Department of Applied Mathematics
Faculty of EEMCS

University of Twente
The Netherlands

P.O. Box 217
7500 AE Enschede
The Netherlands
Phone: +31-53-4893400
Fax: +31-53-4893114
Email: memo@math.utwente.nl
www.math.utwente.nl/publications

Memorandum No. 1793

Spanning trees with many leaves: new extremal results and an improved FPT algorithm

P.S. Bonsma

February, 2006

ISSN 0169-2690

Spanning trees with many leaves: new extremal results and an improved FPT algorithm

Paul S. Bonsma

February 15, 2006

Abstract

We present two lower bounds for the maximum number of leaves in a spanning tree of a graph. For connected graphs without triangles, with minimum degree at least three, we show that a spanning tree with at least $(n+4)/3$ leaves exists, where n is the number of vertices of the graph. For connected graphs with minimum degree at least three, that contain D diamonds induced by vertices of degree three (a diamond is a K_4 minus one edge), we show that a spanning tree exists with at least $(2n - D + 12)/7$ leaves. The proofs use the fact that spanning trees with many leaves correspond to small connected dominating sets. Both of these bounds are best possible for their respective graph classes. For both bounds simple polynomial time algorithms are given that find spanning trees satisfying the bounds.

The second bound is used to find a new fastest FPT algorithm for the Max-Leaf Spanning Tree problem. This problem asks whether a graph G on n vertices has a spanning tree with at least k leaves. The time complexity of our algorithm is $f(k)g(n)$, where $g(n)$ is a polynomial, and $f(k) \in O(8.12^k)$.

Keywords: max-leaf, spanning tree, connected dominating set, FPT.
AMS Subject Classification: 05C35, 05C85, 05C69.

1 Introduction

For standard graph theoretic terminology we refer to [2]. A *tree* is a connected graph without cycles. A *spanning subgraph* of a graph G is a subgraph containing all vertices of G. A *spanning tree* of G is a spanning subgraph of G that is a tree. A *leaf* is a vertex with degree one. The minimum vertex degree of G is denoted by $\delta(G)$, or δ if there is no cause for confusion. See Section 2 for additional definitions.

In this paper we study spanning trees with many leaves, or equivalently, small connected dominating sets (See Section 2.1). We are interested in statements of this type: every graph on n vertices that satisfies a certain set of properties, has a spanning tree with at least $\alpha n + \beta$ leaves $(0 < \alpha < 1)$. One statement of

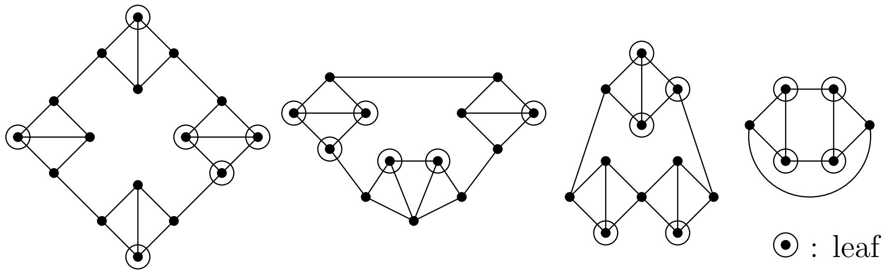

Figure 1: Theorem 1 is best possible for $n = 16, 13, 11, 6$

this type was proved independently by Linial and Sturtevant [11] and Kleitman and West [10]:

Theorem 1 (Linial and Sturtevant, Kleitman and West) *Every connected graph on n vertices with $\delta \geq 3$ has a spanning tree with at least $\lceil \frac{1}{4}n + 2 \rceil$ leaves.*

This bound is best possible for every n: for $n \bmod 4 \in \{0, 1\}$, in [10] examples are given of connected graphs with $\delta = 3$ that do not have spanning trees with more than $\lceil \frac{1}{4}n + 2 \rceil$ leaves. For $n \bmod 4 \in \{2, 3\}$, such examples can also be found: Figure 1 shows some examples for various values of n, which should make clear how to construct examples for every n. The encircled vertices show how leaves can be chosen to construct a spanning tree with maximum number of leaves. One can see that these examples contain many *diamonds*: a diamond is a K_4 minus one edge. An immediate question is whether the bound can be improved when diamonds are forbidden. One result in this direction is by Griggs, Kleitman and Shastri [8]. A *cubic graph* is a graph in which all vertices have degree three.

Theorem 2 (Griggs, Kleitman and Shastri) *Every connected cubic graph without diamonds on n vertices has a spanning tree with at least $\lceil (n + 4)/3 \rceil$ leaves.*

The bound in Theorem 2 is also best possible in some sense: of all the valid bounds of the form $\alpha n + \beta$ for this class, this bound maximizes α, and of all of those bounds that maximize α, this bound maximizes β. Equivalently, we can say that this is the best possible asymptotically sharp linear bound for this class. This is proved in [8] by first constructing examples of graphs with no more than $\lceil (n+4)/3 \rceil$ leaves for every n divisible by 6: this shows that $\alpha = \frac{1}{3}$ is best possible. The single graph Q_3 then shows that $\beta = \frac{4}{3}$ cannot be increased (Q_3 has eight vertices and no spanning tree with more than four leaves). However, for most other values of n no examples are known that show that this bound is sharp. See Section 9 for a discussion on the sharpness of this bound and the other bounds in this paper.

Our first question is what kind of bounds can be obtained when we consider graphs with $\delta \geq 3$ instead of cubic graphs, but still forbid (certain types of) diamonds. In this paper, we present one best possible bound for these graphs. In addition, even though the proof of Kleitman and West for Theorem 1 is very short and elegant, the proof of Theorem 2 uses the same techniques, but consists mainly of a very long and intricate case study. Therefore we are also interested in new techniques for proving results of this kind.

In Section 4 we prove that the bound from Theorem 2 also holds for graphs with $\delta \geq 3$ without triangles:

Theorem 3 *Every connected graph on n vertices without triangles with $\delta \geq 3$, has a spanning tree with at least $\lceil (n+4)/3 \rceil$ leaves.*

Our main result is that for graphs with $\delta \geq 3$, the following result holds. Here a *cubic diamond* is a diamond induced by vertices of degree three.

Theorem 4 *Every connected graph on n vertices with $\delta \geq 3$, that contains D cubic diamonds, has a spanning tree with at least $\lceil (2n - D + 12)/7 \rceil$ leaves.*

Theorem 4 is proved in Section 5. See also Section 5 for a slightly sharper formulation of Theorem 4. The bounds in Theorem 3 and Theorem 4 are best possible, in the same sense as explained above. This is shown in Section 6.

Apart from a small difference of $\frac{2}{7}$ in the constant, Theorem 4 can be seen as a generalization of Theorem 1: since $D \leq \frac{1}{4}n$ (diamonds induced by degree three vertices are vertex disjoint), it follows directly from Theorem 4 that connected graphs with $\delta \geq 3$ have a spanning tree with at least $\lceil (2n - D + 12)/7 \rceil \geq \lceil (\frac{7}{4}n + 12)/7 \rceil = \lceil \frac{1}{4}n + \frac{12}{7} \rceil$ leaves.

The proofs we will present use techniques that are different from previously used techniques. Theorem 3 has a relatively short proof that demonstrates these techniques, but for Theorem 4 a more elaborate proof is needed.

Our proofs are constructive, and correspond to efficient and simple algorithms: Spanning trees that satisfy the bound from Theorem 3 can be found with an algorithm that repeatedly adds vertices or pairs of vertices to a leaf set (according to some simple rules) until no more leaves can be added, and then constructs a spanning tree with those leaves. Spanning trees that satisfy the bound from Theorem 4 can be found using a similar algorithm that also involves local search on these leaf sets. These algorithms are explained in Section 7. The formulation of the algorithms is very basic, and there is a lot of room to customize them for different purposes; whether the goal is to speed them up, find better trees or tune them to specific instances.

An interesting application of this kind of results is in the area of FPT algorithms (see Section 8 for an explanation of FPT algorithms and their parameter functions): Bonsma, Brueggemann and Woeginger [3] use Theorem 1 to find an FPT algorithm with parameter function $O(9.49^k)$, for the problem of deciding whether a graph G has a spanning tree with at least k leaves. In Section 8, we show that Theorem 4 can be used to improve the parameter function of the algorithm to $O(8.12^k)$.

Finally, in Section 9, we discuss some possible ways to extend or improve the current results.

