Faster Connectivity in Low-Rank Hypergraphs via Expander Decomposition

Calvin Beideman[1(✉)], Karthekeyan Chandrasekaran[1], Sagnik Mukhopadhyay[2], and Danupon Nanongkai[3]

[1] University of Illinois, Urbana-Champaign, Champaign, IL, USA
{calvinb2,karthe}@illinois.edu
[2] University of Sheffield, Sheffield, UK
s.mukhopadhyay@sheffield.ac.uk
[3] University of Copenhagen, Copenhagen, Denmark

Abstract. The connectivity of a hypergraph is the minimum number of hyperedges whose deletion disconnects the hypergraph. We design an $\hat{O}_r(p + \min\{\lambda^{\frac{r-3}{r-1}} n^2, n^r/\lambda^{\frac{r}{r-1}}, \lambda^{\frac{5r-7}{4r-4}} n^{\frac{7}{4}}\})$ (The $\hat{O}_r(\cdot)$ notation hides terms that are subpolynomial in the main parameter and terms that depend only on r) time algorithm for computing hypergraph connectivity, where $p := \sum_{e \in E} |e|$ is the input size of the hypergraph, n is the number of vertices, r is the rank (size of the largest hyperedge), and λ is the connectivity of the input hypergraph. Our algorithm also finds a minimum cut in the hypergraph. Our algorithm is faster than existing algorithms if $r = O(1)$ and $\lambda = n^{\Omega(1)}$. The heart of our algorithm is a structural result showing a trade-off between the number of hyperedges taking part in all minimum cuts and the size of the smaller side of any minimum cut. This structural result can be viewed as a generalization of an acclaimed structural theorem for simple graphs [Kawarabayashi-Thorup, JACM 19 (Fulkerson Prize 2021)]. We extend the framework of expander decomposition to hypergraphs to prove this structural result. In addition to the expander decomposition framework, our faster algorithm also relies on a new near-linear time procedure to compute connectivity when one of the sides in a minimum cut is small.

Keywords: Hypergraphs · Connectivity · Expander decomposition

1 Introduction

A hypergraph $G = (V, E)$ is specified by a vertex set V and a collection E of hyperedges, where each hyperedge $e \in E$ is a subset of vertices. In this work, we address the problem of computing connectivity/global min-cut in hypergraphs with low rank (e.g., constant rank). The *rank* of a hypergraph, denoted r, is the size of the largest hyperedge—in particular, if the rank of a hypergraph is 2, then the hypergraph is a graph. In the global min-cut problem, the input

K. Aardal and L. Sanitá (Eds.): IPCO 2022, LNCS 13265, pp. 70–83, 2022.
https://doi.org/10.1007/978-3-031-06901-7_6

is a hypergraph with hyperedge weights $w : E \to \mathbb{R}_+$, and the goal is to find a minimum weight subset of hyperedges whose removal disconnects the hypergraph. Equivalently, the goal is to find a partition of the vertex set V into two non-empty parts $(C, V \setminus C)$ so as to minimize the weight of the set of hyperedges intersecting both parts. For a subset $C \subseteq V$, we will denote the weight of the set of hyperedges intersecting both C and $V \setminus C$ by $d(C)$, the resulting function $d : V \to \mathbb{R}_+$ as the cut function of the hypergraph, and the weight of a min-cut by $\lambda(G)$ (we will use λ when the graph G is clear from context).

If the input hypergraph is *simple*—i.e., each hyperedge has unit weight and no parallel copies—then the weight of a min-cut is also known as the *connectivity* of the hypergraph. We focus on finding connectivity in hypergraphs. We emphasize that, in contrast to graphs whose representation size is the number of edges, the representation size of a hypergraph $G = (V, E)$ is $p := \sum_{e \in E} |e|$. We note that $p \leq rm$, where r is the rank and m is the number of hyperedges in the hypergraph, and moreover, $r \leq n$, where n is the number of vertices. We emphasize that the number of hyperedges m in a hypergraph could be exponential in the number of vertices.

Previous Work. Since the focus of our work is on simple unweighted hypergraphs, we discuss previous work for computing global min-cut in simple unweighted hypergraphs/graphs (i.e., computing connectivity) here. Although global min-cut in weighted graphs has a rich literature, fast computation of global min-cut in simple unweighted graphs was initiated more recently in a seminal work by Kawarabayashi and Thorup (Fulkerson Prize 2021) [20]. The current fastest algorithms to compute graph connectivity (i.e., when $r = 2$) are randomized and run in time $\tilde{O}(m)$ [11,13,15,18,20,25]. In contrast, algorithms to compute hypergraph connectivity are much slower. Furthermore, for hypergraph connectivity/global min-cut, the known randomized approaches are not always faster than the known deterministic approaches. There are two broad algorithmic approaches for global min-cut in hypergraphs: vertex-ordering and random contraction. We discuss these approaches now.

Nagamochi and Ibaraki [26] introduced a groundbreaking vertex-ordering approach to solve global min-cut in graphs in time $O(mn)$. In independent works, Klimmek and Wagner [21] as well as Mak and Wong [24] gave two different generalizations of the vertex-ordering approach to compute hypergraph connectivity in $O(pn)$ time. Queyranne [29] generalized the vertex-ordering approach further to solve *non-trivial symmetric submodular minimization.*[1] Queyranne's algorithm can be implemented to compute hypergraph connectivity in $O(pn)$ time. Thus, all three vertex-ordering based approaches to compute hypergraph connectivity have a run-time of $O(pn)$. This run-time was improved to $O(p+\lambda n^2)$

[1] The input here is a symmetric submodular function $f : 2^V \to \mathbb{R}$ via an evaluation oracle and the goal is to find a partition of V into two non-empty parts $(C, V \setminus C)$ to minimize $f(C)$. We recall that a function $f : 2^V \to \mathbb{R}$ is symmetric if $f(A) = f(V \setminus A)$ for all $A \subseteq V$ and is submodular if $f(A) + f(B) \geq f(A \cap B) + f(A \cup B)$ for all $A, B \subseteq V$. The cut function of a hypergraph $d : V \to \mathbb{R}_+$ is symmetric and submodular.

by Chekuri and Xu [7]: They designed an $O(p)$-time algorithm to construct a *min-cut-sparsifier*, namely a subhypergraph G' of the given hypergraph with size $p' = O(\lambda n)$ such that $\lambda(G') = \lambda(G)$. Applying the vertex-ordering based algorithm to G' gives the connectivity of G within a run-time of $O(p + \lambda n^2)$.

