Lo, On-Hei Solomon; Schmidt, Jens M.:

A cut tree representation for pendant pairs

Original published in: 29th International Symposium on Algorithms and Computation / ISAAC 29, 2018 Jiaoxi, Yilan, Taiwan. - Saarbrücken/Wadern, Germany : Schloss Dagstuhl - Leibniz-Zentrum für Informatik GmbH, Dagstuhl Publishing, December, 2018. - (2018), art. 38, 9 pp.
ISBN 978-3-95977-094-1
(Leibniz International Proceedings in Informatics (LIPIcs) ; 123)

Original published: 2018

ISSN: 1868-8969

DOI: 10.4230/LIPIcs.ISAAC.2018.38

[*Visited:* 2020-03-05]

A Cut Tree Representation for Pendant Pairs

On-Hei S. Lo[1]
Institut für Mathematik, Technische Universität Ilmenau,
Weimarer Strasse 25, D-98693 Ilmenau, Germany
solomon.lo@tu-ilmenau.de

Jens M. Schmidt[2]
Institut für Mathematik, Technische Universität Ilmenau,
Weimarer Strasse 25, D-98693 Ilmenau, Germany

Abstract

Two vertices v and w of a graph G are called a *pendant pair* if the maximal number of edge-disjoint paths in G between them is precisely $\min\{d(v), d(w)\}$, where d denotes the degree function. The importance of pendant pairs stems from the fact that they are the key ingredient in one of the simplest and most widely used algorithms for the minimum cut problem today.

Mader showed 1974 that every simple graph with minimum degree δ contains $\Omega(\delta^2)$ pendant pairs; this is the best bound known so far. We improve this result by showing that every simple graph G with minimum degree $\delta \geq 5$ or with edge-connectivity $\lambda \geq 4$ or with vertex-connectivity $\kappa \geq 3$ contains in fact $\Omega(\delta|V|)$ pendant pairs. We prove that this bound is tight from several perspectives, and that $\Omega(\delta|V|)$ pendant pairs can be computed efficiently, namely in linear time when a Gomory-Hu tree is given. Our method utilizes a new cut tree representation of graphs.

2012 ACM Subject Classification Mathematics of computing → Graph theory, Mathematics of computing → Graph algorithms

Keywords and phrases Pendant Pairs, Pendant Tree, Maximal Adjacency Ordering, Maximum Cardinality Search, Testing Edge-Connectivity, Gomory-Hu Tree

Digital Object Identifier 10.4230/LIPIcs.ISAAC.2018.38

1 Introduction

The study of pendant pairs is motivated by the well-known, simple and widely used minimum cut algorithm of Nagamochi and Ibaraki [11], which refines the work of Mader [8, 7] in the early 70s, and was simplified by Stoer and Wagner [12] and Frank [3]. The key approach of this algorithm is to iteratively contract a pendant pair of the input graph in near-linear time by using *maximal adjacency orderings* (also known as *maximum cardinality search* [13]). Having done that $n-1$ times, one can obtain a minimum cut by just considering the minimum degree of all intermediate graphs. In a break-through result, Kawarabayashi and Thorup [6] succeeded to give a near-linear time deterministic minimum cut algorithm for simple graphs, and this was later made faster by Henzinger et al. [4]. Hence, the algorithm of Nagamochi and Ibaraki is not the most efficient, but its simplicity is unmatched so far.

This motivates the following question: How many (distinct) pendant pairs does a graph with a given minimum degree possess? If there are many and, additionally, these could be computed efficiently, this would lead immediately to an improvement of the running time of

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

the Nagamochi-Ibaraki algorithm. Here, we aim for the fundamental and natural question of finding a good lower bound on the number of distinct pendant pairs in graphs with a given minimum degree. We will mainly consider simple graphs, as these allow us to prove strong lower bounds (we give an example that shows that all bounds for multigraphs must be considerably weaker).

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 at least one pendant pair [8]. 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 $\Omega(\delta^2)$ pendant pairs [9].

Our main result in this paper sets the graph-theoretical prerequisite that the algorithmic approach described above of finding many pendant pairs might actually work out. 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 $\Omega(\delta n)$ pendant pairs; this exhibits a dependency on $n := |V|$ instead of δ, which is usually much larger. We prove that this result is tight with respect to the order of the bound and with respect to every assumption.

We show how to compute $\Omega(\delta n)$ pendant pairs from a Gomory-Hu tree in linear time. Clearly, computing a Gomory-Hu tree in advance does not match the best running time $O(m + n)$, $m := |E|$, for finding *one* pendant pair; however, we conjecture that it is actually possible to compute $\omega(1)$ pendant pairs in linear time. An affirmative answer to this would already imply a speed-up for the Nagamochi-Ibaraki-algorithm.