2 Definitions

In this paper, the *blocks* of a graph are maximal 2-connected subgraphs. So unlike in the usual definition, a K_2 is not a block. Therefore bridges are not contained in any block of the graph. The following lemma is an easy to prove variant of a well-known lemma.

Lemma 5 *A connected graph without leaves and with cut vertices has at least two blocks that contain exactly one cut vertex.*

2.1 Spanning trees and connected dominating sets

$S \subseteq V(G)$ is a *connected dominating set* for G if $G[S]$ is connected, and every vertex of G is part of S or adjacent to a vertex in S. The following close relation between spanning trees and connected dominating sets is well-known. If T is a spanning tree for $G = (V, E)$ with leaf set L, then $V \backslash L$ is a connected dominating set, unless $G = K_1$ or $G = K_2$: $G - L$ is connected, and every vertex in L is adjacent to a vertex not in L. Similarly, for every connected dominating set S, we can easily find a spanning tree for which $V \backslash S$ is a subset of the leaf set: choose any spanning tree of $G[S]$, and for every vertex $u \notin S$ add an edge between u and one of its neighbors in S. This leads to a connected, spanning subgraph of G without cycles. So G has a spanning tree with at least l leaves if and only if G has a connected dominating set with at most $|V| - l$ vertices.

In the remainder of the paper, we will consider connected dominating sets instead of leaf sets of spanning trees. We feel that this is more practical for our purposes, which is illustrated by our methods, and discussed in Section 9. We will call connected dominating sets *CD-sets* for short. If S is a CD-set, we will still call the vertices in $V \backslash S$ the *leaves* of S. A neighbor u of v is called a leaf neighbor or CD-set neighbor of v if u is a leaf or CD-set vertex, respectively. If S is a CD-set for G and $G[S - v]$ is not connected, v is called a *connector*. If S is a CD-set for G and $S - v$ is not dominating, v is called a *dominator*. Observe that if S and $S' \subset S$ are CD-sets for G, and $v \in S$ is a dominator or connector, then $v \in S'$. A leaf v is called an *i-leaf* if it has i CD-set neighbors. A CD-set is called a *2-CD-set* if all its leaves are 1-leaves or 2-leaves.

3 Introduction to the method

We are interested in finding bounds of this form: every (connected) graph G, with possibly some additional properties, has a CD-set S with $|S| \leq \alpha|V(G)| - \beta$, where $0 < \alpha < 1$ and β is any constant. It is clear that when G can have many vertices of degree two, no such bounds can be obtained (consider P_n), so we only consider graphs with $\delta \geq 3$.

The first question one can ask is: how good are the bounds that we can obtain using *minimal* CD-sets? The idea is to start with a CD-set containing all vertices, and removing vertices that are not connectors or dominators, until nothing can be removed anymore, and a minimal CD-set is obtained. It is easy

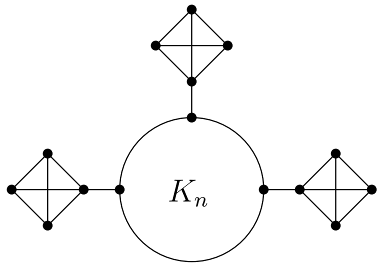

Figure 2: A graph with no small 2-CD-set

to see that with this method we cannot guarantee that a certain fraction of the vertices will become leaves: consider for instance the wheel W_n, where there is only one CD-set containing one vertex, and all other minimal CD-sets contain $n-3$ vertices.

The reason that this method fails is that for some examples, if we make a wrong initial choice, we will have a leaf with many neighbors in the CD-set, all of which have degree three. This then can then again be seen as finding a CD-set in a graph with many degree two vertices.

So the next idea is to only remove vertices from the CD-set that have few neighbors in the CD-set, for instance at most two neighbors. In that case we are looking for minimal 2-CD-sets. The first problem is that some graphs do not have any small 2-CD-sets: see for instance the graph G in Figure 2. G has a CD-set on 6 vertices, but no 2-CD-set on fewer than $|V(G)|-9$ vertices. In addition, for some other graphs (e.g. K_n), we need to remove arbitrarily many vertices to go from one 2-CD-set to the next, which makes it hard to find efficient algorithms and short proofs for results on minimal 2-CD-sets.

Fortunately, in order to prove a good bound it is not necessary that leaves have few neighbors in the CD-set, it is only necessary that they have few neighbors of degree three in the CD-set. To put it differently, we often may ignore edges between two vertices of high degree in our analysis. This leads to the following definitions of certain types of CD-sets.

Definition 6 *$S \subseteq V(G)$ is a* standard CD-set *for graph G if the following properties hold:*

1. *S is a 2-CD-set for G.*
2. *Every $v \in S$ has $d_G(v) \geq 3$.*

Definition 7 *$S \subseteq V(G)$ is a* potential standard CD-set *for G if there exists a spanning subgraph G' of G such that S is a standard CD-set for G'. Such a graph G' is called a* realization *of S. G' is a* maximal realization *of S if there is no other realization of S of which G' is a subgraph.*

We will show that the notion of minimal potential standard CD-sets is strong enough to obtain good bounds, and that potential standard CD-sets are easy to work with within an algorithmic context.

For a given potential standard CD-set S, every maximal realization has the same number of edges: if a leaf v has k neighbors in S, then $k-2$ edges between v and S need to be deleted in order to obtain a realization. In a maximal realization exactly $k-2$ of these edges are deleted. This is true for every leaf, deleted edges count towards only one leaf, and no other edges have to be deleted. So maximal realizations can alternatively be defined as having maximum number of edges.

Using minimal potential standard CD-sets, we will be able to prove the bounds from Theorem 3 and Theorem 4, expressed in terms of CD-sets instead of spanning trees. We first restrict ourselves to connected graphs without triangles with $\delta \geq 3$. In the next section we will show that in this case, for any minimal potential standard CD-set S of G and any minimal core $S' \subseteq S$, $|S'| \leq \frac{2}{3}|V(G)| - \frac{4}{3}$. By imposing two additional restrictions on the minimal potential standard CD-set S, we can prove Theorem 4 in a similar way.

4 Small CD-sets for graphs without triangles

4.1 Construction and properties

In this section we state a number of properties for potential standard CD-sets S and their realizations G'. In Section 4.2 we will prove a bound for pairs of S and G' that satisfy these properties. But first we will show that such a pair S and G' exists for every connected graph G with $\delta(G) \geq 3$.

The next theorem states the essential properties of minimal potential standard CD-sets that we need to prove our bound. Property 2 is added to allow an easier formulation of Properties 4 and 5, though it is not necessary for the method. Similarly, Property 3 is only added to ensure G' is a maximal realization, which simplifies the later proofs. We remark that for every minimal potential standard CD-set S, a realization G' can be found such that G' and S satisfy the properties from Theorem 8. However, from now on we are not concerned with minimality, only with the properties stated in Theorem 8. Note that for this theorem, we do not yet need to exclude triangles.

Theorem 8 *A connected graph G with $\delta(G) \geq 3$ has a spanning subgraph G' and a CD-set S with the following properties. Degrees, neighbors and all other graph related notions are taken with respect to G'.*

1. *S is a standard CD-set (for G').*
2. *If $u, v \in S$, $d(u) \geq d(v) \geq 4$ and $uv \in E(G')$, then uv is a bridge of $G'[S]$.*
3. *Edges in $E(G) \backslash E(G')$ are between vertices in S, or between a vertex in S and a 2-leaf.*
4. *Every vertex $v \in S$ that is neither a dominator nor a connector has at least three neighbors in S, and all of its neighbors in S have degree three.*

5. *If $\{u, v\} \subseteq S$, $uv \in E(G')$ and u and v are both neither dominators nor connectors, then $G'[S] - u - v$ is not connected.*

Proof: We consider a CD-set S and spanning subgraph G' such that S is a standard CD-set for G', that are optimal according to the following priorities:

- Minimize $|S|$.
- Minimize $|E(G'[S])|$.
- Maximize $|E(G')|$.

By this we mean that among all pairs of S and G' that minimize $|S|$, we consider pairs that minimize $|E(G'[S])|$. Among all these pairs, we choose one of those that maximizes $|E(G')|$. Note also that V is a standard CD-set for G, so there is at least one such pair S and G'. We show that for S and G' chosen this way, the above properties hold: if not, we can find an improved S and G' according to the priorities.

Property 2: If $u, v \in S$, $d(u) \geq d(v) \geq 4$ and $uv \in E(G')$, then uv is a bridge of $G'[S]$.
Proof: Consider $u, v \in S$ with $d(u) \geq 4$, $d(v) \geq 4$ and $uv \in E(G')$. If uv is not a bridge in $G'[S]$ then S is a standard CD-set for $G' - uv$. Since this decreases $|E(G'[S])|$, this change is an improvement.