We emphasize that all algorithms discussed in the preceding paragraph are deterministic. Karger [16] introduced the influential random contraction approach to solve global min-cut in graphs which was adapted by Karger and Stein [17] to design an $\tilde{O}(n^2)$ time algorithm[2]. Kogan and Krauthgamer [22] extended the random contraction approach to solve global min-cut in r-rank hypergraphs in time $\tilde{O}_r(mn^2)$. Ghaffari, Karger, and Panigrahi [12] suggested a non-uniform distribution for random contraction in hypergraphs and used it to design an algorithm to compute hypergraph connectivity in $\tilde{O}((m + \lambda n)n^2)$ time. Chandrasekaran, Xu, and Yu [4] refined their non-uniform distribution to obtain an $O(pn^3 \log n)$ time algorithm for global min-cut in hypergraphs. Fox, Panigrahi, and Zhang [10] proposed a branching approach to exploit the refined distribution leading to an $O(p + n^r \log^2 n)$ time algorithm for hypergraph global min-cut, where r is the rank of the input hypergraph. Chekuri and Quanrud [5] designed an algorithm based on isolating cuts which achieves a runtime of $\tilde{O}(\sqrt{pn(m+n)^{1.5}})$ for global min-cut in hypergraphs.

Thus, the current fastest known algorithm to compute hypergraph connectivity is a combination of the algorithms of Chekuri and Xu [7], Fox, Panigrahi, and Zhang [10], and Chekuri and Quanrud [5] with a run-time of

$$\tilde{O}\left(p + \min\left\{\lambda n^2, n^r, \sqrt{pn(m+n)^{1.5}}\right\}\right).$$

1.1 Our Results

In this work, we improve the run-time to compute hypergraph connectivity in low rank simple hypergraphs.

Theorem 1. *[Algorithm] Let G be an r-rank n-vertex simple hypergraph of size p. Then, there exists a randomized algorithm that takes G as input and runs in time*

$$\hat{O}_r\left(p + \min\left\{\lambda^{\frac{r-3}{r-1}} n^2, \frac{n^r}{\lambda^{\frac{r}{r-1}}}, \lambda^{\frac{5r-7}{4r-4}} n^{\frac{7}{4}}\right\}\right)$$

to return the connectivity λ of G with high probability. Moreover, the algorithm returns a min-cut in G with high probability.

Our techniques can also be used to obtain a deterministic algorithm that runs in time

$$\hat{O}_r\left(p + \min\left\{\lambda n^2, \lambda^{\frac{r-3}{r-1}} n^2 + \frac{n^r}{\lambda}\right\}\right).$$

[2] For functions $f(n)$ and $g(n)$ of n, we say that $f(n) = \tilde{O}(g(n))$ if $f(n) = O(g(n)\text{polylog}(n))$ and $f(n) = \hat{O}(g(n))$ if $f(n) = O(g(n)^{1+o(1)})$, where the $o(1)$ is with respect to n. We say that $f(n) = O_r(g(n))$ if $f(n) = O(g(n)h(r))$ for some function h. We define $\tilde{O}_r(f(n))$ and $\hat{O}_r(f(n))$ analogously.

Our deterministic algorithm is faster than Chekuri and Xu's algorithm when r is a constant and $\lambda = \Omega(n^{(r-2)/2})$, while our randomized algorithm is faster than known algorithms if r is a constant and $\lambda = n^{\Omega(1)}$. We summarize the previous fastest algorithms and our results in Table 1.

Table 1. Comparison of results to compute hypergraph connectivity (simple unweighted r-rank n-vertex m-hyperedge p-size hypergraphs with connectivity λ).

	Deterministic	**Randomized**
Previous run-time	$O(p + \lambda n^2)$ [7]	$\tilde{O}(p + \min\{\lambda n^2, n^r, \sqrt{pn(m+n)^{1.5}}\})$ [5,7,10]
Our run-time	$\hat{O}_r\left(p + \min\left\{\lambda n^2, \lambda^{\frac{r-3}{r-1}} n^2 + \frac{n^r}{\lambda}\right\}\right)$	$\hat{O}_r\left(p + \min\left\{\lambda^{\frac{r-3}{r-1}} n^2, \frac{n^r}{\lambda^{\frac{r}{r-1}}}, \lambda^{\frac{5r-7}{4r-4}} n^{\frac{7}{4}}\right\}\right)$

Our algorithm for Theorem 1 proceeds by considering two cases: either (i) the hypergraph has a min-cut where one of the sides is small or (ii) both sides of every min-cut in the hypergraph are large. To account for case (i), we design a near-linear time algorithm to compute a min-cut; to account for case (ii), we perform contractions to reduce the size of the hypergraph without destroying a min-cut and then run known algorithms on the smaller-sized hypergraph leading to savings in run-time. Our contributions in this work are twofold: (1) On the algorithmic front, we design a near-linear time algorithm to find a min-cut where one of the sides is small (if it exists); (2) On the structural front, we show a trade-off between the number of hyperedges taking part in all minimum cuts and the size of the smaller side of any minimum cut (see Theorem 2). This structural result is a generalization of the acclaimed Kawarabayashi-Thorup graph structural theorem [19,20] (Fulkerson prize 2021). We use the structural result to reduce the size of the hypergraph in case (ii). We elaborate on this structural result now.

Theorem 2. *[Structure] Let $G = (V, E)$ be an r-rank n-vertex simple hypergraph with m hyperedges and connectivity λ. Suppose $\lambda \geq r(4r^2)^r$. Then, at least one of the following holds:*

1. *There exists a min-cut $(C, V \setminus C)$ such that*

$$\min\{|C|, |V \setminus C|\} \leq r - \frac{\log\left(\frac{\lambda}{4r}\right)}{\log n},$$

2. *The number of hyperedges in the union of all min-cuts is*

$$O\left(r^{9r^2+2}\left(\frac{6r^2}{\lambda}\right)^{\frac{1}{r-1}} m \log n\right) = \tilde{O}_r\left(\frac{m}{\lambda^{\frac{1}{r-1}}}\right).$$

