

On the Cut Dimension of a Graph

Troy Lee ✉

Centre for Quantum Software and Information, University of Technology Sydney, Australia

Tongyang Li ✉

Center for Theoretical Physics, Massachusetts Institute of Technology, Cambridge, MA, USA
Center on Frontiers of Computing Studies, Peking University, Beijing, China

Miklos Santha ✉

CNRS, IRIF, Université de Paris, France

Centre for Quantum Technologies and MajuLab, National University of Singapore, Singapore

Shengyu Zhang ✉

Tencent Quantum Laboratory, Shenzhen, China

Abstract

Let $G = (V, w)$ be a weighted undirected graph with m edges. The cut dimension of G is the dimension of the span of the characteristic vectors of the minimum cuts of G , viewed as vectors in $\{0, 1\}^m$. For every $n \geq 2$ we show that the cut dimension of an n -vertex graph is at most $2n - 3$, and construct graphs realizing this bound.

The cut dimension was recently defined by Graur et al. [13], who show that the maximum cut dimension of an n -vertex graph is a lower bound on the number of cut queries needed by a deterministic algorithm to solve the minimum cut problem on n -vertex graphs. For every $n \geq 2$, Graur et al. exhibit a graph on n vertices with cut dimension at least $3n/2 - 2$, giving the first lower bound larger than n on the deterministic cut query complexity of computing mincut. We observe that the cut dimension is even a lower bound on the number of *linear* queries needed by a deterministic algorithm to solve mincut, where a linear query can ask any vector $x \in \mathbb{R}^{\binom{n}{2}}$ and receives the answer $w^T x$. Our results thus show a lower bound of $2n - 3$ on the number of linear queries needed by a deterministic algorithm to solve minimum cut on n -vertex graphs, and imply that one cannot show a lower bound larger than this via the cut dimension.

We further introduce a generalization of the cut dimension which we call the ℓ_1 -approximate cut dimension. The ℓ_1 -approximate cut dimension is also a lower bound on the number of linear queries needed by a deterministic algorithm to compute minimum cut. It is always at least as large as the cut dimension, and we construct an infinite family of graphs on $n = 3k + 1$ vertices with ℓ_1 -approximate cut dimension $2n - 2$, showing that it can be strictly larger than the cut dimension.

2012 ACM Subject Classification Mathematics of computing \rightarrow Graph theory; Theory of computation \rightarrow Graph algorithms analysis; Theory of computation \rightarrow Computational complexity and cryptography

Keywords and phrases Query complexity, submodular function minimization, cut dimension

Digital Object Identifier 10.4230/LIPIcs.CCC.2021.15

Related Version *Previous Version:* <https://arxiv.org/abs/2011.05085>

Funding Research at CQT is funded by the National Research Foundation, the Prime Minister's Office, and the Ministry of Education, Singapore under the Research Centres of Excellence programme's research grant R-710-000-012-135. In addition, this work has been supported in part by the QuantERA ERA-NET Cofund project QuantAlgo and the ANR project ANR-18-CE47-0010 QUDATA.

Troy Lee: Troy Lee is supported in part by the Australian Research Council Grant No: DP200100950.

Tongyang Li: Tongyang Li is supported by the ARO contract W911NF-17-1-0433, NSF grant PHY-1818914, and an NSF QISE-NET Triplet Award (grant DMR-1747426).



© Troy Lee, Tongyang Li, Miklos Santha, and Shengyu Zhang;
licensed under Creative Commons License CC-BY 4.0

36th Computational Complexity Conference (CCC 2021).

Editor: Valentine Kabanets; Article No. 15; pp. 15:1–15:35

Leibniz International Proceedings in Informatics



Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



1 Introduction

Let $G = (V, w)$ be a weighted undirected n -vertex graph where w is an $\binom{n}{2}$ -dimensional nonnegative real vector assigning a (possibly zero) weight to each edge slot. For a nontrivial subset $\emptyset \neq X \subsetneq V$, let $\Delta(X)$ be the set of edges of G with one endpoint in X and one endpoint in $\bar{X} = V \setminus X$. A *cut* S in G is a subset of edges of the form $\Delta(X)$ for a nontrivial set X . The sets X and \bar{X} are called the *shores* of the cut. For a cut S , its *weight* is the sum of the weights of the edges in S , denoted $w(S)$. The minimum cut problem is to find the minimum of $w(S)$ over all cuts S . The study of algorithms for the minimum cut problem in theoretical computer science goes back at least to the 1960's and has given rise to a vast and beautiful literature. Minimum cut is also a problem of great practical importance with applications to, for example, clustering algorithms and evaluating network reliability. Randomized algorithms can solve the minimum cut problem in nearly linear time: in 1996 Karger gave an algorithm with running time $O(m \log^3(n))$ to compute the minimum cut of a weighted graph with m edges [20]. This was the best known bound until very recently when two independent works improved on it. Gawrychowski, Mozes, and Weimann [8] gave a randomized algorithm with running time $O(m \log^2(n))$ [8] and Mukhopadhyay and Nanongkai [24] gave a randomized algorithm with time complexity $O(m \frac{\log^2(n)}{\log \log n} + n \log^6(n))$. Gawrychowski, Mozes, and Weimann [9] later improved the running time of the Mukhopadhyay and Nanongkai algorithm to $O(m \frac{\log^2(n)}{\log \log n} + n \log^{3+\varepsilon}(n))$.

For *simple* graphs G , randomized algorithms are known with running times $O(m \log(n))$ and $O(m + n \log^3(n))$ [10]. For simple graphs even nearly linear time *deterministic* algorithms are known. Kawarabayashi and Thorup gave an $O(m \log^{12}(n))$ time algorithm [21], which was subsequently improved to $O(m(\log(n) \log \log n)^2)$ by Henzinger, Rao, and Wang [16].

Our work spans two aspects of the study of the minimum cut problem. The first is to query complexity lower bounds on minimum cut. A natural model in which to study the query complexity of minimum cut is for algorithms allowed to make *cut queries*. A cut query algorithm can query any subset $\emptyset \neq X \subsetneq V$ and receives the answer $w(\Delta(X))$. One motivation to study cut query algorithms comes from submodular function minimization. The cut function $f(X) = w(\Delta(X))$ is a submodular function, and finding the minimum cut value is equivalent to finding the minimum value of f over all nontrivial sets X . The problem of minimizing a submodular function is often studied with respect to an evaluation oracle, which in the case of the cut function is exactly a cut query.

Harvey [15] observed that results on the deterministic communication complexity of deciding graph connectivity [14] imply that any deterministic cut query algorithm to compute minimum cut, or even to decide if the graph is connected or not, must make at least cn cut queries, for a constant $c < 1$. Analogous results on the randomized communication complexity of connectivity [2] imply an $\Omega(n/\log(n))$ lower bound on the number of cut queries needed by a randomized algorithm to compute minimum cut (or even connectivity).

On the algorithms side, Rubinfeld, Shramm, and Weinberg [26] gave a randomized algorithm computing the minimum cut of a simple graph with $\tilde{O}(n)$ many cut queries. Recently, Mukhopadhyay and Nanongkai [24] used a different approach based on Karger's 2-respecting tree algorithm [20] to also give a randomized $\tilde{O}(n)$ cut query algorithm to compute minimum cut in a general undirected weighted graph.

For deterministic cut query algorithms, there remains a large gap between the best upper and lower bounds. We are not aware of any deterministic algorithm for minimum cut better than learning the entire graph, which can take $\Omega(n^2/\log(n))$ cut queries in the worst case. On the lower bound side, Graur, Pollner, Ramaswamy, and Weinberg [13] recently introduced a

very interesting lower bound technique called the *cut dimension*, which we now describe. Let $G = (V, w)$ be a weighted undirected graph with n vertices and m edges, and let $\mathcal{M}(G)$ be the set of minimum cuts of G . For a cut $S \in \mathcal{M}(G)$, let $\chi(S) \in \{0, 1\}^m$ be the characteristic vector of S amongst the m edges of G . Let $\vec{\mathcal{M}}(G) = \{\chi(S) : S \in \mathcal{M}(G)\}$. The cut dimension of G , denoted $\text{cdim}(G)$, is the dimension of $\text{span}(\vec{\mathcal{M}}(G))$. It is shown in [13] that for any n -vertex graph G , the cut dimension $\text{cdim}(G)$ is a lower bound on the deterministic cut query complexity of computing minimum cut on weighted n -vertex graphs. Moreover, for every $n \geq 2$ they construct an n -vertex graph G with cut dimension $3n/2 - 2$.

Besides showing lower bounds on cut query complexity, the cut dimension is a natural measure of the complexity of mincuts in a graph. There is a rich literature on the possible structure of mincuts in a graph. Perhaps the first result of this kind is the cactus representation of mincuts by [6]. A cactus for a graph G is a sparse weighted graph C that represents all the mincuts of G . One consequence of the cactus representation is that the number of possible mincuts in an n -vertex weighted graph is at most $\binom{n}{2}$. This upper bound was later given an algorithmic proof via Karger's famous contraction algorithm ([19], Theorem 6.1). The n -vertex cycle graph has $\binom{n}{2}$ many minimum cuts and shows that this bound can be tight.

While the cycle has $\binom{n}{2}$ many mincuts, these cuts live in an n -dimensional space as the n -vertex cycle only has n edges. Is it possible to construct graphs with many cuts that also have high cut dimension? We show that this is not possible, and in fact the cut dimension of an n -vertex graph is at most $2n - 3$.

► **Theorem 1 (Main Upper Bound).** *For any weighted undirected graph G on $n \geq 2$ vertices it holds that $\text{cdim}(G) \leq 2n - 3$.*

Like the cactus representation, this shows another aspect in which the mincuts of a graph are constrained to have a relatively simple structure. We further show that this bound is tight by constructing graphs with cut dimension $2n - 3$ for every $n \geq 2$.

► **Theorem 2 (Main Lower Bound).** *For every $n \geq 2$ there exists an n -vertex weighted undirected graph G with $\text{cdim}(G) = 2n - 3$.*

In addition to shedding further light on the structure of minimum cuts, this improves the best known lower bound on the deterministic cut query complexity of the minimum cut problem to $2n - 3$. We additionally show that the cut dimension is even a lower bound on a stronger query model called the linear query model, recently studied in [1]. In the linear query model, the algorithm can query any vector $x \in \mathbb{R}^{\binom{n}{2}}$ and receives the answer $\langle w, x \rangle$, the inner product of w and x . Linear queries can be much more powerful than cut queries as one can completely learn an unweighted graph with a single linear query. By an information theoretic argument learning an unweighted graph can require $\Omega(n^2/\log(n))$ many cut queries since each cut query reveals at most $O(\log(n))$ bits.

We further introduce a lower bound technique which is a generalization of the cut dimension that we call the ℓ_1 -approximate cut dimension. This technique looks not just at mincuts in the graph, but all cuts. We again look at the span of the dimension of these cuts with an additional twist. Suppose the weight of a minimum cut in G is λ and cut S has $w(S) = \lambda + \delta$. Abusing notation we will let S represent both a set of edges and the characteristic vector $S \in \{0, 1\}^{\binom{n}{2}}$ of S among all edge slots. The vector S can be *perturbed* to $S - u$ for any vector $u \geq 0$ with $\|u\|_{1,w} \leq \delta$. Here $\|u\|_{1,w} = \sum_i |w(i) \cdot u(i)|$ is the ℓ_1 norm of u weighted by the edge weights of the graph. The ℓ_1 -approximate cut dimension of G is then the minimum over all valid perturbations of the dimension of the span of the perturbed cut vectors.

The minimization over all perturbations makes the ℓ_1 -approximate cut dimension a difficult quantity to lower bound. We are able to show, however, that the ℓ_1 -approximate cut dimension can be strictly larger than the cut dimension. For every $k \in \mathbb{N}$ and $n = 3k + 1$, we construct an unweighted n -vertex graph G whose ℓ_1 -approximate cut dimension is $2n - 2$. This has the following application.

► **Theorem 3.** *Any deterministic linear query algorithm that correctly computes the minimum cut of all n -vertex weighted undirected graphs must make at least $2n - 2$ queries in the worst case.*

Computing the minimum cut of a graph with cut queries is a special case of finding the nontrivial minimum of a symmetric submodular function $f : 2^V \rightarrow \mathbb{R}$ with evaluation queries. That is, to find $\min_{X: \emptyset \neq X \subseteq V} f(X)$ for a submodular f that satisfies $f(X) = f(V \setminus X)$ for all $X \subseteq V$. As linear queries are more powerful than cut queries, Theorem 3 also implies a $2n - 2$ evaluation query lower bound for a deterministic algorithm finding a nontrivial minimum of a symmetric submodular function, which is currently the best known.

1.1 Techniques

We give two different proofs of the $2n - 3$ upper bound on the cut dimension and two different techniques to create graphs with cut dimension $2n - 3$. The first proof is direct and uses the *combinatorial uncrossing* technique, and in particular a key lemma of Jain [18] in his factor of 2 approximation algorithm for the survivable network design problem. The second proof is by induction and follows a framework for constructing a cactus representation of the mincuts of a graph [6, 7]. The second proof uses very few properties of mincuts and seems better suited to also upper bound the ℓ_1 -approximate cut dimension, one of our main open questions.

Key to both proofs is the concept of when cuts *cross* each other. Two cuts $\Delta(X), \Delta(Y)$ are said to *cross* if all four of the intersections $X \cap Y, \bar{X} \cap Y, X \cap \bar{Y}, \bar{X} \cap \bar{Y}$ are non-empty. Note that in the definition of crossing it does not matter which shore we take to define the cut, thus crossing is a property of the cuts themselves. A family \mathcal{L} of cuts is called *cross-free* if for all cuts $S, T \in \mathcal{L}$ it holds that S and T do not cross.

First upper and lower bound proof. In the first upper bound proof, we first show that any cross-free family of cuts has cardinality at most $2n - 3$ (see Section 4.1). We then use Jain's lemma [18] (stated in Lemma 19) to conclude that for a maximal cross-free subset $\mathcal{L} \subseteq \mathcal{M}(G)$ it holds that $\vec{\mathcal{L}} = \{\chi(S) : S \in \mathcal{L}\}$ spans the set $\vec{\mathcal{M}}(G)$. This shows that the cut dimension of a graph is at most $2n - 3$.

In the first lower bound proof we use a tree-representation of a cross-free family of cuts to show that in a complete graph the cut vectors of a cross-free family of cuts are linearly independent (Lemma 27). Thus the lower bound reduces to constructing a graph whose minimum cuts are a cross-free family of cuts of size $2n - 3$. Such a construction has already been given by Chandra and Ram [5]. We go a step further, however. For any \mathcal{L} which is a cross-free family of cuts from a complete n -vertex graph with $|\mathcal{L}| = 2n - 3$, in Theorem 31 we explicitly give the edge weights of a complete weighted graph G such that $\mathcal{M}(G) = \mathcal{L}$ and therefore $\text{cdim}(G) = 2n - 3$. This task is made easier by Lemma 29, which states that if \mathcal{L} is a cross-free family of cuts of size $2n - 3$ that all have the same weight, then this must be the weight of a minimum cut of the graph. This lemma is again shown by the combinatorial uncrossing technique. This reduces the construction problem to solving the linear program of finding a positive vector w that makes all cuts in \mathcal{L} have the same weight. We explicitly give a solution to this linear program by viewing it as a flow problem on the tree-representation of \mathcal{L} .

Second upper and lower bound proof. The second upper bound proof is by induction and follows methods to construct a cactus representation of mincuts [6, 7]. In the base case $n = 2$ it is easy to see that the cut dimension is at most $2n - 3 = 1$. For the inductive step, when G is an $n > 2$ vertex graph, there are 3 cases to consider. We call a cut of the form $\Delta(\{v\})$ a *star cut*, and we will refer to all other cuts as *non-star* cuts. The first case is where all cuts in $\mathcal{M}(G)$ are star cuts. As the graph has n vertices there are at most n star cuts and so in this case the cut dimension is at most $n \leq 2n - 3$. The second case is where for every non-star cut $S \in \mathcal{M}(G)$ there is a cut $T \in \mathcal{M}(G)$ which crosses S . In this case [6] show that the graph must be a cycle and the cut dimension is again at most $n \leq 2n - 3$.

The interesting case is where there is a non-star cut $\Delta(V_0) \in \mathcal{M}(G)$ which is not crossed by any other cut in $\mathcal{M}(G)$. Let $V_1 = \bar{V}_0$. In this case we use a decomposition of G along the cut $\Delta(V_0)$, that we call the *separation* of G , into two smaller graphs G_b , for $b \in \{0, 1\}$. The graph G_b is formed from G by contracting V_{1-b} into a single new vertex v_{1-b} . We show that $\text{cdim}(G) \leq \text{cdim}(G_0) + \text{cdim}(G_1) - 1$ which implies immediately the upper bound. Indeed, let $k = |V_0| \geq 2$. Then G_0 is a graph on $k + 1$ vertices and G_1 is a graph on $n - k + 1$ vertices, both of which are less than n . The inductive hypothesis therefore gives $\text{cdim}(G) \leq 2(k + 1) - 3 + 2(n - k + 1) - 3 - 1 = 2n - 3$.

For the second lower bound proof we use the *merge* operation which creates from two graphs G_b , for $b \in \{0, 1\}$, and a specified vertex v_{1-b} from each, a composed graph G where the vertices v_0, v_1 are not present but the cut $\Delta(V_0)$ reflects the structure of the star cuts at v_0 and v_1 in the original graphs. The operations separation and merge are inverses in the sense that if we apply merge to $\{G_0, G_1\}$ followed by separation on the resulting graph G , we receive back $\{G_0, G_1\}$. We also show that the inequality $\text{cdim}(G) \leq \text{cdim}(G_0) + \text{cdim}(G_1) - 1$ holds with equality if $\Delta(V_0)$ is a connected graph. This enables us to construct inductively a sequence of graphs $G^{(n)}$ on n vertices whose cut dimension is $2n - 3$. In the base case $G^{(3)}$ is the complete graph on 3 vertices where all the edges have the same weight. Then $G^{(n)}$ is defined as the merge of $G^{(3)}$ and $G^{(n-1)}$ where the specified vertices can be chosen arbitrarily. Since the separation of $G^{(n)}$ along the newly constructed complete cut gives back $G^{(3)}$ and $G^{(n-1)}$, from the inductive hypothesis we conclude that $\text{cdim}(G) = \text{cdim}(G_0) + \text{cdim}(G_1) - 1 = 2n - 3$.