Property 3: Edges in $E(G) \backslash E(G')$ are between vertices in S, or between a vertex in S and a 2-leaf.
Proof: Suppose a 1-leaf u is incident with an edge $uv \in E(G) \backslash E(G')$. If $v \in S$, then adding edge uv to G' preserves the standard CD-set; u becomes a 2-leaf. If there is an edge $uv \in E(G) \backslash E(G')$ with $u, v \notin S$, then adding uv clearly also preserves the standard CD-set. In both cases, the values from the first two priorities are not changed, and $|E(G')|$ increases, which is an improvement. We conclude that all edges in $E(G) \backslash E(G')$ are incident with at least one vertex in S and none are incident with 1-leaves.

Property 4: Every vertex $v \in S$ that is neither a dominator nor a connector has at least three neighbors in S, and all of its neighbors in S have degree three.
Proof: Let v be a vertex in S that is not a dominator or connector. If v has at most two neighbors in S, then $S - v$ is again a CD-set, for which v is a 2-leaf, so this is an improvement. Suppose v has a neighbor $u \in S$ with $d(u) \geq 4$. Since v is not a connector, uv is not a bridge of $G'[S]$, so by Property 2, $d(v) = 3$. Then $S - v$ is a standard CD-set for $G' - uv$, an improvement.

Property 5: If $\{u, v\} \subseteq S$, $uv \in E(G')$ and u and v are both neither dominators nor connectors, then $G'[S] - u - v$ is not connected.
Proof: Suppose u and v in S are neighbors which are both not connectors or dominators. By Property 4, u and v have at least three neighbors in S, and all of those neighbors have degree three. It follows that u and v both have degree

three, and therefore they have no neighbors outside of S. So $S - u - v$ is again a dominating set, and u and v are 2-leaves with respect to $S - u - v$ and G'. It follows that $S - u - v$ is an improved standard CD-set for G', unless $S - u - v$ is not a connected set. □

Now that we have proved the existence of a CD-set with these properties, we can use them to establish upper bounds on the size of the CD-set, related to the properties of the input graph.

4.2 An upper bound for the size of the constructed CD-set

Our basic approach behind the deduction of a good upper bound for the size of a CD-set S is based on the following idea. Let S be a potential standard CD-set for G and G' a realization, and suppose they satisfy the properties from Theorem 8. For showing that $|S|$ is roughly bounded from above by $\frac{2}{3}|V(G)|$, it is sufficient to show that $|S| \leq 2|\overline{S}|$. This is easy when $G'[S]$ is a tree: vertices with degree one resp. two in $G'[S]$ have at least two resp. one neighbors in $\overline{S}$, since these vertices have degree at least three in G'. A tree has average degree less than two, so there are at least $|S|$ edges in $[S, \overline{S}]$. Combining this with the fact that vertices in $\overline{S}$ are incident with at most two of these edges, we obtain $|S| \leq 2|\overline{S}|$. This is the basic idea behind the proof of our bound, which will be refined later, also for the case when $G'[S]$ is not a tree.

Observe that when $G'[S]$ is not a tree, there may be vertices in S that are not connectors or dominators, even when S is a *minimal* potential standard CD-set. In this case we can find a minimal CD-set $S' \subset S$. It is for this set S' that we prove the bound. To improve the clarity of the exposition, we first state the theorem which yields this bound, and then prove the two lemmas that are used in its proof.

Theorem 9 *Let $G = (V, E)$ be a connected graph with $\delta(G) \geq 3$, and let G' and S be a spanning subgraph and CD-set for G that satisfy the properties from Theorem 8, such that in addition no block of $G'[S]$ contains a triangle. If $S' \subseteq S$ is a minimal CD-set, then $|S'| \leq (2|V| - 4)/3$.*

Proof: In order to prove this theorem, we assign weights w to the vertices of S such that the total weight $w(S)$ is $|V \backslash S| + \frac{3}{2}|S \backslash S'|$. We show that $w(S) \geq \frac{1}{2}|S| + 2$. Then we can combine these equations to obtain the result.

The weight assignment is as follows: all leaves of S distribute a weight of 1 equally among their neighbors in S' (vertices in $S \backslash S'$ receive no weight yet). This means we have assigned a total weight of $|V \backslash S|$. Next, we assign a weight of $\frac{3}{2}$ to each vertex in $S \backslash S'$.

In order to show that $w(S) \geq \frac{1}{2}|S| + 2$, we prove three lower bounds for this weight assignment w.

1. A vertex v with degree two in $G'[S]$ is a dominator or connector (Property 4 from Theorem 8), so $v \in S'$. Since v has at least one leaf neighbor

$u \notin S$ (Property 1), and u has at most two neighbors in $G'[S']$, u assigns at least $\frac{1}{2}$ to v. This shows that for a degree two vertex v, $w(v) \geq \frac{1}{2}$.

2. A vertex v with degree one in $G'[S]$ is a dominator, since it cannot be a connector, and it must be a connector or dominator (Property 4). From the 1-leaf adjacent to it, it receives a weight of 1. In addition, since v has degree at least three and leaves have at most two CD-set neighbors, it gains an additional weight of at least $\frac{1}{2}$. So for a degree one vertex v, $w(v) \geq \frac{3}{2}$.

3. Let $G'[B]$ be a block of $G'[S]$, and let L denote the set of vertices in B with two neighbors in B, and $H = B \backslash L$ the set of vertices in B with at least three neighbors in B. Then $|L| \geq |H| + 1$ (this is shown below in Lemma 10). Let $C \subset S$ denote the set of connectors, which are the cut vertices of $G'[S]$.

   A vertex in L that is not a connector also has degree two in $G'[S]$, and therefore must be a dominator (Property 4). A dominator in B has weight at least 1. We consider two cases:

   (a) Suppose $|(S \backslash S') \cap B| \geq 1$. Then one of the vertices in H has an additional weight of at least $\frac{3}{2}$. We have $w(B \backslash C) \geq \frac{3}{2} + |L \backslash C| \geq \frac{3}{2} + |L| - |C \cap B|$. Since $|L| \geq |H| + 1$, $|L| \geq \frac{1}{2}|B| + \frac{1}{2}$. Hence $w(B \backslash C) \geq 2 + \frac{1}{2}|B| - |C \cap B|$.

   (b) Suppose $(S \backslash S') \cap B = \emptyset$. Suppose B contains a vertex v that is not a dominator or connector with respect to S. Since $v \in S'$, it is a connector or dominator with respect to S'. But $(S \backslash S') \cap B = \emptyset$, so v is also not a connector for S'. Therefore, in S, v is adjacent to a 2-leaf u, and the other CD-set neighbor of u is in $S \backslash S'$. In this case, v receives a weight of 1 from u. Recall that dominators also receive a weight of at least 1. So all vertices in $B \backslash C$ receive a weight of at least 1. It follows that $w(B \backslash C) \geq |B \backslash C| = |B| - |C \cap B|$. Since $G'[B]$ is not a triangle, $|B| \geq 4$. Hence $w(B \backslash C) \geq \frac{1}{2}|B| + 2 - |C \cap B|$.

   In both cases we have $w(B \backslash C) \geq \frac{1}{2}|B| + 2 - |C \cap B|$.

If $G'[S] \neq K_1$, Lemma 11 below shows that $w(S) \geq \frac{1}{2}|S| + 2$. In that case we have

$$\frac{3}{2}|S \backslash S'| + |V \backslash S| = w(S) \geq \frac{1}{2}|S| + 2 \Leftrightarrow$$

$$\frac{3}{2}|S| - \frac{3}{2}|S'| + |V| - |S| \geq \frac{1}{2}|S| + 2 \Leftrightarrow$$

$$|V| - 2 \geq \frac{3}{2}|S'| \Leftrightarrow |S'| \leq (2|V| - 4)/3.$$

If $G'[S] = K_1$, then this final inequality follows easily since $|V| \geq 4$ (G' is a spanning subgraph of a graph with $\delta \geq 3$). □

We continue to prove the two lemmas that were used in the above proof. We adopt the notation from the proof. In particular, for a block $G'[B]$ of $G'[S]$, the set L is the set of vertices in B that have two neighbors in B, and $H = B \backslash L$ is the set of vertices with at least three neighbors in B. The first lemma gives a lower bound for the number of L-vertices in a block of $G'[S]$.