The Kawarabayashi-Thorup structural theorem for graphs [19,20] states that if every min-cut is non-trivial, then the number of edges in the union of all min-cuts is $O(m/\lambda)$, where a cut is defined to be *non-trivial* if it has at least two vertices on each side. Substituting $r = 2$ in our structural theorem recovers this known Kawarabayashi-Thorup structural theorem for graphs. We emphasize that the Kawarabayashi-Thorup structural theorem for graphs is the backbone of the current fastest algorithms for computing connectivity in graphs and has been proved in the literature via several different techniques [13,15,20,30,31]. Part of the motivation behind our work was to understand whether the Kawarabayashi-Thorup structural theorem for graphs could hold for constant rank hypergraphs and if not, then what would be an appropriate generalization. We discovered that the Kawarabayashi-Thorup graph structural theorem *does not* hold for constant rank hypergraphs: There exist hypergraphs in which (i) the min-cut capacity λ is $\Omega(n)$, (ii) there are no trivial min-cuts, and (iii) the number of hyperedges in the union of all min-cuts is a constant fraction of the number of hyperedges—see the full version of this work [1] for such an example. The existence of such examples suggests that we need an alternative definition of *trivial min-cuts* if we hope to extend the Kawarabayashi-Thorup structural theorem for graphs to r-rank hypergraphs. Conclusion 1 of Theorem 2 can be viewed as a way to redefine the notion of *trivial min-cuts*. We denote the *size* of a cut $(C, V \setminus C)$ to be $\min\{|C|, |V \setminus C|\}$—we emphasize that the size of a cut refers to the size of the smaller side of the cut as opposed to the capacity of the cut. A min-cut is *small-sized* if the smaller side of the cut has at most $r - \log(\lambda/4r)/\log n$ many vertices. With this definition, Conclusion 2 of Theorem 2 can be viewed as a generalization of the Kawarabayashi-Thorup structural theorem to hypergraphs which have no small-sized min-cuts: it says that if there is no small-sized min-cut, then the number of hyperedges in the union of all min-cuts is $\tilde{O}_r(m/\lambda^{\frac{1}{r-1}})$.

We mention that the factor $\lambda^{-1/(r-1)}$ in Conclusion 2 of Theorem 2 cannot be improved: There exist hypergraphs in which every min-cut has at least $\sqrt{n}$ vertices on both sides and the number of hyperedges in the union of all min-cuts is $\Theta(m \cdot \lambda^{-1/(r-1)})$—see the full version of this work [1]. We also note that the structural theorem holds only for *simple* hypergraphs/graphs and is known to fail for weighted graphs. As a consequence, our algorithmic techniques are applicable only in simple hypergraphs and not in weighted hypergraphs.

1.2 Technical Overview

Concepts used in the proof strategy of Theorem 2 will be used in the algorithm of Theorem 1 as well, so it will be helpful to discuss the proof strategy of Theorem 2 before the algorithm. We discuss this now. We define a cut $(C, V \setminus C)$ to be *moderate-sized* if $\min\{|C|, |V \setminus C|\} \in (r - \log(\lambda/4r)/\log n, 4r^2)$ and to be *large-sized* if $\min\{|C|, |V \setminus C|\} \geq 4r^2$; we recall that the cut $(C, V \setminus C)$ is small-sized if $\min\{|C|, |V \setminus C|\} \leq r - \log(\lambda/4r)/\log n$.

Proof Strategy for the Structural Theorem (Theorem 2). We assume that $\lambda > r(4r^2)^r$ as in the statement of Theorem 2. The first step of our proof is to show that every min-cut in a hypergraph is either large-sized or small-sized but not moderate-sized—in particular, we prove that if $(C, V \setminus C)$ is a min-cut with $\min\{|C|, |V \setminus C|\} < 4r^2$, then it is in fact a small-sized min-cut (see

Lemma 2 with the additional assumption that $\lambda > r(4r^2)^r$). Here is the informal argument: For simplicity, we will show that if $(C, V \setminus C)$ is a min-cut with $\min\{|C|, |V \setminus C|\} < 4r^2$, then $\min\{|C|, |V \setminus C|\} \le r$. For the sake of contradiction, suppose that $\min\{|C|, |V \setminus C|\} > r$. The crucial observation is that since the hypergraph has rank r, no hyperedge can contain the smaller side of the min-cut entirely. The absence of such hyperedges means that even if we pack hyperedges in G as densely as possible while keeping $(C, V \setminus C)$ as a min-cut, we cannot pack sufficiently large number of hyperedges to ensure that the degree of each vertex is at least λ. A more careful counting argument extends this approach to show that $\min\{|C|, |V \setminus C|\} \le r - \log \lambda / \log n$.

Now, in order to prove Theorem 2, it suffices to prove Conclusion 2 under the assumption that all min-cuts are large-sized, i.e., $\min\{|C|, |V \setminus C|\} \ge 4r^2$ for every min-cut $(C, V \setminus C)$. Our strategy to prove Conclusion 2 is to find a partition of the vertex set V such that (i) every hyperedge that is completely contained in one of the parts does not cross any min-cut, and (ii) the number of hyperedges that intersect multiple parts (and therefore, possibly cross some min-cut) is small, i.e., $\tilde{O}_r(m \cdot \lambda^{-1/(r-1)})$. To this end, we start by partitioning the vertex set of the hypergraph G into $X_1, \ldots, X_k$ such that the total number of hyperedges intersecting more than one part of the partition is $\tilde{O}_r(m \cdot \lambda^{-1/(r-1)})$ and the subhypergraph induced by each X_i has conductance $\Omega_r(\lambda^{-1/(r-1)})$ (see Sect. 1.3 for the definition of conductance)—such a decomposition is known as an *expander decomposition*. An expander decomposition immediately satisfies (ii) since the number of hyperedges intersecting more than one part is small. Unfortunately, it may not satisfy (i); yet, it is very close to satisfying (i)—we can guarantee that for every min-cut $(C, V \setminus C)$ and every X_i, either C includes very few vertices from X_i, or C includes almost all the vertices of X_i i.e., $\min\{|X_i \cap C|, |X_i \setminus C|\} = O_r(\lambda^{1/(r-1)})$. We note that if $\min\{|X_i \cap C|, |X_i \setminus C|\} = 0$ for every min-cut $(C, V \setminus C)$ and every part X_i then (i) would be satisfied; moreover, if a part X_j is a singleton vertex part (i.e., $|X_j| = 1$), then $\min\{|X_j \cap C|, |X_j \setminus C|\} = 0$ holds. So, our strategy, at this point, is to remove some of the vertices from X_i to form their own singleton vertex parts in the partition in order to achieve $\min\{|X_i \cap C|, |X_i \setminus C|\} = 0$ while controlling the increase in the number of hyperedges that cross the parts. This is achieved by a Trim operation and a series of Shave operations.

