptyset\},$$
$$E(S, T, U) := \{e \in E : e \subseteq S \cup T \cup U \text{ and } e \cap S, e \cap T, e \cap U \neq \emptyset\}, \text{ and}$$
$$E(R, S, T, U) := \{e \in E : e \subseteq R \cup S \cup T \cup U \text{ and } e \cap R, e \cap S, e \cap T, e \cap U \neq \emptyset\}.$$

Our proof approach is similar to the one used in [17] to prove an analogous structural theorem for graphs. Their proof relies on the *submodular triple inequality* which states that for every graph $G = (V, E)$ and every $X, Y, Z \subseteq V$, we have

$$d((Y \cap Z) \setminus X) + d((X \cap Z) \setminus Y) + d((X \cap Y) \setminus Z) + d(V \setminus (X \cup Y \cup Z)) \leq d(X) + d(Y) + d(Z).$$

Unfortunately, the submodular triple inequality fails to hold in hypergraphs (e.g., consider a hypergraph $G = (V, E)$ where $V = \{1, 2, 3, 4\}$, and $E = \{V\}$, with $X = \{1, 2\}$, $Y = \{1, 3\}$, and $Z = \{2, 3\}$). Our proof of Theorem 1.4 instead relies on the following novel Hypergraph 3-cut-set lemma.

Lemma 6.1 (Hypergraph 3-cut-set lemma) *Let $G = (V, E)$ be a hypergraph, and let $X, Y, Z \subseteq V$ be non-empty and pairwise disjoint sets such that $(X, \overline{X})$, $(Y, \overline{Y})$, $(Z, \overline{Z})$, $(X \cup Y, \overline{X \cup Y})$, $(X \cup Z, \overline{X \cup Z})$, $(Y \cup Z, \overline{Y \cup Z})$, and $(X \cup Y \cup Z, \overline{X \cup Y \cup Z})$ are all minimum cuts. Then,*

$$\delta(X) = \delta(Y) = \delta(Z).$$

Proof Let $W := \overline{X \cup Y \cup Z}$. Since $(X \cup Y \cup Z, \overline{X \cup Y \cup Z})$ is a minimum cut, $W \neq \emptyset$. Since $(X, \overline{X})$ and $(X \cup Y, \overline{X \cup Y})$ are both minimum cuts, we have that $d(X) = d(X \cup Y)$. The hyperedges which are in $\delta(X)$ but not $\delta(X \cup Y)$ are the hyperedges of $E(X, Y)$. The hyperedges which are in $\delta(X \cup Y)$ but not $\delta(X)$ are the

hyperedges of $E(Y, Z)$, $E(Y, W)$ and $E(Y, Z, W)$. Thus, we have that

$$|E(X, Y)| = |E(Y, Z)| + |E(Y, W)| + |E(Y, Z, W)|. \tag{17}$$

We can symmetrically derive versions of Eq. (17) for $|E(X, Z)|$, $|E(Y, X)|$, $|E(Y, Z)|$, $|E(Z, X)|$, and $|E(Z, Y)|$. Summing these six equations together we have that

$$\sum_{A,B\in\{X,Y,Z\},A\neq B} |E(A, B)| = \left(\sum_{A,B\in\{X,Y,Z\},A\neq B} |E(A, B)| \right) + \sum_{A,B\in\{X,Y,Z\},A\neq B} (|E(A, W)| + |E(A, B, W)|) .$$

Thus, we have that

$$\sum_{A,B\in\{X,Y,Z\},A\neq B} (|E(A, W)| + |E(A, B, W)|) = 0.$$

Therefore we have that for every $A, B \in \{X, Y, Z\}$ with $A \neq B$, $E(A, W) = E(A, B, W) = \emptyset$. Therefore, we have that

$$\begin{aligned} d(X) &= |E(X, Y)| + |E(X, Z)| + |E(X, W)| \\ &\quad + |E(X, Y, Z)| + |E(X, Y, W)| + |E(X, Z, W)| + |E(X, Y, Z, W)| \\ &= |E(X, Y)| + |E(X, Z)| + |E(X, Y, Z)| + |E(X, Y, Z, W)|. \end{aligned}$$

We also have that

$$\begin{aligned} d(X \cup Y \cup Z) &= |E(X, W)| + |E(Y, W)| + |E(Z, W)| + |E(X, Y, W)| \\ &\quad + |E(X, Z, W)| + |E(Y, Z, W)| + |E(X, Y, Z, W)| = |E(X, Y, Z, W)|. \end{aligned}$$