Our results utilize a new cut tree representation of graphs named *pendant tree*.

2 A Note on the History of Maximal Adjacency Orderings

Mader's proof for the existence of one pendant pair relies strongly on [7, Lemma 1], which in turn uses special orderings on the vertices. Interestingly, these orderings are *maximal adjacency orderings* and this fact exhibits an apparently forgotten variant of them, which existed long before they got 1984 their first name (maximum cardinality search [13]).

We are only aware of one place in literature where this is (briefly) mentioned: [10, p. 443]. Mader's existential proof can in fact be made algorithmic. A direct comparison between the old and the modern variant however shows that the modern maximal adjacency orderings are nicer to describe, as they work on the original graph, while Mader iteratively moves edges in the graph in order to represent the essential connectivity information on the already visited vertex set with a clique.

3 Preliminaries

All graphs considered in this paper are non-empty, finite, unweighted and undirected unless specified otherwise. Let $G := (V, E)$ be a graph. *Contracting* a vertex subset $X \subseteq V$ identifies all vertices in X and deletes occurring self-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 $\overline{X} := V - X$, $d_G(X, Y) := |E_G(X, Y)|$ and $d_G(X) := |E_G(X, \overline{X})|$; if $X = \{v\}$ for some vertex $v \in V$, we simply write $E_G(v, Y)$, $d_G(v, Y)$ and $d_G(v)$. A subset $\emptyset \neq X \subset V$ of a graph G is called a *cut* of G. Let a cut X of G be *trivial* if $|X| = 1$ or $|\overline{X}| = 1$. Let the *length* and *size* of a path be the number of its edges and vertices, respectively. Let $\delta(G) := \min_{v \in V} d_G(v)$ be the *minimum degree* of G. For a vertex $v \in G$, let $N_G(v)$ be the set of neighbors of v in G.

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 $\lambda_G(v, w) \geq k$. The *edge-connectivity* $\lambda(G)$ of G is the greatest integer such that every two distinct vertices are $\lambda(G)$-edge-connected. Let $\kappa(G)$ be the *vertex-connectivity* of G, i.e. the minimum number of vertices U such that $G - U$ is disconnected. We omit parentheses for single elements (like vertices or edges) in set subtractions.

We call a pair $\{v, w\}$ of vertices *pendant* if $\lambda_G(v, w) = \min\{d_G(v), d_G(w)\}$. In order to increase readability, we will omit subscripts whenever the graph is clear from the context.

4 The Pendant Tree

We propose a new cut tree, which can be seen as a refinement of Gomory-Hu trees. The idea is to partition the vertex set such that each part consists only of vertices that are pairwise pendant, and impose a tree structure on these vertex subsets such that edges in this tree correspond to cuts in the graph that separate some non-pendant pair. For the sake of notational clarity, we will call the vertices of such trees *blocks*.

For a tree T whose vertex set partitions V and an edge $AB \in E(T)$, let C_{AB} be the union of the blocks that are contained in the component of $T - AB$ containing A, and symmetrically, $C_{BA} = V - C_{AB}$. We will consider T as a tree with edge weights as follows. For an edge $AB \in E(T)$, let $c(AB) := d_G(C_{AB})$ be the *size* of its corresponding edge-cut in G.

▶ **Definition 1.** A *non-pendant-pair covering tree*, or simply *pendant tree*, T of a graph $G = (V, E)$ is a tree whose vertex set partitions V such that

- **(i)** every two distinct vertices in a common block of this partition are pendant,
- **(ii)** for every edge $AB \in E(T)$, there are vertices $a \in A$ and $b \in B$ such that $\{a, b\}$ is non-pendant, and
- **(iii)** for every edge $AB \in E(T)$, there are vertices $a^* \in A$ and $b^* \in B$ such that $c(AB) = \lambda_G(a^*, b^*)$.

Note that T is an auxiliary tree which is not obtained from G by contracting vertex subsets. The following lemma allows us to find a non-pendant pair for every two adjacent blocks of a pendant tree very efficiently.

