Generalized cut trees for edge-connectivity

On-Hei S. Lo[*†] Jens M. Schmidt[‡†]

Abstract

We present three cut trees of graphs, each of them giving insights into the edge-connectivity structure. All three cut trees have in common that they are defined with respect to a given binary symmetric relation R on the vertex set of the graph, which generalizes Gomory-Hu trees. Applying these cut trees, we prove the following:

- A pair of vertices $\{v, w\}$ of a graph G is *pendant* if $\lambda(v, w) = \min\{d(v), d(w)\}$. Mader showed in 1974 that every simple graph with minimum degree δ contains at least $\delta(\delta + 1)/2$ pendant pairs. We improve this lower bound to $\delta n/24$ for every simple graph G on n vertices with $\delta \geq 5$ or $\lambda \geq 4$ or vertex connectivity $\kappa \geq 3$, and show that this is optimal up to a constant factor with regard to every parameter.
- Every simple graph G satisfying $\delta > 0$ has $O(n/\delta)$ δ-edge-connected components. Moreover, every simple graph G that satisfies $0 < \lambda < \delta$ has $O((n/\delta)^2)$ cuts of size less than $\min\{\frac{3}{2}\lambda, \delta\}$, and $O((n/\delta)^{\lfloor 2\alpha \rfloor})$ cuts of size at most $\min\{\alpha \cdot \lambda, \delta - 1\}$ for any given reel number $\alpha \geq 1$.
- A cut is *trivial* if it or its complement in $V(G)$ is a singleton. We provide an alternative proof of the following recent result of Thorup et al. (to appear in Discrete Applied Mathematics): Given a simple graph G on n vertices that satisfies $\delta > 0$, we can compute vertex subsets of G in near-linear time such that contracting these vertex subsets separately preserves all non-trivial min-cuts of G and leaves a graph having $O(n/\delta)$ vertices and $O(n)$ edges.

1 Introduction

We propose a general notion of cut trees that are defined with respect to a given binary relation R on the vertices. We consider only binary relations R that are irreflexive and symmetric, which allows us to see R as a set of unordered pairs $\{a, b\}$

[*]School of Mathematical Sciences, University of Science and Technology of China, Hefei, Anhui 230026, China.

[†]This work was supported by the grant SCHM 3186/1-1 (270450205) from the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation).

[‡]University of Rostock, Institute of Computer Science, Rostock, Germany.

satisfying $a \neq b$. We will study these generalized cut trees (see Definition 1) for three relations, each of them giving structural insights about the edge-connectivity in graphs.

We call a pair $\{v, w\}$ of vertices of a graph G *pendant* if $\lambda(v, w) = \min\{d(v), d(w)\}$, where d denotes the degree function on G. The study of pendant pairs is motivated by the well-known, simple and widely used min-cut algorithm of Nagamochi and Ibaraki [17], which refines the work of Mader [14, 13] in the early 70s, and was simplified by Stoer and Wagner [18] and Frank [4]. The key approach of this algorithm is to iteratively contract a pendant pair of the input graph $G := (V, E)$ in near-linear time by using *maximal adjacency orderings* (also known as *maximum cardinality search* [19]). Having done that $n - 2$ times, where $n := |V|$, one can obtain a min-cut by just considering the minimum degree of all intermediate graphs.

As early as 1973, and originally motivated by the structure of minimally k-edge-connected graphs, Mader proved that every graph with minimum degree $\delta \geq 1$ contains a pendant pair [14]. This holds also for the vertex-connectivity variant of pendant pairs, which nowadays is most easily proven by using maximal adjacency orderings. Later, Mader improved his result by showing that every simple graph with minimum degree δ contains in fact $\frac{\delta(\delta+1)}{2}$ pendant pairs [15]. Let κ be the *vertex connectivity* of a graph. By considering the cut tree that respects non-pendant pairs in Section 3, we improve Mader's result by showing that every simple graph that satisfies $\delta \geq 5$ or $\lambda \geq 4$ or $\kappa \geq 3$ contains $\frac{1}{24}\delta n$ pendant pairs. We show that this result is optimal up to a constant factor and that every of the three assumptions is best possible. We also show how to compute these $\frac{1}{24}\delta n$ pendant pairs from a Gomory-Hu tree in linear time.

In Section 4, we study two other cut trees, from which results on sparsification done by vertex subset contractions will be derived. In Section 4.1, we consider the cut tree that respects the vertex pairs $\{v, w\}$ for which $\lambda(v, w) < \delta$. We prove that every simple graph G with $\delta > 0$ has $O(n/\delta)$ δ-edge-connected components, and contracting these components leaves $O(n)$ edges. For a simple graph G satisfying $0 < \lambda < \delta$, it was recently shown [12] that G has $O((n/\delta)^2)$ min-cuts. We strengthen this and show that G has $O((n/\delta)^2)$ cuts of size less than $\min\{\frac{3}{2}\lambda, \delta\}$, and $O((n/\delta)^{\lfloor 2\alpha \rfloor})$ cuts of size not larger than $\min\{\alpha\lambda, \delta - 1\}$ for any given α, respectively.

Recently, Kawarabayashi and Thorup [10] gave the first deterministic near-linear time algorithm for finding a min-cut of G. Subsequently, Henzinger, Rao and Wang [5] obtained an improved variant with running time $O(m \log^2 n \log\log^2 n)$ by replacing the diffusion subroutine with a flow-based one. A crucial step in both algorithms is a sparsification routine [10, Theorem 3] for large minimum degree δ that contracts vertex subsets of G such that, after these sparsifications, the remaining graph has only $O((n \log^c n)/\delta)$ vertices and $O(n \log^c n)$ edges (for some constant c) and all non-trivial min-cuts of G are preserved. Thorup and the authors [12] later showed that, for a simple graph G satisfying $\delta > 0$, we can find some vertex subsets in near-linear time such that all non-trivial min-cuts are preserved, and only $O(n/\delta)$

vertices and $O(n)$ edges are left when these vertex subsets are contracted. This eliminates the poly-logarithmic factor needed above. Here, we introduce the cut tree that respects the vertex pairs that are separated by some non-trivial min-cut, and give an alternative proof of this new result (see Section 4.2). We also show that such a cut tree exists and can be computed in near-linear time for every simple graph G satisfying $\lambda \neq 0, 2$ (and that excluding these values 0 and 2 is necessary).

A Note on the History of Maximal Adjacency Orderings. Mader's proof for the existence of one pendant pair relies strongly on [13, Lemma 1], which uses special orderings on the vertices. Interestingly, these orderings are *maximal adjacency orderings* and this fact exhibits a nowadays almost forgotten variant of them, which remained long unnamed until 1984 [19]. We are only aware of one place in literature where this is (briefly) mentioned: [16, p. 443]. Mader's existential proof can in fact be made algorithmic. A direct comparison between his original and the modern variant however shows that the modern maximal adjacency orderings are nicer to describe, as they work directly on the input graph, while Mader iteratively modifies the graph by moving edges in order to represent the essential connectivity information on the already visited vertex set with a clique.

2 Notation

All graphs considered in this paper are non-empty, finite, unweighted and undirected; they may contain parallel edges but no loops. Let $G := (V, E)$ be a graph. *Contracting* a vertex subset $X \subseteq V$ identifies all vertices in X and deletes occurring loops (we do not require that X induces a connected graph in G).

For non-empty and disjoint vertex subsets $X, Y \subset V$, let $E_G(X, Y)$ denote the set of all edges in G that have one endvertex in X and one endvertex in Y. Let further $\overline{X} := V - X$, $E_G(X) := E_G(X, \overline{X})$, $d_G(X, Y) := |E_G(X, Y)|$ and $d_G(X) := |E_G(X)|$; if $X = \{v\}$ for some vertex $v \in V$, we simply write $E_G(v)$, $E_G(v, Y)$, $d_G(v, Y)$ and $d_G(v)$. For any subset $\emptyset \neq X \subset V$ of a graph G, we call $E_G(X)$ an *edge-cut*, or simply a *cut* of G. For a connected graph G and any $\emptyset \neq X, Y \subset V(G)$, $E_G(X) = E_G(Y)$ implies either $X = Y$ or $X = \overline{Y}$. Thus we may see X as a cut (representing $E_G(X, \overline{X})$) if there is no ambiguity. We say vertices $v, w \in V$ are *separated* by a cut X if $|X \cap \{v, w\}| = 1$. A cut X of G is *trivial* if $|X| = 1$ or $|\overline{X}| = 1$. Cuts X and Y are *uncrossing* if $X \subseteq Y$ or $Y \subseteq X$ or $X \cap Y = \emptyset$. The *length* and *size* of a path are the number of its edges and vertices, respectively. For a vertex v in G, let $N_G(v)$ be the set of neighbors of v in G, and $d_G(v) := |N_G(v)|$. The *minimum degree* of G is denoted by $\delta(G)$, or simply δ.