Since $(X \cup Y \cup Z, \overline{X \cup Y \cup Z})$ is a minimum cut, we have that $d(X) = d(X \cup Y \cup Z)$. Thus, by the two equations above, we have that

$$|E(X, Y)| + |E(X, Z)| + |E(X, Y, Z)| + |E(X, Y, Z, W)| = |E(X, Y, Z, W)|.$$

Thus,

$$|E(X, Y)| + |E(X, Z)| + |E(X, Y, Z)| = 0.$$

Therefore, we have that $E(X, Y) = E(X, Z) = E(X, Y, Z) = \emptyset$. By a symmetric argument, we can conclude that $E(Y, Z) = \emptyset$. Then, we have that

$$\delta(X) = \delta(Y) = \delta(Z) = \delta(X \cup Y \cup Z) = E(X, Y, Z, W).$$

□

We will repeatedly use the well-known uncrossing result given in Lemma 6.2 below. We present a proof for the sake of completeness. We emphasize that the lemma relies only on the submodularity of the hypergraph cut function and in fact, holds even for submodular functions. We say that two sets A, B cross if $A \cap B$, $A \cap \overline{B}$, $\overline{A} \cap B$, and $\overline{A} \cap \overline{B}$ are all non-empty.

Lemma 6.2 *Let $G = (V, E)$ be a hypergraph. Let $A, B \subseteq V$ and $\emptyset \subsetneq S \subseteq A \cap B$, $\emptyset \subsetneq T \subseteq \overline{A} \cap \overline{B}$ be sets such that $(A, \overline{A})$ and $(B, \overline{B})$ are minimum (S, T)-terminal cuts. If A and B cross, then $(A \cap B, \overline{A \cap B})$ and $(A \cup B, \overline{A \cup B})$ are also minimum (S, T)-terminal cuts.*

Proof Since $S \subseteq A \cap B$ and $T \subseteq \overline{A} \cap \overline{B}$, we have that $(A \cap B, \overline{A \cap B})$ and $(A \cup B, \overline{A \cup B})$ are (S, T)-terminal cuts. Since $(A, \overline{A})$ and $(B, \overline{B})$ are minimum (S, T)-terminal cuts, we have that

$$d(A) + d(B) \leq d(A \cap B) + d(A \cup B).$$

Since the cut function of a hypergraph is submodular, we also have that

$$d(A \cap B) + d(A \cup B) \leq d(A) + d(B).$$

Thus, we conclude that

$$d(A \cap B) + d(A \cup B) = d(A) + d(B),$$

and therefore $(A \cap B, \overline{A \cap B})$ and $(A \cup B, \overline{A \cup B})$ are both minimum (S, T)-terminal cuts. □

Corollary 6.1 *Let $A, B \subseteq V$ be sets such that $(A, \overline{A})$ and $(B, \overline{B})$ are minimum cuts. If A and B cross, then $(A \cap B, \overline{A \cap B})$ and $(A \cup B, \overline{A \cup B})$ are also minimum cuts.*

We now restate and prove Theorem 1.4.

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 $T \subseteq V$ be arbitrary and let $\mathcal{H} := \{U \subseteq V \setminus T : (U, \overline{U})$ is a minimum cut in $G\}$. Suppose for contradiction that the theorem does not hold for some $V_1 \in \mathcal{H}$. For every $a, b \in V_1$, let H_{ab} be the minimal set in $\mathcal{H}$ such that $a, b \in H_{ab}$ and $\delta(H_{ab}) \neq \delta(V_1)$ (a set H_{ab} with $a, b \in H_{ab}$ and $\delta(H_{ab}) \neq \delta(V_1)$ exists since we have assumed that the theorem does not hold for V_1). □

Claim 6.1 *$H_{ab} \subseteq V_1$ for every $a, b \in V_1$.*

Proof Suppose for contradiction that there exist $a, b \in V_1$ such that $H_{ab} \not\subseteq V_1$. Then $H_{ab} \cap \overline{V_1} \neq \emptyset$. If $\overline{H_{ab}} \cap V_1 = \emptyset$, then $V_1 \subseteq H_{ab}$, and hence, V_1 contradicts the minimality of H_{ab}. So, $\overline{H_{ab}} \cap V_1 \neq \emptyset$. Since $a \in H_{ab} \cap V_1$, and $T \subseteq \overline{H_{ab}} \cap \overline{V_1}$, the sets H_{ab} and V_1 cross. Consequently, by Corollary 6.1, the partition $(H_{ab} \cap V_1, \overline{H_{ab}} \cap \overline{V_1})$ is also a minimum cut, and thus $H_{ab} \cap V_1 \in \mathcal{H}$. This contradicts the minimal choice of H_{ab}. □

Claim 6.2 *There exist distinct vertices $x, y, z \in V_1$ such that $x \notin H_{yz}$, $y \notin H_{xz}$, $z \notin H_{xy}$, and the three hyperedge sets $\delta(H_{xy} \cap H_{xz})$, $\delta(H_{xy} \cap H_{yz})$, and $\delta(H_{xz} \cap H_{yz})$ are not all equal.*