▶ **Lemma 2.** *Let AB be an edge of a pendant tree 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 Condition (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, i.e.

$$\lambda(a, a_{max}) = \min\{d(a), d(a_{max})\} = d(a)$$

and $\lambda(b, b_{max}) = d(b)$, a minimum a-b-cut of size less than $\min\{d(a), d(b)\}$ can neither separate a from a_{max} nor b from b_{max}. Hence,

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

◀

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

▶ **Lemma 3.** *Let AB be an edge of a pendant tree 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 Condition (iii). By transitivity of the edge-connectivity λ, we have

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

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. ◀

We will construct a pendant tree by contracting edges in a Gomory-Hu tree. We recall that, given a graph G, a Gomory-Hu tree T of G is a tree on the vertex set $V(G)$, such that for every pair of vertices $a \neq b$ in G, there is an edge e in the a-b-path in T with that $E_G(V_T(C_e), \overline{V_T(C_e)})$ is a minimum a-b-cut in G, where C_e is a component obtained by deleting e in T and we denote by $V_T(C_e)$ the set of vertices in G which are in the component C_e. In particular, $\lambda_G(a, b) = d_G(V_T(C_e))$. Here we see a Gomory-Hu tree not a tree on the vertex of G, but on the partition of $V(G)$ in which every part is a singleton.

▶ **Proposition 4.** *Given a Gomory-Hu tree of a graph G, a pendant tree of G can be computed in linear time.*

Proof. Let T be a Gomory-Hu tree of G. Throughout the algorithm we maintain that every pair of distinct vertices in a block is pendant. We check iteratively for every edge AB in T, whether there is a non-pendant pair $\{a, b\}$ with $a \in A$ and $b \in B$. We contract AB in 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 is clear (see also Lemma 2), and it suffices 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 $\min\{d_G(a_{max}), d_G(b_{max})\} \leq c(AB)$. 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 property of Gomory-Hu trees, there are 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 comes from the transitivity of local edge-connectivity, the second comes from the fact that every vertex pair of a block is pendant, and the third holds, because $d_G(b^*) \geq d_G(a^*) = c(AB) \geq d_G(a)$.

It is not hard to see that the algorithm has a linear running time. ◀

In particular, Proposition 4 implies that every graph has a pendant tree as it is known that a Gomory-Hu tree always exists.

The best known running time for a deterministic construction of a Gomory-Hu tree is still based on the classical approach that applies $n-1$ times the uncrossing technique to find uncrossing cuts on the input graph, and hence in $O(n\theta_{flow})$, where θ_{flow} is the running time for a maximum flow subroutine (by Dinits' algorithm [2, 5], $\theta_{flow} = O(n^{2/3}m)$). Non-deterministically, Bhalgat et al. [1] showed that a Gomory-Hu tree of a simple unweighted graph can be constructed in expected running time $\tilde{O}(nm)$, where the tilde hides polylogarithmic factors.

Therefore, by our construction above, we conclude the following.

▶ **Corollary 5.** *Given a simple graph G, a pendant tree of G can be computed deterministically in running time $O(n^{5/3}m)$, and randomized in expected running time $\tilde{O}(nm)$.*

The next section gives several helpful lemmas that will be used in counting pendant pairs.

5 Large Blocks of Degree 1 and 2

For a tree T whose vertex set partitions V, let V_k be the set of blocks of T having degree k in T and let $V_{>k} := \bigcup_{k'>k} V_{k'}$. We call the blocks in V_1 *leaf blocks*. In T, the set V_2 induces a family of disjoint paths; we call each such path a *2-path*. We will prove that the leaf blocks of pendant trees as well as the blocks that are contained in 2-paths are large.

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

Proof. Let $p := |A| \geq 1$ and let B be the block adjacent to A in 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, which was first shown by Mader.

▶ **Corollary 7** ([9])**.** *Let 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.*

In simple graphs, we thus know that leaf blocks give us a large number of pendant pairs. Since T is a tree, the number of leaf blocks is completely determined by the number of blocks of degree at least 3, namely $|V_1| = \sum_{A\in V_{>2}}(d_T(A) - 2) + 2$. 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 2-paths. The following two lemmas prove that (i) for every two adjacent blocks A and B in a 2-path with $|A| + |B| > 2$, we have $|A| + |B| \geq \delta - 1 = \Omega(\delta)$ and (ii) if $\delta \geq 5$ and P is a subpath of a 2-path such that all blocks

of P are singletons, then P contains at most two blocks. This will be used later to show that the bad situation of many small blocks of size $o(\delta)$ can actually not occur. We omit the proofs in this extended abstract.