For two vertices $v, w \in V$, let $\lambda_G(v, w)$ be the maximal number of edge-disjoint paths between v and w in G. A *minimum v-w-cut* is a cut X that separates v and w and satisfies $d_G(X) = \lambda_G(v, w)$. Two vertices $v, w \in V$ are called *k-edge-connected* if $\lambda_G(v, w) \geq k$. For any k, the k-edge-connectivity relation on vertices is symmetric and transitive, and thus its reflexive closure is an equivalence relation that partitions V; let the *k-edge-connected components* be the blocks of this partition. The *edge-connectivity* $\lambda(G)$ of G is the greatest integer such that every two distinct vertices are $\lambda(G)$-edge-connected. A *min-cut* X is a cut with $d_G(X, \overline{X}) = \lambda(G)$. The *vertex-connectivity* $\kappa(G)$ of G is defined to be $|V| - 1$ if G is a complete graph and otherwise the smallest integer $\kappa(G)$ such that there exists a vertex subset $S \subseteq V$ of size $\kappa(G)$ and $G - S$ is disconnected. We omit parentheses for single elements (such as vertices or edges) in set subtractions. In order to increase readability, we will omit subscripts and parameters G whenever the graph is clear from the context.

2.1 Cut Trees

Let $\mathcal{T}$ be a tree whose vertex set is a partition of V. We will call the vertices of such trees *blocks*. Let $AB \in E(\mathcal{T})$ and let C_{AB} be the union of the blocks that are contained in the component of $\mathcal{T} - AB$ containing A, and symmetrically, $C_{BA} = \overline{C_{AB}}$. For an edge $AB \in E(\mathcal{T})$, let $c(AB) := d_G(C_{AB})$ be the *size* of its corresponding edge-cut in G.

Definition 1. Given an undirected graph $G := (V, E)$ and a binary relation R on V, a *cut tree $\mathcal{T}$ that respects* R is a tree whose vertex set is a partition of V, such that

(i) for every $A \in V(\mathcal{T})$ and every $a, a' \in A$, we have $\{a, a'\} \notin R$,

(ii) for every tree edge $AB \in E(\mathcal{T})$, there exist $a \in A$ and $b \in B$ that satisfy $\{a, b\} \in R$ and

(iii) for every tree edge $AB \in E(\mathcal{T})$, there exist $a^* \in A$ and $b^* \in B$ that satisfy $\lambda_G(a^*, b^*) = d_G(C_{AB})$, i.e. C_{AB} is a minimum a^*-b^*-cut in G.

This definition generalizes the well-known Gomory-Hu trees, since a Gomory-Hu tree is a cut tree that respects the maximal binary relation $\{\{v, w\} : v, w \in V, v \neq w\}$ on V: to see this, note that Condition (ii) becomes redundant and Condition (i) implies that every $A \in V(\mathcal{T})$ is a singleton. By choosing the binary relation R in the above cut tree appropriately, we will prove new facts about the edge-connectivity structure of graphs in the remaining sections.

3 The Pendant Tree

We call a cut tree $\mathcal{T}$ that respects the set of non-pendant vertex pairs a *pendant tree*. By Definition 1, we have that (i) every pair of two distinct vertices in a common block in $V(\mathcal{T})$ is pendant, (ii) for every edge $AB \in E(\mathcal{T})$, there are vertices $a \in A$ and $b \in B$ such that (a, b) is non-pendant, and (iii) for every edge $AB \in E(\mathcal{T})$, there are vertices $a^* \in A$ and $b^* \in B$ such that $c(AB) = \lambda_G(a^*, b^*)$.

The following lemma allows us to find a non-pendant pair from two adjacent blocks of a pendant tree efficiently.

Lemma 2. *Let AB be an edge of a pendant tree $\mathcal{T}$ and let a_{max} and b_{max} be vertices in A and B of maximum degrees, respectively. Then (a_{max}, b_{max}) is non-pendant.*

Proof. By Property (ii) of Definition 1, there are vertices $a \in A$ and $b \in B$ such that $\lambda(a, b) < \min\{d(a), d(b)\}$. Since (a, a_{max}) and (b, b_{max}) are pendant, a minimum a-b-cut can neither separate a from a_{max} nor b from b_{max}. Hence,

$$\begin{aligned}\lambda(a_{max}, b_{max}) &\leq \lambda(a, b)\\ &< \min\{d(a), d(b)\}\\ &\leq \min\{d(a_{max}), d(b_{max})\}.\end{aligned}$$

□

Property (iii) of pendant trees gives the following lemma.

Lemma 3. *Let AB be an edge of a pendant tree $\mathcal{T}$ and let a_{max} be a vertex in A of maximum degree. Then $c(AB) < d(a_{max})$.*

Proof. Let b_{max} be a vertex of maximum degree in B and let $a^* \in A$ and $b^* \in B$ be such that $c(AB) = \lambda(a^*, b^*)$ due to Property (iii). By transitivity of λ, we have

$$\begin{aligned}\lambda(a_{max}, b_{max}) &\geq \min\{\lambda(a_{max}, a^*), \lambda(a^*, b^*), \lambda(b^*, b_{max})\}\\ &= \min\{d(a^*), \lambda(a^*, b^*), d(b^*)\}\\ &= c(AB),\end{aligned}$$

where the first equality follows from the fact that (a_{max}, a^*) and (b_{max}, b^*) are pendant. According to Lemma 2, $\lambda(a_{max}, b_{max}) < d(a_{max})$, which gives the claim. □

3.1 Constructing Pendant Trees

We will construct a pendant tree by contracting edges in a Gomory-Hu tree. We recall that a *Gomory-Hu tree* $\mathcal{T}$ of a graph G is a tree on the vertex set V of G such that, for every edge ab in $\mathcal{T}$, C_{ab} is a minimum a-b-cut in G.

If we replace each vertex v in a Gomory-Hu tree by the singleton set $\{v\}$, then it is precisely a cut tree that respects the maximal irreflexive binary relation. We see a Gomory-Hu tree as such a cut tree.

Proposition 4. *Given a Gomory-Hu tree of a graph G, a pendant tree of G can be computed in linear time $O(|V(G)| + |E(G)|)$.*

Proof. Let $\mathcal{T}$ be a Gomory-Hu tree of G, seen as a cut tree that respects the maximal irreflexive binary relation. Throughout the algorithm, we maintain the invariant that every pair of distinct vertices that is contained in a block is pendant. Iteratively for every edge AB in $\mathcal{T}$, we check whether there is a non-pendant pair $\{a, b\}$ with $a \in A$ and $b \in B$. We contract AB in $\mathcal{T}$ and set the new block as $A \cup B$ if and only if there is no such non-pendant pair. We claim that there is such a non-pendant pair if and only if $\min\{d_G(a_{max}), d_G(b_{max})\} > c(AB)$, where a_{max} and b_{max} are vertices in A and B with maximum degrees, respectively. The sufficiency follows from the proof of Lemma 2, and it remains to show that if $\min\{d_G(a_{max}), d_G(b_{max})\} \leq c(AB)$, then $\{a,b\}$ is pendant for all $a \in A$ and $b \in B$.

Thus suppose that $\min\{d_G(a_{max}), d_G(b_{max})\} \leq c(AB)$. Without loss of generality, let $d_G(a_{max}) \leq c(AB)$, which implies $d_G(a) \leq c(AB)$ for all $a \in A$. Let $a \in A$ and $b \in B$. By the Gomory-Hu tree properties, $\mathcal{T}$ contains vertices $a^* \in A$ and $b^* \in B$ such that $\lambda_G(a^*, b^*) = c(AB)$; in particular, $d_G(b^*) \geq d_G(a^*) = c(AB)$. Then $\{a,b\}$ is pendant, since

$$\begin{aligned}
\lambda_G(a,b) &= \min\{\lambda_G(a,a^*), \lambda_G(a^*,b^*), \lambda_G(b^*,b)\} \\
&= \min\{d_G(a), d_G(a^*), c(AB), d_G(b^*), d_G(b)\} \\
&= \min\{d_G(a), d_G(b)\}.
\end{aligned}$$

The first equality is implied by the transitivity of local edge-connectivity, the second by the fact that every vertex pair of a block is pendant, and the third by $d_G(b^*) \geq d_G(a^*) = c(AB) \geq d_G(a)$.

By maintaining for every vertex $A \in V(\mathcal{T})$ the maximum degree in G of a vertex in A, we may evaluate $\min\{d_G(a_{max}), d_G(b_{max})\} > c(AB)$ for every edge $AB \in E(\mathcal{T})$ in constant time. As contracting $AB \in E(T)$ and computing the maximum degree of the contracted vertex $A \cup B$ takes also constant time (note that we do not have to represent $A \cup B$ explicitly in every intermediate step), the algorithm has running time $O(|V(G)| + |E(G)|)$. □

Proposition 4 implies in particular that every graph has a pendant tree.

A deterministic construction of a Gomory-Hu tree applies $n-1$ times the uncrossing technique to find uncrossing cuts on the input graph, and hence runs in time $O(n\theta_{flow})$, where θ_{flow} is the time needed for a maximum flow subroutine. By Dinits' classic algorithm, we know that $\theta_{flow} = O(n^{2/3}m)$ [2, 8]. Recent progress on deterministic maximum flow due to Lee and Sidford [11] and Kathuria, Liu and Sidford [9] shows the improved running time bounds $\widetilde{O}(n^{1/2}m)$ and $O(m^{4/3+o(1)})$ for θ_{flow}, where the tilde hides polylogarithmic factors.