Proof Let $a, b \in V_1$ be vertices such that $|H_{ab}|$ is maximum. By the definition of H_{ab}, we have that $\delta(H_{ab}) \neq \delta(V_1)$, and since $(H_{ab}, \overline{H_{ab}})$ is a minimum cut we must have $d(H_{ab}) = d(V_1)$. Therefore, $\delta(V_1) \backslash \delta(H_{ab}) \neq \emptyset$, so let $e \in \delta(V_1) \backslash \delta(H_{ab})$, and let $z \in e \cap V_1$. By Claim 6.1, we have that $H_{ab} \subseteq V_1$. Since $z \in e$ and $e \in \delta(V_1) \backslash \delta(H_{ab})$, we have that $z \notin H_{ab}$. Let $p \in H_{ab}$ be a vertex such that $|H_{pz} \cap H_{ab}|$ is maximum. Since $z \in H_{pz} \backslash H_{ab}$, we have that $H_{ab} \neq H_{pz}$. Thus, by the choice of a and b, we have that $H_{ab} \not\subseteq H_{pz}$. Therefore, $H_{ab} \backslash H_{pz} \neq \emptyset$. Let $y \in H_{ab} \backslash H_{pz}$. By our choice of p, we have that $|H_{pz} \cap H_{ab}| \geq |H_{yz} \cap H_{ab}|$. Since $y \in (H_{yz} \cap H_{ab}) \backslash (H_{pz} \cap H_{ab})$, we have that $(H_{pz} \cap H_{ab}) \backslash H_{yz} \neq \emptyset$. Let $x \in (H_{pz} \cap H_{ab}) \backslash H_{yz}$.

We now analyze relationships between some of the sets that we have defined. □

Proposition 6.1 $y \notin H_{xz}$.

Proof Suppose for contradiction that $y \in H_{xz}$. Then, since $y \notin H_{pz}$, we have that $H_{xz} \cap \overline{H_{pz}} \neq \emptyset$. By the choice of p, we have that $|H_{pz} \cap H_{ab}| \geq |H_{xz} \cap H_{ab}|$. Since $y \in (H_{xz} \cap H_{ab}) \backslash H_{pz}$, we have that $(H_{pz} \cap H_{ab}) \backslash H_{xz} \neq \emptyset$. Thus, $\overline{H_{xz}} \cap H_{pz} \neq \emptyset$. Since $z \in H_{xz} \cap H_{pz}$, we have $H_{xz} \cap H_{pz} \neq \emptyset$. By Claim 6.1, we have $\overline{H_{xz}} \cap \overline{H_{pz}} \neq \emptyset$. Thus, H_{xz} and H_{pz} cross. Therefore, by Corollary 6.1, we have $H_{xz} \cap H_{pz} \in \mathcal{H}$. Since $H_{xz} \cap H_{pz} \subsetneq H_{xz}$ and $x, z \in H_{xz} \cap H_{pz}$, by the choice of x and z, the set $H_{xz} \cap H_{pz}$ contradicts the minimality of H_{xz}. Therefore, we conclude that $y \notin H_{xz}$. □

Proposition 6.2 $H_{xy} \subseteq H_{ab}$.

Proof Suppose for contradiction that $H_{xy} \not\subseteq H_{ab}$. By our choice of a, b, we cannot have $H_{ab} \subseteq H_{xy}$ either. We know that $x, y \in H_{xy} \cap H_{ab}$, and since $H_{xy}, H_{ab} \subseteq V_1$ (by Claim 6.1), we have that $\overline{H_{xy}} \cap \overline{H_{ab}} \neq \emptyset$. Therefore, H_{xy} and H_{ab} cross. Thus, by Corollary 6.1, we have that $H_{ab} \cap H_{xy} \in \mathcal{H}$. Since $H_{xy} \cap H_{ab} \subsetneq H_{xy}$ and $x, y \in H_{ab} \cap H_{xy}$, we have that the set $H_{ab} \cap H_{xy}$ contradicts the minimality of H_{xy}. Thus, we conclude that $H_{xy} \subseteq H_{ab}$. □

We now show that $x \notin H_{yz}$, $y \notin H_{xz}$, and $z \notin H_{xy}$. We have the following facts:

1. By our choice of x, we have that $x \notin H_{yz}$.
2. By Proposition 6.1, we have that $y \notin H_{xz}$.
3. By our choice of z, we have that $z \notin H_{ab}$, and therefore by Proposition 6.2, we have that $z \notin H_{xy}$.

Finally we note that the hyperedge e which we selected when choosing z contains both z and some vertex r in $\overline{V_1}$. By definition, we have that $z \in H_{xz}, H_{yz}$. By Claim 6.1, $H_{xz}, H_{yz} \subseteq V_1$. Therefore, $e \in \delta(H_{xz} \cap H_{yz})$. By Claim 6.1, we have that $H_{ab} \subseteq V_1$, and hence, $r \notin H_{ab}$. However, by choice of e, we have that $e \notin \delta(H_{ab})$, and hence $e \cap H_{ab} = \emptyset$. Therefore, by Proposition 6.2, we have that $e \cap H_{xy} = \emptyset$, and hence $e \cap (H_{xy} \cap H_{xz}) = \emptyset$. Thus, $e \notin \delta(H_{xy} \cap H_{xz})$. Therefore, we conclude that $\delta(H_{xz} \cap H_{yz}) \neq \delta(H_{xy} \cap H_{xz})$, and hence the sets $\delta(H_{xy} \cap H_{xz})$, $\delta(H_{xy} \cap H_{yz})$, and $\delta(H_{xz} \cap H_{yz})$ are not all equal. This completes the proof of Claim 6.2. □