The crucial parameter underlying Trim and Shave operations is the notion of degree within a subset: We will denote the degree of a vertex v as $d(v)$ and define the degree contribution of a vertex v inside a vertex set X, denoted by $d_X(v)$, to be the number of hyperedges containing v that are completely contained in X. The Trim operation on a part X_i repeatedly removes from X_i vertices with small degree contribution inside X_i, i.e., $d_{X_i}(v) < d(v)/2r$ until no such vertex can be found. Let X_i' denote the set obtained from X_i after the Trim operation. We note that our partition now consists of $X_1', \ldots, X_k'$ as well as singleton vertex parts for each vertex that we removed with the Trim operation. This operation alone makes a lot of progress towards our goal—we show that $\min\{|X_i' \cap C|, |X_i' \setminus C|\} = O(r^2)$, while the number of hyperedges crossing the partition blows up only by an $O(r)$ factor (see Claims 3 and 4). The little progress that is left to our final goal is achieved by a series of ($O(r^2)$ many) Shave operations. The Shave operation finds the set of vertices in each X_i' whose degree contribution inside X_i' is not very large, i.e., $d_{X_i'}(v) \leq (1-r^{-2})d(v)$ and removes this set of vertices from X_i' in one shot—such vertices are again declared as singleton vertex parts in the partition. We show that the Shave operation strictly reduces $\min\{|X_i' \cap C|, |X_i' \setminus C|\}$ without adding too many hyperedges across the parts (see Claims 3 and 5)—this argument crucially uses the assumption that all min-cuts are large-sized (i.e., $\min\{|C|, |V \setminus C|\} \geq 4r^2$). Because of our guarantee from the Trim operation regarding $\min\{|X_i' \cap C|, |X_i' \setminus C|\}$, we need to perform the Shave operation $O(r^2)$ times to obtain a partition that satisfies conditions (i) and (ii) stated in the preceding paragraph.

Algorithm from Structural Theorem (Theorem 1). We now briefly describe our algorithm: Given an r-rank hypergraph G, we estimate the connectivity λ to within a constant factor in $O(p)$ time using an algorithm of Chekuri and Xu [7]. Next, we use the estimated connectivity value $k = \Theta(\lambda)$ to obtain a subhypergraph G' with size $p' = O_r(\lambda n)$ such that all min-cuts are preserved in time $O(p)$. The rest of the steps are run on this subhypergraph G'. We have two possibilities as stated in Theorem 2. We account for these two possibilities by running two different algorithms: (i) Assuming that some min-cut has size less than $r - \log(\lambda/4r)/\log n$, we design a near-linear time algorithm to find a min-cut. This algorithm is inspired by recent vertex connectivity algorithms, in particular the local vertex connectivity algorithm of [9,28] and the sublinear-time kernelization technique of [23]. This algorithm runs in $\tilde{O}_r(p)$ time. (ii) Assuming that every min-cut is large-sized, we design a fast algorithm to find a min-cut. For this, we find an expander decomposition $\mathcal{X}$ of G', perform a Trim operation followed by a series of $O(r^2)$ Shave operations, and then contract each part of the trimmed and shaved expander decomposition to obtain a hypergraph G''. This reduces the number of vertices in G'' to $O_r(n/\lambda^{1/(r-1)})$ and consequently, running the global min-cut algorithm of either [10] or [6] or [5] (whichever is faster) on G'' leads to an overall run-time of $\hat{O}_r(p + \min\{\lambda^{(r-3)/(r-1)}n^2, n^r/\lambda^{r/(r-1)}, \lambda^{(5r-7)/(4r-4)}n^{7/4}\})$ for step (ii). We return the cheaper of the two cuts found in steps (i) and (ii). The correctness of the algorithm follows by the structural theorem and the total run-time is $\hat{O}_r(p + \min\{n^r/\lambda^{r/(r-1)}, \lambda^{(r-3)/(r-1)}n^2, \lambda^{(5r-7)/(4r-4)}n^{7/4}\})$.

We note here that the expander decomposition framework for graphs was developed in a series of works for the dynamic connectivity problem [8,27,32,33]. Very recently, it has found applications for other problems [2,3,14]. Closer to our application, Saranurak [31] used expander decomposition to give an algorithm to compute edge connectivity in graphs via the use of Trim and Shave operations. The Trim and Shave operations were introduced by Kawarabayashi and Thorup [20] to compute graph connectivity in deterministic $O(m \log^{12} n)$ time. Our line of attack is an adaptation of Saranurak's approach. Since our structural theorem is meant for hypergraph connectivity (and is hence, more complicated than what is used by [31]), we have to work more.

Organization. We prove the structural theorem in Sect. 2. We defer the proof of the algorithmic result and all missing proofs to the full version of the work [1] due to space limitations. We also elaborate on relevant previous work in the full version.

1.3 Preliminaries

Let $G = (V, E)$ be a hypergraph. Let $S, T \subseteq V$ be subsets of vertices. We define $E[S]$ to be the set of hyperedges completely contained in S, $E(S, T)$ to be the set of hyperedges contained in $S \cup T$ and intersecting both S and T, and $E^o(S, T)$ to be the set of hyperedges intersecting both S and T. With this notation, if S and T are disjoint, then $E(S, T) = E[S \cup T] - E[S] - E[T]$ and moreover, if the hypergraph is a graph, then $E(S, T) = E^o(S, T)$. A cut is a partition $(S, V \setminus S)$ where both S and $V \setminus S$ are non-empty. Let $\delta(S) := E(S, V \setminus S)$. For a vertex $v \in V$, we let $\delta(v)$ represent $\delta(\{v\})$. We define the capacity of $(S, V \setminus S)$ as $|\delta(S)|$, and call a cut as a min-cut if it has minimum capacity among all cuts in G. The connectivity of a simple hypergraph G is the capacity of a min-cut in G.

We recall that the *size* of a cut $(S, V \setminus S)$ is $\min\{|S|, |V \setminus S|\}$. We emphasize the distinction between the size of a cut and the capacity of a cut: size is the cardinality of the smaller side of the cut while capacity is the number of hyperedges crossing the cut.