For randomized algorithms, it was shown in [1] that a Gomory-Hu tree of a simple unweighted graph can be constructed in expected running time $\widetilde{O}(nm)$. Therefore, by our construction above, we conclude that:

Corollary 5. *Given a simple graph G, a pendant tree of G can be constructed deterministically in running times $\widetilde{O}(n^{3/2}m)$ and $O(nm^{4/3+o(1)})$, respectively, and randomized in expected running time $\widetilde{O}(nm)$.*

3.2 Large Blocks of Degree 1 and 2

In this section, we show that the blocks of a pendant tree have large sizes on average. This implies a lower bound on the number of pendant pairs, as every pair of vertices in a block is pendant. For any tree $\mathcal{T}$ whose vertex set partitions V, let $\mathcal{V}_k$ be the set of blocks of $\mathcal{T}$ having degree k in $\mathcal{T}$ and let $\mathcal{V}_{>k} := \bigcup_{k'>k} \mathcal{V}_{k'}$. We call the blocks in $\mathcal{V}_1$ *leaf blocks*. In $\mathcal{T}$, the set $\mathcal{V}_2$ induces a family of disjoint paths; we call each such path a *straight path*. We will prove that the leaf blocks of pendant trees as well as the blocks that are contained in straight paths are large.

Lemma 6. *Let $\mathcal{T}$ be a pendant tree of a simple graph G. Then every leaf block A of $\mathcal{T}$ satisfies $|A| > \delta(G)$.*

Proof. Let $p := |A| \geq 1$ and let B be the block adjacent to A in $\mathcal{T}$. By Lemma 3, we have $\max_{v \in A} d(v) > c(AB) \geq \sum_{v \in A}(d(v)-(p-1)) \geq \max_{v \in A} d(v) + \delta(p-1) - p(p-1)$, where the last inequality singles out the maximum degree. Therefore, $p > 1$ and $p > \delta$. □

Let a_{max} be a vertex of maximal degree in a leaf block A with neighbor B. Since $c(AB) < d(a_{max})$, A must actually contain a vertex that has all its neighbors in A, as otherwise each of the $d(a_{max})$ incident edges of a_{max} would contribute at least one edge to the edge-cut, either directly or by an incident edge of the corresponding neighbor of a_{max}. This gives the following corollary of Lemma 6, of which the existence of one such vertex subset A was first shown by Mader (without using pendant trees).

Corollary 7 ([15])**.** *Let $\mathcal{T}$ be a pendant tree of a simple graph G. Then every leaf block A contains a vertex v with $N(v) \subseteq A$. Hence, every pair in $\{v\} \cup N(v)$ is pendant.*

This already implies that simple graphs contain $\binom{\delta+1}{2} = \Omega(\delta^2)$ pendant pairs. Note that Lemma 6 and Corollary 7 do not hold for graphs having parallel edges: for example, consider a block A that consists of two vertices of degree δ, which are joined by $\delta - 1$ parallel edges. However, even if the graph is not simple, a leaf block A must always contain at least two vertices due to Lemma 3.

Corollary 8. *Every leaf block of a pendant tree of a graph contains at least two vertices.*

For simple graphs, we thus know that leaf blocks give us a large number of pendant pairs. Since $\mathcal{T}$ is a tree, the number of leaf blocks is completely determined by the number of blocks of degree at least 3, namely $|\mathcal{V}_1| = \sum_{A \in \mathcal{V}_{>2}}(d_T(A) - 2) + 2$. Thus, in order to prove a better lower bound on the number of pendant pairs, we have to consider the case that there are many small blocks of size $o(\delta)$ contained in straight paths. The following two lemmas prove that (i) for every two adjacent blocks A and B in a straight path that satisfy $|A|+|B| > 2$, we have $|A|+|B| \geq \delta - 1 = \Omega(\delta)$ and (ii) if $\delta \geq 5$ or $\lambda \geq 4$ or $\kappa \geq 3$ and $\mathcal{P}$ is a subpath of a straight path such that all blocks of $\mathcal{P}$ are singletons, then $\mathcal{P}$ contains at most two blocks (see Corollary 11). This will be used later to show that the bad situation of many small blocks of size $o(\delta)$ can actually not occur.

Lemma 9. *Let $\mathcal{T}$ be a pendant tree of a simple graph G. Let AB be an edge in $\mathcal{T}$ with $A, B \in \mathcal{V}_2$. If $|A| + |B| > 2$, $|A| + |B| \geq \delta(G) - 1$.*

Proof. Let $p := |A|$ and $q := |B|$, and let $A'A, BB'$ be edges in $\mathcal{T}$ with $A' \neq B$ and $B' \neq A$. By Lemma 3, we have $\sum_{v \in A \cup B} d(v, C_{A'A}) \leq c(A'A) \leq \max_{v \in A} d(v) - 1$ and $\sum_{v \in A \cup B} d(v, C_{B'B}) \leq \max_{v \in B} d(v) - 1$. For $v \in A \cup B$, there are at most $p + q - 1$ edges that are incident to v and $A \cup B$, which implies $d(v, C_{A'A}) + d(v, C_{B'B}) \geq d(v) - (p + q - 1)$ (see Figure 1). Therefore,

$$\begin{aligned}
&\max_{v \in A} d(v) + \max_{v \in B} d(v) - 2 \\
\geq &\sum_{v \in A \cup B} (d(v, C_{A'A}) + d(v, C_{B'B})) \\
\geq &\sum_{v \in A \cup B} (d(v) - (p + q - 1)) \\
\geq &\max_{v \in A} d(v) + \max_{v \in B} d(v) + (p + q - 2)\delta - (p + q)(p + q - 1),
\end{aligned}$$

which gives $(p + q)(p + q - 1) \geq (p + q - 2)\delta + 2$ and thus

$$(p + q)(p + q - 2) \geq (p + q - 2)(\delta - 1).$$

Hence, $p + q \geq \delta - 1$ if $p + q > 2$. $\square$

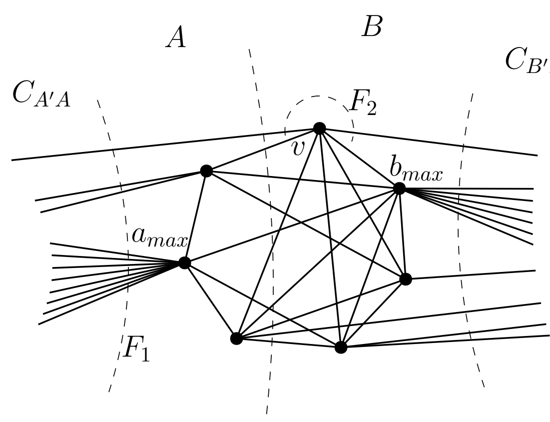

Figure 1: A graph G with $\delta = 6$ and adjacent blocks $A, B \in \mathcal{V}_2$ of sizes 3 and 4. Here, $d(a_{max}) = d(b_{max}) = 12$, $|F_1| := |c(AA')| = 11 \leq d(a_{max}) - 1$ and $|F_2| := d(v, C_{A'A}) + d(v, C_{B'B}) = 2 \geq d(v) - (|A| + |B| - 1)$.

Lemma 10. *Let $\mathcal{T}$ be a pendant tree of a simple graph G with $|V(\mathcal{T})| > 1$. Let $A = \{v_0\}$ be a block in $\mathcal{V}_r$ with neighborhood $B_1, \ldots, B_r \in \mathcal{V}_2$ in $\mathcal{T}$ such that $|A| = |B_1| = \cdots = |B_r| = 1$. Let $B_i' \neq A$ be the block that is adjacent to B_i in*

$\mathcal{T}$. *Then* $d(v_0) \le r^2 - 2\gamma$, *where* $\gamma := \sum_{1\le i<j\le r} d(C_{B'_iB_i}, C_{B'_jB_j})$ *is the number of* cross-edges. *In particular, we have* $\delta(G) \le r^2$ *and* $\lambda(G) < r^2$. *Moreover, if* $r = 2$, $\kappa(G) \le 2$.