Claim 6.3 *Let $x, y, z \in V_1$ be such that $x \notin H_{yz}$, $y \notin H_{xz}$, and $z \notin H_{xy}$. Then*

$$H_{xy} \setminus (H_{xz} \cup H_{yz}) = H_{xz} \setminus (H_{xy} \cup H_{yz}) = H_{yz} \setminus (H_{xy} \cup H_{xz}) = \emptyset.$$

Proof We will prove that $H_{xy} \setminus (H_{xz} \cup H_{yz}) = \emptyset$. The arguments for $H_{xz} \setminus (H_{xy} \cup H_{yz})$ and $H_{yz} \setminus (H_{xy} \cup H_{xz})$ are similar.

Let $Q := H_{xy} \setminus (H_{xz} \cup H_{yz})$. We note that $x \in H_{xy} \cap H_{xz}$, and therefore $H_{xy} \setminus Q$ is non-empty. Therefore $(H_{xy} \setminus Q, \overline{H_{xy} \setminus Q})$ is a cut. Since $(H_{xy}, \overline{H_{xy}})$ is a minimum cut, this means that

$$d(H_{xy}) \leq d(H_{xy} \setminus Q). \tag{18}$$

The hyperedges which are in $\delta(H_{xy})$ but not in $\delta(H_{xy} \setminus Q)$ are the hyperedges in $E(Q, \overline{H_{xy}})$. The hyperedges which are in $\delta(H_{xy} \setminus Q)$ but not $\delta(H_{xy})$ are the hyperedges in $E(Q, H_{xy} \setminus Q)$. Thus, inequality (18) implies that

$$|E(Q, \overline{H_{xy}})| \leq |E(Q, H_{xy} \setminus Q)|. \tag{19}$$

Next we note that $x \in H_{xz} \setminus H_{yz}$, $y \in H_{yz} \setminus H_{xz}$, $z \in H_{xz} \cap H_{yz}$, and, by Claim 6.1, $\overline{V_1} \subseteq \overline{H_{xz}} \cap \overline{H_{yz}}$. Therefore, H_{xz} and H_{yz} cross. Thus, by Corollary 6.1, we have that $H_{xz} \cup H_{yz} \in \mathcal{H}$. If $Q = \emptyset$ we are done. Otherwise, we have that $H_{xy} \cap \overline{(H_{xz} \cup H_{yz})} = Q \neq \emptyset$, $y \in H_{xy} \cap (H_{xz} \cup H_{yz})$, $z \in \overline{H_{xy}} \cap (H_{xz} \cup H_{yz})$, and $\overline{V_1} \subseteq \overline{H_{xy}} \cap \overline{(H_{xz} \cup H_{yz})}$. Thus, H_{xy} and $H_{xz} \cup H_{yz}$ cross. Therefore, by Corollary 6.1, we have that $H_{xy} \cup H_{xz} \cup H_{yz} \in \mathcal{H}$. Therefore, we have that

$$d(H_{xz} \cup H_{yz}) = d(H_{xy} \cup H_{xz} \cup H_{yz}) = d(Q \cup (H_{xz} \cup H_{yz})). \tag{20}$$

Since Q and $H_{xz} \cup H_{yz}$ are disjoint, we have that the hyperedges which are in $\delta(H_{xz} \cup H_{yz})$ but not in $\delta(Q \cup (H_{xz} \cup H_{yz}))$ are the hyperedges of $E(Q, H_{xz} \cup H_{yz})$, and the hyperedges which are in $\delta(Q \cup (H_{xz} \cup H_{yz}))$, but not $\delta(H_{xz} \cup H_{yz})$ are the hyperedges of $E(Q, \overline{H_{xy} \cup H_{xz} \cup H_{yz}})$. Thus, Eq. (20) implies that

$$|E(Q, H_{xz} \cup H_{yz})| = |E(Q, \overline{H_{xy} \cup H_{xz} \cup H_{yz}})|. \tag{21}$$

Therefore, we have that

$$\begin{aligned} |E(Q, \overline{H_{xy}})| &\geq |E(Q, \overline{H_{xy} \cup H_{xz} \cup H_{yz}})| \\ &= |E(Q, H_{xz} \cup H_{yz})| \end{aligned}$$

$$\begin{aligned}
&\geq |E(Q, (H_{xz} \cup H_{yz}) \cap H_{xy})| \\
&= |E(Q, H_{xy} \setminus Q)| \\
&\geq |E(Q, \overline{H_{xy}})|.
\end{aligned}$$