ℓ_1 -approximate cut dimension. As the cut dimension is at most $2n - 3$, we have to look to other methods in order to show larger lower bounds, if possible. We propose a generalization of the cut dimension which we call the ℓ_1 -approximate cut dimension. In order to motivate this, we quickly explain why the cut dimension is a lower bound on the linear query complexity of mincut. The main idea behind the cut dimension lower bound on query complexity is to answer all queries of the algorithm according to an n -vertex graph $G = (V, w)$. Supposing the algorithm makes k queries, we package these into a k -by- $\binom{n}{2}$ matrix A whose rows are the query vectors. If there is a cut $S \in \mathcal{M}(G)$ which is not in the row space of A , then by the Fredholm alternative there is a vector z such that $Az = \mathbf{0}$, where $\mathbf{0}$ is the all-zero vector, but $\langle S, z \rangle > 0$ and furthermore $z(i) = 0$ whenever $w(i) = 0$. Thus for a sufficiently small $\varepsilon > 0$ we have that $w - \varepsilon z \geq \mathbf{0}$ and so $G' = (V, w - \varepsilon z)$ defines a valid non-negatively weighted graph that has all the same answers to the queries of the algorithm as G . On the other hand, the weight of a minimum cut in G' is strictly smaller than that of G and thus as the algorithm cannot distinguish G and G' it cannot correctly compute the weight of a minimum cut in all n -vertex graphs.

The ℓ_1 -approximate cut dimension extends this adversary argument to include *all* the cuts of G instead of just the mincuts. If the minimum cut weight of G is λ and S is a cut with weight $\lambda + \delta$, then the algorithm will still fail if there is a z such that

15:6 On the Cut Dimension of a Graph

1. $w - z \geq \mathbf{0}$
2. $Az = \mathbf{0}$
3. $\langle S, z \rangle > \delta$.

The reason is the same: the graph $G' = (V, w - z)$ has all the same answers to the queries made by the algorithm as G yet has a cut with weight strictly smaller than λ .

Taking the dual of the corresponding linear program shows that such a vector z will not exist iff $S - u$ is in the rowspan of A for a vector $u \geq \mathbf{0}$ with $\|u\|_{1,w} \leq \delta$. This leads us to define the (w, c) one-sided row-by-row ℓ_1 approximate rank of a matrix. For a matrix $Y \in \mathbb{R}^{M \times N}$ this is defined by a weight vector $w \in \mathbb{R}^N$ and a cost vector $c \in \mathbb{R}^M$ with $c \geq \mathbf{0}$. It is the minimum rank of a matrix \tilde{Y} such that $\tilde{Y} \leq Y$ and $\|Y(i, \cdot) - \tilde{Y}(i, \cdot)\|_{1,w} \leq c(i)$ for every row i , where $Y(i, \cdot)$ denotes the i^{th} row of Y . Let $G = (V, w)$ be a graph and the weight of a minimum cut in G be λ . The ℓ_1 -approximate cut dimension of a graph $G = (V, w)$, denoted $\widehat{\text{cdim}}(G)$, is the (w, c) one-sided row-by-row ℓ_1 -approximate rank of the matrix Y whose rows are the vectors $S \in \{0, 1\}^{\binom{n}{2}}$ for every cut S of G , and where $c = Yw - \lambda \mathbf{1}$, and $\mathbf{1}$ is the all-one vector.

Lower bounding the rank under such an ℓ_1 perturbation is a difficult task. However, we are able to show an infinite family of graphs whose ℓ_1 -approximate cut dimension is $2n - 2$, thereby showing the ℓ_1 -approximate cut dimension can be strictly larger than the cut dimension. This lower bound is of a “direct sum” type. We show that the ℓ_1 -approximate cut dimension of K_4 , the complete graph on 4 vertices, is 6, giving a tight lower bound of 6 on the number of linear queries needed to compute minimum cut on a 4 vertex graph. We then show that the direct union (see Definition 6) of k copies of K_4 has ℓ_1 -approximate cut dimension $6k$. The proof is tailored to the specific properties of the cut vectors of K_4 , and makes use of Gaussian elimination and properties of diagonally dominant matrices.

Near-mincuts. Related to the ℓ_1 -approximate cut dimension is the question of the cut dimension of *near-mincuts*. For $\alpha \geq 1$ call a cut S of a graph G an α -near-mincut if its weight is at most α times the weight of a minimum cut of G . Let $\mathcal{M}_\alpha(G) = \{S : S \text{ is an } \alpha\text{-near-mincut of } G\}$. It is known that $|\mathcal{M}_\alpha(G)| \leq \binom{n}{2}$ for $\alpha < 4/3$ [25] (see also the beautiful proof given in Theorem 15 of [12]). Even for $\alpha < 3/2$ the number of α -near-mincuts is $O(n^2)$ [17], which is a sharp threshold as there exist graphs with $\Omega(n^3)$ many $3/2$ -mincuts. There is also a generalization of the cactus representation of mincuts in terms of a tree of deformable polygons that applies to α -near-mincuts for $\alpha < 6/5$ [3]. In Section 8 we show that if G is a *simple* graph then $\dim(\text{span}(\vec{\mathcal{M}}_\alpha(G))) = O(n)$ for any $\alpha < 2$ (Theorem 41). This bound is tight as for $\alpha = 2$ the unweighted complete graph K_n witnesses $\dim(\text{span}(\vec{\mathcal{M}}_2(K_n))) = \binom{n}{2}$. For weighted graphs, on the other hand, we show that for any $\alpha > 1$ there exists an n -vertex weighted graph G with $\dim(\text{span}(\vec{\mathcal{M}}_\alpha(G))) = \binom{n}{2}$.

1.2 Open Problems

Several interesting open problems remain from this work.

- There is still a large gap between the known upper and lower bounds on the deterministic cut/linear query complexity of minimum cut. What is the right answer? We conjecture there is a deterministic cut query algorithm for minimum cut making $O(n^{2-\varepsilon})$ many queries for some $\varepsilon > 0$.
- Is the ℓ_1 -approximate cut dimension $O(n)$ for any n -vertex graph? Also can one show a general direct sum theorem for the ℓ_1 -approximate cut dimension?

1.3 Organization

The rest of the paper is organized as follows. We review necessary backgrounds about graphs, operations on graphs, and query models in Section 2. In Section 3, we show that the cut dimension is a lower bound on the deterministic linear query complexity of computing minimum cut. We then prove that the cut dimension is at most $2n - 3$ in Section 4, and give an explicit construction of graphs with cut dimension $2n - 3$ in Section 5. In Section 6, we give another proof for both the upper and lower bounds on $2n - 3$ using graph operations. In Section 7 we show a $2n - 2$ lower bound on ℓ_1 -approximate cut dimension which implies Theorem 3. Finally, in Section 8 we show that for a simple graph G and $1 \leq \alpha < 2$ it holds that $\dim(\text{span}(\vec{\mathcal{M}}_\alpha(G))) = O(n)$.

2 Preliminaries

For every natural number n , we denote by $[n]$ the set $\{1, 2, \dots, n\}$. For a vector $z \in \mathbb{R}^n$ we write $z \geq \mathbf{0}$ if every coordinate of the vector is at least 0, and similarly we write $z = \mathbf{0}$ if z is the all-zero vector. We denote the scalar product of two vectors $z, z' \in \mathbb{R}^n$ by $\langle z, z' \rangle$. For any matrix, denote the rank of A by $\text{rk}(A)$. We denote the disjoint union of sets X and Y by $X \sqcup Y$.

2.1 Graphs, cuts, sets

An undirected *weighted graph* on n vertices is a couple $G = (V, w)$, where V is the set of vertices with $|V| = n$, the set of edge slots $V^{(2)}$ is the set of subsets of V with cardinality 2, and the weight function $w : V^{(2)} \rightarrow \mathbb{R}$ is non-negative. We refer to the vertex set of G as $V(G)$. The set of edges of G is defined as $E = \{e \in V^{(2)} : w(e) > 0\}$. When in a graph $G = (V, w)$ the weight of every edge is 1, we say that the graph is *unweighted*, and we refer to it also as $G = (V, E)$; such graph is also called a *simple graph*. For an edge $e = \{u, v\}$, we say that u and v are the endpoints of e . For a subset $X \subseteq V$ of the vertices, we denote by $E(X)$ the set of edges in E which have both endpoints in X , and for disjoint subsets $X, Y \subseteq V$, we denote by $E(X, Y)$ the set of edges with exactly one endpoint in each of the two sets. We extend the weight function w to any subset E' of the edges by $w(E') = \sum_{e \in E'} w(e)$. We will deal only with graphs which have at least 2 vertices.

We fix an ordering $v_1 < v_2 < \dots < v_n$ of the vertices which induces also an ordering $\{v_1, v_2\}, \{v_1, v_3\}, \dots, \{v_{n-1}, v_n\}$ of the edge slots as well as an ordering $e_1 < e_2 < \dots < e_m$ of the $m = |E|$ edges. We view $w \in \mathbb{R}^{\binom{n}{2}}$ as a vector whose i^{th} coordinate gives the (possibly zero) weight of the i^{th} edge slot according to this ordering, and we define $\vec{w} \in \mathbb{R}^m$ as the restriction of w to the edges. With some slight abuse of notation, for a set of edges $S \subseteq E$, we use the same symbol S to also denote the characteristic vector in $\{0, 1\}^{\binom{n}{2}}$ of S among all edge slots. We further need the characteristic vector of $S \subseteq E$ among the m edges E , for which we use the notation $\chi(S) \in \{0, 1\}^m$. For a family \mathcal{F} of subsets of the edges, we use the notation $\vec{\mathcal{F}} = \{\chi(S) \in \{0, 1\}^m : S \in \mathcal{F}\}$.

For $X \subseteq V$, we denote by \bar{X} the set $V \setminus X$. A *cut* S is a set $E(X, \bar{X})$ for some $\emptyset \neq X \subsetneq V$. We call X and \bar{X} the *shores* of S , and we denote the cut by $\Delta(X)$. A cut is a *star cut* if one of its shores is a singleton, otherwise it is *non-star cut*. If the singleton shore of a star cut S is $\{v\}$, then we say that S is a star cut at v . The *weight* of a cut is the sum of the weights of its edges. For a cut S we define the *graph of the cut* S as the unweighted graph $G(S) = (V', E')$ where V' is the set of vertices in V that are endpoints of at least one edge in S , and $E' = S$. We say that a cut S is *connected* if $G(S)$ is a connected graph. A cut is a *minimum cut*, or *mincut*, for short, if no other cut has smaller weight. We denote by $\mathcal{M}(G)$ be the set of minimum cuts of G . The *cut dimension* of G is $\text{cdim}(G) = \dim(\text{span}(\vec{\mathcal{M}}(G)))$.

15:8 On the Cut Dimension of a Graph

Let V be a set of size n . Two sets $X, Y \subseteq V$ are said to *overlap* if $X \cap Y \neq \emptyset$, $\bar{X} \cap Y \neq \emptyset$, $X \cap \bar{Y} \neq \emptyset$. A family \mathcal{G} of subsets of V is said to be *laminar* if for all $X, Y \in \mathcal{G}$ it holds that X and Y do not overlap. A set family $\mathcal{G} \subseteq 2^V$ is said to be *closed under overlaps* if for every $X, Y \in \mathcal{G}$ that overlap it holds that $X \cap Y, X \cup Y \in \mathcal{G}$. A laminar subset $\mathcal{L} \subseteq \mathcal{G}$ is said to be *maximal in \mathcal{G}* if for every $X \in \mathcal{G} - \mathcal{L}$ there is a $Y \in \mathcal{L}$ such that X, Y overlap. We say a laminar subset \mathcal{L} is maximal if it is maximal in 2^V .

The sets $X, Y \subseteq V$ *cross* if they overlap and additionally $\bar{X} \cap \bar{Y} \neq \emptyset$. Note that if X, Y cross then so do X, \bar{Y} . A set family $\mathcal{G} \subseteq 2^V$ is said to be *cross-free* if for all $X, Y \in \mathcal{G}$ it holds that X and Y do not cross. Observe that if X and Y do not cross then either Y or \bar{Y} is a subset of X or \bar{X} . Let $G = (V, w)$ be a graph with n vertices. Two cuts $\Delta(X)$ and $\Delta(Y)$ of G are *crossing* if X and Y are crossing. Let $\mathcal{F} = \{\Delta(X_1), \dots, \Delta(X_k)\}$ be a set of cuts of G . We say that \mathcal{F} is *cross-free family of cuts* if $\mathcal{G} = \{X_1, \dots, X_k\}$ is cross-free. Note that it does not matter which shore we take to be in \mathcal{G} .

There is a close relationship between cross-free families of cuts and laminar sets. Let $\mathcal{F} = \{\Delta(X_1), \dots, \Delta(X_k)\}$ be a cross-free family of cuts where each $X_i \subseteq V$, and let $X'_i = X_i$ if $v_1 \notin X_i$ and $X'_i = \bar{X}_i$ otherwise. The *beach* of \mathcal{F} is the set $\mathcal{G} = \{X'_1, \dots, X'_k\}$. For a family of sets $\mathcal{G} \subseteq 2^V$ we say that it is *proper* if $\emptyset, V \notin \mathcal{G}$, and we say that it is *complement free* if it does not contain X, Y with $Y = \bar{X}$.

▷ **Claim 4.** Let \mathcal{F} be a cross-free family of distinct cuts and \mathcal{G} its beach. Then \mathcal{G} is proper, complement free and laminar.

Proof. First, \mathcal{G} does not contain \emptyset or V because these are not shores of cuts. It is complement free because \mathcal{F} contains distinct cuts, and its beach contains exactly one representative shore from each cut. Finally, we show that it is laminar. Let $X_1, X_2 \in \mathcal{G}$. By definition of a beach, neither of these sets contain v_1 , thus $\bar{X}_1 \cap \bar{X}_2 \neq \emptyset$. Therefore if X_1, X_2 overlapped they would also cross, in contradiction to \mathcal{F} being a cross-free family of cuts. ◁

A mincut is *crossless* if no other mincut crosses it. Observe that a star mincut is always crossless. Also, if a mincut $\Delta(X)$ is crossless then for every mincut $\Delta(Y)$, either Y or \bar{Y} is a subset of X or \bar{X} . Crossing mincuts have a nice structural property which was already observed by [6].

▷ **Claim 5.** Let $G = (V, w)$ be a weighted graph. If $\Delta(X), \Delta(Y) \in \mathcal{M}(G)$ cross then $\Delta(X \cap Y), \Delta(X \cup Y) \in \mathcal{M}(G)$.

Proof. We have $\Delta(X \cap Y) \neq \emptyset$ and $\Delta(X \cup Y) \neq V$ because $\Delta(X)$ and $\Delta(Y)$ cross. The cut function is submodular therefore we have

$$w(\Delta(X \cap Y)) + w(\Delta(X \cup Y)) \leq w(\Delta(X)) + w(\Delta(Y)).$$

Let c be the weight of a minimum cut in G . Then the right hand side of the above inequality is equal to $2c$, while its left hand side is at least $2c$. Therefore $w(\Delta(X \cap Y)) + w(\Delta(X \cup Y)) = 2c$ from which the statement follows. ◁

2.2 Operations on graphs

We will use several operations on graphs. The first of these is the *direct union*.

► **Definition 6** (direct union). For two graphs $G_0 = (V_0, w_0), G_1 = (V_1, w_1)$ with disjoint vertex sets, and for vertices $v_0 \in V_0$ and $v_1 \in V_1$, the direct union of G_0 and G_1 at vertices v_0, v_1 is the fusion of the two by identifying v_0 and v_1 . Formally, the direct union is $G_0^{v_0} \oplus G_1^{v_1} = (V, w)$ where $V = (V_0 \cup V_1 \cup \{v\}) \setminus \{v_0, v_1\}$, for a new vertex $v \notin V_0 \cup V_1$. The weight function of $G_0^{v_0} \oplus G_1^{v_1}$ is defined by

$$w(\{x, y\}) = \begin{cases} w_b(\{x, y\}) & \text{if } x, y \in V_b \setminus \{v_b\}, b \in \{0, 1\}, \\ w_b(\{x, v_b\}) & \text{if } x \in V_b \setminus \{v_b\}, y = v_b, b \in \{0, 1\}, \\ 0 & \text{otherwise.} \end{cases}$$

The cut dimension of a direct union is a simple function of the cut dimensions of its components.

▷ **Claim 7.** Let $G = G_0^{v_0} \oplus G_1^{v_1}$ be the direct union of G_0 and G_1 at vertices v_0, v_1 . Let c_b be the weight of a minimum cut in G_b , for $b = 0, 1$. Then $\text{cdim}(G) = \text{cdim}(G_0) + \text{cdim}(G_1)$ if $c_0 = c_1$, and $\text{cdim}(G) = \text{cdim}(G_b)$ if $c_b < c_{1-b}$.