Proof. Note that $r \ge 2$, since there is no singleton leaf block by Corollary 8. For every $1 \le i \le r$, let $B_i = \{v_i\}$ and $C_i := C_{B'_iB_i}$. Since every v_i can have at most r neighbors in $\{v_0, v_1, \ldots, v_r\}$, we have $d(v_i) \le r + \sum_{j=1}^{r} d(v_i, C_j)$ for every $i \in \{0, 1, \ldots r\}$. On the other hand, by Lemma 3, we have, for every $i \in \{1, 2, \ldots r\}$, $\sum_{j=0}^{r} d(v_j, C_i) + \sum_{j\neq i} d(C_j, C_i) = d(C_i) \le d(v_i) - 1$ (see Figure 2). Therefore,

$$\begin{aligned}
&\sum_{i=1}^{r}\left(d(v_i) + \sum_{j=0}^{r} d(v_j, C_i) + \sum_{j\neq i} d(C_j, C_i)\right) \le \sum_{i=1}^{r}\left(r + \sum_{j=1}^{r} d(v_i, C_j) + d(v_i) - 1\right)\\
\iff &\sum_{i=1}^{r}\left(d(v_0, C_i) + \sum_{j\neq i} d(C_j, C_i)\right) \le r^2 - r\\
\iff &\sum_{i=1}^{r} d(v_0, C_i) \le r^2 - r - 2\gamma,
\end{aligned}$$

and hence,

$$d(v_0) \le r + \sum_{i=1}^{r} d(v_0, C_i) \le r^2 - 2\gamma.$$

In particular, this gives $\delta \le r^2$ and, according to Lemma 3, $\lambda \le c(AB_1) < d(v_0) \le r^2$.

Now, we claim that, if $r = 2$, then $\kappa \le 2$. If $\gamma > 0$, then $\kappa \le \delta \le d(v_0) \le r^2 - 2\gamma \le 2$. If $\gamma = 0$, we have $d(v_0) \le 4$. Let $S := \{v_0, v_1, v_2\}$, which is a separator of G of size 3. If a vertex $z \in S$ has no neighbor in C_i for some $i \in \{1, 2\}$, $S - z$ is a separator of size 2, which gives the claim. Otherwise, we have $c(AB_i) \ge 3$ for every $i \in \{1, 2\}$. Since $d(v_0) \le 4$ and $c(AB_i) < d(v_0)$ (Lemma 3), we must have $c(AB_i) = 3$ for every $i \in \{1, 2\}$. Hence, v_0 is of degree 2 in G, which gives the claim. □

Setting $r = 2$ in Lemma 10 gives the following corollary for adjacent blocks of straight paths.

Corollary 11. *Let G be simple and let AB and BC be edges in a straight path of $\mathcal{T}$. If $\delta(G) \ge 5$ or $\lambda(G) \ge 4$ or $\kappa(G) \ge 3$, then $|A| + |B| + |C| > 3$.*

Let $\mathcal{V}_2^{in}$ denote the set of blocks A such that A is in $\mathcal{V}_2$ and its two neighbors are also in $\mathcal{V}_2$, and let $\mathcal{V}_2^{out}$ denote $\mathcal{V}_2 - \mathcal{V}_2^{in}$.

The following lemma holds for general trees.

Lemma 12. *Let $\mathcal{T}$ be a tree. If $|V(\mathcal{T})| > 1$, then $|\mathcal{V}_{>2}| \le |\mathcal{V}_1| - 2$ and $|\mathcal{V}_2^{out}| \le 4|\mathcal{V}_1| - 6$.*

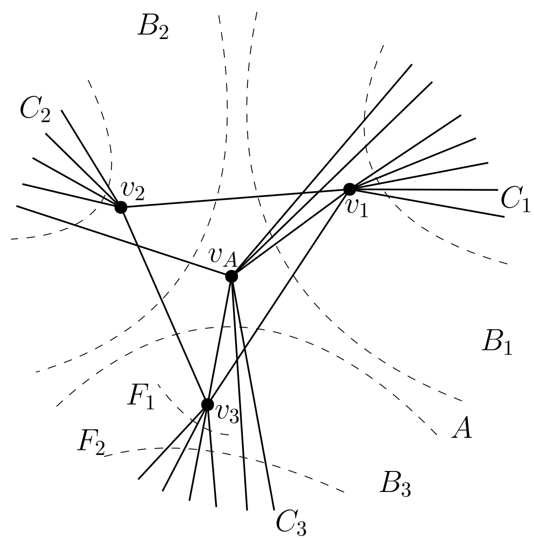

Figure 2: A graph with $\delta = 6$. Here, $r = 3$, $|F_1| := \sum_{i=1}^{3} d(v_3, C_i) = 4 \leq d(v_3) - r$ and $|F_2| := c(B_3 C_3) = 6 \leq d(v_3) - 1$.

Proof. As $\mathcal{T}$ is a tree, $\sum_{A \in V(\mathcal{T})} d_{\mathcal{T}}(A) = 2|E(\mathcal{T})| = 2(|V(\mathcal{T})| - 1)$, which yields $-2 = \sum_{A \in V(\mathcal{T})}(d_{\mathcal{T}}(A) - 2) = \sum_{A \in \mathcal{V}_1}(d_{\mathcal{T}}(A) - 2) + \sum_{A \in \mathcal{V}_2}(d_{\mathcal{T}}(A) - 2) + \sum_{A \in \mathcal{V}_{>2}}(d_{\mathcal{T}}(A) - 2) \geq -|\mathcal{V}_1| + 0 + |\mathcal{V}_{>2}|$, i.e. $|\mathcal{V}_{>2}| \leq |\mathcal{V}_1| - 2$. Since every straight path contains at most two blocks in $\mathcal{V}_2^{out}$ and contracting every straight path along with one of its neighbors gives a tree $\mathcal{T}'$ with $V(\mathcal{T}') = \mathcal{V}_1 \cup \mathcal{V}_{>2}$, we have $|\mathcal{V}_2^{out}| \leq 2E(\mathcal{T}') = 2(|\mathcal{V}_1| + |\mathcal{V}_{>2}| - 1)$. Thus, $|\mathcal{V}_2^{out}| \leq 4|\mathcal{V}_1| - 6$. □

Now we are ready to show that the blocks of straight paths contain many vertices if $\delta(G) \geq 5$ or $\lambda(G) \geq 4$ or $\kappa(G) \geq 3$.

Lemma 13. *Let $\mathcal{T}$ be a pendant tree of a simple graph G satisfying $\delta(G) \geq 5$ or $\lambda(G) \geq 4$ or $\kappa(G) \geq 3$. Let $\mathcal{P}$ be a straight path of $\mathcal{T}$. Then*

$$\sum_{S \in V(\mathcal{P})} |S| \geq (|V(\mathcal{P})| - 2)\frac{\max\{4, \delta(G)\}}{3} + 2.$$

Proof. For any two consecutive edges AB and BC in $\mathcal{P}$, applying Corollary 11 gives $|A| + |B| + |C| > 3$. Due to Lemma 9, this implies $|A| + |B| + |C| \geq \max\{4, \delta\}$. Since there may be at most two singletons that are not contained in such a triple, we conclude $\sum_{S \in V(\mathcal{P})} |S| \geq (|V(\mathcal{P})| - 2)\frac{\max\{4,\delta\}}{3} + 2$. □

3.3 Many Pendant Pairs

We will use the results on large blocks of the previous section to obtain our main Theorem 14, which shows the existence of many pendant pairs.

Theorem 14. *Let G be a simple graph that satisfies $\delta \geq 5$ or $\lambda \geq 4$ or $\kappa \geq 3$. Then G contains at least $\frac{\delta n}{24}$ pendant pairs.*

Proof. Note that $n > \delta \geq 3$. If G does not contain a non-pendant pair, there are $\binom{n}{2} \geq \frac{\delta n}{24}$ pendant pairs in G. Otherwise, G contains a non-pendant pair. Let $\mathcal{T}$ be a pendant tree of G; then $|V(T)| \geq 2$. For every straight path $\mathcal{P}$ with $|V(\mathcal{P})| \geq 3$, let $\mathcal{P}^*$ be a subpath obtained from $\mathcal{P}$ by deleting at most two endvertices of $\mathcal{P}$ (i.e. nodes in $\mathcal{P} \cap \mathcal{V}_2^{out}$) such that $|V(\mathcal{P}^*)|$ is a multiple of 3. Then, we split $\mathcal{P}^*$ into subpaths $\mathcal{P}_1^*, \ldots, \mathcal{P}_{|V(\mathcal{P}^*)|/3}^*$, each of size 3. By Corollary 11 and Lemma 9, $\sum_{S \in V(\mathcal{P}_i^*)} |S| \geq \max\{4, \delta\}$ for every $i = 1, \ldots, |V(\mathcal{P}^*)|/3$.