Here the equation on the second line comes from Eq. (21) and the inequality on the last line comes from Inequality (19). Since the first and last terms in the inequality chain are the same, equality holds throughout. Therefore, $|E(Q, \overline{H_{xy}})| = |E(Q, H_{xy} \setminus Q)|$, and so $d(H_{xy}) = d(H_{xy} \setminus Q)$. Thus, $H_{xy} \setminus Q \in \mathcal{H}$. Since $x, y \in H_{xz} \cup H_{yz}$, $x, y \in H_{xy} \setminus Q$. Furthermore $H_{xy} \setminus Q \subseteq H_{xy}$. Since H_{xy} is a minimial element of $\mathcal{H}$ containing x and y, it must be that $H_{xy} \setminus Q = H_{xy}$, and thus $Q = \emptyset$. □

The next claim completes the proof of Theorem 1.4, because the conclusion of this claim is in direct contradiction with the conclusion of Lemma 6.1. □

Claim 6.4 *There exist non-empty and pairwise disjoint sets $X, Y, Z \subseteq V_1$ such that $X, Y, Z, X \cup Y, X \cup Z, Y \cup Z, X \cup Y \cup Z \in \mathcal{H}$, and $\delta(X), \delta(Y), \delta(Z)$ are not all equal.*

Proof Let $x, y, z \in V_1$ be such that $x \notin H_{yz}$, $y \notin H_{xz}$, and $z \notin H_{xy}$, and the three hyperedge sets $\delta(H_{xy} \cap H_{xz})$, $\delta(H_{xy} \cap H_{yz})$, and $\delta(H_{xz} \cap H_{yz})$ are not all equal. Such x, y, z exist by Claim 6.2. Let

$$\begin{aligned}
X' &:= H_{xy} \cap H_{xz}, \\
Y' &:= H_{xy} \cap H_{yz}, \text{ and} \\
Z' &:= H_{xz} \cap H_{yz}.
\end{aligned}$$

By Claim 6.3, we have that $H_{xy} \setminus (H_{xz} \cup H_{yz})$, $H_{xz} \setminus (H_{xy} \cup H_{yz})$, and $H_{yz} \setminus (H_{xy} \cup H_{xz})$ are all empty, and hence $H_{xy} = X' \cup Y'$, $H_{xz} = X' \cup Z'$, and $H_{yz} = Y' \cup Z'$. Let

$$\begin{aligned}
U &:= H_{xy} \cap H_{xz} \cap H_{yz}, \\
W &:= \overline{H_{xy} \cup H_{xz} \cup H_{yz}}, \\
X &:= X' \setminus U, \\
Y &:= Y' \setminus U, \text{ and} \\
Z &:= Z' \setminus U.
\end{aligned}$$

Since H_{xy} and H_{xz} cross (because $x \in H_{xy} \cap H_{xz}$, $y \in H_{xy} \setminus H_{xz}$, and $z \in H_{xz} \setminus H_{xy}$), we have that $X' \in \mathcal{H}$. Since $x \notin H_{yz}$, we have that $x \notin U$, and hence $x \in X$, so $(X, \overline{X})$ is a cut. Therefore,

$$d(X) \geq d(X'). \tag{22}$$

The hyperedges which are in $\delta(X)$ but not $\delta(X')$ are the hyperedges of $E(X, U)$. The hyperedges which are in $\delta(X')$ but not $\delta(X)$ are the hyperedges of $E(U, Y)$, $E(U, Z)$,

$E(U, W)$, $E(U, Y, Z)$, $E(U, Y, W)$, $E(U, Z, W)$, and $E(U, Y, Z, W)$. Thus, Eq. (22) implies that

$$|E(X, U)| \geq |E(U, Y)| + |E(U, Z)| + |E(U, W)| + |E(U, Y, Z)| + |E(U, Y, W)| + |E(U, Z, W)| + |E(U, Y, Z, W)|.$$

We can derive symmetric inequalities for $|E(Y, U)|$ and $|E(Z, U)|$ by exchanging variable names in the preceding argument. Summing these three inequalities together, we have that

$$\begin{aligned}\sum_{A \in \{X,Y,Z\}} |E(A, U)| \geq 2 \sum_{A \in \{X,Y,Z\}} (|E(A, U)| + |E(A, U, W)|) + 3|E(U, W)| \\ + |E(U, X, Y)| + |E(U, X, Z)| + |E(U, Y, Z)| \\ + |E(U, X, Y, W)| + |E(U, X, Z, W)| + |E(U, Y, Z, W)|.\end{aligned}$$