Lemma 10 *Let CD-set S and graph G' have the properties stated in Theorem 8. For every block $G'[B]$ in $G'[S]$, $|L| \geq |H|$.*

If $G'[B]$ contains no triangles, then $|L| \geq |H| + 1$.

Proof: Let $G'' = G'[B]$.

Consider two neighbors $u, v \in H \cap B$. If $d_{G'}(u) \geq 4$, then $d_{G'}(v) = 3 = d_{G''}(v)$ since uv is not a bridge of $G'[S]$ (Property 2 from Theorem 8). Then v is not a dominator or connector since all its neighbors are in B. This is a contradiction with Property 4. So both u and v have degree three in G', and therefore are not dominators or connectors. So by Property 5, they form a 2-cut in $G'[S]$, and therefore also in G''.

We show that it follows that every vertex in H has at most one neighbor in H. Suppose $u \in H$ has two neighbors v and w in H. $G'' - u - v$ is disconnected, and u and v both have exactly one neighbor in both components. Therefore if u and v share a neighbor, this must be a vertex in L since G'' is 2-connected. So w is not a neighbor of v. Now it can be checked that $G'' - u - w$ is still connected, a contradiction with the statement from the previous paragraph.

So every vertex in H has at most one neighbor in H, and therefore at least two neighbors in L. Clearly, every vertex in L has at most two neighbors in H. It follows that $|L| \geq |H|$.

In addition, if $|L| = |H|$, then every vertex in H has exactly one neighbor in H, and every vertex in L has two neighbors in H. We show that it follows that G'' contains a triangle. Consider two neighbors $u, v \in H$. Let C_1 and C_2 be the two components of $G'' - u - v$, such that $|V(C_1)| \leq |V(C_2)|$. If C_1 contains vertices of H, then let u' and v' be two neighbors in $H \cap V(C_1)$. Let C_1' and C_2' be the components of $G'' - u' - v'$, such that $\{u, v\} \subseteq V(C_2')$. Now $V(C_2) \subset V(C_2')$, so $|V(C_1')| < |V(C_1)|$. We can continue to find new neighbor pairs in H like this, until the smallest component contains no vertices of H anymore. Since every vertex in L has two neighbors in H, this component consists of a single vertex, and we have found a triangle. □

In the proof of Theorem 9 we started with a certain weight assignment and showed that in $G'[S]$: a degree two vertex has weight at least $\frac{1}{2}$, a degree one vertex has weight at least $\frac{3}{2}$, and for a block B and cut vertex set C (with respect to $G'[S]$), $w(B \backslash C) \geq \frac{1}{2}|B| + 2 - |C \cap B|$. We anticipated in the proof of Theorem 9 that this yields that the total weight of S is at least $\frac{1}{2}|S| + 2$, so slightly more than $\frac{1}{2}$ on average.

The proof of the following lemma shows our counting method: leaves of $G'[S]$ contain excess weight, and blocks of $G'[S]$ have a weight deficit. The proof shows how weights are moved inwards along the branches, and how the

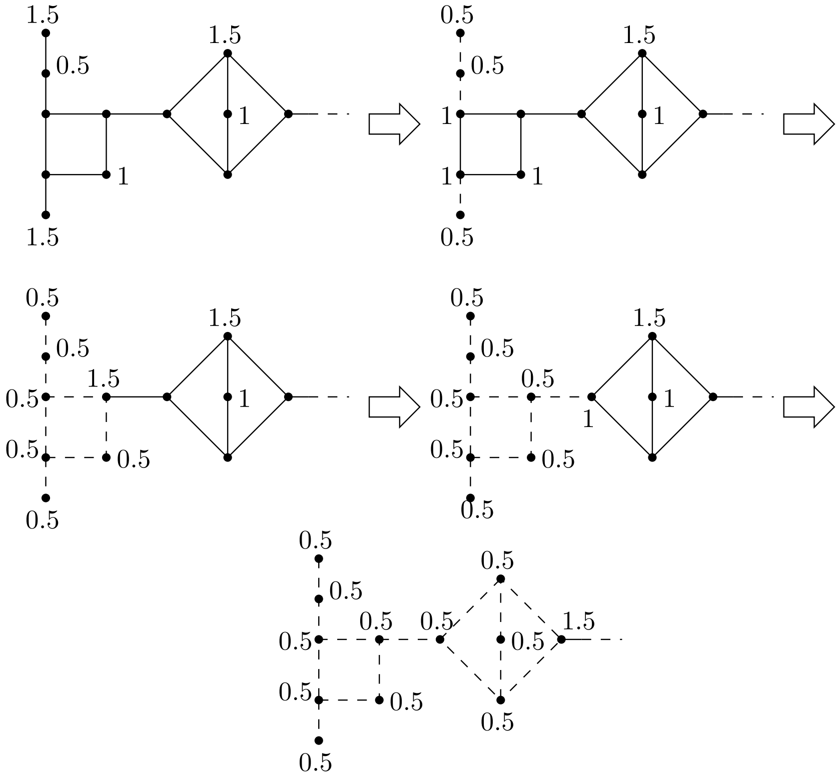

Figure 3: Moving the weights in $G'[S]$

weight deficit for blocks (given by the $|C \cap B|$ term) becomes an excess when all but one of the branches surrounding a block are dealt with this way. See Figure 3 for an example illustrating this reassignment of weights.

Lemma 11 *Let $G = (V, E) \neq K_1$ be a connected graph with non-negative weights w on the vertices such that:*

1. *If $d(v) = 2$ then $w(v) \geq \frac{1}{2}$.*
2. *If $d(v) = 1$ then then $w(v) \geq \frac{3}{2}$.*
3. *If $G[B]$ is a block of G, and C is the set of cut vertices of G, then $w(B \backslash C) \geq \frac{1}{2}|B| + 2 - |C \cap B|$.*

For this graph, $w(V) \geq \frac{1}{2}|V| + 2$.

Proof: We prove the statement by induction on $|V|$. If G is 2-connected, then the statement follows immediately from Property 3. If $G = K_2$ the statement is clearly true.

In any other case, G has at least one leaf u such that $G - u$ is not K_1, or it has at least one block with exactly one cut vertex (Lemma 5). We consider these two cases.

Suppose first that u is a leaf of G, with neighbor v. Consider the graph $G' = G - u$ with weight function $w'(v) = w(v) + w(u) - \frac{1}{2}$, and $w'(x) = w(x)$ for all other vertices. We prove that for G' and w', the three properties hold again, so we can use induction. Note that since $w(u) \geq \frac{3}{2}$, $w'(v) \geq w(v) + 1$.

If v has degree two in G' then since $w'(v) \geq 1$, the first property holds for v. If v has degree one in G', then $w(v) \geq \frac{1}{2}$ (since v has degree two in G), so $w'(v) \geq \frac{3}{2}$, and the second property holds for v. For all other vertices, the weights and degrees do not change so these two properties hold for all vertices.

For a block $G'[B]$, it is obvious that Property 3 still holds, unless $v \in B$ and v is not a cut vertex anymore in G'. In this case, let C_G resp. $C_{G'}$ denote the set of cut vertices of G and G' ($C_G = C_{G'} + v$).

$$w'(B \backslash C_{G'}) \geq w(B \backslash C_G) + 1 \geq \frac{1}{2}|B| - |C_G \cap B| + 3 = \frac{1}{2}|B| - |C_{G'} \cap B| + 2,$$

so Property 3 holds.

We conclude that if G has a leaf u, then we can construct a new graph $G' = G - u$ with weight function w' such that $w'(V(G')) = w(V(G)) - \frac{1}{2}$, for which the three properties hold. $G' \neq K_1$ and G' is connected, so by induction, $w(V(G)) = w'(V(G') + \frac{1}{2} = \frac{1}{2}|V(G')| + \frac{5}{2} = \frac{1}{2}|V(G)| + 2$, which proves the lemma for this case.

Now we consider the case that G has no leaves. Then G has a block B which contains only a single cut vertex u. Consider the graph $G' = G - (B - u)$ with weight function $w'(u) = \frac{3}{2}$ and $w'(x) = w(x)$ for all other vertices. $G' \neq K_1$, so if we can show that the three properties hold for G' and w', we can use induction.

If $d_{G'}(u) = 1$ or $d_{G'}(u) = 2$, then w' clearly satisfies the corresponding properties. For a block $G'[B']$, Property 3 clearly holds when $u \notin B'$ or u is still a cut vertex in G'. In the other case, let C_G resp. $C_{G'}$ denote the set of cut vertices of G and G'. Now

$$w'(B' \backslash C_{G'}) = w(B' \backslash C_G) + \frac{3}{2} \geq \frac{1}{2}|B'| - |C_G \cap B'| + \frac{7}{2} > \frac{1}{2}|B'| - |C_{G'} \cap B'| + \frac{5}{2},$$

which is even better than necessary.