Let $\mathcal{V}_2^* := \mathcal{V}_2 - \bigcup_{\text{straight path } \mathcal{P}, |V(\mathcal{P})| \geq 3} V(\mathcal{P}^*) \subseteq \mathcal{V}_2^{out}$. For every leaf node $S \in \mathcal{V}_1$, let Y_S be a collection of nodes that consists of S, at most four nodes from $\mathcal{V}_2^*$ and at most one node from $\mathcal{V}_{>2}$ such that the collections Y_S $(S \in \mathcal{V}_1)$ form a partition of $\mathcal{V}_1 \cup \mathcal{V}_2^* \cup \mathcal{V}_{>2}$; such allocation exists, as $|\mathcal{V}_2^*| \leq |\mathcal{V}_2^{out}| \leq 4|\mathcal{V}_1|$ and $|\mathcal{V}_{>2}| \leq |\mathcal{V}_1|$ due to Lemma 12. For every $S \in \mathcal{V}_1$, let D_S be a node in Y_S of maximum size. Then $|D_S| \geq |S| > \delta$ by Lemma 6. Thus, the number of pendant pairs in G is at least

$$
\begin{aligned}
&\sum_{S \in V(\mathcal{T})} \binom{|S|}{2} \\
\geq &\sum_{S \in \mathcal{V}_1} \frac{|D_S|(|D_S| - 1)}{2} + \sum_{\text{straight path } \mathcal{P},\ |V(\mathcal{P})| \geq 3} \ \sum_{S \in V(\mathcal{P}^*)} \frac{|S|(|S| - 1)}{2} \\
\geq &\frac{\delta}{2} \sum_{S \in \mathcal{V}_1} |D_S| + \frac{1}{2} \sum_{\text{straight path } \mathcal{P},\ |V(\mathcal{P})| \geq 3} \ \sum_{i=1}^{|V(\mathcal{P}^*)|/3} \ \sum_{S \in V(\mathcal{P}_i^*)} |S|(|S| - 1) \\
\geq &\frac{\delta}{2} \sum_{S \in \mathcal{V}_1} |D_S| + \frac{1}{2} \sum_{\text{straight path } \mathcal{P},\ |V(\mathcal{P})| \geq 3} \ \sum_{i=1}^{|V(\mathcal{P}^*)|/3} 3 \left(\frac{\sum_{S \in V(\mathcal{P}_i^*)} |S|}{3} \left(\frac{\sum_{S \in V(\mathcal{P}_i^*)} |S|}{3} - 1 \right) \right) \\
\geq &\frac{\delta}{2} \sum_{S \in \mathcal{V}_1} |D_S| + \frac{\delta}{24} \sum_{\text{straight path } \mathcal{P},\ |V(\mathcal{P})| \geq 3} \ \sum_{S \in V(\mathcal{P}^*)} |S| \\
\geq &\frac{\delta}{12} \sum_{S \in \mathcal{V}_1 \cup \mathcal{V}_2^* \cup \mathcal{V}_{>2}} |S| + \frac{\delta}{24} \sum_{S \in \mathcal{V}_2 - \mathcal{V}_2^*} |S| \\
\geq &\frac{\delta n}{24}.
\end{aligned}
$$

The third inequality follows from Jensen's inequality, as the function $f(x) := x(x-1)$ is convex for $x \geq 1$. The fourth inequality follows from $\frac{1}{3}\left(\sum_{S \in V(\mathcal{P}_i^*)} |S|\right) - 1 \geq \frac{\delta}{12}$ for every $i = 1, \ldots, |V(\mathcal{P}^*)|/3$, which holds since $\delta \geq 3$ and $\sum_{S \in V(\mathcal{P}_i^*)} |S| \geq \max\{4, \delta\}$ for every $i = 1, \ldots, |V(\mathcal{P}^*)|/3$. $\square$

While our main theorem shows the existence of $\Omega(\delta n)$ pendant pairs, we will in addition need the existence of many pendant pairs of a special type: consider the

$\binom{n}{2}$ pendant pairs of the complete graph K_n. We cannot contract these one-by-one and expect a sparsification result, as there are multiple cyclic dependencies among the pendant pairs such as the three pendant pairs that consist only of the vertices of a cycle of length three. In order to argue about contraction-based sparsification (Theorem 18), we therefore define the following variant of pendant pairs without such cyclic dependencies. Compared to our main theorem, lower bounds on the number of these pairs are necessarily weaker (e.g. K_n has at most $n-1$ such pairs); our lower bounds will however be best possible up to constant factors.

Definition 15. A set F of pendant pairs is *dependent* if V contains at least three distinct vertices $v_1, \dots, v_k$ such that $\{v_i, v_{i+1}\} \in F$ for all $i = 1, \dots, k$, where $v_{k+1} := v_1$; otherwise, F is *independent*.

We now identify an independent set of pendant pairs whose contraction implies not only an additive decrease of the number of vertices by at least $\frac{5}{17}n$, but also a multiplicative decrease by the factor δ (so that the number of vertices left is $O(n/\delta)$).

Lemma 16. *Let G be a simple graph that satisfies $\delta \geq 5$ or $\lambda \geq 4$ or $\kappa \geq 3$. Then G has an independent set of at least $\frac{\delta}{\delta+12}n = \Omega(n)$ pendant pairs whose pairwise contraction leaves $O(n/\delta)$ vertices in the graph.*

Proof. Note that $n > \delta \geq 3$ and, for this reason, $\frac{\delta}{\delta+12}n \geq \frac{1}{5}n = \Omega(n)$. First assume that G does not contain a non-pendant pair. For an arbitrary spanning tree of G, consider the pair of endvertices of every edge of it. These are $n-1 \geq \frac{\delta}{\delta+12}n$ pendant pairs which constitute an independent set and whose pairwise contraction leaves only $1 = O(n/\delta)$ vertex.

Now assume that G contains a non-pendant pair. Let $\mathcal{T}$ be a pendant tree of G, we have $|V(\mathcal{T})| \geq 2$. Using the previous results, we can relate n with the number of blocks in $\mathcal{T}$ by considering the blocks in $\mathcal{V}_2$ separately as follows.

$$
\begin{aligned}
n &= \sum_{S \in V(\mathcal{T})} |S| \\
&= |V(\mathcal{T})| + \sum_{S \in \mathcal{V}_1 \cup \mathcal{V}_{>2}} (|S|-1) + \sum_{\text{straight path } \mathcal{P}} \left(\sum_{S \in V(\mathcal{P})} |S| - |V(\mathcal{P})| \right) \\
&\geq |V(\mathcal{T})| + |\mathcal{V}_1|\delta + \sum_{\text{straight path } \mathcal{P},\ |V(\mathcal{P})| \geq 3} (|V(\mathcal{P})|-2) \left(\frac{\max\{4,\delta\}}{3} - 1 \right) \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{(by Lemmas 6 and 13)} \\
&\geq |V(\mathcal{T})| + |\mathcal{V}_1|\delta + \frac{1}{12}|\mathcal{V}_2^{in}|\delta \ \ \text{(as } \delta = 3 \Rightarrow (\tfrac{4}{3}-1) > \tfrac{\delta}{12} \text{ and } \delta \geq 4 \Leftrightarrow \tfrac{\delta}{3} - 1 \geq \tfrac{\delta}{12}\text{)} \\
&\geq |V(\mathcal{T})| + \frac{1}{6}(|\mathcal{V}_1| + |\mathcal{V}_2^{out}| + |\mathcal{V}_{>2}|)\delta + \frac{1}{12}|\mathcal{V}_2^{in}|\delta \\
&\qquad\qquad\qquad\qquad \text{(since } 6|\mathcal{V}_1| \geq |\mathcal{V}_1| + |\mathcal{V}_2^{out}| + |\mathcal{V}_{>2}| \text{ by Lemma 12)} \\
&\geq \left(1 + \frac{1}{12}\delta\right) |V(\mathcal{T})|.
\end{aligned}
$$

Now let F be any forest on the vertex set $V(G)$ such that, for every $S \in V(\mathcal{T})$, S is the vertex set of a component of F. Then $\{\{u,v\} : uv \in E(F)\}$ is an independent set of pendant pairs. Therefore, G has an independent set of at least $|E(F)| = n - |V(\mathcal{T})| \geq (1 - \frac{12}{\delta+12})n = \frac{\delta}{\delta+12}n > \frac{n}{5}$ pendant pairs. Furthermore, contracting every edge in F leaves at most $|V(\mathcal{T})| \leq (1 + \frac{1}{12}\delta)^{-1}n = O(n/\delta)$ vertices, which gives the second claim. □

We remark that the constants 1/24 and 1/12 that appear in the proofs of Theorem 14 and Lemma 16 can both be improved for larger δ than 5.

3.4 Tightness

We call a bound *tight* if it is optimal up to a constant factor. The unions of $\frac{n}{\delta+1}$ many disjoint cliques $K_{\delta+1}$ show that the number of pendant pairs in Theorem 14 and the number of vertices left after contraction in Lemma 16 are tight. Clearly, an independent set of pendant pairs is of size at most $n-1$, hence also the lower bound on the pendant pairs in Lemma 16 is tight.

Each of the conditions $\delta \geq 5$, $\lambda \geq 4$ and $\kappa \geq 3$ in Theorem 14 and Lemma 16 is tight, as the graph in Figure 3 can be arbitrarily large and satisfies $\delta = 4$, $\lambda = 3$ and $\kappa = 2$, but has only a constant number of pendant pairs. Also the simpleness condition in both results is indispensable: Consider the path graph on n vertices whose two end edges have multiplicity δ such that all other edges have multiplicity $\delta/2$. This graph has precisely two pendant pairs, each at one of its ends.

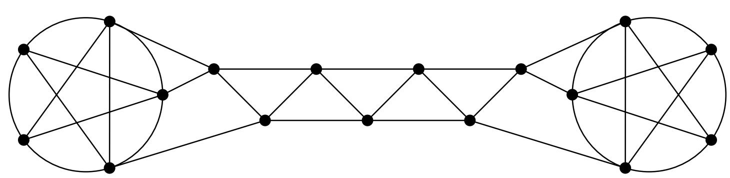

Figure 3: The *bone graph* G, whose only pendant pairs are the ones contained in the two K_5 (those form the only leaf blocks of the pendant pair tree). Hence, G has exactly 20 pendant pairs.