Subtracting $|E(X, U)| + |E(Y, U)| + |E(Z, U)|$ from both sides of this inequality, we have that $E(U, X)$, $E(U, Y)$, $E(U, Z)$, $E(U, W)$, $E(U, Y, Z)$, $E(U, Y, W)$, $E(U, Z, W)$, and $E(U, Y, Z, W)$ are all empty. Thus, $\delta(X) = \delta(X')$, so $X \in \mathcal{H}$. Similarly, all of the hyperedges in $\delta(X' \cup Y')$ which are not in $\delta(X \cup Y)$ are from sets that we have concluded are empty, so $\delta(X \cup Y) = \delta(X' \cup Y') = \delta(H_{xy})$. Thus, $X \cup Y \in \mathcal{H}$. Symmetrically, Y, Z, $X \cup Z$, and $Y \cup Z$ are all in $\mathcal{H}$ as well. Since $X \cup Y$ and $Y \cup Z$ cross, we have that $X \cup Y \cup Z \in \mathcal{H}$, by Corollary 6.1. Since $x \in X$, $y \in Y$, $z \in Z$, X, Y, and Z are all non-empty. By definition, X, Y, and Z are disjoint. Since $\{\delta(X), \delta(Y), \delta(Z)\} = \{\delta(X'), \delta(Y'), \delta(Z')\} = \{\delta(H_{xy} \cap H_{xz}), \delta(H_{xy} \cap H_{yz}), \delta(H_{xz} \cap H_{yz})\}$, we have that the three hyperedge sets $\delta(X)$, $\delta(Y)$, $\delta(Z)$ are not all equal, by the guarantee that we obtained from Claim 6.2. Thus, X, Y, Z satisfy the conditions of the claim we are proving. □

7 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 [3, 4, 17, 18, 35]). Our work adds to this rich literature by showing that Enum- Hypergraph- k- Cut can be solved via minimum (S, T)-terminal cuts. In addition to the first deterministic polynomial-time algorithm for solving Enum- Hypergraph- k- Cut, one of the main contributions of this work is an elegant explanation for the number of minimum cut-sets in a n-vertex hypergraph being at most $\binom{n}{2}$ and a simple approach to enumerate all of them in a given hypergraph in deterministic polynomial time based on minimum (S, T)-terminal cuts.

We note that our deterministic run-time to solve Enum- Hypergraph- k- Cut is $n^{O(k^2)}p$, where p is the size of the input hypergraph. Subsequent to the present work, we improved the deterministic run-time to $n^{O(k)}p$ in a separate paper [2]. In that paper, we strengthened Theorem 1.3 in a way that allowed it to solve Enum- Hypergraph- k- Cut as well as to enumerate all minmax k-cut-sets (a k-cut-set $F \subseteq E$ of a hypergraph $H = (V, E)$ is a minmax k-cut-set if there exists a partition $V_1, V_2, \ldots, V_k$

of V such that $\max_{i\in[k]} |\delta(V_i)| \leq \max_{i\in[k]} |\delta(P_i)|$ for every partition $P_1, P_2, \ldots, P_k$ of V and $F = \delta(V_1, \ldots, V_k)$).

We mention an open question raised by our work. For a long time, the known upper bound on the number of minimum k-partitions in connected graphs was $O(n^{2k-2})$ [6, 29, 39] 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 $O(n^k)$ for fixed k which also resulted in a faster randomized algorithm to solve GRAPH- k- CUT for fixed k [19, 23]. 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 $O(n^k)$?

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 [17] via email.

Funding Calvin Beideman, Karthekeyan Chandrasekaran, and Weihang Wang were supported in part by NSF grants CCF-1814613 and CCF-1907937.

Declarations

Conflict of interests No Conflict of interest.

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 [4] 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 [4]. Let $(Y_1, \ldots, Y_p, W, Z)$ be a partition of V. We recall that $\text{cost}(Y_1, \ldots, Y_p, W, Z)$ 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 $\alpha(Y_1, \ldots, Y_p, W, Z)$ be the number of hyperedges that intersect Z and at least two of the sets in $\{Y_1, \ldots, Y_p, W\}$.
3. Let $\beta(Y_1, \ldots, Y_p, Z)$ be the number of hyperedges that are disjoint from Z but intersect at least two of the sets in $\{Y_1, \ldots, Y_p\}$.

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

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

The precise interpretation of the quantity $\sigma(Y_1, \ldots, Y_p, W, Z)$ will not be important for our purposes—see [4] for the interpretation.

The following result from [4] 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 Fig. 1 for an illustration of the sets that appear in the statement of Lemma A.1.

Lemma A.1 [4]. *Let $G = (V, E)$ be a hypergraph and $\emptyset \neq R \subsetneq U \subsetneq V$. Let $S = \{u_1, \ldots, u_p\} \subseteq U \setminus R$ for $p \geq 2$. Let $(\overline{A_i}, A_i)$ be a minimum $((S \cup R)\setminus\{u_i\}, \overline{U})$-terminal cut. Suppose that $u_i \in A_i \setminus (\cup_{j\in[p]\setminus\{i\}} A_j)$ for every $i \in [p]$. Let*

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

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

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

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

The next lemma from [4] 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 [4]. *Let $G = (V, E)$ be a hypergraph, $k \geq 2$ be an integer, and $(Y_1, \ldots, Y_p, W, Z)$ be a partition of V for some integer $p \geq 2k - 2$. Then, there exist distinct $i_1, \ldots, i_{k-1} \in [p]$ such that*