For a vertex $v \in V$ and a subset $S \subseteq V$, we define the degree of v by $d(v) := |\delta(v)|$ and its degree inside S by $d_S(v) := |e \in \delta(v) : e \subseteq S|$. We define $\delta := \min_{v \in V} d(v)$ to be the minimum degree in G. We define $\mathrm{vol}(S) := \sum_{v \in S} d(v)$ and for $T \subseteq V$, $\mathrm{vol}_S(T) := \sum_{v \in T} d_S(v)$. We define the *conductance* of a set $X \subseteq V$ as $\min_{\emptyset \neq S \subsetneq X} \{\frac{|E^o(S, X \setminus S)|}{\min\{\mathrm{vol}(S), \mathrm{vol}(X \setminus S)\}}\}$. For positive integers, $i < j$, we let $[i, j]$ represent the set $\{i, i+1, \ldots, j-1, j\}$. The following proposition will be useful while counting hyperedges within nested sets.

Proposition 1. *Let $G = (V, E)$ be an r-rank n-vertex hypergraph and let $T \subseteq S \subseteq V$. Then,*

$$|E(T, S \setminus T)| \geq \left(\frac{1}{r-1}\right) \left(\mathrm{vol}_S(T) - r\,|E[T]|\right).$$

2 Structural Theorem

We prove Theorem 2 in this section. We call a min-cut $(C, V \setminus C)$ *moderate-sized* if its size $\min\{|C|, |V \setminus C|\}$ is in the range $(r - \log(\lambda/4r)/\log n, (\lambda/2)^{1/r})$. In Sect. 2.1, we show that a hypergraph has no moderate-sized min-cuts. In Sect. 2.2, we define Trim and Shave operations and prove properties about these operations. We prove Theorem 2 in Sect. 2.3. We begin with the following lemma showing the existence of an expander decomposition for low-rank hypergraphs (which follows from the existence of an expander decomposition for graphs).

Lemma 1 (Existential hypergraph expander decomposition). *For every r-rank n-vertex hypergraph $G=(V,E)$ with $p := \sum_{e\in E}|e|$ and every positive real value $\phi \le 1/(r-1)$, there exists a partition $\{X_1,\ldots,X_k\}$ of the vertex set V such that the following hold:*

1. $\sum_{i=1}^{k}|\delta(X_i)| = O(r\phi p\log n)$, and
2. For every $i\in[k]$ and every non-empty set $S\subset X_i$, we have that

$$|E^o(S, X_i\setminus S)| \ge \phi\cdot\min\{\text{vol}(S), \text{vol}(X_i\setminus S)\}.$$

2.1 No Moderate-Sized Min-Cuts

The following lemma is the main result of this section. It shows that there are no moderate-sized min-cuts.

Lemma 2. *Let $G=(V,E)$ be an r-rank n-vertex hypergraph with connectivity λ such that $\lambda \ge r2^{r+1}$. Let $(C, V\backslash C)$ be an arbitrary min-cut. If $\min\{|C|,|V\backslash C|\} > r-\log(\lambda/4r)/\log n$, then $\min\{|C|,|V\setminus C|\}\ge(\lambda/2)^{1/r}$.*

Proof. Without loss of generality, let $|C|=\min\{|C|,|V\setminus C|\}$. Let $t:=|C|$ and $s := r-\log(\lambda/4r)/\log n$. We know that $s<t$. Suppose for contradiction that $t<(\lambda/2)^{1/r}$. We will show that there exists a vertex v with $|\delta(v)|<\lambda$, thus contradicting the fact that λ is the min-cut capacity. We classify the hyperedges of G which intersect C into three types as follows: $E_1 := \{e\in E\colon e\subseteq C\}$, $E_2 := \{e\in E\colon C\subsetneq e\}$, and $E_3 := \{e\in E\colon \emptyset\ne e\cap C\ne C \text{ and } e\cap(V\setminus C)\ne\emptyset\}$. We distinguish two cases:

Case 1: Suppose $t<r$. Then, the number of hyperedges that can be fully contained in C is at most 2^r, so $|E_1|\le 2^r$. Since $(C, V\setminus C)$ is a min-cut, we have that $\lambda = |\delta(C)| = |E_2|+|E_3|$. We note that the number of hyperedges of size i that contain all of C is at most $\binom{n-t}{i-t}$. Hence,

$$|E_2| \le \sum_{i=t+1}^{r}\binom{n-t}{i-t} = \sum_{i=1}^{r-t}\binom{n-t}{i} \le \sum_{i=1}^{r-t} n^i \le 2n^{r-t}.$$

Since each hyperedge in E_3 contains at most $t-1$ vertices of C, a uniform random vertex of C is in such a hyperedge with probability at most $(t-1)/t$. Therefore, if we pick a uniform random vertex from C, the expected number of hyperedges from E_3 incident to it is at most $(\frac{t-1}{t})|E_3|$. Hence, there exists a vertex $v\in C$ such that

$$|\delta(v)\cap E_3| \le \left(\frac{t-1}{t}\right)|E_3| \le \left(\frac{t-1}{t}\right)|\delta(C)| \le \left(\frac{r-1}{r}\right)\lambda.$$

Combining the bounds for E_1, E_2, and E_3, we have that

$$\begin{aligned}|\delta(v)| &= |\delta(v)\cap E_1| + |E_2| + |\delta(v)\cap E_3| \le |E_1| + |E_2| + |\delta(v)\cap E_3|\\ &\le 2^r + 2n^{r-t} + \left(\frac{r-1}{r}\right)\lambda < 2^r + 2n^{r-s} + \left(\frac{r-1}{r}\right)\lambda\\ &= 2^r + \frac{\lambda}{2r} + \left(\frac{r-1}{r}\right)\lambda = \lambda + \frac{r2^{r+1}-\lambda}{2r} \le \lambda.\end{aligned}$$

Case 2: Suppose $t \geq r$. Then, no hyperedge can contain C as a proper subset, so $|E_2| = 0$. For each $v \in C$, the number of hyperedges e of size i such that $v \in e \subseteq C$ is at most $\binom{t-1}{i-1}$. Hence,

$$|\delta(v) \cap E_1| \leq \sum_{i=2}^{r} \binom{t-1}{i-1} = \sum_{i=1}^{r-1} \binom{t-1}{i} \leq \sum_{i=1}^{r-1} t^i \leq 2t^{r-1}.$$