4 Contraction-Based Sparsification

In the recent algorithm of Kawarabayashi and Thorup [10], a crucial sparsification step is to contract vertex subsets of G such that $O((n \log^c n)/\delta)$ vertices and $O(n \log^c n)$ edges remain for some constant c and all non-trivial min-cuts are preserved. We will show the existence of two such contraction-based sparsifications by considering two cut tree called *δ-edge-connectedness tree*, which respects the vertex pairs $\{v,w\}$ with $\lambda_G(v,w) < \delta$, and *non-trivial min-cut tree*, which respects the vertex pairs that are separated by some non-trivial min-cut.

By inheriting the argument using large leaf- and $\mathcal{V}_2$-blocks for pendant trees to these new cut trees, we will prove that the contraction of every block in these trees leaves only $O(n/\delta)$ vertices and $O(n)$ edges. In this way, all cuts of size less than δ and all non-trivial min-cuts will be preserved, respectively.

4.1 The δ-Edge-Connectedness Tree

By definition, every pendant pair of a graph G is $\delta(G)$-edge-connected. Hence, most of the results about pendant pairs can be transferred directly to statements about δ-edge-connected pairs. In particular, Lemma 6 gives the following corollary.

Corollary 17. *Every simple graph G contains a set S of at least $\delta+1$ vertices such that $\lambda(v,w) \geq \delta$ for every $v, w \in S$.*

More generally, Theorem 14 and Lemma 16 still hold without further ado when we replace the binary pendant pair relation with the δ-edge-connectedness relation on vertex pairs. We now show how these arguments give the first sparsification result described above. To this end, we will use the following relaxation of both Gomory-Hu and pendant trees. A *k-edge-connectedness tree* (also known as *partial Gomory-Hu tree* [1]) is a cut tree that respects $\{\{v,w\} : \lambda_G(v,w) < k\}$.

A k-edge-connectedness tree $\mathcal{T}$ exists for every graph, as we can contract all edges that induce cuts of size larger than k in a Gomory-Hu tree. Moreover, $\mathcal{T}$ can be computed similarly as in the approach that was used in Proposition 4, and hence in deterministic time $O(n\theta_{flow})$. For randomized algorithms, [1] showed that $\mathcal{T}$ can be constructed in expected running time $\widetilde{O}(m + nk^2)$.

In this section, we focus on the case that $k = \delta$, i.e. on the δ-edge-connectedness tree $\mathcal{T}$. By Property (i) of Definition 1, every block of $\mathcal{T}$ is δ-edge-connected and therefore a subset of a δ-edge-connected component of G. By Property (ii) and (iii), no δ-edge-connected component intersects two (not necessarily adjacent) blocks A and B of $\mathcal{T}$, as A and B are separated by a cut of size less than δ. Hence, the blocks of every $\delta(G)$-edge-connectedness tree are precisely the δ-edge-connected components of G.

4.1.1 Contractions Preserving Small Cuts

Now we relate $\mathcal{T}$ to any pendant tree $\mathcal{T}'$ of G. Since $\mathcal{T}'$ is pendant, every block of $\mathcal{T}'$ is δ-edge-connected and therefore a subset of some block of $\mathcal{T}$. Hence, the vertex partition of every pendant tree *refines* the partition of V into δ-edge-connected components. It is also not hard to see that, given a δ-edge-connectedness tree $\mathcal{T}$, there is a pendant tree $\mathcal{T}'$, such that contracting all edges $e \in E(\mathcal{T}')$ with $c(e) \geq \delta$ gives $\mathcal{T}$.

Theorem 18. *Contracting every δ-edge-connected component of a simple graph G satisfying $\delta > 0$ leaves $O(n/\delta)$ vertices and $O(n)$ edges.*

Proof. If $1 \le \delta \le 4$, then $\delta n \le 4n$ and hence there are trivially at most $n \le 4n/\delta = O(n/\delta)$ many vertices left after the contractions. If $\delta \ge 5$, consider any pendant tree $\mathcal{T}'$ of G and contract every of its blocks. Since the partition of any pendant tree refines the partition of V into the δ-edge-connected components of G, Lemma 16 implies that these contractions leave only $O(n/\delta)$ vertices.

Now let $\mathcal{T}$ be a δ-edge-connectedness tree of G and consider the edges that are left after the contractions. Every remaining edge is contained in some edge-cut of G that is induced by an edge of $\mathcal{T}$. Since $\mathcal{T}$ is a δ-edge-connectedness tree, every such edge-cut has size at most $\delta - 1$. By the result above, $\mathcal{T}$ has $O(n/\delta)$ blocks. Hence, there are at most $O(n/\delta) \cdot (\delta - 1) = O(n)$ edges left. $\square$

Note that the graph after contraction may have multiedges. The following is a fundamental corollary of Theorem 18. Despite its generality, it appears to be unknown so far.

Corollary 19. *Every simple graph G with $\delta > 0$ has $O(n/\delta)$ many δ-edge-connected components.*

It was recently shown [12] that every simple graph G that satisfies $0 < \lambda(G) < \delta(G)$ has $O((n/\delta)^2)$ min-cuts. We strengthen this to cuts of size *not much larger* than $\lambda(G)$ as follows.

Theorem 20. *Every simple graph G that satisfies $0 < \lambda < \delta$ has $O((n/\delta)^2)$ cuts of size less than $\min\{\frac{3}{2}\lambda, \delta\}$.*

Proof. Henzinger and Williamson [6] proved that in any connected graph H the number of cuts of size less than $\frac{3}{2}\lambda(H)$ is at most $O(|V(H)|^2)$. Let G' be the graph obtained from G by contracting every δ-edge-connected component. This preserves every cut of size less than δ. As contractions do not decrease the edge-connectivity of any vertex pair, G' has precisely the same cuts of size less than δ as G. Thus, we can count the number of these cuts in G' instead of in G. Since $\lambda(G') = \lambda(G) > 0$, $\delta(G') > 0$ and G' is connected. Applying Theorem 18 to G and [6] to $H := G'$ therefore shows that G has $O((n/\delta)^2)$ cuts of size less than $\min\{\frac{3}{2}\lambda, \delta\}$. $\square$

Theorem 21. *Given any real number $\alpha \ge 1$, every simple graph G that satisfies $0 < \lambda < \delta$ has $O((n/\delta)^{\lfloor 2\alpha \rfloor})$ cuts of size at most $\min\{\alpha\lambda, \delta - 1\}$.*

Proof. Karger [7] proved that in any connected graph H the number of cuts of size at most $\alpha \cdot \lambda(H)$ is in $O(|V(H)|^{\lfloor 2\alpha \rfloor})$. Again, let G' be the graph obtained from G by contracting every δ-edge-connected component. Applying Theorem 18 to G and [7] to $H := G'$ shows that G has $O((n/\delta)^{\lfloor 2\alpha \rfloor})$ cuts of size at most $\min\{\alpha\lambda, \delta - 1\}$. $\square$

The same approach can also be used to strengthen various other upper bounds known on the number of small cuts.

4.2 The Non-Trivial Min-Cut Tree

Although the $\delta(G)$-edge-connectedness tree preserves all (not necessarily minimum) cuts of size less than δ, it does not preserve cuts of size δ. However, one could not expect to preserve all cuts of size δ by contracting vertex subsets leaving say $O(n/\delta)$ vertices, as the complete graph $K_{\delta+1}$ shows. Hence, we will preserve only non-trivial min-cuts. To this end we consider a new cut tree, which can be used to obtain an upper bound on the number of non-trivial min-cuts.

A *non-trivial min-cut tree* $\mathcal{T}$ is a cut tree that respects the vertex pairs separated by some non-trivial min-cut, and satisfies the following additional property:

(iv) for every $AB \in E(\mathcal{T})$, C_{AB} is a non-trivial min-cut

It is clear that Property (iv) implies Properties (ii) and (iii). Property (i) implies that all non-trivial min-cuts will be preserved if every block is contracted. Property (iv) is equivalent to saying that no leaf block is a singleton and $c(AB) = \lambda$ for all $AB \in E(\mathcal{T})$.

Unlike pendant trees, non-trivial min-cut trees do not exist for every graph. To see this, consider any cycle of length at least four. As every edge is contained in a non-trivial min-cut, every leaf block A of a non-trivial min-cut tree $\mathcal{T}$ is an independent set of size at least two in G. Then the tree edge $AB \in E(\mathcal{T})$ satisfies $c(AB) \geq 4$, which contradicts Property (iv). We conclude that not every graph with $\lambda(G) = 2$ has a non-trivial min-cut tree. However, we will show that non-trivial min-cut trees exist for all simple graphs G with $\lambda(G) \neq 0, 2$.