Proof. Let $\Delta(X)$ be an arbitrary cut of G where $v \notin X$. If $X \not\subseteq V_b$, for $b \in \{0, 1\}$, then the weight of the cut $\Delta(X)$ is at least $c_0 + c_1$, and therefore it is not a minimum cut. If $X \subseteq V_b$, for some $b \in \{0, 1\}$ then the weight of $\Delta(X)$ in G is the same as the weight of $\Delta(X)$ in G_b . Therefore if $c_0 = c_1$ then every mincut in G_0 and every mincut of G_1 is a mincut of G , and these are the only mincuts. Since their supports are disjoint, we have $\text{cdim}(G) = \text{cdim}(G_0) + \text{cdim}(G_1)$. If $c_b < c_{1-b}$ then only the mincuts of G_b are mincuts of G , and therefore $\text{cdim}(G) = \text{cdim}(G_b)$. ◁

The next two operations, which are inverses of each other, give a decomposition of a graph along a cut into two smaller graphs, and a composition of two graphs into a bigger one by unfolding a star cut in each components. The decomposition operation was essentially defined in [7]. Let $G = (V, w)$ be a weighted graph and let Z be a cut in G with shores X_0 and $X_1 = V \setminus X_0$. The *separation* of G along the cut Z , denoted by $\text{sep}(G, Z)$, is the set of two graphs $\{G_0 = (V_0, w_0), G_1 = (V_1, w_1)\}$, where $V_b = X_b \cup \{v_{1-b}\}$, for $b = 0, 1$ with new vertices v_0, v_1 . The respective weight functions are defined by $w_b(\{x, y\}) = w(\{x, y\})$ for any $x, y \in X_b$, and $w_b(\{x, v_{1-b}\}) = \sum_{y \in V_{1-b}} w(\{x, y\})$ for any $x \in X_b$.

Let $G_0 = (V_0, w_0), G_1 = (V_1, w_1)$ be two graphs on disjoint vertex sets, and let $v_b \in V_{1-b}$ be arbitrary vertices for $b \in \{0, 1\}$. The *merge* of G_0 and G_1 along the vertices v_1, v_0 , denoted by $\text{mer}(\{(G_0, v_1), (G_1, v_0)\})$, is the graph $G = (V, w)$, where $V = (V_0 \cup V_1) \setminus \{v_0, v_1\}$. The weight function in G is defined by $w(\{x, y\}) = w_b(\{x, y\})$ if $x, y \in V_b$, for $b \in \{0, 1\}$, and

$$w(\{x, y\}) = w_0(\{x, v_1\})w_1(\{v_0, y\}), \text{ if } x \in V_0 \text{ and } y \in V_1.$$

It follows from the definitions sep is the left inverse of mer if the star cut at v_1 in V_0 and the star cut at v_0 in V_1 both have weight one, and sep is the right inverse of mer if the weight of the cut Z is one. We formally state the former property.

▷ **Claim 8.** Let $G_0 = (V_0, w_0)$ and $G_1 = (V_1, w_1)$ have disjoint vertex sets, and let $v_b \in V_{1-b}$ such that $w_b(\Delta(v_{1-b})) = 1$, for $b = 0, 1$. Let Z be the cut in $\text{mer}(\{(G_0, v_1), (G_1, v_0)\})$ whose shores are $V_0 \setminus \{v_1\}$ and $V_1 \setminus \{v_0\}$. Then $w(Z) = 1$ and

$$\text{sep}(\text{mer}(\{(G_0, v_1), (G_1, v_0)\}), Z) = \{G_0, G_1\}.$$

2.3 Query models

► **Definition 9** (MINCUT_n). *The input in the MINCUT_n problem is an n -vertex weighted undirected graph $G = (V, w)$. The required output on G is the weight of a minimum cut in G .*

15:10 On the Cut Dimension of a Graph

A deterministic algorithm correctly solves the MINCUT_n problem if it outputs the correct mincut weight for every n -vertex input graph G . We consider algorithms given two models of query access to the input graph $G = (V, w)$, linear queries and cut queries. A *linear query* for G is a vector $x \in \mathbb{R}^{\binom{n}{2}}$, and the query is answered by $\langle x, w \rangle$. A *cut query* is a vector $x \in \{0, 1\}^{\binom{n}{2}}$ which is the characteristic vector of a cut in the complete n -vertex graph. The answer to a cut query is again $\langle x, w \rangle$. Clearly any cut query algorithm can be simulated by a linear query algorithm.

We use $D_{\text{cut}}(\text{MINCUT}_n)$ to denote the minimum, over all deterministic query algorithms \mathcal{A} that correctly solve MINCUT_n , of the maximum over all n -vertex input graphs $G = (V, w)$ of the number of cut queries made by \mathcal{A} on G . $D_{\text{lin}}(\text{MINCUT}_n)$ is defined analogously for linear queries.

Some authors instead define the output of the minimum cut problem to be a cut S that achieves the minimum weight, rather than the weight itself. Over n -vertex weighted graphs let us denote this problem as ARGMINCUT_n . For linear and cut queries, an algorithm that finds a minimum cut S can also return the weight of S with one additional query. Thus $D_{\text{lin, cut}}(\text{ARGMINCUT}_n) \geq D_{\text{lin, cut}}(\text{MINCUT}_n) - 1$, and the lower bounds we prove for MINCUT_n can be applied, minus 1, to ARGMINCUT_n as well.

3 Lower bounds on the linear query complexity of MINCUT

Graur et al. [13] introduce the cut dimension as a means to show lower bounds on the deterministic cut query complexity of computing minimum cut.

► **Theorem 10** ([13]). *If there is an n -vertex weighted graph $G = (V, w)$ with $\text{cdim}(G) = k$ then $D_{\text{cut}}(\text{MINCUT}_n) \geq k$.*

We show that this theorem even holds with respect to a stronger computational model where the algorithm is able to make linear queries. We also give a generalization of the cut dimension to a quantity which is at least as large, and can be strictly larger, that we call the ℓ_1 -approximate cut dimension. We now give an overview of the Graur et al. [13] argument in the context of linear queries and how we can extend it.

The proof of Theorem 10 is based on an adversary argument. Suppose a deterministic algorithm makes k linear queries and consider the execution of the algorithm on a fixed n -vertex graph $G = (V, w)$ whose set of minimum cuts is $\mathcal{M}(G)$. Make a k -by- $\binom{n}{2}$ matrix A whose rows are the query vectors asked by the algorithm. Suppose we can find a vector $z \in \mathbb{R}^{\binom{n}{2}}$ such that

1. $w - z \geq \mathbf{0}$,
2. $Az = \mathbf{0}$,
3. There is a cut $S \in \mathcal{M}(G)$ such that $\langle S, z \rangle > 0$.

The existence of such a vector z means the algorithm cannot correctly compute minimum cut weight on all weighted n -vertex graphs. The reason is that $G' = (V, w - z)$ is a valid non-negatively weighted graph by (1), has the same answers on all queries asked by the algorithm by (2), and by (3) has minimum cut weight at most $\langle S, w - z \rangle = \langle S, w \rangle - \langle S, z \rangle < \langle S, w \rangle$, which is strictly less than the minimum cut weight of G . As with k queries the algorithm cannot distinguish whether the input is G or G' , it cannot correctly output the minimum cut weight for all n -vertex weighted graphs.

A weaker condition than (3) suffices for this argument to work. Suppose that the minimum cut weight in G is c^* . Then the argument still goes through with the condition

3'. There is a cut S such that $\langle S, z \rangle > \langle S, w \rangle - c^*$.

This is because the algorithm cannot distinguish the graph G with minimum cut weight c^* from the graph $G' = (V, w - z)$ which has minimum cut weight at most $\langle S, w - z \rangle < c^*$.

In order to understand what kind of bound this argument gives, for fixed w, A, S we define the quantity $\alpha(w, A, S)$ which is given by the following linear program.

$$\begin{aligned} \alpha(w, A, S) = \underset{z}{\text{maximize}} \quad & \langle S, z \rangle \\ \text{subject to} \quad & w - z \geq 0 \\ & Az = \mathbf{0} \end{aligned}$$

Taking the dual of this program gives

$$\begin{aligned} \alpha(w, A, S) = \underset{v}{\text{minimize}} \quad & \langle S - A^T v, w \rangle \\ \text{subject to} \quad & S - A^T v \geq 0 \end{aligned}$$

The dual tells us that a vector z having large overlap with S and satisfying items (1), (2) above exists iff the vector S is *far away* from the rowspace of A . The notion of far away here is a one-sided ℓ_1 distance weighted by w . It is one-sided because the condition $S - A^T v \geq 0$ tells us we are looking to approximate S by vectors in the rowspace of A that are entrywise at most S . As $S - A^T v \geq 0$ and $w \geq 0$ this means $\langle S - A^T v, w \rangle = \sum_i |w(i) \cdot (S(i) - A^T v)| = \|S - A^T v\|_{1,w}$, where $\|u\|_{1,w}$ is defined to be $\sum_i |u(i)w(i)|$. Thus the value of the dual can be interpreted as the one-sided $\|\cdot\|_{1,w}$ distance between S and the rowspace of A .

This leads us to define an ℓ_1 approximate version of the cut dimension. The notion we need is given by the following definitions.

► **Definition 11** (one-sided row-by-row ℓ_1 -approximate rank). *Let $Y \in \mathbb{R}^{M \times N}$ be a matrix, $w \in \mathbb{R}^N$ a weight vector and $c \in \mathbb{R}^M$ a cost vector. We define the (w, c) one-sided row-by-row ℓ_1 -approximate rank of Y to be the minimum rank of a matrix \tilde{Y} such that $\tilde{Y} \leq Y$ and $\|Y(i, :) - \tilde{Y}(i, :)\|_{1,w} \leq c(i)$, for all $1 \leq i \leq M$.*

► **Definition 12** (ℓ_1 -approximate cut dimension). *Let $G = (V, w)$ be an n -vertex weighted undirected graph with minimum cut weight c^* . Let M be $(2^{n-1} - 1)$ -by- $\binom{n}{2}$ matrix whose rows are $S \in \{0, 1\}^{\binom{n}{2}}$ for all cuts S of G . Let $c = Mw - c^* \mathbf{1}$, where $\mathbf{1}$ is the all one vector. Then the ℓ_1 -approximate cut dimension of G , denoted $\widetilde{\text{cdim}}(G)$, is the (w, c) one-sided row-by-row ℓ_1 -approximate rank of M .*

► **Theorem 13.** *If there is an n -vertex graph weighted graph $G = (V, w)$ with $\widetilde{\text{cdim}}(G) = k$ then $D_{\text{lin}}(\text{MINCUT}_n) \geq k$.*

Proof. Let $G = (V, w)$ be a graph with $\widetilde{\text{cdim}}(G) = k$ and let c^* be the minimum cut weight of G . Suppose for contradiction there is a deterministic $k - 1$ linear query algorithm that correctly computes the minimum cut of any n -vertex graph. Run this algorithm answering queries according to G and package the queries into a $(k - 1)$ -by- $\binom{n}{2}$ matrix A .

As the algorithm is correct, for every cut S of G it must be the case that $\alpha(w, A, S) \leq \langle S, w \rangle - c^*$. If not, the graph $G' = (V, w - z)$, where z is an optimal solution to the primal of $\alpha(w, A, S)$, has minimum cut weight strictly smaller than c^* , yet G' cannot be distinguished from G by the algorithm. Thus by the dual formulation of $\alpha(w, A, S)$, this means that for every cut S of G there is a vector $\tilde{S} = A^T v$ in the rowspace of A such that $\tilde{S} \leq S$ and $\|S - \tilde{S}\|_{1,w} \leq \langle S, w \rangle - c^*$. The matrix \tilde{M} whose rows are \tilde{S} for all cuts S therefore witnesses that $\widetilde{\text{cdim}}(G) \leq \text{rk}(A) \leq k - 1$, a contradiction. ◀

15:12 On the Cut Dimension of a Graph

► **Lemma 14.** *For any weighted graph $G = (V, w)$ we have $\text{cdim}(G) \leq \widetilde{\text{cdim}}(G)$.*

Proof. Suppose that $G = (V, w)$ has minimum cut weight c^* , and let $\mathcal{M}(G)$ be the set of minimum cuts of G . Let M be the $(2^{n-1} - 1)$ -by- $\binom{n}{2}$ matrix whose rows are $S \in \{0, 1\}^{\binom{n}{2}}$ for all cuts S of G and let $c = Mw - c^*$.

Let Y be the submatrix of M where rows are restricted to cuts in $\mathcal{M}(G)$ and columns are restricted to the edge slots e where $w(e) > 0$. Thus the rows of Y are exactly the vectors $\chi(S)$ for $S \in \mathcal{M}(G)$, and the rank of Y is $\text{cdim}(G)$. Any matrix \tilde{M} which satisfies $\tilde{M} \leq M$ and $\|M(i, :) - \tilde{M}(i, :)\|_{1, w} \leq c(i)$ for all i must contain Y as a submatrix, as $c(i) = 0$ for rows i that correspond to minimum cuts and w is positive on the edge slots labeling the columns of Y . Thus $\text{rk}(\tilde{M}) \geq \text{rk}(Y)$ for any (w, c) one-sided row-by-row ℓ_1 approximation \tilde{M} of M , giving the lemma. ◀

In Section 7 we will see that $\widetilde{\text{cdim}}(G)$ can be strictly larger than $\text{cdim}(G)$. From Theorem 13 and Lemma 14 we obtain the following corollary.

► **Corollary 15.** *If there is an n -vertex weighted graph $G = (V, w)$ with $\text{cdim}(G) = k$ then $D_{\text{lin}}(\text{MINCUT}_n) \geq k$.*

4 The cut dimension is at most $2n - 3$

In this section we prove Theorem 1 that $\text{cdim}(G) \leq 2n - 3$ for any undirected weighted graph G on $n \geq 2$ vertices. This will follow from two facts:

1. For $n \geq 2$ a cross-free family of cuts in an n -vertex graph has cardinality at most $2n - 3$.
2. If $\mathcal{L} \subseteq \mathcal{M}(G)$ is a maximal cross-free subset of the mincuts of G then $\text{span}(\vec{\mathcal{L}}) = \text{span}(\vec{\mathcal{M}}(G))$.

We remind the reader that $\vec{\mathcal{L}} = \{\chi(S) : S \in \mathcal{L}\}$ where $\chi(S) \in \{0, 1\}^{|E|}$ is the characteristic vector of the cut S amongst the edges of G .

These two facts are presented in the next two subsections.

4.1 Cardinality of a cross-free family of cuts

Recall from Claim 4 that if \mathcal{L} is a cross-free family of cuts then the beach \mathcal{G} of \mathcal{L} is a laminar family of sets. A standard inductive proof shows that a laminar family of subsets of a universe of cardinality n that contains no singletons has size at most $n - 1$, and thus a laminar family in general has size at most $2n - 1$. A beach has the additional properties of being proper and complement free which allows one to prove an upper bound of $2n - 3$. This is mentioned by Goemans [11] in the paragraph after Theorem 4 under the heading “Size of a Laminar Family”, who observes that the standard inductive proof also implies the bound is attained only if the family includes the universe and at least one set and its complement. See also Corollary 2.15 of [22], where it is shown that a proper laminar family has cardinality at most $2n - 2$.

► **Lemma 16.** *Let $n \geq 2$, V a set of cardinality n , and $\mathcal{G} \subseteq 2^V$ be a family of sets which is proper and laminar. Then $|\mathcal{G}| \leq 2n - 2$. If \mathcal{G} is proper, laminar, and complement free then $|\mathcal{G}| \leq 2n - 3$.*

Proof. First we show the $2n - 2$ upper bound. We prove by induction. Consider first the base case where $n = 2$ and $V = \{v_1, v_2\}$. As $\emptyset, V \notin \mathcal{G}$ the only possible elements to include in \mathcal{G} are $\{v_1\}, \{v_2\}$ and $|\mathcal{G}| \leq 2 = 2n - 2$.

Now we assume the statement is true for families of sets on a universe of $n - 1$ elements and show it holds for families of sets on a universe of size n . Let $\mathcal{G} \subseteq 2^V$ be a proper laminar family. We say that $X \in \mathcal{G}$ is maximal if there is no set $Y \in \mathcal{G}$ with $X \subset Y$. Let X_1, \dots, X_m be the maximal sets in \mathcal{G} . Note that we must have $X_i \cap X_j = \emptyset$ for all $i \neq j \in [m]$. This is because for distinct maximal sets $X_i - X_j, X_j - X_i \neq \emptyset$ thus if $X_i \cap X_j \neq \emptyset$ they would be overlapping. If $\cup_{i=1}^m X_i \subsetneq V$ then the result already holds by the induction hypothesis. Thus we may assume $m \geq 2$ and X_1, \dots, X_m form a partition of V . The family $\mathcal{F}_1 = \{Y : Y \subsetneq X_1\}$ is a laminar family on the universe X_1 which does not contain X_1 . Hence by the induction hypothesis it has at most $2|X_1| - 2$ many sets. This holds for all $i = 1, \dots, m$, thus including X_1, \dots, X_m the total number of sets is $\sum_{i=1}^m 2|X_i| - m \leq 2n - 2$.

Now we show the $2n - 3$ upper bound additionally assuming the family is complement free. We show this result directly using the upper bound of $2n - 2$ we have just shown on the size of proper laminar families. Let $\mathcal{G} \subseteq 2^V$ be proper, laminar, and complement free, and let X_1, \dots, X_m be the maximal sets in \mathcal{G} , which again must be disjoint. The number of subsets strictly contained in X_i is at most $2|X_i| - 2$ by the previous result. Thus, including X_1, \dots, X_m we can upper bound the size of \mathcal{G} by $\sum_{i=1}^m 2|X_i| - m$. If $m > 2$ then the upper bound of $2n - 3$ already holds. If $m = 1$ then as \mathcal{G} is a proper family we must have $|X_1| \leq n - 1$ in which case the upper bound of $2n - 3$ holds as well. Finally, consider the case $m = 2$. In this case, if $|X_1 \cup X_2| < n$ then the bound already holds. If $X_1 \cup X_2 = V$ then $X_2 = \bar{X}_1$ and we must exclude one of these sets, giving a bound of $2n - 2 - 1 = 2n - 3$. ◀

► **Remark 17.** From the proof in the proper, laminar, complement-free case we can observe for what maximal sets equality in the upper bound can hold. The first is the case where there are three maximal sets X_1, X_2, X_3 that form a partition of $[n]$. With $V = [6]$ an example of this type saturating the bound is $\mathcal{G} = \{\{1\}, \dots, \{6\}, \{1, 2\}, \{3, 4\}, \{5, 6\}\}$. The second is the case where there are two maximal sets X_1, X_2 that form a partition of $[n]$ and exactly one of X_1, X_2 is not included. The latter includes the case where there is a single maximal set X_1 of size $|X_1| = n - 1$. For $V = [6]$, an example of this type is $\mathcal{G} = \{\{2\}, \dots, \{6\}, \{2, 3\}, \{2, 3, 4\}, \{2, 3, 4, 5\}, \{2, 3, 4, 5, 6\}\}$.