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

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 $S = \{u_1, \ldots, u_p\} \subseteq U \setminus R$ for $p \geq 2k - 2$. Let $(\overline{A_i}, A_i)$ be a minimum $((S \cup R)\setminus\{u_i\}, \overline{U})$-terminal cut. Suppose that $u_i \in A_i \setminus (\cup_{j\in[p]\setminus\{i\}} A_j)$ for every $i \in [p]$. Then, the following two hold:*

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

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

2. *Moreover, if there exists a hyperedge $e \in E$ such that e intersects $W := \cup_{1\leq i<j\leq p}(A_i \cap A_j)$, e intersects $Z := \cap_{i\in[p]} \overline{A_i}$, and e is contained in $W \cup Z$, then the inequality in the previous conclusion is strict.*

Proof For the first conclusion of the theorem, we will use the same proof that appeared in [4]. 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 $(Y_1, \ldots, Y_p, W, Z)$ such that

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

and moreover, $\overline{U} \subseteq W$, 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 $(Y_1, \ldots, Y_p, W, Z)$, we obtain a k-partition $(P_1, \ldots, P_k)$ of V such that $W \cup Z \subseteq P_k$ and

$$\begin{aligned}
\text{cost}(P_1, \ldots, P_k) &\leq \frac{1}{2}\left(\text{cost}(Y_1, \ldots, Y_p, W, Z) + \alpha(Y_1, \ldots, Y_p, W, Z) + \beta(Y_1, \ldots, Y_p, Z)\right) && (23)\\
&\leq \frac{1}{2}\left(\text{cost}(Y_1, \ldots, Y_p, W, Z) + \text{cost}(W, Z) + \alpha(Y_1, \ldots, Y_p, W, Z)\right.\\
&\quad \left. + \beta(Y_1, \ldots, Y_p, Z)\right) && (24)\\
&= \frac{1}{2}\sigma(Y_1, \ldots, Y_p, W, Z) && (25)\\
&\leq \frac{1}{2}\min\{d(A_i) + d(A_j) : i, j \in [p], i \neq j\}. && (26)
\end{aligned}$$

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

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 (24) in the above sequence of inequalities should be strict. □

References

1. Aissi, H., Mahjoub, A., McCormick, T., Queyranne, M.: Strongly polynomial bounds for multiobjective and parametric global minimum cuts in graphs and hypergraphs. Math. Program. **154**(1–2), 3–28 (2015)
2. Beideman, C., Chandrasekaran, K., Wang, W.: Counting and enumerating optimum cut sets for hypergraph k-partitioning problems for fixed k. In: Proceedings of the 49th International Colloquium on Automata, Languages and Programming, ICALP, pp. 16:1–16:22 (2022)
3. Chandrasekaran, K., Chekuri, C.: Min–max partitioning of hypergraphs and symmetric submodular functions. In: Proceedings of the 32nd Annual ACM-SIAM Symposium on Discrete Algorithms, SODA, pp. 1026–1038 (2021)
4. Chandrasekaran, K., Chekuri, C.: Hypergraph k-cut for fixed k in deterministic polynomial time. Math. Oper. Res. **47**, 3380–3399 (2022)
5. Chandrasekaran, K., Xu, C., Yu, X.: Hypergraph k-cut in randomized polynomial time. Math. Program. **186**, 85–113 (2019)
6. Chekuri, C., Quanrud, K., Xu, C.: LP relaxation and tree packing for minimum k-cut. SIAM J. Discrete Math. **34**(2), 1334–1353 (2020)
7. Chekuri, C., Xu, C.: Minimum cuts and sparsification in hypergraphs. SIAM J. Comput. **47**(6), 2118–2156 (2018)
8. Cheng, E.: Edge-augmentation of hypergraphs. Math. Program. **84**, 443–465 (1999)
9. Cunningham, W.: Decomposition of submodular functions. Combinatorica **3**, 53–68 (1980)