Now for G' all weight properties are satisfied and we can use induction: let $V = V(G)$ and $V' = V(G')$.

$$w(V) \geq w'(V') + w(B - u) - \frac{3}{2} \geq (\frac{1}{2}|V'| + 2) + (\frac{1}{2}|B| + 1) - \frac{3}{2} =$$

$$\frac{1}{2}|V'| + \frac{1}{2}|B| + \frac{3}{2} = \frac{1}{2}(|V| - |B| + 1) + \frac{1}{2}|B| + \frac{3}{2} = \frac{1}{2}|V| + 2.$$

This proves the lemma for the case where G has no leaves, so the lemma is true in all cases. □

In Theorem 9, there is an additional condition that no block of $G'[S]$ contains triangles. Unfortunately, such a G' and S do not always exist for every connected graph G with $\delta(G) \geq 3$. Consider graphs with triangles consisting of three cut vertices: every standard CD-set and actually even every CD-set contains triangles. In fact, we can even construct planar, 3-connected, cubic graphs that do not have a CD-set without triangles:

Theorem 12 *Deciding whether a given graph has a CD-set without triangles is $\mathcal{NP}$-complete, even when restricted to planar, cubic, 3-connected graphs. The same is true for deciding whether a given graph has a potential standard CD-set without triangles.*

Proof: The problem of deciding whether a given graph has a spanning subgraph that is a path is called the Hamiltonian Path problem. This problem is $\mathcal{NP}$-complete even when restricted to planar, cubic, 3-connected graphs [7]. Let $G = (V, E)$ be such a Hamiltonian Path instance. Replacing every vertex with a triangle in the straightforward way gives a graph G', which is again planar, cubic and 3-connected. Suppose S is a CD-set without triangles for G'. Define $F \subset E$ as follows: if the edge corresponding to $uv \in E$ is part of $G'[S]$, add uv to F. Since S contains at most two vertices of every triangle in G', $\Delta((V, F)) \leq 2$. Suppose a vertex v with $d(v) = 0$ exists in (V, F). Then, because S is a dominating set, every neighbor of v has degree one in (V, F). So (V, F) has maximum degree two and at least three vertices with degree one, and therefore has multiple non-trivial components, contradicting the fact that $G'[S]$ is connected. Therefore (V, F) has minimum degree one and maximum degree two and is connected, which gives a Hamiltonian path for G. Similarly, a Hamiltonian path for G can be used to find a CD-set without triangles for G'. This shows that the first decision problem of Theorem 12 is $\mathcal{NP}$-complete.

In the graph constructed above, a CD-set without triangles exists if and only if a potential standard CD-set without triangles exists, so the second problem is also $\mathcal{NP}$-complete. □

So we cannot hope to improve our CD-set construction method to always find CD-sets that do not contain triangles. However, for graphs without triangles Theorem 8 and Theorem 9 immediately lead to the following positive statement:

Theorem 13 *A connected, triangle-free graph on n vertices with $\delta \geq 3$ has a CD-set S with $|S| \leq (2n - 4)/3$.*

This statement is equivalent with stating that a connected, triangle-free graph on n vertices with $\delta \geq 3$ has a spanning tree with at least $(n + 4)/3$ leaves (see Section 2.1), so we have proved Theorem 3. We have proved this by considering minimal CD-sets with a number of properties, and showing that for any such CD-set the bound holds. This differs from the techniques that were used previously to prove this kind of results.

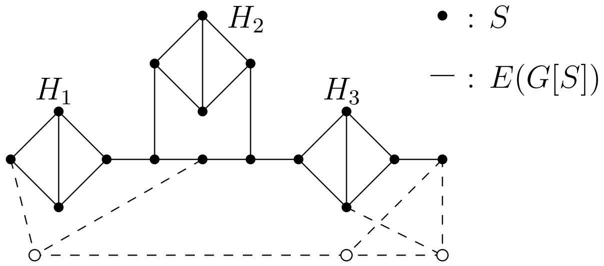

Figure 4: $G[S]$ contains one cubic diamond block

For graphs with triangles, we use the CD-set construction method presented in the next section. The idea is to find a CD-set in which blocks and therefore blocks containing triangles are sparse. This leads to an upper bound for the CD-set size, and corresponding lower bound for the number of leaves of a spanning tree, that is better than the bound from Theorem 1 (except in some very special cases given by the worst case examples for that bound), but worse than the bound from Theorem 13.

5 Small CD-sets for graphs without cubic diamonds

5.1 Construction and properties

The bound given in Theorem 1 is best possible for its class (connected graphs with $\delta \geq 3$). So if we want to improve the bound, we have to restrict the graph class. Fortunately, only a small restriction is needed.

Definition 14 *A subgraph H of G is a* cubic diamond *if H is a diamond that is induced by four vertices of degree three in G.*
If u and v are the two vertices in H that are not adjacent, it is called a cubic diamond between u *and* v.
Let $S \subseteq V(G)$. Subgraph H of $G[S]$ is a cubic diamond block *of $G[S]$ if H is a cubic diamond with respect to G and a block with respect to $G[S]$. (Vertices of H may have degree two in $G[S]$.)*

Figure 4 shows an illustration of cubic diamond blocks: H_1 is a cubic diamond block of $G[S]$ that contains one vertex with degree two in $G[S]$, but H_2 and H_3 are not cubic diamond blocks.

All of the examples showing that Theorem 1 is best possible contain many cubic diamonds. We will show that for graphs without cubic diamonds, the bound can be improved. Actually our main result is stronger: we give a new bound that depends on the number of cubic diamonds. The new bound is considerably better if this number is small, and it almost coincides with the

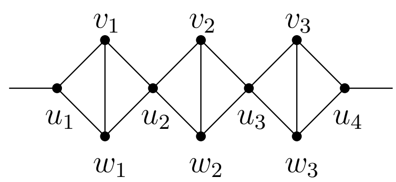

Figure 5: A diamond necklace

bound from Theorem 1 when this number is maximum. The only difference is a discrepancy of $\frac{2}{7}$ in the constant.

Below we again prove the existence of a potential standard CD-set S for G and a realization G' that have a number of properties. The first five properties are the same as those stated in Theorem 8. In addition, since cubic diamonds are the only obstruction to an improved bound, we need to make sure that cubic diamonds in $G'[S]$ only occur when they cannot be avoided. This is the case when the graph G contains a *diamond necklace* (see Figure 5).

Definition 15 *An induced subgraph H of G is a* diamond necklace *in G if for some $k \geq 1$, the vertices of H can be labeled $u_1, \ldots, u_{k+1}$, $v_1, \ldots, v_k$, $w_1, \ldots, w_k$, such that the edge set of H consists exactly of u_iv_i, u_iw_i, v_iw_i, v_iu_{i+1}, w_iu_{i+1} for $i = 1, \ldots, k$, and such that $u_2, \ldots, u_k$ have degree four in G, and all other vertices of H have degree three in G.*

Note that a cubic diamond is also a diamond necklace. Property 6 in Theorem 18 below shows that in the CD-set we consider, the only cubic diamond blocks that occur are part of a diamond necklace. Before we can explain Property 7 of Theorem 18, we need the following definitions.

Definition 16 *A vertex $v \in S$ is a* simple CD-set vertex *with respect to the CD-set S for graph G if $d(v) = 3$ and*

- *$G[S - v]$ has two components, and $S - v$ is a dominating set for G, or*
- *$G[S - v]$ is connected, and v has exactly one 1-leaf neighbor.*

Definition 17 *A 1-leaf v is a* block-leaf *with respect to the CD-set S for graph G if there is a path P in $G[S]$ with end vertices x and y ($x = y$ is possible), consisting only of simple CD-set vertices, such that*

- *x is a neighbor of v.*
- *y is part of a block of $G[S]$, but none of the other vertices of P are part of a block.*

Sloppily speaking, in the proof of Theorem 9 and Lemma 11, a lower bound for $|\overline{S}|$ was found as follows. First weights were assigned to vertices in S, mainly from leaves that distributed a weight of one equally among their neighbors in

S. Then we showed that these weights can be redistributed over the vertices of S such that every vertex receives a weight of at least $\frac{1}{2}$.