Chandran and Ram (Lemma 2.13 in [5]) show that if the set $\mathcal{M}(G)$ of minimum cuts of a graph G is cross-free, then $|\mathcal{M}(G)| \leq 2n - 3$. This is an easy corollary of Lemma 16, which gives something more general.

► **Corollary 18.** *Let $G = (V, w)$ be a graph on $n \geq 2$ vertices. Let $\mathcal{L} \subseteq \mathcal{M}(G)$ be a subset of minimum cuts that is cross-free. Then $|\mathcal{L}| \leq 2n - 3$.*

4.2 Spanning

Let $\mathcal{L} \subseteq \mathcal{M}(G)$ be a maximal cross-free subset of $\mathcal{M}(G)$. Here maximal means that for any cut $S \in \mathcal{M}(G) \setminus \mathcal{L}$ there is a cut $T \in \mathcal{L}$ that crosses S . The fact that $\text{span}(\vec{\mathcal{L}}) = \text{span}(\vec{\mathcal{M}}(G))$ essentially follows from a key lemma of Jain in his factor of 2 approximation algorithm for the survivable network design problem (Lemma 4.2 in [18]). Another application of a similar lemma can be found in Goeman's approximation algorithm for the bounded-degree minimum spanning tree problem [11].

The context of Jain's lemma is slightly different than ours, as we now explain. Instead of mincuts, Jain considers the set of cuts \mathcal{T} which saturate the inequalities of a particular linear program. He shows that the set \mathcal{T} has the property that if $\Delta(X), \Delta(Y) \in \mathcal{T}$ cross then either

1. $\Delta(X \cap Y), \Delta(X \cup Y) \in \mathcal{T}$ and $\chi(\Delta(X)) + \chi(\Delta(Y)) = \chi(\Delta(X \cap Y)) + \chi(\Delta(X \cup Y))$, or
2. $X \setminus Y, Y \setminus X \in \mathcal{T}$ and $\chi(\Delta(X)) + \chi(\Delta(Y)) = \chi(\Delta(X \setminus Y)) + \chi(\Delta(Y \setminus X))$.

15:14 On the Cut Dimension of a Graph

As shown by Dinitz, Karzanov, and Lomonosov [6], for crossing mincuts $\Delta(X), \Delta(Y)$ both items (1), (2) hold (see Proposition 45 for a proof). Thus Jain's lemma applies to $\mathcal{M}(G)$ as well.

► **Lemma 19** ([18]). *Let $G = (V, w)$ be a graph and $\mathcal{L} \subseteq \mathcal{M}(G)$ be a maximal cross-free family of mincuts. Then $\text{span}(\vec{\mathcal{L}}) = \text{span}(\vec{\mathcal{M}}(G))$.*

For completeness, we include a full proof of Lemma 19 in Appendix A.

We now can give the first proof of our main upper bound that for any $n \geq 2$ an n -vertex graph $G = (V, w)$ has $\text{cdim}(G) \leq 2n - 3$.

Proof of Theorem 1. Follows from Corollary 18 and Lemma 19. ◀

5 Explicit construction of graphs with cut dimension $2n - 3$

In this section we prove Theorem 2 by giving a general technique to explicitly construct graphs of cut dimension $2n - 3$. We focus on constructing graphs $G = (V, w)$ where w is strictly positive, i.e. where G is a complete weighted graph. The main lemma of this section, Lemma 27, shows that, in a complete weighted graph, for any cross-free family of cuts \mathcal{L} the vectors in $\vec{\mathcal{L}}$ are linearly independent.

Thus to construct a graph with cut dimension $2n - 3$ it suffices to construct a complete weighted graph whose set of mincuts is a cross-free family of cuts of cardinality $2n - 3$. Such a graph is constructed for every $n \geq 2$ in Theorem 5.2 of [5]. Combining this construction with our linear independence result Lemma 27 gives a proof of our main lower bound Theorem 2.

In Section 5.3 we go further and show for any maximal cross-free family $\mathcal{F} \subseteq 2^{[n]}$ there is a complete weighted graph $G = ([n], w)$ with $\mathcal{M}(G) = \{\Delta(X) : X \in \mathcal{F}\}$. Moreover, we give an explicit formula for the weight vector w . Part of this construction is a lemma, Lemma 29, which may be of independent interest: it says that if \mathcal{L} is a maximal family of cross-free cuts in a graph G , and all cuts in \mathcal{L} have the same weight c , then c is the weight of the minimum cut in G .

A key tool for showing the linear independence of cuts from a cross-free family is the tree representation of a laminar family, which we go over next.

5.1 Tree representation

► **Definition 20.** *For an unweighted directed graph $G = (V, E)$ we let $\delta^+(X) = \{(x, y) \in E : x \in X, y \in V - X\}$. For a singleton $v \in V$ we write $\delta^+(v)$ instead of $\delta^+(\{v\})$.*

► **Definition 21** (Arborescence). *An arborescence is a directed rooted tree where all edges point away from the root. A vertex of an arborescence which is not the root or a leaf we call an internal vertex.*

► **Definition 22** (Tree representation). *Let T be a directed graph whose underlying undirected graph is a tree. Let U be a finite set and $\phi : U \rightarrow V(T)$. For $e = (x, y) \in E(T)$ define S_e as*

$$S_e = \{s \in U : \phi(s) \text{ is in the same connected component of } T - e \text{ as } y\} .$$

Then (T, ϕ) defines a set family $\mathcal{F} = \mathcal{F}(T, \phi)$ where $\mathcal{F} = \{S_e : e \in E(T)\}$. We say that (T, ϕ) is a tree representation of (U, \mathcal{F}) . We call (T, ϕ) a faithful tree representation if $|E(T)| = |\mathcal{F}|$. For $v \in V(T)$, if there is a $u \in U$ such that $\phi(u) = v$ then we say that v has a label.

We will need the fact that a laminar set family has a faithful tree representation by an arborescence. A textbook proof of this fact can be found in Korte and Vygen Proposition 2.14 [22]. While they do not explicitly say the tree representation they construct is faithful, this is clear from the proof.

► **Proposition 23.** *Let (U, \mathcal{F}) be laminar family. Then there is a faithful tree representation (T, ϕ) of (U, \mathcal{F}) where T is an arborescence.*

Recall from Claim 4 that if \mathcal{L} is a cross-free family of cuts then its beach \mathcal{G} is laminar, and thus has a tree representation.

► **Lemma 24** (Tree structure of maximal cross-free families). *Let \mathcal{L} be a maximal family of cross-free cuts of a graph $G = ([n], w)$ and $\mathcal{G} \subseteq 2^{[n]}$ its beach. Then in a faithful tree representation (T, ϕ) of \mathcal{G} it holds that*

1. *The root r is labeled by 1 and has $|\delta^+(r)| = 1$*
2. *There are $n - 1$ leaves of T each with a distinct label in $\{2, \dots, n\}$.*
3. *Every internal vertex v has $|\delta^+(v)| = 2$.*

Proof. As by the definition of a beach, sets do not contain 1, this means that 1 must be the label of the root. As star cuts do not cross any other cut, if \mathcal{L} is maximal it must contain all the star cuts. This means that \mathcal{G} contains the sets $\{2\}, \dots, \{n\}, \{2, \dots, n\}$. Thus the outdegree of the root must be 1, as this outgoing edge represents the set $\{2, \dots, n\}$. Further there must be $n - 1$ leaves which are labeled by $2, \dots, n$. We have now accounted for all the labels, thus no internal vertex has a label. Further, if there was a leaf v with parent u such that v did not have a label, then (u, v) would represent the empty set, which by definition is not in \mathcal{G} . Thus there are exactly $n - 1$ leaves.

It remains to show that every internal vertex v of T which is not the root has $|\delta^+(v)| = 2$. Let v be an internal vertex, and as v is not the root, let u be its parent, and as v is not a leaf let w be a child of v . If $|\delta^+(v)| = 1$ then the edges $(u, v), (v, w)$ would represent the same set, as v is not labeled. This contradicts the fact that (T, ϕ) is a faithful tree representation. Now suppose $|\delta^+(v)| > 2$ and let w, x, y be three of its children. Consider the sets $X_1, X_2, X_3 \in \mathcal{G}$ represented by the edges $(v, w), (v, x), (v, y)$. Further the edge (u, v) represents a set $A \in \mathcal{G}$ with $X_1 \cup X_2 \cup X_3 \subseteq A$. We claim that in this case \mathcal{L} is not maximal because the cut $\Delta(X_1 \cup X_2)$ does not cross any cut in \mathcal{L} . Indeed, $X_1 \cup X_2$ is contained in all the sets represented by edges on the path from v to the root, and is disjoint from the sets represented by any other edge of T . Thus we have a contradiction. ◀

► **Corollary 25.** *Let \mathcal{L} be a maximal family of cross-free cuts of a graph $G = ([n], w)$. Then $|\mathcal{L}| = 2n - 3$.*

Proof. Let \mathcal{G} be the beach of \mathcal{L} and (T, ϕ) a faithful tree representation of \mathcal{G} . As (T, ϕ) is faithful $|E(T)| = |\mathcal{L}|$. Let T' be the undirected graph underlying T . Clearly $|E(T')| = |E(T)|$. We use Lemma 24 to count $|E(T')|$. Let i be the number of internal vertices of T' , each of which has degree 3. There are also n non-internal vertices each of which has degree 1. Thus $|E(T')| = (3i + n)/2$. Also as T' is a tree $|E(T')| = |V(T')| - 1 = n + i - 1$. Hence $i = n - 2$ and $|\mathcal{L}| = |E(T)| = |E(T')| = 2n - 3$. ◀

5.2 Linear independence

We now show the main theorem of this section that in a complete weighted graph any set $\vec{\mathcal{L}}$ of cut vectors of a cross-free family of cuts \mathcal{L} is linearly independent. We will use the tree representation (T, ϕ) of the beach \mathcal{G} of \mathcal{L} to do this via the following lemma.

15:16 On the Cut Dimension of a Graph

► **Lemma 26.** *Let T be an arborescence with root r and $\psi : E(T) \rightarrow \mathbb{R}$. Let U be a finite set and $\phi : U \rightarrow V(T)$. Suppose that T, ϕ, ψ have the property that*

1. *The root r is labeled and has $|\delta^+(r)| = 1$.*
2. *Every internal vertex v is unlabeled and has $|\delta^+(v)| = 2$.*
3. *Every leaf of T has a label.*
4. *For every $s, t \in U$ it holds that $\sum_{e \in \phi(s) - \phi(t)} \psi(e) = 0$, where $\phi(s) - \phi(t)$ is the set of edges on the undirected path from $\phi(s)$ to $\phi(t)$.*

Then ψ is identically 0.

Proof. We will prove by induction on the depth of the arborescence. We need a slightly different statement for the inductive hypothesis since when considering a sub-arborescence T' of T we do not know that the root of T' has property (1).

Inductive hypothesis. Let T be an arborescence with root r that is unlabeled and has $|\delta^+(r)| = 2$, and further suppose T, ϕ, ψ satisfy conditions (2)-(4) of the proposition. Then letting u, v be the children of r it holds that $\psi((r, u)) = -\psi((r, v))$ and for any other edge $e \in E(T), e \neq (r, u), (r, v)$ it holds that $\psi(e) = 0$.

For the base case consider a tree of depth 1, with root r and two children u, v which are leaves. As they are leaves, u, v are labeled which, considering the path from u to v , means $\psi((r, u)) + \psi((r, v)) = 0$. This concludes the base case.

Now we prove the inductive step. Let r be the root of a tree with children u, v . We consider two cases:

Case 1: one of u, v is a leaf. Suppose without loss of generality that u is a leaf and v is an internal node with children v_1, v_2 . By the inductive hypothesis $\psi((v, v_1)) + \psi((v, v_2)) = 0$ and ψ is identically 0 on the subtrees rooted at v_1, v_2 . Let y_1, y_2 be leaves that are descendants of v_1, v_2 respectively (and can possibly be y_1, y_2 themselves). Considering the path from u to y_1 and y_2 we have the equations

$$\begin{aligned}\psi((r, u)) + \psi((r, v)) + \psi((v, v_1)) &= 0 \\ \psi((r, u)) + \psi((r, v)) + \psi((v, v_2)) &= 0\end{aligned}$$

As $\psi((v, v_1)) + \psi((v, v_2)) = 0$, adding these equations shows that $\psi((r, u)) + \psi((r, v)) = 0$, as desired. Substituting this back into the equations further implies that $\psi((v, v_1)) = \psi((v, v_2)) = 0$ so ψ is identically 0 on the subtree rooted at v completing this case.

Case 2: both u, v are internal vertices. Let the children of u be u_1, u_2 and the children of v be v_1, v_2 . By the inductive hypothesis, $\psi(\cdot)$ is identically zero on the sub-trees rooted at u_1, u_2, v_1, v_2 and we have $\psi((u, u_1)) + \psi((u, u_2)) = \psi((v, v_1)) + \psi((v, v_2)) = 0$. We must show that $\psi((u, u_1)) = \psi((u, u_2)) = \psi((v, v_1)) = \psi((v, v_2)) = 0$ and that $\psi((r, u)) + \psi((r, v)) = 0$.

Let x_1, x_2 be a leaves that are descendants of u_1, u_2 , respectively, and similarly let y_1, y_2 be leaves that are descendants of v_1, v_2 , respectively. By assumption all of these leaves are labeled. Considering the paths from $x_b - y_{b'}$ for $b, b' \in \{0, 1\}$ we obtain the following four constraints on ψ :

$$\begin{aligned}\psi((u, u_1)) + \psi((r, u)) + \psi((r, v)) + \psi((v, v_1)) &= 0 \\ \psi((u, u_1)) + \psi((r, u)) + \psi((r, v)) + \psi((v, v_2)) &= 0 \\ \psi((u, u_2)) + \psi((r, u)) + \psi((r, v)) + \psi((v, v_1)) &= 0 \\ \psi((u, u_2)) + \psi((r, u)) + \psi((r, v)) + \psi((v, v_2)) &= 0\end{aligned}$$

Adding all four equations and using $\psi((u, u_1)) + \psi((u, u_2)) = \psi((v, v_1)) + \psi((v, v_2)) = 0$ shows that $\psi((r, u)) + \psi((r, v)) = 0$. Taking this into account, adding the first two equations then shows $\psi((u, u_1)) = 0$, and adding the last two equations shows $\psi((u, u_2)) = 0$. This then also means $\psi((v, v_1)) = \psi((v, v_2)) = 0$.

We have now shown the inductive statement holds. It remains to see why this implies the lemma. Let r be the root of the tree, let u be the child of r , and let u_1, u_2 be the children of u . By the inductive statement we have that $\psi((u, u_1)) + \psi((u, u_2)) = 0$ and ψ is identically zero on the subtree rooted at u_1 and the subtree rooted at u_2 . Let x_1, x_2 be leaves which are descendants of u_1, u_2 , respectively. As the root has a label, considering the path from r to u_1 implies that $\psi((r, u)) + \psi((u, u_1)) = 0$ and considering the path from r to u_2 implies $\psi((r, u)) + \psi((u, u_2)) = 0$. Adding these equations implies that $\psi((r, u)) = 0$, from which it then follows that $\psi((u, u_1)) = \psi((u, u_2)) = 0$. ◀

► **Lemma 27.** *Let $G = ([n], w)$ be a complete weighted graph and let \mathcal{L} be a cross-free family of cuts. Then $\vec{\mathcal{L}} = \{\chi(S) : S \in \mathcal{L}\}$ form a linearly independent set of vectors.*

Proof. We may assume that \mathcal{L} is a maximal cross-free family, as showing that a superset of $\vec{\mathcal{L}}$ is linearly independent implies that $\vec{\mathcal{L}}$ is as well. Thus suppose \mathcal{L} is a maximal cross-free family and let \mathcal{G} be its beach. Let (T, ϕ) be a faithful tree representation of \mathcal{G} . By Lemma 24 we have that (T, ϕ) satisfy conditions (1)-(3) of Lemma 26.

Now we ask the question: for an edge $\{i, j\} \in E(G)$ which sets $S \in \mathcal{L}$ contain it? This has a very nice description in terms of the tree decomposition. Let $u, v \in V(T)$ be the vertices with $\phi(i) = u, \phi(j) = v$. Then the sets containing i are the sets represented by edges from the root to u ; the sets containing j are the sets represented by the edges on the path from the root to v . Therefore the sets which contain i but not j or j but not i , are exactly those represented by the edges on the path from u to v in the undirected tree underlying T . Thus the cuts which contain the edge $\{i, j\}$ are exactly those with a shore which is represented by an edge on the path from u to v in undirected graph underlying T .

Consider a linear combination $\sum_{S \in \mathcal{L}} \alpha_S \chi(S) = \mathbf{0}$ which is equal to the all zero vector. The $\{i, j\}$ coordinate of this equation says that $\sum_{S \in \mathcal{L}, \{i, j\} \in S} \alpha_S \chi(S)(\{i, j\}) = 0$. This sum is exactly over the sets represented by edges on the path from $\phi(i)$ to $\phi(j)$. As this sum must be zero for every edge $\{i, j\}$, this says that if we let $\psi(e) = \alpha_S$ where the edge e represents a shore of S then for any two labeled vertices $u, v \in V(T)$ the sum of $\psi(e)$ over the edges on the path from u to v is zero. Thus also condition (4) of Lemma 26 is satisfied. Hence all of the conditions of Lemma 26 hold which implies that ψ must be identically zero and therefore all coefficients $\alpha_S = 0$. This shows that $\{\chi(S) : S \in \mathcal{L}\}$ is a linearly independent set. ◀

We can now give the first proof of our main lower bound result on the cut dimension Theorem 2, which says that for every integer $n \geq 2$ there is an n -vertex weighted graph $G = (V, w)$ with $\text{cdim}(G) \geq 2n - 3$.