▶ **Lemma 9.** *Let T be a pendant tree of a simple graph G. Let AB be an edge in T with $A, B \in V_2$. If $|A| + |B| > 2$, $|A| + |B| \geq \delta(G) - 1$.*

▶ **Lemma 10.** *Let T be a pendant tree of a simple graph G with $|V(T)| > 1$. Let $A = \{v_A\}$ be a block in V_r with neighborhood $B_1, \dots, B_r \in V_2$ in 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 T. Then $d(v_A) \leq r^2 - 2\gamma$, where $\gamma := \sum_{1 \leq i < j \leq r} d(C_{B_i'B_i}, C_{B_j'B_j})$ is the number of* cross-edges*. In particular, we have $\delta(G) \leq r^2$ and $\lambda(G) < r^2$. Moreover, if $r = 2$, $\kappa(G) \leq 2$.*

Setting $r = 2$ in Lemma 10 gives the following corollary for adjacent blocks of 2-paths. Note that the proof of Lemma 10 allows to weaken the conditions of this corollary further if the number of cross-edges is large.

▶ **Corollary 11.** *Let G be simple and let AB and BC be edges in a 2-path of T. If $\delta(G) \geq 5$ or $\lambda(G) \geq 4$ or $\kappa(G) \geq 3$, then $|A| + |B| + |C| > 3$.*

For every block $A \in V_2$, let A be in V_2^{in} if all of its neighbors are also in V_2; otherwise, let A be in V_2^{out}. The blocks in V_2^{out} are exactly the endblocks of 2-paths.

▶ **Lemma 12.** *Let T be a tree. If $|V(T)| > 1$, then $|V_{>2}| \leq |V_1| - 2$ and $|V_2^{out}| \leq 4|V_1| - 6$.*

Now we are ready to show that the blocks of 2-paths contain many vertices if $\delta(G) \geq 5$ or $\lambda(G) \geq 4$ or $\kappa(G) \geq 3$.

▶ **Lemma 13.** *Let 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 P be a 2-path of T. Then*

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

We will use these lemmas to count pendant pairs in the next section.

6 Many Pendant Pairs

We will use the results on large blocks of the previous section to obtain our main theorems, Theorems 15 and 16. While the latter shows the existence of $\Omega(\delta n)$ pendant pairs, as mentioned in the introduction, the former gives the slightly weaker bound $\Omega(n)$, but in return counts only pendant pairs of a special type.

▶ **Definition 14.** Let a set F of pendant pairs be *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 we set $v_{k+1} := v_1$; otherwise, F is called *independent*.

Counting only independent pendant pairs allows us to deduce statements about the number of vertices in the graph that is obtained from contracting these pairs (these are not true for arbitrary sets of pendant pairs): Theorem 15 will prove for $\delta \geq 5$ that there are at least $\frac{\delta}{\delta+12}n \geq \frac{5}{17}n = \Omega(n)$ such independent pendant pairs. We will show that the contractions imply 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 (i.e. the number of vertices left is $O(n/\delta)$). We omit the proof in this extended abstract.

▶ **Theorem 15.** *Let G be a simple graph that satisfies $\delta(G) \geq 5$ or $\lambda(G) \geq 4$ or $\kappa(G) \geq 3$. Let T be a pendant tree of G. Then G has at least $\frac{\delta}{\delta+12}n = \Omega(n)$ independent pendant pairs each of which is in some block of T and whose pairwise contraction leaves $O(n/\delta)$ vertices in the graph.*

For arbitrary pendant pairs not requiring independence, we improve the lower bound $\Omega(n)$ of Theorem 15 to $\Omega(\delta n)$ in the following theorem. This is done by grouping the blocks more precisely. The main idea is that the blocks are of average size $\Omega(\delta)$ and therefore contain $\Omega(\delta^2)$ pendant pairs on average. As the number of blocks is $O(n/\delta)$, we thus expect that the number of pendant pairs is $\Omega(\frac{n}{\delta} \cdot \delta^2) = \Omega(\delta n)$.

▶ **Theorem 16.** *Let G be a simple graph that satisfies $\delta(G) \geq 5$ or $\lambda(G) \geq 4$ or $\kappa(G) \geq 3$. Then G contains at least $\frac{1}{30}\delta n = \Omega(\delta n)$ pendant pairs.*

Proof. Note that $n > \delta \geq 3$. If G does not contain a non-pendant pair, there are $\binom{n}{2} \geq \frac{\delta n}{30}$ pendant pairs in G. Otherwise, G contains a non-pendant pair. Let T be a pendant tree of G; then $|V(T)| \geq 2$.

For each 2-path P with $|V(P)| \geq 3$, let P^* be a subpath obtained from P by deleting at most two endblocks (i.e. blocks in $P \cap V_2^{out}$) of P such that $|V(P^*)|$ is a multiple of 3. Then, we split P^* into subpaths $P_1^*, \ldots, P_{|V(P^*)|/3}^*$, each of size 3. Now, let M_P be a collection of blocks that contains exactly one block $S_i \in V(P_i^*)$ for every $i = 1, \ldots, \frac{|V(P^*)|}{3}$, such that S_i is of maximum size amongst other blocks in $V(P_i^*)$. By Corollary 11 and Lemma 9, every block $S \in M_P$ is of size at least $\max\{2, (\delta-1)/2\}$.