If blocks of $G'[S]$ may contain triangles, then this is not always possible. However, for every block that is not a diamond, there is usually one additional block-leaf (there are exceptions to this statement, but those are easy to work with). If we can ensure that there are relatively few block-leaves (and thus that blocks are sparse), then we can change the weight assignment: every block-leaf will assign a weight of a little more than 1 to its CD-set neighbors, and other leaves will distribute a weight of a little less than 1 among its CD-set neighbors. This can be done such that the total weight is still at most $|\overline{S}|$. With this weight assignment, non-diamond blocks receive enough weight to prove the desired bound. Property 7 below shows that there are relatively few block-leaves.

Theorem 18 *A connected graph G with $\delta(G) \geq 3$ has a spanning subgraph G' and a CD-set S with the following properties. Degrees, neighbors and all other graph related notions are taken with respect to G'.*

1. *S is a standard CD-set (for G').*
2. *If $u, v \in S$, $d(u) \geq d(v) \geq 4$ and $uv \in E(G')$, then uv is a bridge of $G'[S]$.*
3. *Edges in $E(G) \backslash E(G')$ are between vertices in S, or between a vertex in S and a 2-leaf.*
4. *Every vertex $v \in S$ that is neither a dominator nor a connector has at least three neighbors in S, and all of its neighbors in S have degree three.*
5. *If $\{u, v\} \subseteq S$, $uv \in E(G')$ and u and v are both neither dominators nor connectors, then $G'[S] - u - v$ is not connected.*
6. *The number of cubic diamond blocks in $G'[S]$ is at most the number of diamond necklaces in G.*
7. *The number of block-leaves is at most half the total number of leaves.*

In addition, if $|S| = 2$, then $|\overline{S}| \geq 4$.

Proof: We consider a CD-set S and spanning subgraph G' such that S is a standard CD-set for G', that are optimal according to the following priorities:

- Minimize $|S|$.
- Maximize the number of 2-leaves in $\overline{S}$.
- Minimize the number of cubic diamond blocks in $G'[S]$.
- Minimize $|E(G'[S])|$.
- Maximize $|E(G')|$.

By this we mean that among all pairs of S and G' that minimize $|S|$, we consider pairs that maximize the number of 2-leaves. Among all these pairs, we choose one that minimizes the number of cubic diamond blocks in $G'[S]$, etc. We show that for S and G' chosen this way, the above properties hold. The proof is by contradiction: we consider a standard CD-set S for G'. If one of the properties does not hold, we find an improved S and G' according to the priorities.

Property 2: If $u, v \in S$, $d(u) \geq d(v) \geq 4$ and $uv \in E(G')$, then uv is a bridge of $G'[S]$.
Proof: Consider $u, v \in S$ with $d(u) \geq 4$, $d(v) \geq 4$ and $uv \in E(G')$. If uv is not a bridge in $G'[S]$, then we can delete uv such that S still is a standard CD-set for G'. Since this decreases $|E(G'[S])|$, this change is an improvement unless it introduces a new cubic diamond block D. In the latter case, assume w.l.o.g. $v \in V(D)$, and let $z \notin \{u, v\}$ be a vertex in D adjacent to the three other vertices of D. Consider $S - z$ and $G' - vz$ (with $uv \in E(G')$). This is again a standard CD-set, so an improvement is found.

The proofs of Property 4 and Property 5 are exactly the same as in the proof of Theorem 8, and the proof of Property 3 is very similar. We leave the details to the reader.

Property 6: The number of cubic diamond blocks in $G'[S]$ is at most the number of diamond necklaces in G.
Proof: We show that every cubic diamond block in $G'[S]$ is part of a diamond necklace in G, and that every diamond necklace in G contains only one cubic diamond block of $G'[S]$.

First we define an operation on a standard CD-set S for subgraph G' of G, which changes S and G'. This operation is illustrated in Figure 6. Let D be a cubic diamond block in $G'[S]$. Edges in $E(G) \backslash E(G')$ will be called *removed edges*. Suppose a removed edge xy exists with $x \in V(D)$ and y a 2-leaf. A *2-leaf change* using xy consists of the following steps: let $w \in V(D) - x$ be a vertex with three neighbors in $V(D)$. Add y to S, remove w from S, add xy to $E(G')$, and remove wx from $E(G')$. If the addition of y to S introduces a 3-leaf z, then in addition remove yz from $E(G')$, for every such leaf z. Since y now has three neighbors in S, this last operation does not reduce its degree below three. After this change, x has degree three and w is a 2-leaf. It follows that the result is a standard CD-set S^* for the new graph G^* and that D is not a block in $G^*[S^*]$. Note that $G^*[S^*]$ may contain a cubic diamond block that was not present in $G'[S]$, as illustrated in Figure 6(b).

Using 2-leaf changes, we will prove the statements from the beginning of the proof. Let $D_1 = G'[\{u_1, v_1, w_1, u_2\}]$ be a cubic diamond block in $G'[S]$, with $u_1u_2 \notin E(G')$. If D_1 is also a cubic diamond in G, the statements are obvious. Otherwise, there is a removed edge xy such that $x \in V(D_1)$. If $y \in S$, then add xy to G'. S is again a standard CD-set with respect to G', and $G'[S]$ contains one cubic diamond block less. This is an improvement. So we may assume