Proof of Theorem 2. For every integer $n \geq 2$, Theorem 5.2 of [5] constructs a complete weighted graph $G = (V, w)$ on n vertices such that $\mathcal{M}(G)$ is a cross-free family of size $|\mathcal{M}(G)| = 2n - 3$. By Lemma 27 the vectors in $\vec{\mathcal{M}}$ form a linearly independent set, thus $\text{cdim}(G) \geq 2n - 3$. ◀

5.3 Constructing graphs with a cross-free set of mincuts

In this subsection we explicitly construct, for any maximal cross-free family $\mathcal{F} \subseteq 2^{[n]}$, a complete weighted graph $G = ([n], w)$ with $\mathcal{M}(G) = \{\Delta(X) : X \in \mathcal{F}\}$. This task is made easier by the next lemma. We first need a definition.

15:18 On the Cut Dimension of a Graph

► **Definition 28.** Let $\mathcal{F} \subseteq 2^V$. For a subset $X \subseteq V$, let $\text{overlap}_{\mathcal{F}}(X) = \{Y \in \mathcal{F} : X, Y \text{ overlap}\}$.

► **Lemma 29.** Let $G = (V, w)$ be a graph and \mathcal{L} be a maximal cross-free family of cuts. Suppose that for all $S \in \mathcal{L}$ it holds that $w(S) = c$. Then the weight of a minimum cut in G is c .

Proof. Let \mathcal{G} be the beach of \mathcal{L} . Suppose for a contradiction that the weight of a minimum cut of G is $< c$. Let $\mathcal{T} = \{Z : \emptyset \neq Z \subsetneq V, v_1 \notin Z, Z \notin \mathcal{G}, w(\Delta(Z)) < c\}$ and

$$X = \underset{Z}{\operatorname{argmin}}\{|\text{overlap}_{\mathcal{G}}(Z)| : Z \in \mathcal{T}\}.$$

In the following we always use $\text{overlap}(\cdot)$ with respect to \mathcal{G} and drop the subscript. As $|\text{overlap}(X)| \geq 1$, let $Y \in \text{overlap}(X)$. As shown in Appendix A Lemma 46, both $|\text{overlap}(X \cap Y)|$ and $|\text{overlap}(X \cup Y)|$ are strictly smaller than $|\text{overlap}(X)|$. Thus it must be the case that $X \cap Y, X \cup Y \notin \mathcal{T}$. Let us take the case of $X \cap Y$. It does not contain v_1 , as neither X nor Y do, and it is a nonempty set by the definition of overlap . Thus it must be the case that either $w(\Delta(X \cap Y)) \geq c$ or that $X \cap Y \in \mathcal{G}$, which implies $w(\Delta(X \cap Y)) = c$. The same argument holds for $X \cup Y$, thus both $w(\Delta(X \cap Y)), w(\Delta(X \cup Y)) \geq c$.

However by submodularity of the cut function we have $w(\Delta(X \cap Y)) + w(\Delta(X \cup Y)) \leq w(\Delta(X)) + w(\Delta(Y))$, which implies that at least one of $\Delta(X \cap Y), \Delta(X \cup Y)$ must have weight $< c$. Hence we have a contradiction and the lemma holds. ◀

We will additionally need the following theorem which follows from Theorem 5.1 in [5].

► **Theorem 30** ([5]). Let $G = (V, w)$ be a complete weighted graph. Then $\mathcal{M}(G)$ is a cross-free family of cuts.

► **Theorem 31.** Let $n \geq 2$ and \mathcal{L} be a maximal cross-free family of cuts in the n -vertex complete graph. Let A be an $|\mathcal{L}|$ -by- $\binom{n}{2}$ matrix whose rows are the vectors $\chi(S)$ for $S \in \mathcal{L}$ and let $z = A^T \mathbf{1}$. Define $w(e) = 2^{-z(e)+1}$ for $e \in \binom{[n]}{2}$. Then $G = ([n], w)$ is a complete weighted graph with $\text{cdim}(G) = 2n - 3$ and $\mathcal{M}(G) = \mathcal{L}$.

Proof. It is clear from the definition that $w > 0$ and so defines a complete weighted graph. We will show that $Aw = \mathbf{1}$. By Lemma 29 this shows that the minimum cut weight of G is 1 and so the set of minimum cuts includes \mathcal{L} . As w defines a complete weighted graph, by Theorem 30 the set of minimum cuts in G is cross-free and therefore must be exactly $\mathcal{M}(G) = \mathcal{L}$, since \mathcal{L} is maximal. Further, $|\mathcal{L}| = 2n - 3$ by Corollary 25 and the vectors in $\vec{\mathcal{L}}$ are linearly independent by Lemma 27, thus $\text{cdim}(G) = 2n - 3$.

It remains to show $Aw = \mathbf{1}$. We do this using an alternative way of viewing the assignment of edge weights. Let $\mathcal{G} \subseteq 2^{[n]}$ be the beach of \mathcal{L} , and (T, ϕ) be a faithful tree representation of \mathcal{G} . For vertices $u, v \in V(T)$ let $d(u, v)$ be the length of the shortest path between u, v in the undirected graph underlying T . Now let $\{i, j\} \in \binom{[n]}{2}$ and suppose $\phi(i) = u, \phi(j) = v$. We claim that $w(\{i, j\}) = 2^{-d(u, v)+1}$. The sets of \mathcal{G} containing i are the sets represented by edges from the root to u ; the sets of \mathcal{G} containing j are the sets represented by the edges on the path from the root to v . Therefore the sets which contain i but not j or j but not i , are exactly those represented by the edges on the path from u to v in the undirected tree underlying T . As (T, ϕ) is faithful, each of these edges represents a different set, and therefore the number of edges on the path from u to v is exactly the number of sets of \mathcal{L} which contain $\{i, j\}$.

We now continue with the proof that $Aw = \mathbf{1}$ using this interpretation of the weights. For any cut $S \in \mathcal{L}$ with shore $X \in \mathcal{G}$, take the edge $(u, v) \in E(T)$ representing X . Now imagine we remove the edge (u, v) from T which disconnects T into two components. Let T_u be the

component containing u and T_v the component containing v . From T_u , which contains the root r of T , we create a graph T'_u whose underlying undirected graph is the same as T_u , but for which all edges are directed away from u . Thus in T'_u , vertex u becomes the root and r becomes a leaf. Now by item (2) of Lemma 24, every non-leaf vertex in T_v and T'_u has out-degree 2. We inject a unit of flow into u in the graph T'_u and let it propagate according to the rule that at every non-leaf vertex half of the flow is routed along each outgoing edge. We similarly inject a unit of flow into v in the graph T_v and let it propagate according to the same rule. Thus in the tree T_v , each leaf a gets $f(a) = 2^{-d(a,v)}$ amount of flow, where $d(a,v)$ is the number of edges along the path from v to a in T_v . Similarly, if b is a leaf in the tree T'_u , the amount of flow arriving at b is $f(b) = 2^{-d(b,u)}$. Now let $\{i, j\} \in [n]^{\binom{2}{}}$ with $i \in X, j \in \bar{X}$ and observe that the way we defined $w(\{i, j\})$ satisfies

$$w(\{i, j\}) = 2^{-d(\phi(i), \phi(j))+1} = 2^{-d(\phi(i), v) - d(\phi(j), u)} = f(\phi(i)) \cdot f(\phi(j)) .$$

Thus the weight of the cut S is

$$\sum_{i \in X, j \in \bar{X}} w(\{i, j\}) = \sum_{i \in X, j \in \bar{X}} f(\phi(i)) \cdot f(\phi(j)) = \left(\sum_{i \in X} f(\phi(i)) \right) \cdot \left(\sum_{j \in \bar{X}} f(\phi(j)) \right) = 1 \cdot 1 = 1 . \blacktriangleleft$$

6 Another proof using graph operations

In this section we give another proof of our main theorems: we prove that the cut dimension of any n -vertex graph is at most $2n - 3$ and we also prove that this upper bound is tight. An important role will be played by the following lemma, giving an explicit characterization of graphs having at least one non-star mincut, where none of these mincuts is crossless. This characterization has originally appeared in [4, 6]. More modern presentations can be found in Lemma 2.9 of [5] or Lemma 2 of [7].

► **Lemma 32.** *Suppose that $G = (V, w)$ is a graph which has a non-star mincut, and every non-star mincut is crossed by a non-star mincut. Then G is a cycle where all edges have the same weight.*

Let us denote by C_n the cycle on the n vertex set $V = \{v_1, \dots, v_n\}$ and with edge set $E = \{\{v_1, v_2\}, \dots, \{v_{n-1}, v_n\}, \{v_n, v_1\}\}$, where the weight of every edge is the same. We also need that the cut dimension of C_n is at most n . In fact, it is easy to prove that its cut dimension is exactly n when $n \geq 3$.

► **Lemma 33.** *The cut dimension of C_2 is 1, and $\text{cdim}(C_n) = n$, for $n \geq 3$.*

Proof. The statement for $n = 2$ is obvious. For $n \geq 3$ we have $\text{cdim}(C_n) \leq n$ as the graph only has n edges and thus the cut vectors are elements of \mathbb{R}^n which has dimension n .

For the lower bound we construct a set of n linearly independent minimum cut vectors in C_n . Label the coordinates of the vectors by the edges $\{v_1, v_2\}, \dots, \{v_{n-1}, v_n\}, \{v_n, v_1\}$. We define the sets $X_1 = \{v_1, v_2\}$ and $X_k = \{v_2, \dots, v_k\}$, for $2 \leq k \leq n$.

We claim that the cut vectors $\xi_k = \chi(\Delta(X_k))$, for $1 \leq k \leq n$, are linearly independent. Let e_i be the i^{th} standard basis vector in \mathbb{R}^n . Then we see that $\xi_1 = e_2 + e_n$ and $\xi_k = e_1 + e_k$, for $2 \leq k \leq n$. Thus $\xi_2 + \xi_n - \xi_1 = 2e_1$, so e_1 is in the span of these vectors. Also $e_k = \xi_k - e_1$ is in the span for $2 \leq k \leq n$. Hence these n vectors span all of \mathbb{R}^n and therefore must be linearly independent. ◀

6.1 Two lemmas on graph operations

The main technical part of the second proof of our main theorems is played by the two lemmas in this section. The second lemma gives an upper bound on the cut dimension of a graph G in function of the cut dimension of the smaller graphs obtained when G is separated along a crossless non-star minimum cut Z . Moreover, this upper bound becomes an equality when in addition the cut Z is connected. Our upper and lower bounds for the cut dimension are respectively almost immediate consequences of these results.

► **Lemma 34.** *Let $G = (V, w)$ be a weighted graph and let $Z \in \mathcal{M}(G)$ be a crossless non-star minimum cut defined by shores $X_0, X_1 = V \setminus X_0$. For $b \in \{0, 1\}$, let $\mathcal{M}_b = \{S \in \mathcal{M}(G) : S \subseteq Z \cup E(X_b)\}$. Let $\text{sep}(G, Z) = \{G_0 = (V_0, w_0), G_1 = (V_1, w_1)\}$ as defined in Section 2, where $V_b = X_b \cup \{v_{1-b}\}$, for $b \in \{0, 1\}$, with $v_0, v_1 \notin X_0 \cup X_1$. Then $\dim(\text{span}(\vec{\mathcal{M}}_b)) = \text{cdim}(G_b)$, for $b \in \{0, 1\}$.*

Proof. We prove the statement for $b = 0$, the other case follows in exactly the same manner. Let $m = |E|$ and partition E into three disjoint sets $E = E(X_0) \sqcup Z \sqcup E(X_1)$. Call a vertex $x \in X_0$ *friendly* if it has a neighbor in X_1 , that is there exists an edge $\{x, y\} \in Z$ for some $y \in X_1$. The edges in Z can then be partitioned into the disjoint union of sets Z_x , over all friendly x , where $Z_x = \{e \in Z : x \in e\}$.

Let $\mathcal{M}(G_0)$ be the set of all minimum cuts of G_0 . The set $\vec{\mathcal{M}}(G_0)$ is composed of m_0 dimensional vectors where $m_0 = |E(X_0)| + \deg(v_1)$. Observe that $\deg(v_1)$ is the number of friendly vertices in X_0 . We can partition the edges of G_0 into two sets $E(X_0) \sqcup Z_1$ where $Z_1 = \{\{x, v_1\} : x \text{ is friendly}\}$.

We define a natural bijection $\psi : \mathcal{M}_0 \rightarrow \mathcal{M}(G_0)$ as follows. Let S be a mincut in \mathcal{M}_0 with shores X' and $V \setminus X'$, where $X' \subseteq X_0$. Note that we can assume this because Z is crossless. Then $\psi(S)$ is the mincut in $\mathcal{M}(G_0)$ whose shores are X' and $(X_0 \setminus X') \cup \{v_1\}$. Let $k = |\mathcal{M}_0| = |\mathcal{M}(G_0)|$.

We now consider two matrices C and D , where C is a k -by- m matrix and D is a k -by- m_0 matrix. Fix an ordering S_1, \dots, S_k of \mathcal{M}_0 and let the i^{th} row of C be $\chi(S_i)$, the characteristic vector of the cut S_i . Likewise the i^{th} row of D is $\chi(\psi(S_i))$. We have $\text{rk}(C) = \dim(\text{span}(\vec{\mathcal{M}}_0))$ and $\text{rk}(D) = \text{cdim}(G_0)$.

The columns of C, D are labeled by edges. For C , we label the edges according to the partition $E = E(X_0) \sqcup Z \sqcup E(X_1)$, with edges in $E(X_0)$ coming first, then edges from Z , then edges from $E(X_1)$. For D , we label the edges according to the partition $E(X_0) \sqcup Z_1$, again with edges from $E(X_0)$ coming first and then those from Z_1 . We observe the following facts:

- The edges in $E(X_0)$ are common in G and G_1 , and $\chi(\psi(S_i))(e) = \chi(S_i)(e)$, for every $S_i \in \mathcal{M}_0$ and edge $e \in E(X_0)$. This means that columns of C and D labeled by an edge $e \in E(X_0)$ are identical.
- For an edge $e \in E(X_1)$, we have that $\chi(S_i)(e) = 0$, for every $S_i \in \mathcal{M}_0$. Thus columns of C labeled by an edge $e \in E(X_1)$ are all zero.
- Finally, for a friendly $x \in X_0$ consider any edge $e = \{x, y\} \in Z_x$ and the edge $f = \{x, v_1\} \in Z_1$. Then the e^{th} column of C and the f^{th} column of D are identical because for every $S_i \in \mathcal{M}_0$ we have $\chi(S_i)(e) = 1$ iff $x \in X'$ iff $\chi(\psi(S_i))(f) = 1$.

These points together imply that D is actually a submatrix of C , which can be obtained by taking the columns labeled by edges in $E(X_0)$ and then taking $|Z_1|$ more columns of C by choosing one $e \in Z_x$ for every friendly $x \in X_0$. Therefore $\text{rk}(D) \leq \text{rk}(C)$.

We can also see that $\text{rk}(C) \leq \text{rk}(D)$ as C can be obtained from D by repeating columns labeled by edges in Z_1 several times and adding all zero columns, and neither of these operations increase the rank. ◀

► **Lemma 35.** *Let G, Z, G_0, G_1 as in Lemma 34. Then $\text{cdim}(G) \leq \text{cdim}(G_0) + \text{cdim}(G_1) - 1$, and if Z is connected then the equality holds.*

Proof. We first prove that $\text{cdim}(G) \leq \text{cdim}(G_0) + \text{cdim}(G_1) - 1$. The important fact is that $\mathcal{M}(G) \subseteq \mathcal{M}_0 \cup \mathcal{M}_1$ because Z is a crossless mincut. Also since $\mathcal{M}_0, \mathcal{M}_1 \subseteq \mathcal{M}(G)$ we in fact have $\mathcal{M}(G) = \mathcal{M}_0 \cup \mathcal{M}_1$. Therefore

$$\begin{aligned} \text{cdim}(G) &= \dim(\text{span}(\vec{\mathcal{M}}(G))) \\ &= \dim(\text{span}(\vec{\mathcal{M}}_0 \cup \vec{\mathcal{M}}_1)) \\ &= \dim(\text{span}(\text{span}(\vec{\mathcal{M}}_0) \cup \text{span}(\vec{\mathcal{M}}_1))) \\ &= \dim(\text{span}(\vec{\mathcal{M}}_0)) + \dim(\text{span}(\vec{\mathcal{M}}_1)) - \dim(\text{span}(\vec{\mathcal{M}}_0) \cap \text{span}(\vec{\mathcal{M}}_1)) \\ &= \text{cdim}(G_0) + \text{cdim}(G_1) - \dim(\text{span}(\vec{\mathcal{M}}_0) \cap \text{span}(\vec{\mathcal{M}}_1)) . \end{aligned}$$

We use Lemma 34 to obtain the last equality. Notice that $Z \in \mathcal{M}_0 \cap \mathcal{M}_1$, which implies that $\dim(\text{span}(\vec{\mathcal{M}}_0) \cap \text{span}(\vec{\mathcal{M}}_1)) \geq 1$, and thus $\text{cdim}(G) \leq \text{cdim}(G_0) + \text{cdim}(G_1) - 1$.

We now prove the inequality in the reverse direction, when Z is connected. Let $d_b = \text{cdim}(G_b) - 1$, for $b = 0, 1$. Let Z_b be the star cut at v_{1-b} in G_b . Since these are mincuts, we can extend them to a basis in the respective graphs. Therefore there exist $A_1, \dots, A_{d_0} \subset X_0$ and $B_1, \dots, B_{d_1} \subset X_1$ such that the family $\{\chi(\Delta(A_1)), \dots, \chi(\Delta(A_{d_0})), \chi(Z_0)\}$ is independent in $\text{span}(\vec{\mathcal{M}}(G_0))$ and the family $\{\chi(\Delta(B_1)), \dots, \chi(\Delta(B_{d_1})), \chi(Z_1)\}$ is independent in $\text{span}(\vec{\mathcal{M}}(G_1))$. We claim that in $\text{span}(\vec{\mathcal{M}}(G))$ the set

$$\{\chi(\Delta(A_1)), \dots, \chi(\Delta(A_{d_0})), \chi(\Delta(B_1)), \dots, \chi(\Delta(B_{d_1})), \chi(Z)\}$$

of size $d_0 + d_1 + 1$ is independent.