Since each hyperedge in E_3 contains at most $r-1$ vertices of C, a random vertex of C is in such a hyperedge with probability at most $(r-1)/t$. Therefore, if we pick a random vertex from C, the expected number of hyperedges from E_3 incident to it is at most $(\frac{r-1}{t})|E_3|$. Hence, there exists a vertex $v \in C$ such that

$$|\delta(v) \cap E_3| \leq \left(\frac{r-1}{t}\right)|E_3| \leq \left(\frac{r-1}{t}\right)\lambda.$$

Since $t < (\lambda/2)^{1/r}$ and $t \geq r$, we have that $2t^r/\lambda < t - r + 1$. Combining this with our bounds on $|\delta(v) \cap E_1|$ and $|\delta(v) \cap E_3|$, we have that

$$|\delta(v)| = |\delta(v) \cap E_1| + |\delta(v) \cap E_3| \leq 2t^{r-1} + \left(\frac{r-1}{t}\right)\lambda = \left(r - 1 + \frac{2t^r}{\lambda}\right)\frac{\lambda}{t} < \lambda.$$

2.2 Trim and Shave Operations

In this section, we define the trim and shave operations and prove certain useful properties about them. Throughout this section, let $G = (V, E)$ be an r-rank, n-vertex hypergraph with minimum degree δ and min-cut capacity λ. For $X \subseteq V$, let $\textsc{Trim}(X)$ be the set obtained by repeatedly removing from X a vertex v with $d_X(v) < d(v)/2r$ until no such vertices remain, $\textsc{Shave}(X) := \{v \in X : d_X(v) > (1-1/r^2)d(v)\}$, and $\textsc{Shave}_k(X) := \textsc{Shave}(\textsc{Shave}\cdots(\textsc{Shave}(X)))$ be the result of applying k consecutive shave operations to X. We emphasize that $\textsc{Trim}$ is an adaptive operation while $\textsc{Shave}$ is a non-adaptive operation and $\textsc{Shave}_k(X)$ is a sequence of shave operations. The next claim shows that $\textsc{Trim}$ and $\textsc{Shave}$ operations could increase the cut value only by a small factor.

Claim 3. *Let X be a subset of V, $X' := \textsc{Trim}(X)$, and $X'' := \textsc{Shave}(X)$. Then*

1. $|E[X] - E[X']| \leq |\delta(X)|$, $|E[X] - E[X'']| \leq r^2(r-1)|\delta(X)|$, *and*
2. $|\delta(X')| \leq 2|\delta(X)|$, *and* $|\delta(X'')| \leq r^3|\delta(X)|$.

The following claim shows that the $\textsc{Trim}$ operation on a set X that has small intersection with a min-cut further reduces the intersection.

Claim 4. *Let $(C, V \setminus C)$ be a min-cut. Let X be a subset of V and $X' := \textsc{Trim}(X)$. If $\min\{|X \cap C|, |X \cap (V \setminus C)|\} \leq (\delta/6r^2)^{1/(r-1)}$, then*

$$\min\{|X' \cap C|, |X' \cap (V \setminus C)|\} \leq 3r^2.$$

The following claim shows that the Shave operation on a set X which has small intersection with a large-sized min-cut further reduces the intersection.

Claim 5. *Suppose* $\lambda \geq r(4r^2)^r$. *Let* $(C, V \setminus C)$ *be a min-cut with* $\min\{|C|, |V \setminus C|\} \geq 4r^2$. *Let* X' *be a subset of* V *and* $X'' := \textsc{Shave}(X')$. *If* $0 < \min\{|X' \cap C|, |X' \cap (V \setminus C)|\} \leq 3r^2$, *then*

$$\min\{|X'' \cap C|, |X'' \cap (V \setminus C)|\} \leq \min\{|X' \cap C|, |X' \cap (V \setminus C)|\} - 1.$$

Proof. Without loss of generality, we assume that $|X' \cap C| = \min\{|X' \cap C|, |X' \cap (V \setminus C)|\}$. Since $X'' \subseteq X'$, we have that $|X'' \cap C| \leq |X' \cap C|$. Thus, we only need to show that this inequality is strict. Suppose for contradiction that $|X'' \cap C| = |X' \cap C|$. We note that $0 < |X'' \cap C| \leq 3r^2$.

Let $Z := X' \cap C = X'' \cap C$, and let $C' := C - X'$. Since $|C| \geq \min\{|C|, |V \setminus C|\} \geq 4r^2$ and $|Z| \leq 3r^2$, we know that C' is nonempty.

We note that $Z \subseteq X''$. By definition of Shave, we have that $\mathrm{vol}_{X'}(Z) = \sum_{v \in Z} d_{X'}(v) > \sum_{v \in Z} \left(1 - \frac{1}{r^2}\right) d(v) = \left(1 - \frac{1}{r^2}\right) \mathrm{vol}(Z)$.

We note that $|E(Z, V \setminus C)| \geq |E(Z, X' \setminus C)| = |E(Z, X' \setminus Z)|$, so by Proposition 1, we have that $|E(Z, V \setminus C)| \geq |E(Z, X' \setminus Z)| \geq \left(\frac{1}{r-1}\right)(\mathrm{vol}_{X'}(Z) - r|E[Z]|) > \left(\frac{1}{r-1}\right)\left(\left(1 - \frac{1}{r^2}\right)\mathrm{vol}(Z) - r|Z|^r\right)$. We also know from the definition of Shave that $|E(Z, C \setminus Z)| \leq \sum_{v \in Z} |E(\{v\}, C \setminus Z)| \leq \sum_{v \in Z} \frac{1}{r^2} d(v) = \frac{\mathrm{vol}(Z)}{r^2}$.