10. Cunningham, W., Edmonds, J.: A combinatorial decomposition theory. Can. J. Math **32**, 734–765 (1980)
11. Dahlhaus, E., Johnson, D., Papadimitriou, C., Seymour, P., Yannakakis, M.: The complexity of multiterminal cuts. SIAM J. Comput. **23**(4), 864–894 (1994)
12. Dinitz, E.A., Karzanov, A.V., Lomonosov, M.V.: On the structure of a family of minimum weighted cuts in a graph. In: Fridman, A.A. (ed.) Studies in Discrete Optimization. Nauka Publishers, Moscow (1976)
13. Fox, K., Panigrahi, D., Zhang, F.: Minimum cut and minimum k-cut in hypergraphs via branching contractions. In: Proceedings of the 30th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA, pp. 881–896 (2019)
14. Fujishige, S.: Canonical decompositions of symmetric submodular functions. Discrete Appl. Math. **5**, 175–190 (1983)
15. Fukunaga, T.: Computing minimum multiway cuts in hypergraphs. Discrete Optim. **10**(4), 371–382 (2013)
16. Ghaffari, M., Karger, D., Panigrahi, D.: Random contractions and sampling for hypergraph and hedge connectivity. In: Proceedings of the 28th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA, pp. 1101–1114 (2017)
17. Goemans, M.X., Ramakrishnan, V.S.: Minimizing submodular functions over families of sets. Combinatorica **15**, 499–513 (1995)
18. Goldschmidt, O., Hochbaum, D.: A polynomial algorithm for the k-cut problem for fixed k. Math. Oper. Res. **19**(1), 24–37 (1994)
19. Gupta, A., Harris, D., Lee, E., Li, J.: Optimal Bounds for the k-cut Problem. Preprint in arXiv arXiv:2005.08301 (2020)
20. Gupta, A., Lee, E., Li, J.: An FPT algorithm beating 2-approximation for k-cut. In: Proceedings of the 29th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA, pp. 2821–2837 (2018)
21. Gupta, A., Lee, E., Li, J.: Faster exact and approximate algorithms for k-cut. In: Proceedings of the 59th IEEE Annual Symposium on Foundations of Computer Science, FOCS, pp. 113–123 (2018)
22. Gupta, A., Lee, E., Li, J.: The number of minimum k-cuts: improving the Karger–Stein bound. Proceedings of the 51st ACM Symposium on Theory of Computing, STOC, pp. 229–240 (2019)
23. Gupta, A., Lee, E., Li, J.: The Karger–Stein algorithm is optimal for k-cut. In: Proceedings of the 52nd Annual ACM Symposium on Theory of Computing, STOC, pp. 473–484 (2020)
24. Hamacher, J.W., Picard, J.-C., Queyranne, M.: Ranking the cuts and cut-sets of a network. N. Holl. Math. Stud. **95**, 183–200 (1984)
25. Henzinger, M., Williamson, D.: On the number of small cuts in a graph. Inf. Process. Lett. **59**, 41–44 (1996)
26. Kamidoi, Y., Yoshida, N., Nagamochi, H.: A deterministic algorithm for finding all minimum k-way cuts. SIAM J. Comput. **36**(5), 1329–1341 (2007)
27. Karger, D.: 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, pp. 21–30 (1993)
28. Karger, D.: Minimum cuts in near-linear time. J. ACM **47**(1), 46–76 (2000)
29. Karger, D., Stein, C.: A new approach to the minimum cut problem. J. ACM **43**(4), 601–640 (1996)
30. Lawler, E.: Cutsets and partitions of hypergraphs. Networks **3**, 275–285 (1973)
31. Lokshtanov, D., Saurabh, S., Surianarayanan, V.: A parameterized approximation scheme for Min k- Cut. In: Proceedings of the 61st IEEE Annual Symposium on Foundations of Computer Science, FOCS, pp. 798–809 (2020)
32. Manurangsi, P.: Inapproximability of maximum biclique problems, minimum k-cut and densest at-least-k-subgraph from the small set expansion hypothesis. Algorithms **11**(1), 10 (2018)
33. Nagamochi, H., Ibaraki, T.: Algorithmic Aspects of Graph Connectivity. Cambridge University Press, Cambridge (2008)
34. Nagamochi, H., Nishimura, K., Ibaraki, T.: Computing all small cuts in an undirected network. SIAM J. Discrete Math. **10**(3), 469–481 (1997)
35. Nägele, M., Sudakov, B., Zenklusen, R.: Submodular minimization under congruency constraints. Combinatorica **39**, 1351–1386 (2019)
36. Okumoto, K., Fukunaga, T., Nagamochi, H.: Divide-and-conquer algorithms for partitioning hypergraphs and submodular systems. Algorithmica **62**(3), 787–806 (2012)
37. Ravi, R., Sinha, A.: Approximating k-cuts using network strength as a Lagrangean relaxation. Eur. J. Oper. Res. **186**(1), 77–90 (2008)

38. Saran, H., Vazirani, V.: Finding k cuts within twice the optimal. SIAM J. Comput. **24**(1), 101–108 (1995)
39. Thorup, M.: Minimum *k*-way cuts via deterministic greedy tree packing. In: Proceedings of the 40th Annual ACM Symposium on Theory of Computing, STOC, pp. 159–166 (2008)
40. Vazirani, V., Yannakakis, M.: Suboptimal cuts: their enumeration, weight and number (extended abstract). In: Proceedings of the 19th International Colloquium on Automata, Languages and Programming, ICALP '92, pp. 366–377 (1992)
41. Xiao, M.: An improved divide-and-conquer algorithm for finding all minimum k-way cuts. In: Proceedings of 19th International Symposium on Algorithms and Computation, ISAAC, pp. 208–219 (2008)
42. Xiao, M.: Finding minimum 3-way cuts in hypergraphs. Inf. Process. Lett. **110**(14), 554–558 (2010)
43. Zhao, L.: Approximation algorithms for partition and design problems in networks, Ph.D. thesis, Graduate School of Informatics, Kyoto University, Japan (2002)
44. Zhao, L., Nagamochi, H., Ibaraki, T.: Greedy splitting algorithms for approximating multiway partition problems. Math. Program. **102**(1), 167–183 (2005)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.