Let $V_2^* := V_2 - \bigcup_{\text{2-path } P, |V(P)|\geq 3} V(P^*) \subseteq V_2^{out}$. For every leaf block $S \in V_1$, let Y_S be a collection of blocks that consists of S, at most four blocks from V_2^* and at most one block from $V_{>2}$ such that the collections Y_S ($S \in V_1$) form a partition of $V_1 \cup V_2^* \cup V_{>2}$; such allocation exists as $|V_2^*| \leq |V_2^{out}| \leq 4|V_1|$ and $|V_{>2}| \leq |V_1|$ (Lemma 12). For every $S \in V_1$, let D_S be a block in Y_S of maximum size. Then, by Lemma 6, $|D_S| \geq |S| > \delta$.

Now we can count the number of pendant pairs to obtain the desired lower bound, as the blocks have average size $\Omega(\delta)$. The number of pendant pairs in G is at least

$$
\begin{aligned}
&\sum_{S \in V(T)} \binom{|S|}{2} \\
&\geq \sum_{S \in V_1} \frac{|D_S|(|D_S|-1)}{2} + \sum_{\text{2-path } P,\ |V(P)|\geq 3} \ \sum_{S \in M_P} \frac{|S|(|S|-1)}{2} \\
&\geq \frac{\delta}{2} \sum_{S \in V_1} |D_S| + \frac{\delta}{10} \sum_{\text{2-path } P,\ |V(P)|\geq 3} \ \sum_{S \in M_P} |S| \\
&\qquad\qquad \text{(as } |D_S| > \delta,\ |S| \geq \max\{2, \tfrac{\delta-1}{2}\} \text{ and } \delta \geq 3) \\
&\geq \frac{\delta}{2} \cdot \frac{1}{6} \sum_{S \in V_1 \cup V_2^* \cup V_{>2}} |S| + \frac{\delta}{10} \cdot \frac{1}{3} \sum_{S \in V_2 - V_2^*} |S| \\
&\geq \frac{1}{30}\delta n = \Omega(\delta n).
\end{aligned}
$$

◀

We remark that the constants 1/12 and 1/30 in the proofs of the bounds of Theorems 15 and 16 can be improved for larger δ.

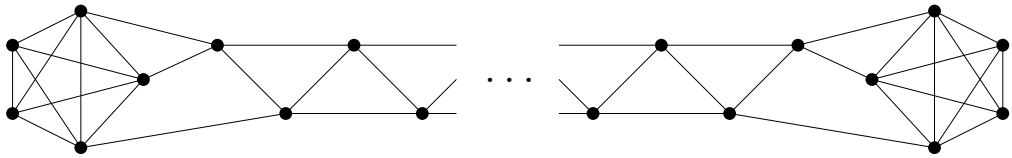

Figure 1 The *bone graph* G, whose only pendant pairs are the ones contained in the two K_5 (those form the only leaf blocks of the pendant pair tree). Hence, G has exactly 20 pendant pairs.

7 Tightness

Clearly, any graph G contains at most $n-1$ independent pendant pairs, hence the order of the lower bound in Theorem 15 is best possible. The order of the number of vertices left after contraction in Theorem 15 and that of the number of pendant pairs in Theorem 16 are also tight; consider the unions of $\frac{n}{\delta+1}$ many disjoint cliques $K_{\delta+1}$.

Each of the conditions $\delta \geq 5$, $\lambda \geq 4$ and $\kappa \geq 3$ in Theorems 15 and 16 is tight, as the graph in Figure 1 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 and all other edges have multiplicity $\delta/2$. This graph has precisely 2 pendant pairs, each at one of its ends.

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. doi:10.1145/1250790.1250879.

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

4 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. doi:10.1137/1.9781611974782.125.

5 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.

6 K. Kawarabayashi and M. Thorup. Deterministic Global Minimum Cut of a Simple Graph in Near-Linear Time. In *Proceedings of the 47th Annual Symposium on Theory of Computing (STOC'15)*, pages 665–674, 2015. doi:10.1145/2746539.2746588.

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

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

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

10 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.

11 H. Nagamochi and T. Ibaraki. Computing Edge-Connectivity in Multigraphs and Capacitated Graphs. *SIAM Journal on Discrete Mathematics*, 5(1):54–66, 1992.

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

13 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.