4.2.1 Construct Non-Trivial Min-Cut Tree from Cactus Representation

We call a multigraph $\mathcal{K}$ a *cactus* if it is 2-edge-connected, contains no loops, and each edge in $\mathcal{K}$ belongs to exactly one cycle (which may be of length 2, i.e. a pair of parallel edges). This is equivalent to saying that all maximal 2-connected subgraphs of $\mathcal{K}$ are cycles. Note that an edge min-cut in $\mathcal{K}$ is exactly two edges from a cycle in $\mathcal{K}$. Let C be a cycle in $\mathcal{K}$ and v be a vertex in C. We denote by $\mathcal{K}[C, v]$ the component containing v of the graph that is obtained from $\mathcal{K}$ by deleting the two edges incident to v in C. We denote by $\mathcal{C}(G)$ the set of all min-cuts of G and by $\mathcal{NC}(G)$ that of all non-trivial min-cuts of G.

A *cactus representation* $(\mathcal{K}, \varphi)$ of G consists of a cactus $\mathcal{K}$ and a mapping φ from $V(G)$ to $V(\mathcal{K})$ such that (a) for every min-cut X in G, there is a min-cut Y in $\mathcal{K}$ with $X = \varphi^{-1}(Y)$ and (b) for every min-cut Y in $\mathcal{K}$, $\varphi^{-1}(Y)$ is a min-cut in G. A vertex v in $\mathcal{K}$ is *empty* if $\varphi^{-1}(v)$ is empty, a *singleton* if $\varphi^{-1}(v)$ consists of exactly one vertex of G, and a *k-junction* if v is contained in exactly k cycles of $\mathcal{K}$. A cactus representation $(\mathcal{K}, \varphi)$ of G is *minimal* if one cannot get another cactus representation by contracting an edge of $\mathcal{K}$ and revising the mapping correspondingly.

It has been proven by Dinits et al. [3] that every graph G admits a cactus representation. Furthermore, Kawarabayashi and Thorup [10] show that a cactus representation can be computed in near-linear time. We first consider some lemmas which help us to show that a non-trivial min-cut tree can be constructed from a cactus representation in near-linear time for simple graphs G with $\lambda(G) \neq 0, 2$.

Lemma 22. *Let $(\mathcal{K}, \varphi)$ be a cactus representation of G. If $\mathcal{K}$ contains a cycle C of length larger than two, let u and v be adjacent vertices in C, then G has exactly $\lambda/2$ edges between $\varphi^{-1}(\mathcal{K}[C,u])$ and $\varphi^{-1}(\mathcal{K}[C,v])$; in particular, λ is even.*

Proof. Let $X_1 := \varphi^{-1}(\mathcal{K}[C,u])$, $X_2 := \varphi^{-1}(\mathcal{K}[C,v])$ and $X_3 := V - X_1 - X_2$. As $(\mathcal{K}, \varphi)$ is a cactus representation, X_1, X_2 and X_3 are min-cuts in G, respectively. This implies that $d(X_1, X_2) + d(X_1, X_3) = d(X_2, X_3) + d(X_2, X_1) = d(X_3, X_1) + d(X_3, X_2) = \lambda$. Therefore, $d(X_1, X_2) = \lambda/2$. □

Proposition 23. *Let G be a simple graph with $\lambda \neq 0, 2$. Then a non-trivial min-cut tree $\mathcal{T}$ of G can be computed in time $\widetilde{O}(|E(G)|)$.*

Proof. Let $(\mathcal{K}, \varphi)$ be a minimal cactus representation of G. We are going to construct a collection $\mathcal{U} := \mathcal{U}_1 \cup \mathcal{U}_2$ of uncrossing min-cuts of $\mathcal{K}$, which will then be used to construct a non-trivial min-cut tree $\mathcal{T}$.

For every cycle $C := (\{v_1, \ldots, v_l\}, \{v_1v_2, \ldots, v_{l-1}v_l, v_lv_1\})$ of length l in $\mathcal{K}$, we proceed as follows.

If there are two distinct cactus vertices say v_1 and v_i $(1 < i \leq l)$ in C with $|\varphi^{-1}(\mathcal{K}[C, v_1])| > 1$ and $|\varphi^{-1}(\mathcal{K}[C, v_i])| > 1$. We put the following $l-1$ cuts into $\mathcal{U}_1$: $\bigcup_{k=1}^{j} \mathcal{K}[C, v_k]$ for $j = 1, \ldots, i-1$, and $\bigcup_{k=i}^{j} \mathcal{K}[C, v_k]$ for $j = i, \ldots, l-1$. Each of these $l-1$ cuts represents some non-trivial min-cut of G.

If, for every cactus vertex v in C except v_1, we have $|\varphi^{-1}(\mathcal{K}[C, v])| = 1$, we claim that $l = 2$. Otherwise, if $l > 2$, then by Lemma 22, λ must be even. By our condition $\lambda \neq 0, 2$, we have $\lambda \geq 4$. By Lemma 22, there are $\lambda/2 \geq 2$ edges between $\varphi^{-1}(\mathcal{K}[C, v_2])$ and $\varphi^{-1}(\mathcal{K}[C, v_3])$, which is not possible since $|\varphi^{-1}(\mathcal{K}[C, v_2])| = |\varphi^{-1}(\mathcal{K}[C, v_3])| = 1$ and G is simple. By the minimality of $(\mathcal{K}, \varphi)$, we know that v_2 is a 1-junction singleton in C. We put the cut $\{v_2\}$ into $\mathcal{U}_2$. Note that $\varphi^{-1}(\{v_2\})$ is a trivial min-cut of G.

After collecting cuts for every cycle in $\mathcal{K}$, we now have a collection $\mathcal{U}$ of $|V(\mathcal{K})| - 1$ uncrossing min-cuts of $\mathcal{K}$. It is known that such a collection of uncrossing cuts can be *represented* by a tree $\mathcal{T}_0$ on $V(\mathcal{K})$. More precisely, two vertices are adjacent in $\mathcal{T}_0$ if they are separated by only one cut from $\mathcal{U}$. So, the min-cuts in $\mathcal{U}$ are one-to-one corresponded to the edges of $\mathcal{T}_0$. We replace every vertex $v \in V(\mathcal{T}_0)$ by $\varphi^{-1}(v)$, so it becomes a tree whose vertex set is "almost" a partition of $V(G)$, as the vertices form a disjoint union of $V(G)$ but some of them can be empty. To obtain a non-trivial min-cut tree $\mathcal{T}$, we first contract the edges corresponding to the cuts in $\mathcal{U}_2$, and then iteratively contract edges which contain some empty endvertex. Recall that the new vertex after an edge contraction is defined to be the union of the endvertices of the edge to be contracted. It is straightforward to justify that the tree satisfies Properties (i) and (iv) since the first step has been done, and its vertex set becomes a partition of $V(G)$ after the second step. Thus we obtain a non-trivial min-cut tree $\mathcal{T}$.

We use the result of Kawarabayashi and Thorup [10] to find a cactus representation of G in near-linear time. All subsequent steps such as verifying the minimality of $(\mathcal{K}, \varphi)$, collecting cuts to form $\mathcal{U}$ and contracting tree edges in $\mathcal{T}_0$ can be done in linear time. We conclude that a non-trivial min-cut tree can be constructed in near-linear time. $\square$

4.2.2 Contractions Preserving Non-Trivial Min-Cuts

The following lemma assures that leaf blocks must have size $\Omega(\delta)$.

Lemma 24. *Every non-trivial min-cut $A \subset V(G)$ of a simple graph G satisfies $|A| \geq \delta$. In particular, every leaf block A of a non-trivial min-cut tree of G satisfies $|A| \geq \delta$.*

Proof. For the first claim, let $p := |A|$. Then $\delta \geq \lambda \geq \sum_{v \in A}(d(v) - (p-1)) \geq p\delta - p(p-1)$ implies $p \geq \delta$, as $p > 1$. The second claim follows directly from the first. $\square$

We remark that a non-trivial min-cut tree would also exist for graphs that satisfy $\lambda \in \{0, 2\}$ if we would omit Property (iv); however, for such cut trees, Lemma 24 does not hold in general.

The following analogues of Lemmas 9 and 10 will ensure that the number of vertices will decrease by a factor of $\Omega(\delta)$ when contracting all blocks of a non-trivial min-cut tree.