Thus, using our assumption that $\lambda \geq r(4r^2)^r$, we have that $|E(Z, (V \setminus C))| >$

$$\begin{aligned}
&\left(\frac{1}{r-1}\right)\left(\left(1 - \frac{1}{r^2}\right)\mathrm{vol}(Z) - r|Z|^r\right) = \left(\frac{r^2-1}{r^2(r-1)}\right)\mathrm{vol}(Z) - \left(\frac{r}{r-1}\right)|Z|^r \\
&= \frac{\mathrm{vol}(Z)}{r^2} + \frac{\mathrm{vol}(Z)}{r} - \left(\frac{r}{r-1}\right)|Z|^r \geq \frac{\mathrm{vol}(Z)}{r^2} + \frac{\sum_{v \in Z} d(v)}{r} - r|Z|^r \\
&\geq \frac{\mathrm{vol}(Z)}{r^2} + \frac{|Z|\lambda}{r} - r|Z|^r \geq \frac{\mathrm{vol}(Z)}{r^2} + (4r^2)^r|Z| - r|Z|^r \\
&\geq \frac{\mathrm{vol}(Z)}{r^2} + (4r^2)|Z|^r - r|Z|^r \geq \frac{\mathrm{vol}(Z)}{r^2} \geq |E(Z, C \setminus Z)|.
\end{aligned}$$

We note that $E(Z, (V \setminus C))$ is the set of hyperedges which are cut by C but not C', while $E(Z, C \setminus Z)$ is the set of hyperedges which are cut by C' but not C. Since we have shown that $|E(Z, V \setminus C)| > |E(Z, C \setminus Z)|$, we conclude that $|\delta(C)| > |\delta(C')|$. Since $(C, V \setminus C)$ is a min-cut and $\emptyset \neq C' \subseteq C \subsetneq V$, this is a contradiction.

2.3 Proof of Theorem 2

Proof (Proof of Theorem 2). Suppose the first conclusion does not hold. Then, by Lemma 2, the smaller side of every min-cut has size at least $(\lambda/2)^{1/r} \geq 4r^2$. Let $(C, V \setminus C)$ be an arbitrary min-cut. We use Lemma 1 with $\phi = (6r^2/\lambda)^{1/(r-1)}$ to get an expander decomposition $\mathcal{X} = \{X_1, \ldots, X_k\}$. We note that $\phi \leq 1/(r-1)$ holds by the assumption that $\lambda \geq r(4r^2)^r$. For $i \in [k]$, let $X_i' := \textsc{Trim}(X_i)$ and $X_i'' := \textsc{Shave}_{3r^2}(X_i')$.

Let $i \in [k]$. By the definition of the expander decomposition and our choice of $\phi = (6r^2/\lambda)^{1/(r-1)}$, we have that $\lambda \geq |E^o(X_i \cap C, X_i \cap (V \setminus C))| \geq \left(\frac{6r^2}{\lambda}\right)^{\frac{1}{r-1}} \min\{\text{vol}(X_i \cap C), \text{vol}(X_i \setminus C)\} \geq \left(\frac{6r^2}{\lambda}\right)^{\frac{1}{r-1}} \delta \min\{|X_i \cap C|, |X_i \setminus C|\}$.

Thus, $\min\{|X_i \cap C|, |X_i \setminus C|\} \leq (\lambda/\delta)(\lambda/6r^2)^{1/(r-1)} \leq (\lambda/6r^2)^{1/(r-1)} \leq (\delta/6r^2)^{1/(r-1)}$. Therefore, by Claim 4, we have that $\min\{|X_i' \cap C|, |X_i' \cap (V \setminus C)|\} \leq 3r^2$. We recall that $\lambda \geq r(4r^2)^r$ and every min-cut has size at least $4r^2$. By $3r^2$ repeated applications of Claim 5, we have that $\min\{|X_i'' \cap C|, |X_i'' \cap (V \setminus C)|\} = 0$.

Let $\mathcal{X}'' := \{X_1'', \ldots, X_k''\}$. Since $\min\{|X_i'' \cap C|, |X_i'' \cap (V \setminus C)|\} = 0$ for every min-cut $(C, V \setminus C)$ and every $X_i'' \in \mathcal{X}''$, it follows that no hyperedge crossing a min-cut is fully contained within a single part of $\mathcal{X}''$. Thus, it suffices to show that $|E - \bigcup_{i=1}^k E[X_i'']|$ is small—i.e., the number of hyperedges not contained in any of the parts of $\mathcal{X}''$ is $\tilde{O}_r(m/\lambda^{\frac{1}{r-1}})$.

By the first part of Claim 3, we have that $|E[X_i] - E[X_i']| \leq 2|\delta(X_i)|$ and $|\delta(X_i')| \leq 2|\delta(X_i)|$ for each $i \in [k]$. By the second part of Claim 3, we have that $|\delta(\textsc{Shave}_{j+1}(X_i'))| \leq r^3|\delta(\textsc{Shave}_j(X_i'))|$ for every non-negative integer j. Therefore, by repeated application of the second part of Claim 3, for every $j \in [3r^2]$, we have that $|\delta(\textsc{Shave}_j(X_i'))| \leq 2r^{3j}|\delta(X_i)|$. By the first part of Claim 3, for every $j \in [3r^2]$, we have that $|E[\textsc{Shave}_{j-1}(X_i')] - E[\textsc{Shave}_j(X_i')]| \leq r^3|\delta(\textsc{Shave}_{j-1}(X_i'))| \leq 2r^{3j}|\delta(X_i)|$.

Therefore, $\left|E - \bigcup_{i=1}^k E[X_i'']\right| - \left|E - \bigcup_{i=1}^k E[X_i]\right|$

$$\begin{aligned}
&\leq \sum_{i=1}^{k} |E[X_i] - E[X_i'']| \\
&= \sum_{i=1}^{k} \left(|E[X_i] - E[X_i']| + \sum_{j=1}^{3r^2} |E[\textsc{Shave}_{j-1}(X_i')] - E[\textsc{Shave}_j(X_i')]| \right) \\
&\leq \sum_{i=1}^{k} \left(2|\delta(X_i)| + \sum_{j=1}^{3r^2} 2r^{3j}|\delta(X_i)| \right) = \sum_{i=1}^{k} |\delta(X_i)| \left(2 + \sum_{j=1}^{3r^2} 2r^{3j} \right) \\
&\leq \sum_{i=1}^{k} 3r^{9r^2}|\delta(X_i)| \leq 3r^{9r^2} \sum_{i=1}^{k} |E(X_i, V \setminus X_i)|.
\end{aligned}$$

Hence, $\left|E - \bigcup_{i=1}^k E[X_i'']\right| \leq 4r^{9r^2} \sum_{i=1}^k |E(X_i, V \setminus X_i)|$. By Lemma 1, since $\mathcal{X}$ is an expander decomposition for $\phi = (6r^2/\lambda)^{1/(r-1)}$ and since $p = \sum_{e \in E} |e| \leq mr$, we have that

$$\sum_{i=1}^{k} |E(X_i, V \setminus X_i)| = O(r\phi p \log n) = O(r) \left(\frac{6r^2}{\lambda}\right)^{\frac{1}{r-1}} p \log n = O(r^2) \left(\frac{6r^2}{\lambda}\right)^{\frac{1}{r-1}} m \log n.$$

Thus, $|E - \bigcup_{i=1}^k E[X_i'']| = O(r^{9r^2+2}(6r^2/\lambda)^{1/(r-1)} m \log n)$, thus proving the second conclusion.