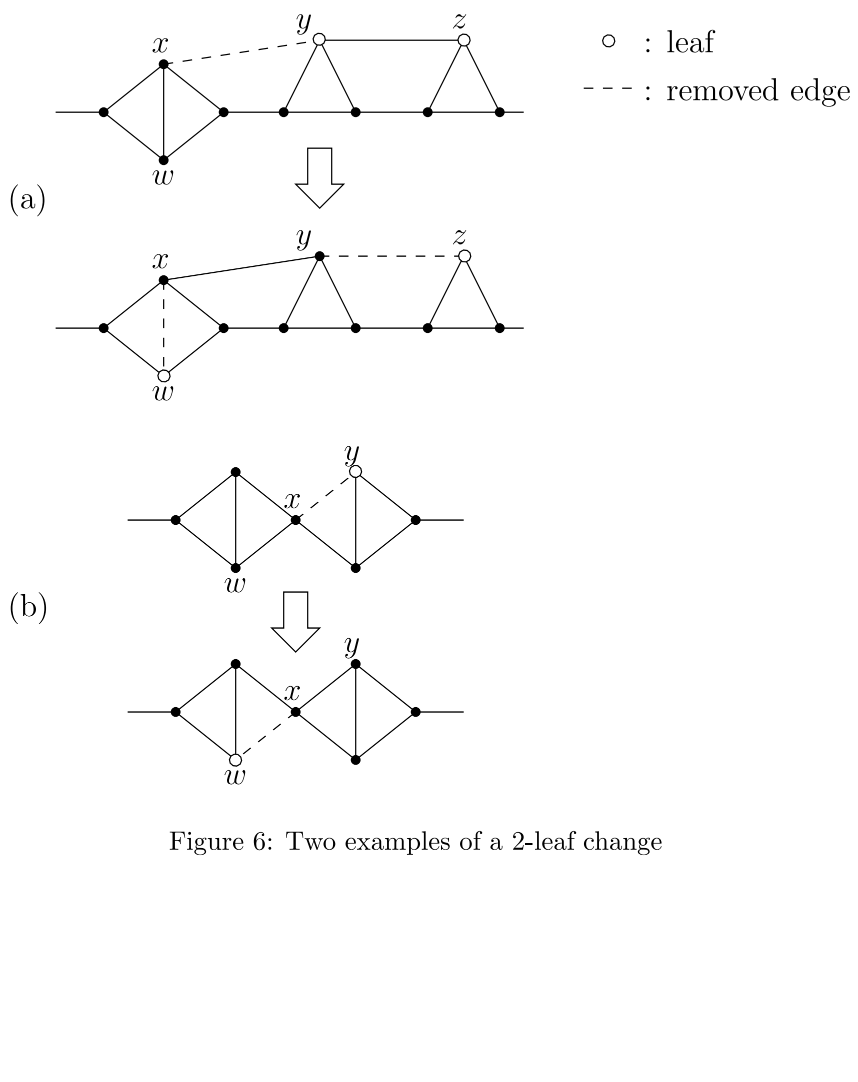

Figure 6: Two examples of a 2-leaf change

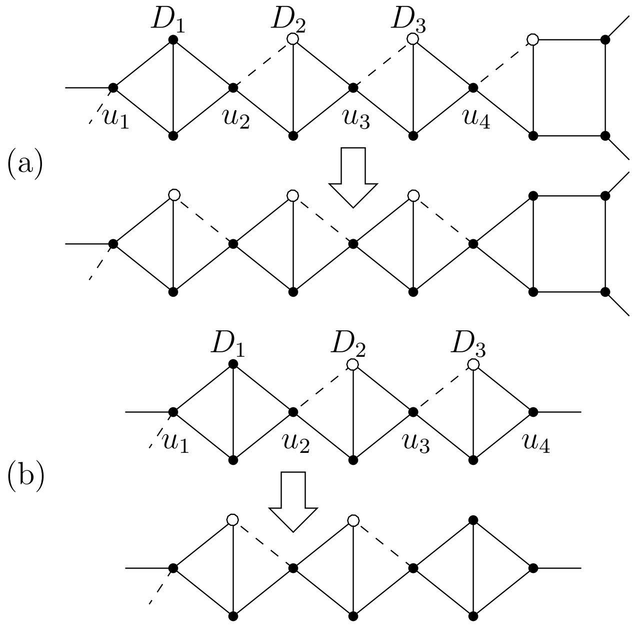

Figure 7: Two possible results of a series of 2-leaf changes

that $y \notin S$. Then y is a 2-leaf (Property 3). We can apply a 2-leaf change using the edge xy. If the number of cubic diamond blocks decreases, this is an improvement (as indicated in Figure 6(a)). Otherwise, x and y are part of a new cubic diamond block D_2. In this case, D_1 and D_2 only have x in common (the other two remaining vertices of D_1 cannot be part of a cubic diamond block since they are now adjacent to a 2-leaf), and w.l.o.g. $x = u_2$ (otherwise x would have degree at least four, and D_2 would not be cubic). This case is shown in Figure 6(b). We label the vertices of D_2 as $V(D_2) = \{u_2, v_2, w_2, v_3\}$, with $v_2v_3 \notin E(G')$.

If a 2-leaf change is possible for D_1, we first try to apply a 2-leaf change that decreases the number of cubic diamond blocks. If this is not possible, we apply an other 2-leaf change. So in this case, v_1 and w_1 are not incident with removed edges. We also know that w.l.o.g. u_2 is incident with exactly one removed edge (otherwise the neighbor of u_2 in $V(G')\backslash V(D_1)$ would have at least two 2-leaf neighbors and therefore would have degree at least four), and u_1 is incident with at most one removed edge. We apply a 2-leaf change using the removed edge incident with u_2, which gives a new cubic diamond block D_2, with $V(D_2) = \{u_2, v_2, w_2, u_3\}$ such that u_2 and u_3 are not adjacent. Now we try to find an improvement again: if an edge can be added such that D_2 is not a cubic diamond block anymore, or if a 2-leaf change can be applied such that the number of cubic diamond blocks decreases, we have found such an improvement. Otherwise, if a 2-leaf change is possible that does not introduce D_1 again, we apply this 2-leaf change introducing a cubic diamond block D_3 with $V(D_3) =$

$\{u_3, v_3, w_3, u_4\}$. We continue to apply 2-leaf changes (without returning to a previous state) until either an improvement is found (see Figure 7(a)), or no new 2-leaf change can be applied anymore. In the latter case, we have found k diamonds in G labeled $D_1, \dots, D_k$ such that $V(D_i) = \{u_i, v_i, w_i, u_{i+1}\}$, where u_i and u_{i+1} are not adjacent, and we know that $d_G(v_i) = d_G(w_i) = 3$ for all i, $d_G(u_{k+1}) = 3$, $d_G(u_i) = 4$ for $i = 2, \dots, k$ (see Figure 7(b)). If $d_G(u_1) = 3$, then we have found a diamond necklace in G which contains only one cubic diamond block of $G'[S]$. If $d_G(u_1) > 3$, then we consider the original G' and S again, and apply the same strategy starting with a 2-leaf change using an edge incident with u_1. If this also does not lead to an improvement, a structure like the one shown in Figure 7(b) is present on both sides of D_1, and we have again found a diamond necklace in G.

In every case we have either found an improvement, or have shown that the cubic diamond block is part of a cubic diamond necklace in G that contains only one cubic diamond block. This concludes the proof of Property 6.

Property 7: The number of block-leaves is at most half the total number of leaves.
Proof: We will show that if there are leaves with at least two block-leaf neighbors, we can make an improvement according to our priorities, except in one very specific case. Combined with the fact that 1-leaves have at least two leaf neighbors (Property 3), this enables us to prove that there are at least as many non-block-leaves as block-leaves in S and G' (a *non-block-leaf* is a leaf but not a block-leaf).

Below we consider a number of cases corresponding to structures in S and G'. In every case we will define a new pair S^* (or S_2) and G^* such that S^* is a standard CD-set for subgraph G^* of G, and S^*, G^* is an improvement. We use the following shorthand to indicate the four features we will check for S^* and G^*:

Connected: S^* induces a connected graph.
Dominating: S^* is a dominating set.
2-CD-set: There are only 1-leaves and 2-leaves with respect to S^* and G^*.
Minimum degree 3: Vertices in S^* have degree at least three in G^*.
Improvement: In every case we can show that $|S^*| < |S|$ or $|S^*| = |S|$ and S^* has more 2-leaves than S.

First we study these cases separately, and afterwards we use these cases to prove Property 7.

Suppose there is a leaf u with two block-leaf neighbors v and w. Let x and y be the CD-set neighbors of v resp. w. Consider $S^* = S + u - x - y$.

Case 1: S^* is again a CD-set.
We will show to construct graph G^*. It is clear that this change is an improvement.
2-CD-set: If there is a 2-leaf a with respect to S that becomes a 3-leaf because of the addition of u to the CD-set, then $d(u) > 3$, since v, w and at least one

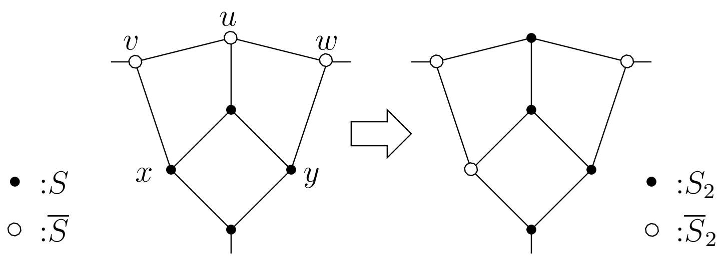

Figure 8: The change from case 2.1

CD-set vertex are also neighbors of u. So the edge au can be deleted from G' without decreasing the degree of a CD-set vertex below three. This holds for every such edge. Since v and w are block-leaves for S, $d(x) = d(y) = 3$, and they both have at least one leaf neighbor with respect to S^*, so the two new leaves are also not 3-leaves.
Minimum degree 3: Only vertex u with $d(u) \geq 3$ was added to the CD-set.

Case 2: $G'[S^*]$ is not connected. We consider two subcases which cover this case.

Case 2.1: $G'[S^*]$ and $G'[S - x - y]$ are not connected.
Since v and w are block-leaves and x and y are dominators, x and y cannot be connectors. So x and y form a minimal 2-cut for $G'[S]$. So both have at least two CD-set neighbors. Consider $S_2 = S + u - x$ (See Figure 8 for an example).
Connected: x is not a connector, so $G'[S - x]$ is connected. Since v is a block-leaf, x has degree three, and thus u is not adjacent to x. So $G'[S_2]$ is also connected.
Dominating: Since v is a block-leaf, only v is not dominated by $S - x$. u is a neighbor of v, so S_2 is a dominating set.
2-CD-set: If a 2-leaf becomes a 3-leaf because of the addition of u, we can delete an edge from G' (as in case 1). x becomes a 2-leaf.
Minimum degree 3: u has degree at least three.
Improvement: $|S_2| = |S|$. We gain a 2-leaf (x), and lose a leaf that may have been a 1-leaf or 2-leaf (u). v remains a 1-leaf, but with a new CD-set neighbor. w was a 1-leaf but becomes a 2-leaf after the addition of u. x has degree three, so it has no leaf neighbors other than v, and therefore no 2-leaves can become 1-leaves. It is possible that some other 1-leaves become 2-leaves after the addition of u. We see that there is at least one more 2-leaf.