Let us suppose on the contrary that a non-trivial linear combination of these $d_0 + d_1 + 1$ vectors gives $\mathbf{0}$. Then there exist non all zero real numbers $a_1, \dots, a_{d_0}, b_1, \dots, b_{d_1}$ and $\varepsilon \in \{0, 1\}$ such that

$$\sum_{i=1}^{d_0} a_i \chi(\Delta(A_i)) + \sum_{j=1}^{d_1} b_j \chi(\Delta(B_j)) = \varepsilon \chi(Z). \quad (1)$$

We define the function $S : V \rightarrow \mathbb{R}$ by

$$S(x) = \begin{cases} \sum_{x \in A_i} a_i & \text{if } x \in X_0, \\ \sum_{x \in B_j} b_j & \text{if } x \in X_1. \end{cases}$$

If $x \in X_0$ and $y \in X_1$ are arbitrary elements and $\{x, y\} \in Z$, then $\chi(\Delta(A_i))(\{x, y\}) = 1$ iff $x \in A_i$ and $\chi(\Delta(B_j))(\{x, y\}) = 1$ iff $y \in B_j$. Therefore for every $\{x, y\} \in Z$, the coordinate $\{x, y\}$ of Equation (1) gives

$$S(x) + S(y) = \varepsilon. \quad (2)$$

From Equation (1) we can also deduce that for every $\{x, x'\} \in E(X_0)$ we have

$$\sum_{i=1}^{d_0} a_i \chi(\Delta(A_i))(\{x, x'\}) = 0, \quad (3)$$

and for every $\{y, y'\} \in E(X_1)$ we have

$$\sum_{j=1}^{d_1} b_j \chi(\Delta(B_j))(\{y, y'\}) = 0. \quad (4)$$

15:22 On the Cut Dimension of a Graph

Let $\{x_0, y_0\}$ be an arbitrary edge in Z , where $x_0 \in X_0$ and $y_0 \in X_1$. We set $s_0 = S(x_0)$ and $s_1 = S(y_0)$. We know from Equation (2) that

$$s_0 + s_1 = \varepsilon.$$

We claim that for every $\{x, y\} \in Z$, where $x \in X_0$ and $y \in X_1$, we have $S(x) = s_0$ and $S(y) = s_1$. For this consider an arbitrary breadth first search tree with root x_0 . Since the graph of the cut Z , the graph $G(Z) = (V', Z)$, is a connected bipartite graph, every vertex in $V' \cap X_0$ will be at some even depth of the tree, and every vertex in $V' \cap X_1$ at some odd depth of the tree. Going through all the vertices depth by depth starting with x_0 at depth 0, Equation (2) gives the claim.

We now distinguish two cases. In the first case at least one of s_0 and s_1 is non-zero, say without loss of generality that $s_0 \neq 0$. For $i = 1, \dots, d_0$, we define

$$a'_i = a_i/s_0.$$

Then Equation (3) implies that in G_0 , for every $\{x, x'\} \in E(X_0)$, we have

$$\sum_{i=1}^{d_0} a'_i \chi(\Delta(A_i))(\{x, x'\}) = 0. \quad (5)$$

Also in G_0 , if $x \in X_0$ then $\chi(\Delta(A_i))(\{x, v_1\}) = 1$ iff $x \in A_i$. Therefore

$$\sum_{i=1}^{d_0} a'_i \chi(\Delta(A_i))(\{x, v_1\}) = s_0/s_0 = 1. \quad (6)$$

Therefore Equations (5) and (6) imply that

$$\sum_{i=1}^{d_0} a'_i \chi(\Delta(A_i)) = \chi(Z_0), \quad (7)$$

which contradicts the linear independence of $\{\chi(\Delta(A_1)), \dots, \chi(\Delta(A_{d_0})), \chi(Z_0)\}$.

In the second case $s_0 = s_1 = 0$, and thus for all $\{x, y\} \in Z$, with $x \in X_0$ and $y \in X_1$, we have $S(x) = S(y) = 0$. Therefore in G_0 , for every edge $\{x, v_1\}$,

$$\sum_{i=1}^{d_0} a_i \chi(\Delta(A_i))(\{x, v_1\}) = 0, \quad (8)$$

and similarly in G_1 , for every edge $\{y, v_0\}$,

$$\sum_{j=1}^{d_1} b_j \chi(\Delta(B_j))(\{y, v_0\}) = 0. \quad (9)$$

Since $a_1, \dots, a_{d_0}, b_1, \dots, b_{d_1}$ are not all zero, either a_1, \dots, a_{d_0} is not all zero or b_1, \dots, b_{d_1} is not all zero. If a_1, \dots, a_{d_0} is not all zero then from Equations (3) and (8) it follows that the family $\{\chi(\Delta(A_1)), \dots, \chi(\Delta(A_{d_0}))\}$ is dependent in $\text{span}(\vec{\mathcal{M}}(G_0))$. If b_1, \dots, b_{d_1} is not all zero then similarly from Equations (4) and (9) it follows that the family $\{\chi(\Delta(B_1)), \dots, \chi(\Delta(B_{d_1}))\}$ is dependent in $\text{span}(\vec{\mathcal{M}}(G_1))$. In either case, we reach a contradiction. \blacktriangleleft

6.2 The upper bound

We now can give our second proof of the upper bound on the cut dimension Theorem 1.

Proof of Theorem 1. The proof is by induction. For the base case $n = 2$, the only graph to be considered consists of a single edge and the cut dimension is $1 = 2n - 3$.

Now let $n \geq 3$, and we assume the inductive hypothesis holds for all graphs on at most $n - 1$ vertices. We consider 3 cases.

Case 1: The graph G has only star mincuts, say at vertices v_1, \dots, v_k , for some $1 \leq k \leq n$. As there are only k mincuts here we have $\text{cdim}(G) \leq k \leq n \leq 2n - 3$ for $n \geq 3$.

Case 2: There is a non-star mincut in G , and every non-star mincut is crossed by a non-star mincut. Then by Lemma 32, the graph G is a cycle where the edges have all the same weight. In this case by Lemma 33, we have $\text{cdim}(G) = \text{cdim}(C_n) = n \leq 2n - 3$ for $n \geq 3$.

Case 3 is where we use the induction hypothesis: Suppose that G has a non-star crossless mincut Z with shores X_0 and $X_1 = V \setminus X_0$. Let $|X_0| = k$. Then by Lemma 35 there are graphs G_0, G_1 such that $\text{cdim}(G) \leq \text{cdim}(G_0) + \text{cdim}(G_1) - 1$, where G_0 is a graph on $k + 1$ vertices, and G_1 is a graph on $n - k + 1$ vertices. Therefore by the inductive hypothesis

$$\text{cdim}(G) \leq 2(k + 1) - 3 + 2(n - k + 1) - 3 - 1 = 2n - 3 . \quad \blacktriangleleft$$

6.3 The lower bound

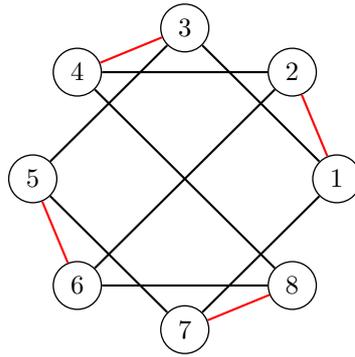
We now give our second proof of Theorem 2 that for every $n \geq 2$ there exist graphs G with $\text{cdim}(G) = 2n - 3$. We need a slightly more detailed statement for the inductive hypothesis which is given in the following theorem.

► **Theorem 36.** *For every integer $n \geq 2$, there is a complete weighted graph $G = (V, w)$ on n vertices with cut dimension $2n - 3$ and minimum cut weight 1, and where for every $v \in V$, the star cut $\Delta(\{v\})$ is a minimum cut.*

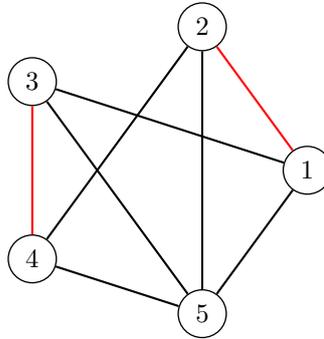
Proof. For $n = 2$ the statement is satisfied by the graph consisting of a single edge of weight one which has cut dimension one and where the two star cuts are minimum cuts. For $n = 3$ we may take the complete graph $G^{(3)} = (V^{(3)}, w^{(3)})$ with all weights $1/2$, which has cut dimension 3.

Now assume that there exists a graph $G^{(n-1)} = (V^{(n-1)}, w^{(n-1)})$ on $n - 1$ vertices satisfying the inductive hypothesis. Let us consider a copy of $G^{(3)} = (V^{(3)}, w^{(3)})$ where $V^{(3)} = \{t, u, v_0\}$ and $V^{(n-1)} \cap V^{(3)} = \emptyset$. We choose $v_1 \in V^{(n-1)}$ arbitrarily. We claim that the n -vertex graph $G^n = (V^{(n)}, w^{(n)})$ defined as $\text{mer}(\{(G^{(n-1)}, v_1), (G^{(3)}, v_0)\})$ satisfies the statement. It follows from the definition of the merge operation that $G^{(n)}$ is a complete weighted graph and that its star cuts are of weight one. In addition Claim 8 asserts that if Z is the cut in G^n whose shores are $V^{(n-1)} \setminus \{v_1\}$ and $V^{(3)} \setminus \{v_0\}$ then $w(Z) = 1$.

We now claim that the weight of a minimum cut of $G^{(n)}$ is one and that the mincut Z is crossless. Consider a non-star cut $\Delta(X)$. If both vertices t, u are on the same shore then the weight of $\Delta(X)$ is the same as the analogous cut in $G^{(n-1)}$ and therefore is at least one. If $\Delta(X)$ crosses Z , then we suppose without loss of generality that $t \in X, u \in \bar{X}$. We show that the weight of $\Delta(X)$ is greater than one, which then implies both claims. The cut contains the edge $\{t, u\}$ which has weight $1/2$. For every $y \in V^{(n-1)} \setminus \{v_1\}$, the cut either contains the edge $\{t, y\}$ or the edge $\{u, y\}$, and these edges have the same weight. Thus the total weight of such edges is half of the weight of Z , that is $1/2$. In addition, the cut contains also at least one edge from $G^{(n-1)}$, therefore its total weight is greater than one.



■ **Figure 1** Example graph G showing the necessity of the connected condition in Lemma 34. Red edges have weight 2 and black edges have weight 1. The minimum cut weight is 4 and the cuts achieving this are all the star cuts and $\Delta(\{1, 2\}), \Delta(\{3, 4\}), \Delta(\{5, 6\}), \Delta(\{7, 8\}), \Delta(\{1, 2, 3, 4\})$.



■ **Figure 2** The graph G_0 . Red edges have weight 2 and black edges have weight 1. The minimum cut weight is 4 and the cuts achieving this are all the star cuts and $\Delta(\{1, 2\}), \Delta(\{3, 4\})$. The cut dimension is 7.

Finally Claim 8 says that $\text{sep}(G^{(n)}, Z) = \{G^{(n-1)}, G^{(3)}\}$. Since Z is a crossless non-star minimum cut that is also connected, Lemma 35 implies that $\text{cdim}(G^{(n)}) = \text{cdim}(G^{(n-1)}) + \text{cdim}(G^{(3)}) - 1$, which is $2n - 3$ by the inductive hypothesis. ◀

6.4 On the tightness of Lemma 35

One can wonder whether the connectedness of Z is a necessary hypothesis in Lemma 35. In fact it is, when $Z \in \mathcal{M}(G)$ is not connected then we can have $\text{cdim}(G) < \text{cdim}(G_0) + \text{cdim}(G_1) - 1$. An example is given in Figure 1. The mincuts in this graph are all the star cuts and

$$\Delta(\{1, 2\}), \Delta(\{3, 4\}), \Delta(\{5, 6\}), \Delta(\{7, 8\}), \Delta(\{1, 2, 3, 4\}) .$$

Thus no mincuts cross each other. Also none of the non-star mincuts are connected.

Consider the case where $Z = \Delta(\{1, 2, 3, 4\})$. When we separate G along this cut we see that $G_0 = G_1$ and they are equal to the graph in Figure 2. The mincuts in G_0 are all star cuts and $\Delta(\{1, 2\}), \Delta(\{3, 4\})$. All non-star mincuts in G_0 are connected so one can use Lemma 34 to compute that $\text{cdim}(G_0) = 7$, i.e. all these mincut vectors are linearly independent. However, the cut dimension of G is clearly at most 12 as it only has 12 edges. Direct computation shows that in fact $\text{cdim}(G) = 11$.

7 ℓ_1 -approximate cut dimension

In this section, we use the ℓ_1 -approximate cut dimension method to show Theorem 3 that for any $k \in \mathbb{N}$ and $n = 3k + 1$, it holds that $D_{\text{lin}}(\text{MINCUT}_n) \geq 2n - 2$.

Let K_4 be the complete graph on 4 vertices with all edge weights equal to 1. The theorem will follow from showing that the ℓ_1 -approximate cut dimension of the direct union of k copies of K_4 has ℓ_1 -approximate cut dimension $6k$. We start with the base case $k = 1$ to build up the notation and intuition that will be needed for the general case. The following definition and fact will be useful.

► **Definition 37** (Strictly diagonally dominant). *Let $A \in \mathbb{R}^{n \times n}$ be a matrix. We say that the i^{th} row of A is strictly diagonally dominant if $|A(i, i)| > \sum_{j \neq i} |A(i, j)|$. We say that A is strictly diagonally dominant iff all of its rows are.*

It is well known that a strictly diagonally dominant matrix has full rank. One way to prove this is via the following fact, which we will make use of in the proof of Theorem 3.

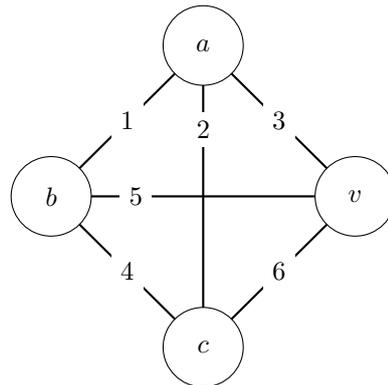
► **Fact 38.** *Let $A \in \mathbb{R}^{n \times n}$ be a matrix whose i^{th} row is strictly diagonally dominant. If $Au = \mathbf{0}$ for a vector $u \neq \mathbf{0}$ then $|u_i| < \|u\|_\infty$.*

Proof. Suppose for a contradiction that for some $u \neq \mathbf{0}$ it holds that $Au = \mathbf{0}$ and $|u_i| = \|u\|_\infty$ where the i^{th} row of A is strictly diagonally dominant. By normalizing and flipping the sign of u if necessary we may assume $\|u\|_\infty = 1$ and $A(i, i)u_i = |A(i, i)|$. Thus

$$\sum_j A(i, j)u_j = |A(i, i)| + \sum_{j \neq i} A(i, j)u_j \geq |A(i, i)| - \sum_{j \neq i} |A(i, j)| > 0 ,$$

a contradiction. ◀

7.1 ℓ_1 -approximate cut dimension of K_4



■ **Figure 3** The complete graph on 4 vertices with all edge weights equal to 1. The labels on edges indicate the ordering of edges used to represent cut vectors in the proof.

We label the vertices of K_4 by a, b, c, v , and use the ordering of edges indicated in Figure 3. Let X be the 7-by-6 matrix whose rows correspond to the cut vectors of all the nontrivial cuts

15:26 On the Cut Dimension of a Graph

$$X = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 \end{bmatrix} . \quad (10)$$

The cut vectors in X are given in the order

$$\Delta(\{a\}), \Delta(\{b\}), \Delta(\{c\}), \Delta(\{a, b, c\}), \Delta(\{a, b\}), \Delta(\{a, c\}), \Delta(\{b, c\}) .$$

The first 4 rows correspond to star cuts which are minimum cuts of weight 3 in K_4 . The last three rows correspond to cuts which have weight 4 in K_4 . Thus to show a lower bound of 6 on the number of linear queries needed to compute the minimum cut of a 4 vertex graph, we need to show that the $w = (1, 1, 1, 1, 1, 1)$, $c = (0, 0, 0, 0, 1, 1, 1)$ one-sided ℓ_1 approximate rank of X is 6.

▷ **Claim 39.** Let $w = \mathbf{1} \in \mathbb{R}^6$, and $c = (0, 0, 0, 0, 1, 1, 1)$. The (w, c) one-sided ℓ_1 approximate rank of X is 6.

Proof. The rank of X at most 6 as this is the number of columns, which takes care of the upper bound.

Now consider the lower bound. To do this we need to lower bound the rank of the matrix

$$Z = X - \begin{bmatrix} \mathbf{0}_{4,2} & \mathbf{0}_{4,2} & \mathbf{0}_{4,2} \\ A_1 & A_2 & A_3 \end{bmatrix}$$

where each of $A_1, A_2, A_3 \geq 0$ are 3-by-2 matrices and every row of $A_1 + A_2 + A_3$ sums to at most 1. As the first 4 rows of X correspond to vectors of minimum cuts, no error is allowed on the first 4 rows.

The first 4 rows of Z are equal to the first 4 rows of X , as there is no perturbation allowed on these rows. By doing elementary row operations on the first four rows, which do not change the rank, we can transform the first four rows of Z into the reduced row echelon form of $X(1 : 4, :)$. Thus we arrive at the following matrix.