Lemma 25. *Let $\mathcal{T}$ be a non-trivial min-cut tree of a simple graph G. Let $A'A, AB, BB'$ be edges in $\mathcal{T}$ such that $A, B \in \mathcal{V}_2$. If $|A| + |B| > 2$, $|A| + |B| \geq \delta(G)/2$.*

Proof. Let $p := |A|$ and $q := |B|$. It is clear that $\sum_{v \in A \cup B} d(v, C_{A'A}) \leq \lambda \leq \delta$, $\sum_{v \in A \cup B} d(v, C_{B'B}) \leq \delta$ and $d(v, C_{A'A}) + d(v, C_{B'B}) \geq d(v) - (p+q-1)$. Therefore, $2\delta \geq \sum_{v \in A \cup B}(d(v, C_{A'A}) + d(v, C_{B'B})) \geq \sum_{v \in A \cup B}(d(v) - (p+q-1)) \geq (p+q)(\delta - (p+q-1))$, which gives $p + q \geq \frac{p+q-2}{p+q-1} \cdot \delta \geq \frac{1}{2} \cdot \delta$ if we assume $p + q > 2$. $\square$

Lemma 26. *Let $\mathcal{T}$ be a non-trivial min-cut tree of a simple graph G. Let A be a block in $\mathcal{V}_r$ with neighborhood $B_1, \ldots, B_r \in \mathcal{V}_2$ in $\mathcal{T}$ such that $|A| = |B_1| = \cdots = |B_r| = 1$. Then $\delta(G) \leq r^2 + r$.*

Proof. Let $A := \{v_0\}$. For every $i \in \{1, \ldots, r\}$, let $B_i := \{v_i\}$ and $B'_i \neq A$ be the block that is adjacent to B_i in $\mathcal{T}$. Let $C_i := C_{B'_i B_i}$. Since every v_i $(i \in \{0, 1, \ldots, r\})$ can have at most r neighbors in $\{v_0, v_1, \ldots, v_r\}$, we have, for every $i \in \{0, 1, \ldots, r\}$, $d(v_i) \leq r + \sum_{j=1}^{r} d(v_i, C_j)$. On the other hand, we have, for every $i \in \{1, \ldots, r\}$, $\sum_{j=0}^{r} d(v_j, C_i) \leq \lambda \leq \delta$. Therefore, $\sum_{i=1}^{r}(\delta + \sum_{j=0}^{r} d(v_j, C_i)) \leq \sum_{i=1}^{r}(d(v_i) + \delta) \leq \sum_{i=1}^{r}(r + \sum_{j=1}^{r} d(v_i, C_j) + \delta)$, which implies $r^2 \geq \sum_{i=1}^{r} d(v_0, C_i) \geq d(v_0) - r \geq \delta - r$. $\square$

Now we present an alternative proof of the sparsification result in [12].

Theorem 27. *Let G be a simple graph with $\delta > 0$ and let $\mathcal{T}$ be a non-trivial min-cut tree of G. Then contracting every block of $\mathcal{T}$ leaves $O(n/\delta)$ vertices and $O(n)$ edges.*

Proof. We can assume $\delta \geq 7$, as otherwise $\delta n \leq 6n$, which implies that there are at most $n \leq 6n/\delta = O(n/\delta)$ vertices left after the contractions. We can also assume that $|V(\mathcal{T})| > 1$, as otherwise the contraction leaves precisely $1 = O(n/\delta)$ vertex; in particular, we have $\mathcal{V}_0 = \emptyset$. Since $\delta \geq 7$, Lemma 26 implies that there are no distinct blocks $B_1, B_2, B_3 \in \mathcal{V}_2$ satisfying $B_1B_2, B_2B_3 \in E(\mathcal{T})$. We conclude by Lemma 25 that, for every straight path $\mathcal{P}$, $\sum_{S \in \mathcal{V}_2^{in} \cap V(\mathcal{P})} |S| = |\mathcal{V}_2^{in} \cap V(\mathcal{P})| \cdot \Omega(\delta)$. Now the number of vertices can be bounded as follows.

$$\begin{aligned}
n &= \sum_{S \in V(\mathcal{T})} |S| \\
&\geq \sum_{S \in \mathcal{V}_1 \cup \mathcal{V}_{>2}} |S| + \sum_{\text{straight path } \mathcal{P}} \left(\sum_{S \in \mathcal{V}_2^{in} \cap V(\mathcal{P})} |S| \right) \\
&\geq |\mathcal{V}_1| \cdot \Omega(\delta) + \sum_{\text{straight path } \mathcal{P}} \Big(|\mathcal{V}_2^{in} \cap V(\mathcal{P})| \cdot \Omega(\delta) \Big) && \text{(by Lemma 24)} \\
&= (|\mathcal{V}_1| + |\mathcal{V}_2^{out}| + |\mathcal{V}_{>2}|) \cdot \Omega(\delta) + |\mathcal{V}_2^{in}| \cdot \Omega(\delta) && \text{(by Lemma 12)} \\
&= |V(\mathcal{T})| \cdot \Omega(\delta).
\end{aligned}$$

Therefore, $|V(\mathcal{T})| = O(n/\delta)$ vertices and at most $(|V(\mathcal{T})| - 1) \cdot \lambda = O(n\lambda/\delta) \leq O(n)$ edges will be left if all blocks of $\mathcal{T}$ are contracted. $\square$

4.3 Tightness

We show that the above results are tight, except for the cases in which this was already shown. The following graph shows that the bounds of Corollaries 17 and 19 and Theorems 18 and 27 (vertex- and edge-bound) are tight. Let $n \geq 3(\delta + 1)$, $\delta \geq 2$ and assume that n is a multiple of $\delta + 1$ (the last assumption can be avoided by a simple modification of the construction). Then the graph G obtained from the cycle on $n/(\delta + 1)$ vertices by replacing all vertices with a copy of $K_{\delta+1}$ shows tightness.

Although this fixes $\lambda(G) = 2$, this example can be readily generalized to tight graphs having larger and even λ such that $\lambda < \delta/2$. To do so, obtain a graph G' from G by adding $\lambda/2 - 1$ cycles that are vertex-disjoint from the first initial cycle C, but visit exactly the same complete subgraphs in the same order as C.

References

[1] A. Bhalgat, R. Hariharan, T. Kavitha, and D. Panigrahi. An $\tilde{O}(mn)$ Gomory-Hu tree construction algorithm for unweighted graphs. In *Proceedings of the 39th Annual Symposium on Theory of Computing (STOC'07)*, pages 605–614, 2007.

[2] E. A. Dinic. Algorithm for Solution of a Problem of Maximum Flow in a Network with Power Estimation. *Soviet Math Doklady*, 11:1277–1280, 1970.

[3] E. A. Dinits, A. V. Karzanov, and M. V. Lomonosov. On the structure of a family of minimal weighted cuts in a graph. In A. A. Fridman, editor, *Studies in Discrete Optimization (in Russian)*, pages 290–306, Nauka, Moscow, 1976.

[4] A. Frank. On the edge-connectivity algorithm of Nagamochi and Ibaraki. Laboratoire Artemis, IMAG, Université J. Fourier, Grenoble, March 1994.

[5] M. Henzinger, S. Rao, and D. Wang. Local flow partitioning for faster edge connectivity. In *Proceedings of the 28th Annual Symposium on Discrete Algorithms (SODA'17)*, pages 1919–1938, 2017.

[6] M. Henzinger and D. P. Williamson. On the number of small cuts in a graph. *Information Processing Letters*, 59:41–44, 1996.

[7] D. R. Karger. Minimum cuts in near-linear time. *Journal of the ACM*, 47(1):46–76, 2000.

[8] A. V. Karzanov. On finding a maximum flow in a network with special structure and some applications. *Matematicheskie Voprosy Upravleniya Proizvodstvom (in Russian)*, pages 81–94, 1973.

[9] T. Kathuria, Y. P. Liu, and A. Sidford. Unit capacity maxflow in almost $O(m^{4/3})$ time. In *Proceedings of the 61st Annual Symposium on Foundations of Computer Science (FOCS'20)*, 2020.

[10] K. Kawarabayashi and M. Thorup. Deterministic edge connectivity in near-linear time. *Journal of the ACM*, 66(1):4:1–4:50, 2018.

[11] Y. T. Lee and A. Sidford. Path-finding methods for linear programming: Solving linear programs in $\tilde{O}(\sqrt{\text{rank}})$ iterations and faster algorithms for maximum flow. In *Proceedings of the 55th Annual Symposium on Foundations of Computer Science (FOCS'14)*, 2014.

[12] O.-H. S. Lo, J. M. Schmidt, and M. Thorup. Compact cactus representations of all non-trivial min-cuts. *Discrete Applied Mathematics*, to appear.

[13] W. Mader. Existenz gewisser Konfigurationen in n-gesättigten Graphen und in Graphen genügend großer Kantendichte. *Mathematische Annalen*, 194:295–312, 1971.

[14] W. Mader. Grad und lokaler Zusammenhang in endlichen Graphen. *Mathematische Annalen*, 205:9–11, 1973.

[15] W. Mader. Kantendisjunkte Wege in Graphen. *Monatshefte für Mathematik*, 78(5):395–404, 1974.

[16] W. Mader. On vertices of degree n in minimally n-connected graphs and digraphs. *Bolyai Society Mathematical Studies (Combinatorics, Paul Erdős is Eighty, Keszthely, 1993)*, 2:423–449, 1996.

[17] H. Nagamochi and T. Ibaraki. Computing edge-connectivity in multigraphs and capacitated graphs. *SIAM Journal on Discrete Mathematics*, 5(1):54–66, 1992.

[18] M. Stoer and F. Wagner. A simple min-cut algorithm. *Journal of the ACM*, 44(4):585–591, 1997.

[19] R. E. Tarjan and M. Yannakakis. Simple linear-time algorithms to test chordality of graphs, test acyclicity of hypergraphs, and selectively reduce acyclic hypergraphs. *SIAM Journal on Computing*, 13(3):566–579, 1984.