Acknowledgement. This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme under grant agreement No 715672. The last two authors are also supported by the Swedish Research Council (Reg. No. 2015-04659 and 2019-05622). Karthekeyan and Calvin are supported in part by NSF grants CCF-1814613 and CCF-1907937.

References

1. Beideman, C., Chandrasekaran, K., Mukhopadhyay, S., Nanongkai, D.: Faster connectivity in low-rank hypergraphs via expander decomposition. CoRR abs/2011.08097 (2021)
2. Bernstein, A., et al.: Fully-dynamic graph sparsifiers against an adaptive adversary. CoRR abs/2004.08432 (2020)
3. Bernstein, A., Gutenberg, M.P., Saranurak, T.: Deterministic decremental reachability, SCC, and shortest paths via directed expanders and congestion balancing. In: FOCS. IEEE Computer Society (2020)
4. Chandrasekaran, K., Xu, C., Yu, X.: Hypergraph k-cut in randomized polynomial time. Mathematical Programming (Preliminary version in SODA 2018), November 2019
5. Chekuri, C., Quanrud, K.: Isolating cuts, (Bi-)submodularity, and faster algorithms for connectivity. In: ICALP, pp. 50:1–50:20 (2021)
6. Chekuri, C., Xu, C.: Computing minimum cuts in hypergraphs. In: SODA, pp. 1085–1100. SIAM (2017)
7. Chekuri, C., Xu, C.: Minimum cuts and sparsification in hypergraphs. SIAM J. Comput. **47**(6), 2118–2156 (2018)
8. Chuzhoy, J., Gao, Y., Li, J., Nanongkai, D., Peng, R., Saranurak, T.: A deterministic algorithm for balanced cut with applications to dynamic connectivity, flows, and beyond. In: FOCS. IEEE Computer Society (2020)
9. Forster, S., Nanongkai, D., Yang, L., Saranurak, T., Yingchareonthawornchai, S.: Computing and testing small connectivity in near-linear time and queries via fast local cut algorithms. In: SODA, pp. 2046–2065. ACM/SIAM (2020)
10. Fox, K., Panigrahi, D., Zhang, F.: Minimum cut and minimum k-cut in hypergraphs via branching contractions. In: SODA, pp. 881–896. SIAM (2019)
11. Gawrychowski, P., Mozes, S., Weimann, O.: Minimum cut in $O(m \log^2 n)$ time. In: ICALP, pp. 57:1–57:15 (2020)
12. Ghaffari, M., Karger, D., Panigrahi, D.: Random contractions and sampling for hypergraph and hedge connectivity. In: SODA, pp. 1101–1114. ACM/SIAM (2017)
13. Ghaffari, M., Nowicki, K., Thorup, M.: Faster algorithms for edge connectivity via random 2-out contractions. In: SODA. ACM/SIAM (2020)
14. Goranci, G., Räcke, H., Saranurak, T., Tan, Z.: The expander hierarchy and its applications to dynamic graph algorithms. In: SODA, pp. 2212–2228 (2021)
15. Henzinger, M., Rao, S., Wang, D.: Local flow partitioning for faster edge connectivity. In: SODA, pp. 1919–1938. ACM/SIAM (2017)
16. Karger, D.: Global min-cuts in RNC, and other ramifications of a simple min-cut algorithm. In: SODA, pp. 21–30. ACM/SIAM (1993)
17. Karger, D., Stein, C.: A new approach to the minimum cut problem. J. ACM **43**(4), 601–640 (1996)
18. Karger, D.R.: Minimum cuts in near-linear time. J. ACM **47**(1), 46–76 (2000). Announced at STOC 1996

19. Kawarabayashi, K., Thorup, M.: Deterministic edge connectivity in near-linear time. In: STOC, pp. 665–674. ACM (2015)
20. Kawarabayashi, K., Thorup, M.: Deterministic edge connectivity in near-linear time. J. ACM **66**(1), 4:1–4:50 (2019)
21. Klimmek, R., Wagner, F.: A simple hypergraph min cut algorithm. Technical report B 96–02, Institute of Computer Science, Freie Universitat (1996)
22. Kogan, D., Krauthgamer, R.: Sketching cuts in graphs and hypergraphs. In: ITCS, pp. 367–376 (2015)
23. Li, J., Nanongkai, D., Panigrahi, D., Saranurak, T., Yingchareonthawornchai, S.: Vertex connectivity in poly-logarithmic max-flows (2021, unpublished)
24. Mak, W.K., Wong, M.D.F.: A fast hypergraph min-cut algorithm for circuit partitioning. Integr.: VLSI J. **30**(1), 1–11 (2000)
25. Mukhopadhyay, S., Nanongkai, D.: Weighted min-cut: sequential, cut-query, and streaming algorithms. In: STOC, pp. 496–509. ACM (2020)
26. Nagamochi, H., Ibaraki, T.: Computing edge-connectivity in multigraphs and capacitated graphs. SIAM J. Discret. Math. **5**(1), 54–66 (1992)
27. Nanongkai, D., Saranurak, T.: Dynamic spanning forest with worst-case update time: adaptive, Las Vegas, and $O(n^{1/2-\epsilon})$-time. In: STOC, pp. 1122–1129. ACM (2017)
28. Nanongkai, D., Saranurak, T., Yingchareonthawornchai, S.: Breaking quadratic time for small vertex connectivity and an approximation scheme. In: STOC, pp. 241–252. ACM (2019)
29. Queyranne, M.: Minimizing symmetric submodular functions. Math. Program. **82**(1–2), 3–12 (1998)
30. Rubinstein, A., Schramm, T., Weinberg, S.M.: Computing exact minimum cuts without knowing the graph. In: ITCS, pp. 39:1–39:16 (2018)
31. Saranurak, T.: A simple deterministic algorithm for edge connectivity. In: SOSA. SIAM (2021)
32. Saranurak, T., Wang, D.: Expander decomposition and pruning: faster, stronger, and simpler. In: SODA, pp. 2616–2635. SIAM (2019)
33. Wulff-Nilsen, C.: Fully-dynamic minimum spanning forest with improved worst-case update time. In: STOC, pp. 1130–1143. ACM (2017)