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & -1 \\ 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 \end{bmatrix} - \begin{bmatrix} \mathbf{0}_{4,2} & \mathbf{0}_{4,2} & \mathbf{0}_{4,2} \\ A_1 & A_2 & A_3 \end{bmatrix} .$$

Now we do column operations to zero out the entries in the first four rows and last two columns. For a m -by- 2 matrix A we will use the notation A° to denote the matrix A with the order of the columns swapped. We arrive at

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & -2 \\ 1 & 0 & 1 & 1 & -2 & 0 \\ 1 & 1 & 0 & 0 & 2 & 2 \end{bmatrix} - \begin{bmatrix} \mathbf{0}_{4,2} & \mathbf{0}_{4,2} & \mathbf{0}_{4,2} \\ A_1 & A_2 & A_1^\circ - A_2 - A_2^\circ + A_3 \end{bmatrix}.$$

Finally, we can do row operations to zero out the first four columns in the last three rows.

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -2 \\ 0 & 0 & 0 & 0 & -2 & 0 \\ 0 & 0 & 0 & 0 & 2 & 2 \end{bmatrix} - \begin{bmatrix} \mathbf{0}_{4,2} & \mathbf{0}_{4,2} & \mathbf{0}_{4,2} \\ \mathbf{0}_{3,2} & \mathbf{0}_{3,2} & A_1^\circ - A_2 - A_2^\circ + A_3 \end{bmatrix}.$$

The task has now reduced to showing the matrix

$$Z' = \begin{bmatrix} 2 & 0 \\ 0 & 2 \\ -2 & -2 \end{bmatrix} + A_1 - A_2 - A_2^\circ + A_3^\circ$$

has rank 2 for any A_1, A_2, A_3 satisfying the constraints. Let us simplify the matrix $A_1 - A_2 - A_2^\circ + A_3^\circ$. First, let $A_1' = A_1 + A_3^\circ$. Next, note that $D = A_2 + A_2^\circ$ has the property that $D(i, 1) = D(i, 2)$ for $i \in [3]$. In the sequel we call this the *partner property*.

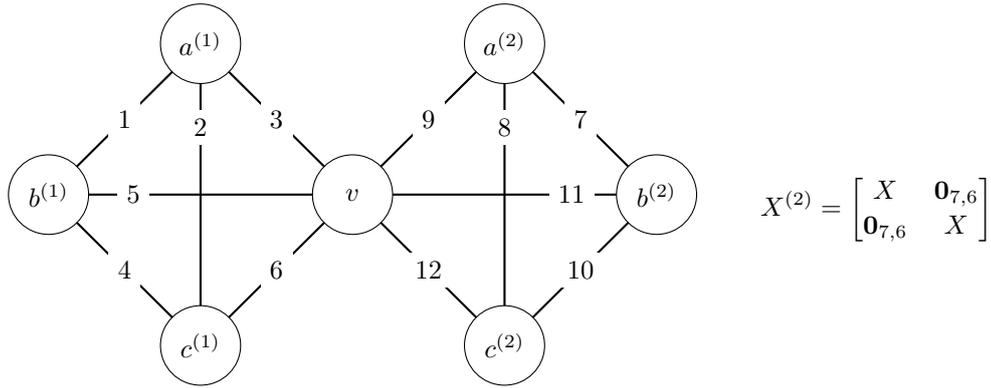
As the row sum of $A_1' + A_2$ is at most 1, unless $A_1'(1 : 2, 1 : 2) = \mathbf{0}_{2,2}$ and at least one row sum of $A_2(1 : 2, 1 : 2)$ is equal to 1 the first two rows of Z' will be strictly diagonally dominant. If the first two rows of Z' are strictly diagonally dominant then the rank of Z' must be 2, thus we now handle the “unless” case.

First, suppose exactly one row sum of $A_2(1 : 2, 1 : 2)$ is equal to 1. Say without loss of generality it is the second one, thus the first row of Z' is strictly diagonally dominant. Then for a sufficiently small ε we can multiply the first column by $1 - \varepsilon$ so that the first row remains strictly diagonally dominant and the second row becomes strictly diagonally dominant as well. This does not increase the rank and thus shows again that the rank of Z' is 2.

The remaining case is where both rows of $A_2(1 : 2, 1 : 2)$ sum to one. In this case by the partner property we have

$$Z'(1 : 2, 1 : 2) = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}.$$

On the other hand, the last row of Z' must have both entries ≤ -1 . Thus the determinant of the submatrix formed by the first row and the third is strictly negative and so Z' has rank 2. \triangleleft



■ **Figure 4** Example of the direct union of two copies of K_4 . With the ordering of the edges given by the edge labels, the matrix of cut vectors of the cuts $\Delta(\{a^{(i)}\})$, $\Delta(\{b^{(i)}\})$, $\Delta(\{c^{(i)}\})$, $\Delta(\{a^{(i)}, b^{(i)}, c^{(i)}\})$, $\Delta(\{a^{(i)}, b^{(i)}\})$, $\Delta(\{a^{(i)}, c^{(i)}\})$, $\Delta(\{b^{(i)}, c^{(i)}\})$ for $i \in [2]$ becomes the matrix $X^{(2)}$ on the right.

7.2 Direct union of K_4 with itself

Now we prove the general case. The key to the proof is the following lemma.

▶ **Lemma 40.** Let $k \in \mathbb{N}$ and B be the $3k$ -by- $2k$ matrix

$$B = \begin{bmatrix} 2\mathbf{I}_{2k} \\ -2\mathbf{I}_k \otimes [1, 1] \end{bmatrix}.$$

For any matrices $3k$ -by- $2k$ matrices A_1, A_2 satisfying the conditions

1. $A_1, A_2 \geq 0$
 2. (partner property) For all $i \in [3k]$ and $j \in [k]$ it holds that $A_2(i, 2j - 1) = A_2(i, 2j)$.
 3. Every row of $A_1 + A_2/2$ sums to at most 1
- it holds that $B + A_1 - A_2$ has rank $2k$.

Proof. The rank is at most $2k$ as that is the number of columns; we focus on showing the columns are linearly independent.

Let $Z = B + A_1 - A_2$. We call the first $2k$ rows of Z rows of type I, and the last k rows of type II. If a type I row is not strictly diagonally dominant, we call it *full*. Notice that a type I row i is full if and only if the i^{th} row of A_1 is zero and the i^{th} row of A_2 sums to 2. In this case, $Z(i, j) \leq 0$ for every $j \neq i$ and it holds that $Z(i, i) = -\sum_{j \neq i} Z(i, j)$. For $i \in [k]$ we call $2i - 1$ and $2i$ *partners*.

Suppose for contradiction there is a vector $\vec{u} \neq \mathbf{0}$ such that $A\vec{u} = \mathbf{0}$. As $\vec{u} \neq \mathbf{0}$ by normalizing and multiplying by -1 as needed we may assume that $\|u\|_\infty = 1$ and i is a coordinate with $\vec{u}(i) = 1$. By Fact 38 the i^{th} row of Z , which is a type I row, cannot be strictly diagonally dominant. Thus the i^{th} row must be full. Therefore for $Z(i, \cdot)\vec{u} = 0$ to hold it must be the case that $\vec{u}(j) = 1$ for every j where $A_2(i, j) > 0$. Such a j must exist as the i^{th} row of A_2 sums to 2. So let j be a coordinate with $A_2(i, j) > 0$ and let j' be the partner of j . By the partner property we also have $A_2(i, j') > 0$ and therefore $\vec{u}(j) = \vec{u}(j') = 1$.

Now consider the type II row ℓ for which $B(\ell, j) = B(\ell, j') = -2$. As $B(\ell, t) = 0$ for $t \notin \{j, j'\}$ this means

$$\begin{aligned} Z(\ell, \cdot) \vec{u} &= Z(\ell, j) + Z(\ell, j') + \sum_{t \notin \{j, j'\}} Z(\ell, t) \vec{u}(t) \\ &\leq B(\ell, j) + A_1(\ell, j) + B(\ell, j') + A_1(\ell, j') + \|\vec{u}\|_\infty \sum_{t \notin \{j, j'\}} |Z(\ell, t)| \\ &\leq -4 + \sum_t A_1(\ell, t) + A_2(\ell, t) \\ &\leq -2, \end{aligned}$$

and we have arrived at a contradiction. ◀

With Lemma 40 in hand we are now ready to prove Theorem 3.

Proof of Theorem 3. Let $G^{(1)}, \dots, G^{(k)}$ be k copies of K_4 where the vertices in $G^{(i)}$ are labeled by $a^{(i)}, b^{(i)}, c^{(i)}, v^{(i)}$ for $i \in [k]$. The graph G is formed by taking the direct union of $G^{(1)}, \dots, G^{(k)}$ at the vertices $v^{(1)}, \dots, v^{(k)}$. That is, the vertices $v^{(1)}, \dots, v^{(k)}$ are all identified by a common vertex denoted v . See Figure 4 for an illustration of the graph for $k = 2$.

The cuts of G we focus on are the $7k$ cuts given by

$$\Delta(\{a^{(i)}\}), \Delta(\{b^{(i)}\}), \Delta(\{c^{(i)}\}), \Delta(\{a^{(i)}, b^{(i)}, c^{(i)}\}), \Delta(\{a^{(i)}, b^{(i)}\}), \Delta(\{a^{(i)}, c^{(i)}\}), \Delta(\{b^{(i)}, c^{(i)}\},$$

for $i \in [k]$. For any $i \in [k]$ the cuts $\Delta(\{a^{(i)}\}), \Delta(\{b^{(i)}\}), \Delta(\{c^{(i)}\}), \Delta(\{a^{(i)}, b^{(i)}, c^{(i)}\})$ achieve the minimum cut weight of G , which is 3, and the cuts $\Delta(\{a^{(i)}, b^{(i)}\}), \Delta(\{a^{(i)}, c^{(i)}\}), \Delta(\{b^{(i)}, c^{(i)}\})$ have weight 4.

With an ordering of the edges as exemplified in Figure 4, the matrix of cut vectors of these cuts is $X^{(k)} = \mathbf{I}_k \otimes X$, where X is the matrix from Equation (10). In every nonzero block of $X^{(k)}$ the first four rows are minimum cuts with weight 3 and the last 3 rows are cuts with weight 4. Let $c' = (0, 0, 0, 0, 1, 1, 1)$. The theorem will follow from Theorem 13 by showing that the $w = \mathbf{1}_{6k}, c = \mathbf{1}_k \otimes c'$ one-sided ℓ_1 approximate rank of $X^{(k)}$ is $6k$.

To do this, we must show that $X^{(k)} - A$ has rank $6k$ for any matrix $A \geq 0$ which is all zero on any row of $I_k \otimes X$ corresponding to a minimum cut, and where the row sum of A is at most 1 on any row of $I_k \otimes X$ corresponding to a cut of weight 4. In order to make reference to the base case, it will be useful to partition the columns into k blocks of 6 columns, where the i^{th} block is further partitioned into blocks of size 2 represented by the $7k$ -by-2 matrices $A_1^{(i)}, A_2^{(i)}, A_3^{(i)}$. In other words, we view A as follows

$$A = \begin{bmatrix} A_1^{(1)} & A_2^{(1)} & A_3^{(1)} & \dots & A_1^{(k)} & A_2^{(k)} & A_3^{(k)} \end{bmatrix}$$

where each $A_j^{(i)}$ for $j \in [3], i \in [k]$ is a $7k$ -by-2 matrix.

As in the base case, we begin by doing Gauss-Jordan elimination on the rows corresponding to mincuts of each X block in $X^{(k)}$. These operations only touch rows corresponding to mincuts where A is zero, thus they do not change A . After these operations we arrive at the matrix $\mathbf{I}_k \otimes X' - A$ where

$$X' = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & -1 \\ 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 \end{bmatrix}$$

15:30 On the Cut Dimension of a Graph

Next, as in the base case, we do column operations to zero out the last two columns in the first four rows of each block of X' . This gives us the matrix $\mathbf{I}_k \otimes X'' - A'$ where

$$X'' = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & -2 \\ 1 & 0 & 1 & 1 & -2 & 0 \\ 1 & 1 & 0 & 0 & 2 & 2 \end{bmatrix}$$

and the i^{th} block of A' looks like

$$[A_1^{(i)} \mid A_2^{(i)} \mid A_1^{(i)\circ} - A_2^{(i)} - A_2^{(i)\circ} + A_3^{(i)}]. \quad .$$

Here $A_1^{(i)\circ}$ denotes the matrix $A_1^{(i)}$ with the order of the columns swapped. Finally, we use $X''(1 : 4, 1 : 4)$ to zero out all other entries of $\mathbf{I}_k \otimes X'' - A'$ in the first 4 columns of each block. This brings us to the matrix $\mathbf{I}_k \otimes X''' - A''$ where

$$X''' = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -2 \\ 0 & 0 & 0 & 0 & -2 & 0 \\ 0 & 0 & 0 & 0 & 2 & 2 \end{bmatrix}$$

and the i^{th} block of A'' is

$$[\mathbf{0}_{7k,2} \mid \mathbf{0}_{7k,2} \mid A_1^{(i)\circ} - A_2^{(i)} - A_2^{(i)\circ} + A_3^{(i)}]. \quad .$$

Again, each of $A_1^{(i)}, A_2^{(i)}, A_3^{(i)}$ is zero on rows corresponding to minimum cuts. Thus by multiplying the last two columns of each block by -1 and permuting rows and columns we can transform $\mathbf{I}_k \otimes X''' - A''$ into the form

$$\begin{bmatrix} \mathbf{I}_{4k} & \mathbf{0}_{4k,2k} \\ \mathbf{0}_{3k,4k} & B + A_1 - A_2 \end{bmatrix}$$

where B, A_1, A_2 satisfy the conditions of Lemma 40. Thus a rank lower bound of $6k$ follows from the lower bound of $2k$ on the rank of $B + A_1 - A_2$ given in Lemma 40. \blacktriangleleft

8 The dimension of approximate mincuts

Let G be a weighted graph and λ the weight of a minimum cut in G . For $\alpha \geq 1$ define an α -near-mincut of G to be a cut S whose weight is at most $\alpha\lambda$. Let $\mathcal{M}_\alpha(G)$ be the set of all α -near-mincuts of G and $\vec{\mathcal{M}}_\alpha(G) = \{\chi(S) : S \in \mathcal{M}_\alpha(G)\}$. In this section, we look at $\text{cdim}_\alpha(G) = \dim(\text{span}(\vec{\mathcal{M}}_\alpha(G)))$.

The first observation is that if $\alpha = 2$ then the unweighted complete graph K_n satisfies $\text{cdim}_\alpha(K_n) = \binom{n}{2}$. For *simple* graphs we can show $\alpha = 2$ is a sharp threshold.

► Theorem 41. *Let $1 \leq \alpha < 2$ be a constant and G be a simple n -vertex graph. Then $\text{cdim}_\alpha(G) = O(n)$.*

The key to this theorem is the following lemma of Rubinstein, Schramm, and Weinberg [26].

► **Lemma 42** (Lemma 2.6 [26]). *Let G be a simple graph with minimum degree d_{\min} and minimum cut value λ . For constant $0 \leq \epsilon < 1$ let \mathcal{T} be the set of non-star cuts of G whose weight is at most $\lambda + \epsilon d_{\min}$. Then $|\cup_{T \in \mathcal{T}} T| = O(n)$.*

Proof of Theorem 41. Let G be a simple graph. To prove the theorem we create a set of $O(n)$ vectors that span $\vec{\mathcal{M}}_\alpha(G)$. Let $\mathcal{M}_\alpha(G) = \mathcal{T} \sqcup \mathcal{S}$, where \mathcal{T} is the set of non-star cuts of $\mathcal{M}_\alpha(G)$ and \mathcal{S} is the set of star cuts of $\mathcal{M}_\alpha(G)$. Let $E' = \cup_{T \in \mathcal{T}} T$ be the set of edges involved in the cuts in \mathcal{T} . Let $\vec{\mathcal{L}} = \{e_i : i \in E'\}$. Note that from the definition of d_{\min} , there is a star cut with cut value d_{\min} , which implies that $\lambda \leq d_{\min}$. As a result, every α -near-mincut has cut value at most $\alpha\lambda \leq \lambda + (\alpha - 1)d_{\min}$, and hence by Lemma 42 we have $|\vec{\mathcal{L}}| = O(n)$. Also $\text{span}(\vec{\mathcal{T}}) \subseteq \text{span}(\vec{\mathcal{L}})$. Thus $\text{span}(\vec{\mathcal{M}}_\alpha(G)) \subseteq \text{span}(\vec{\mathcal{L}} \cup \vec{\mathcal{S}})$. As $|\mathcal{S}| \leq n$ this is a spanning set of size $O(n)$. ◀

In a previous version of this work we conjectured that for an n -vertex *weighted* graph G it holds that $\text{cdim}_\alpha(G) = O(n)$ for any $\alpha < 4/3$. This turns out to be false, however. The reason is that, on the one hand, in a graph $G = (V, w)$ the characteristic vector of a cut $\chi(S)$ depends only on the set of edges, but not the weight of these edges. On the other hand, $w(S)$ does of course depend on the weight of the edges. We can utilize this difference to construct an example as follows. Let us start with a cycle C_n with all edge weights being 1. While C_n has $\binom{n}{2}$ mincuts with weight 2, these mincuts live in an n -dimensional space as C_n only has n edges. We can then turn C_n into a complete weighted graph G by adding a tiny weight $\epsilon = 2(\alpha - 1)/\binom{n}{2}$ edge to all pairs of vertices that are not adjacent in the cycle. As adding edges cannot decrease the minimum cut weight, the weight of a minimum cut in G is at least 2. Further, if X is the shore of a minimum cut in C_n then in the graph G we have $w(\Delta(X)) \leq 2 + \binom{n}{2}\epsilon = 2\alpha$, as the weight is at most its weight in C_n plus the weight of all added edges. Thus $\Delta(X)$ is an α -near-mincut in G . Further, the characteristic vectors $\chi(\Delta(X)) \in \{0, 1\}^{\binom{n}{2}}$ of these cuts in G now live in an $\binom{n}{2}$ -dimensional space and become linearly independent. This example demonstrates that a reasonable extension of the cut dimension to near-mincuts should take into account the magnitude of the edge weights, as the ℓ_1 -approximate cut dimension does.

We now give the formal proof that the graph G mentioned above has the correct properties.

► **Lemma 43.** *Let $n \in \mathbb{N}$. Let C_n be the cycle on n vertices and \mathcal{G} the beach of $\mathcal{M}(C_n)$. Let K_n be the complete graph on n vertices. Let $\mathcal{T} = \{\Delta(X) : X \in \mathcal{G}\}$, where here $\Delta(X) \in \{0, 1\}^{\binom{n}{2}}$ is the cut in K_n with shore X . Then $\dim(\text{span}(\vec{\mathcal{T}})) = \binom{n}{2}$.*

Proof. For this proof we assume the vertices are labeled by $0, \dots, n - 1$ and use addition modulo n . We will show that all of the standard basis vectors $e_{\{i,j\}}$ are in $\text{span}(\vec{\mathcal{T}})$. For concreteness, we show how to construct the vectors $e_{\{0,j\}}$; by symmetry the same argument can then be used for any $e_{\{i,j\}}$.

We will actually construct the vectors $E_j = \sum_{k=1}^j e_{\{0,k\}}$. This suffices as $e_{\{0,j\}} = E_j - E_{j-1}$. First note that $e_{\{0,1\}} = \frac{1}{2}(\chi(\Delta(\{0\}) + \chi(\Delta(\{1\})) - \chi(\Delta(\{0, 1\})))$, and thus is in $\text{span}(\vec{\mathcal{T}})$ as all the vectors on the right hand side are in $\vec{\mathcal{T}}$.

Now let $j > 1$ and $X = \{1, \dots, j\}, X' = X \cup \{0\}$. Then

$$\chi(\Delta(X))(e) - \chi(\Delta(X'))(e) = \begin{cases} 1 & \text{if } e = \{0, k\}, k \in X \\ -1 & \text{if } e = \{0, k\}, k \in \bar{X}' \\ 0 & \text{otherwise} \end{cases} .$$

Thus $E_j = \frac{1}{2}(\Delta(\{0\}) + \chi(\Delta(X)) - \chi(\Delta(X')))$. ◀

► **Theorem 44.** Let $n \in \mathbb{N}$. For any $\alpha > 1$ there exists a graph $G = (\{0, \dots, n-1\}, w)$ such that $\text{cdim}_\alpha(G) = \binom{n}{2}$.

Proof. We again use addition modulo n on the labels of the vertices. Let $\varepsilon = 2(\alpha - 1)/\binom{n}{2}$. Define $w(\{i, i+1\}) = 1$ for $i \in \{0, \dots, n-1\}$ and for any other i, j let $w(\{i, j\}) = \varepsilon$. Let $G = (\{0, \dots, n-1\}, w)$. Thus G is the graph of the cycle C_n with edges of weight ε added between all pairs of vertices that are not adjacent in the cycle. The weight of a minimum cut of G is at least that of C_n , which is 2, as adding edges cannot decrease the weight of a cut. Further, if X is the shore of a minimum cut in C_n then in the graph G we have $w(\Delta(X)) \leq 2 + \binom{n}{2}\varepsilon = 2\alpha$, as the weight is at most its weight in C_n plus the weight of all added edges. Thus $\Delta(X)$ is an α -near-minicut in G and $\text{cdim}_\alpha(G)$ is at least $\binom{n}{2}$ by Lemma 43. It also clearly cannot be larger than $\binom{n}{2}$ and so the theorem is proved. ◀

References

- 1 Sepehr Assadi, Deeparnab Chakrabarty, and Sanjeev Khanna. Graph connectivity and single element recovery via linear and OR queries. *CoRR*, abs/2007.06098, 2020. [arXiv:2007.06098](#).
- 2 László Babai, Peter Frankl, and Janos Simon. Complexity classes in communication complexity theory (preliminary version). In *27th Annual Symposium on Foundations of Computer Science, Toronto, Canada, 27-29 October 1986*, pages 337–347, 1986. [doi:10.1109/SFCS.1986.15](#).
- 3 András A. Benczúr and Michel X. Goemans. Deformable polygon representations and near-mincuts. In Martin Grötschel and Gyula O. H. Katona, editors, *Building Bridges: Between Mathematics and Computer Science*, volume 19 of *Bolyai Society Mathematical Studies*, pages 103–135. Springer, 2008.
- 4 R. E. Bixby. The minimum number of edges and vertices in a graph with edge connectivity n and m n -bonds. *Netw.*, 5(3):253–298, 1975. [doi:10.1002/net.1975.5.3.253](#).
- 5 L. Sunil Chandran and L. Shankar Ram. On the number of minimum cuts in a graph. *SIAM J. Discret. Math.*, 18(1):177–194, 2004. [doi:10.1137/S0895480103427138](#).
- 6 Efim A. Dinitz, Alexander V. Karzanov, and Michael V. Lomonosov. On the structure of the system of minimum edge cuts of a graph. *Studies in discrete optimization*, 1976.
- 7 Tamás Fleiner and András Frank. A quick proof for the cactus representation of mincuts. *EGRES Quick Proof*, 2009-03, 2009.
- 8 Pawel Gawrychowski, Shay Mozes, and Oren Weimann. Minimum cut in $O(m \log^2 n)$ time. In Artur Czumaj, Anuj Dawar, and Emanuela Merelli, editors, *47th International Colloquium on Automata, Languages, and Programming, ICALP 2020, July 8-11, 2020, Saarbrücken, Germany (Virtual Conference)*, volume 168 of *LIPICs*, pages 57:1–57:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. [doi:10.4230/LIPICs.ICALP.2020.57](#).
- 9 Pawel Gawrychowski, Shay Mozes, and Oren Weimann. A note on a recent algorithm for minimum cut. In Hung Viet Le and Valerie King, editors, *4th Symposium on Simplicity in Algorithms, SOSA 2021, Virtual Conference, January 11-12, 2021*, pages 74–79. SIAM, 2021. [doi:10.1137/1.9781611976496.8](#).
- 10 Mohsen Ghaffari, Krzysztof Nowicki, and Mikkel Thorup. Faster algorithms for edge connectivity via random 2-out contractions. In Shuchi Chawla, editor, *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020*, pages 1260–1279. SIAM, 2020. [doi:10.1137/1.9781611975994.77](#).
- 11 Michel X. Goemans. Minimum bounded degree spanning trees. In *47th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2006), 21-24 October 2006, Berkeley, California, USA, Proceedings*, pages 273–282. IEEE Computer Society, 2006. [doi:10.1109/FOCS.2006.48](#).
- 12 Michel X. Goemans and V. S. Ramakrishnan. Minimizing submodular functions over families of sets. *Comb.*, 15(4):499–513, 1995. [doi:10.1007/BF01192523](#).

- 13 Andrei Graur, Tristan Pollner, Vidhya Ramaswamy, and S. Matthew Weinberg. New query lower bounds for submodular function minimization. In Thomas Vidick, editor, *11th Innovations in Theoretical Computer Science Conference, ITCS 2020, January 12-14, 2020, Seattle, Washington, USA*, volume 151 of *LIPICs*, pages 64:1–64:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPICs.ITCS.2020.64.
- 14 András Hajnal, Wolfgang Maass, and György Turán. On the communication complexity of graph properties. In *Proceedings of the 20th Annual ACM Symposium on Theory of Computing, May 2-4, 1988, Chicago, Illinois, USA*, pages 186–191, 1988. doi:10.1145/62212.62228.
- 15 Nicholas J. A. Harvey. *Matchings, matroids and submodular functions*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 2008. URL: <http://hdl.handle.net/1721.1/44416>.
- 16 Monika Henzinger, Satish Rao, and Di Wang. Local flow partitioning for faster edge connectivity. *SIAM J. Comput.*, 49(1):1–36, 2020. doi:10.1137/18M1180335.
- 17 Monika Rauch Henzinger and David P. Williamson. On the number of small cuts in a graph. *Inf. Process. Lett.*, 59(1):41–44, 1996. doi:10.1016/0020-0190(96)00079-8.
- 18 Kamal Jain. A factor 2 approximation algorithm for the generalized Steiner network problem. *Comb.*, 21(1):39–60, 2001. doi:10.1007/s004930170004.
- 19 David R. Karger. Global min-cuts in RNC, and other ramifications of a simple min-cut algorithm. In Vijaya Ramachandran, editor, *Proceedings of the Fourth Annual ACM/SIGACT-SIAM Symposium on Discrete Algorithms, 25-27 January 1993, Austin, Texas, USA*, pages 21–30. ACM/SIAM, 1993. URL: <http://dl.acm.org/citation.cfm?id=313559.313605>.
- 20 David R. Karger. Minimum cuts in near-linear time. *J. ACM*, 47(1):46–76, 2000. doi:10.1145/331605.331608.
- 21 Ken-ichi Kawarabayashi and Mikkel Thorup. Deterministic edge connectivity in near-linear time. *J. ACM*, 66(1):4:1–4:50, 2019. doi:10.1145/3274663.
- 22 Bernhard Korte and Jens Vygen. *Combinatorial Optimization: Theory and Algorithms*. Springer, 2018.
- 23 László Lovász. *Combinatorial problems and exercises (2. ed.)*. North-Holland, 1993.
- 24 Sagnik Mukhopadhyay and Danupon Nanongkai. Weighted min-cut: sequential, cut-query, and streaming algorithms. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC 2020, Chicago, IL, USA, June 22-26, 2020*, pages 496–509, 2020.
- 25 Hiroshi Nagamochi, Kazuhiro Nishimura, and Toshihide Ibaraki. Computing all small cuts in an undirected network. *SIAM J. Discret. Math.*, 10(3):469–481, 1997. doi:10.1137/S0895480194271323.
- 26 Aviad Rubinfeld, Tselil Schramm, and S. Matthew Weinberg. Computing exact minimum cuts without knowing the graph. In *9th Innovations in Theoretical Computer Science Conference, ITCS 2018, January 11-14, 2018, Cambridge, MA, USA*, pages 39:1–39:16, 2018. doi:10.4230/LIPICs.ITCS.2018.39.

A Jain’s spanning lemma

In this appendix we prove Lemma 19. The proof uses the following key property of mincuts which goes back at least to work of Dinitz, Karzanov, and Lomonosov [6].

► **Proposition 45** ([6] “Lemma on a quadrangle”). *Let $G = (V, w)$ be a graph. For any crossing mincuts $\Delta(X), \Delta(Y)$ of G it holds that*

$$\chi(\Delta(X)) + \chi(\Delta(Y)) = \chi(\Delta(X \cap Y)) + \chi(\Delta(X \cup Y)) .$$

Proof. If $\Delta(X), \Delta(Y)$ cross then $\Delta(X \cap Y), \Delta(X \cup Y)$ are mincuts of G by Claim 5. Further, by counting the number of times an edge appears on each side it can be seen (eg. Ex. 6.48 in [23]) that

$$\chi(\Delta(X)) + \chi(\Delta(Y)) = \chi(\Delta(X \cap Y)) + \chi(\Delta(X \cup Y)) + 2\chi(E(X - Y, Y - X)) . \quad (11)$$

15:34 On the Cut Dimension of a Graph

Let the minimum cut value of G be λ . Let m be the number of edges in G and $\vec{w} \in \mathbb{R}^m$ be the positive vector resulting from restricting w to the edges of G . The inner product of \vec{w} with the left hand side of Equation (11) is 2λ , and with the righthand side is $2\lambda + 2\langle \vec{w}, \chi(E(X - Y, Y - X)) \rangle$. Thus $\langle \vec{w}, \chi(E(X - Y, Y - X)) \rangle = 0$, which implies $\chi(E(X - Y, Y - X)) = \mathbf{0}$ since \vec{w} is positive and $\chi(E(X - Y, Y - X))$ is nonnegative. ◀

Jain's proof uses the technique of *combinatorial uncrossing*. Recall the definition of $\text{overlap}_{\mathcal{G}}(X)$ from Definition 28. A key to the proof is the following simple lemma about $\text{overlap}_{\mathcal{G}}(X)$.

► **Lemma 46** ([18]). *Let $\mathcal{F} \subseteq 2^V$ be a set family closed under overlaps and $\mathcal{G} \subseteq \mathcal{F}$ be a maximal laminar subset of \mathcal{F} . Then for any $X \in \mathcal{F} - \mathcal{G}$ and $Y \in \text{overlap}_{\mathcal{G}}(X)$*

$$\text{overlap}_{\mathcal{G}}(X \cap Y) \subset \text{overlap}_{\mathcal{G}}(X) \quad (12)$$

$$\text{overlap}_{\mathcal{G}}(X \cup Y) \subset \text{overlap}_{\mathcal{G}}(X) \quad (13)$$

Proof. In the following we always refer to $\text{overlap}(X)$ with respect to \mathcal{G} and drop the subscript. We first show Equation (12). First note that $Y \in \text{overlap}(X) - \text{overlap}(X \cap Y)$. Thus to show Equation (12) it suffices to show $\text{overlap}(X \cap Y) \subseteq \text{overlap}(X)$. Let $W \in \text{overlap}(X \cap Y)$. We want to show that $W \in \text{overlap}(X)$, i.e. that it cannot be the case that $W \subseteq X, X \subseteq W$, or $X \cap W = \emptyset$. We know that the last one cannot hold because $W \in \text{overlap}(X \cap Y)$ implies $W \cap (X \cap Y) \neq \emptyset$.

Also as $W, Y \in \mathcal{G}$ they do not overlap and thus either $Y \subseteq W, W \subseteq Y$, or $Y \cap W = \emptyset$. Again the last one cannot hold as $W \cap (X \cap Y) \neq \emptyset$. The following table shows that assuming $W \notin \text{overlap}(X)$ leads to a contradiction in all 4 remaining cases.

	$Y \subseteq W$	$W \subseteq Y$
$W \subseteq X$	$Y \subseteq X$ $Y \notin \text{overlap}(X)$	$W \subseteq X \cap Y$ $W \notin \text{overlap}(X \cap Y)$
$X \subseteq W$	$X \cap Y \subseteq W$ $W \notin \text{overlap}(X \cap Y)$	$X \subseteq Y$ $Y \notin \text{overlap}(X)$

We now show Equation (13), which follows similarly. Again $Y \in \text{overlap}(X) - \text{overlap}(X \cup Y)$ thus it suffices to show $\text{overlap}(X \cup Y) \subseteq \text{overlap}(X)$. Let $W \in \text{overlap}(X \cup Y)$. We want to show that $W \in \text{overlap}(X)$, i.e. that is not the case that either $W \cap X = \emptyset, X \subseteq W$, or $W \subseteq X$. We cannot have $W \subseteq X$ because this means $W \subseteq X \cup Y$ which contradicts $W \in \text{overlap}(X \cup Y)$. As $W, Y \in \mathcal{G}$ they do not overlap, so we also know either $Y \subseteq W, W \cap Y = \emptyset$, or $W \subseteq Y$. The last one again cannot hold as it implies $W \subseteq X \cup Y$. The following table shows that assuming $W \notin \text{overlap}(X)$ leads to a contradiction in the remaining 4 cases.

	$Y \subseteq W$	$W \cap Y = \emptyset$
$X \subseteq W$	$X \cup Y \subseteq W$ $W \notin \text{overlap}(X \cup Y)$	$X \cap Y = \emptyset$ $Y \notin \text{overlap}(X)$
$X \cap W = \emptyset$	$Y \cap X = \emptyset$ $Y \notin \text{overlap}(X)$	$W \cap (X \cup Y) = \emptyset$ $W \notin \text{overlap}(X \cup Y)$

◀

We are now ready to show the key lemma of Jain.

► **Lemma 19** ([18]). *Let $G = (V, w)$ be a graph and $\mathcal{L} \subseteq \mathcal{M}(G)$ be a maximal cross-free family of mincuts. Then $\text{span}(\vec{\mathcal{L}}) = \text{span}(\vec{\mathcal{M}}(G))$.*

Proof. It is clear that $\text{span}(\vec{\mathcal{L}}) \subseteq \text{span}(\vec{\mathcal{M}}(G))$ so we focus on the other direction.

Let \mathcal{F} be the beach of $\mathcal{M}(G)$. By Claim 5 \mathcal{F} is closed under overlaps. Let $\mathcal{G} \subseteq \mathcal{F}$ be the beach of \mathcal{L} . As \mathcal{L} is a maximal cross-free subset of $\mathcal{M}(G)$ it follows that \mathcal{G} is a maximal laminar subset of \mathcal{F} . Thus $|\text{overlap}_{\mathcal{G}}(X)| \geq 1$ for all $X \in \mathcal{F} - \mathcal{G}$. In the following we will always refer to $\text{overlap}(X)$ with respect to \mathcal{G} and drop the subscript.

Suppose for a contradiction that $\text{span}(\vec{\mathcal{L}})$ is a strict subset of $\text{span}(\vec{\mathcal{M}}(G))$. Let

$$X = \underset{Z \in \mathcal{F} - \mathcal{G}}{\text{argmin}} \{ |\text{overlap}(Z)| : \chi(\Delta(Z)) \notin \text{span}(\vec{\mathcal{L}}) \} .$$

As $\text{overlap}(X) \geq 1$, let $Y \in \text{overlap}(X)$. By Lemma 46

$$|\text{overlap}(X \cap Y)| < |\text{overlap}(X)| \tag{14}$$

$$|\text{overlap}(X \cup Y)| < |\text{overlap}(X)| . \tag{15}$$

By the definition of X , and as \mathcal{F} is closed under overlaps, we must have $\chi(\Delta(X \cap Y)), \chi(\Delta(X \cup Y)) \in \text{span}(\vec{\mathcal{L}})$. Also as $Y \in \mathcal{G}$ we have $\chi(\Delta(Y)) \in \vec{\mathcal{L}}$ which implies by Proposition 45 that

$$\chi(\Delta(X)) = \chi(\Delta(X \cap Y)) + \chi(\Delta(X \cup Y)) - \chi(\Delta(Y)) .$$

This implies $\chi(\Delta(X)) \in \text{span}(\vec{\mathcal{L}})$, a contradiction. ◀