Case 2.2: $G'[S^*]$ is not connected, but $G'[S - x - y]$ is connected.
In this case, u is an isolated vertex in $G'[S^*]$. This means that in S its CD-set neighbors are exactly x and y (not just one of them, since v and w are block-leaves). Consider $S_2 = S + u - x$ (See Figure 9).
Connected: x is not a connector, so $G'[S - x]$ is connected. u has y as a

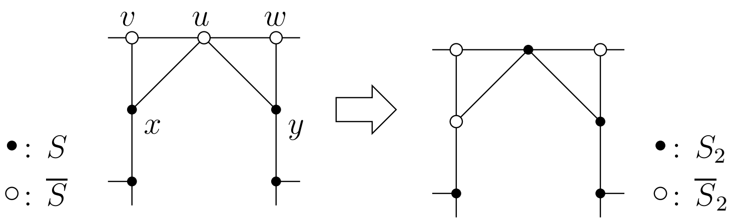

Figure 9: The change from case 2.2

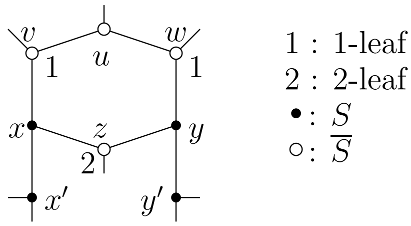

Figure 10: The situation and labeling in case 3

CD-set neighbor, so $G'[S_2]$ is connected.
Dominating: Only v is not dominated by $S - x$. v is adjacent to u, so S_2 is dominating.
2-CD-set: If the addition of u makes 2-leaves into 3-leaves, we can again delete edges from G' to prevent this (as in case 1). x has degree three and one leaf neighbor (v), so it is not a 3-leaf.
Minimum degree 3: u has degree at least four.
Improvement: $|S_2| = |S|$. We gain a 2-leaf (x), and lose a 2-leaf (u). v remains a 1-leaf, with a new CD-set neighbor. w was a 1-leaf but becomes a 2-leaf after the addition of u. x has degree three, so it has no leaf neighbors other than v, and therefore no 2-leaves can become 1-leaves. It is possible that some other 1-leaves become 2-leaves after the addition of u. We see that there is at least one more 2-leaf.

Case 3: Suppose S^* is not a dominating set.
This case is more complicated than the previous cases. We study the situation in more detail, and introduce some additional vertex labels, and then proceed to study six subcases. See Figure 10. The only 1-leaf neighbors of x and y are v resp. w, so there is a 2-leaf (with respect to S) that has CD-set neighbors x and y. Let z be this 2-leaf. $d(z) = 2$ only if in G, z is adjacent to another vertex $a \in S$ (Property 3). In this case, we can add the edge az to G', and find that there is a graph for which S^* is a standard CD-set (see case 1 for details). So we may now assume that z has at least one leaf neighbor. Since S^* is not dominating, this leaf neighbor is not equal to u, but can be equal to v or w. x and y have degree one in $G'[S]$. Let x' and y' be their respective CD-set

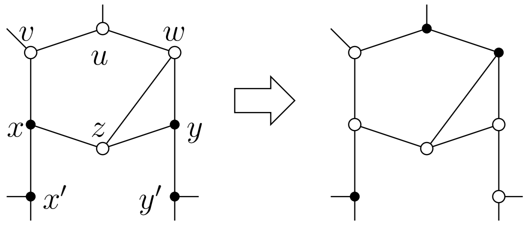

Figure 11: The change from case 3.1

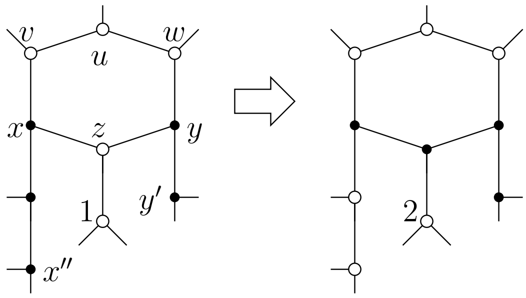

Figure 12: The change from case 3.2

neighbors. $x' \neq y'$ since v and w are block-leaves.

Case 3.1: Suppose z is adjacent to w. Now consider $S_2 := S+u+w-x-y-y'$ (See Figure 11).
Connected: $G'[S-x]$ and $G'[S-y-y']$ are connected since v and w are block-leaves. x is not part of a block (of $G'[S]$), so $G'[S-x-y-y']$ is also connected. y' is not a dominator, so u has a neighbor in S not in $\{x, y, y'\}$. v is adjacent to u. So $G'[S_2]$ is connected.
Dominating: The only vertices not dominated by $S-x-y-y'$ are y, v, w and z, since y' is not a dominator. u dominates v, w is in S_2, and w dominates z and y.
2-CD-set: z becomes a 1-leaf, so u and w both are adjacent to at least three vertices that will not become 3-leaves. So if for S_2 there would be 3-leaves adjacent to u or w, we can prevent this by deleting the corresponding edges, without decreasing the degrees of u and w below three. The new leaves have degree three and are adjacent to other leaves.
Minimum degree 3: u and w have degree at least three.
Improvement: $|S_2| < |S|$.

Case 3.2: Suppose the following properties hold:

- z has a 1-leaf neighbor.
- x' is not part of a block (of $G'[S]$), so it has one other simple CD-set

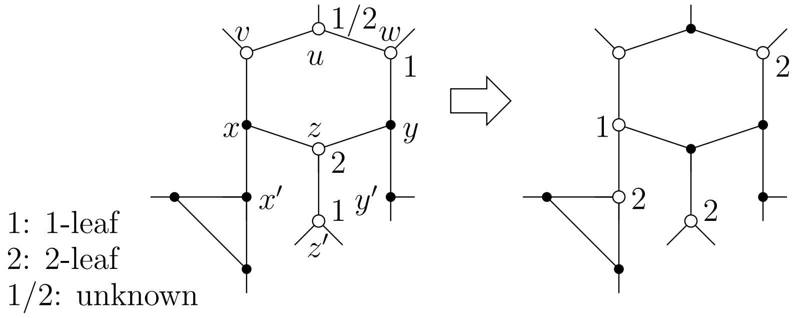

Figure 13: The change from case 3.3

neighbor x''.

Consider $S_2 = S+z-x'-x''$ (See Figure 12). If x' and x'' share a 2-leaf neighbor, then x'' is not part of a block and has one other simple CD-set neighbor x''' and we consider $S_2 = S + z - x'' - x'''$ instead.
Connected: $G'[S - x' - x'']$ has two components, one consisting of the single vertex x. z is adjacent to y and x, so $G[S_2]$ is connected. In the case where we consider $G'[S - x'' - x''']$, one component consists of vertices x and x', and the reasoning is the same.
Dominating: x' and x'' (or x'' and x''') are not dominators, and do not share a 2-leaf neighbor, and both have neighbors in $S - x' - x''$ ($S - x'' - x'''$), so $S - x' - x''$ ($S - x'' - x'''$) is already a dominating set.
2-CD-set: z has at least three neighbors that are not 2-leaves with respect to S, so if 3-leaves would be introduced by the addition of z, we can prevent this by deleting the corresponding edges. x' and x'' (x'' and x''') have degree three and are adjacent, so these will not become 3-leaves.
Minimum degree 3: z has degree at least three.
Improvement: $|S_2| < |S|$.

Case 3.3: Suppose the following properties hold:

- z has a 1-leaf neighbor $z' \neq v$, $z' \neq w$.
- x' is part of a block.

Consider $S_2 := S + u + z - x - x'$ (See Figure 13).
Connected: $G'[S - x - x']$ is connected. Since x' is not a dominator and $N(x) = \{v, z, x'\}$, u has a CD-set neighbor other than x' or x. z is adjacent to y. So $G'[S_2]$ is connected.
Dominating: Only v and x are not dominated by $S - x - x'$. u dominates v, and z dominates x.
2-CD-set: x and x' do not become 3-leaves since they have degree three. u is adjacent to at least three vertices that will not become 3-leaves. z is adjacent to at least three vertices that will not become 3-leaves.

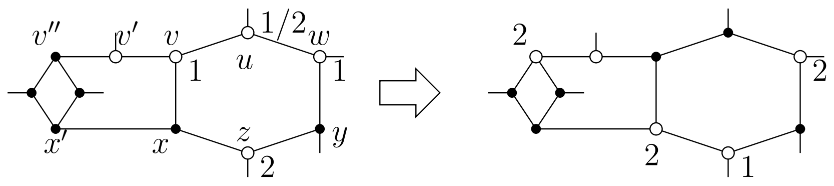

Figure 14: The change from case 3.4

Minimum degree 3: u and z have degree at least three.
Improvement: $|S_2| = |S|$. We lose u which can be a 1-leaf or 2-leaf, and we lose the 2-leaf z. We gain a 2-leaf x' (since it is part of a block) and a 1-leaf x. w becomes a 2-leaf, v remains a 1-leaf (with a new CD-set neighbor), and z' becomes a 2-leaf. Since x and x' have no other leaf neighbors in $\overline{S}$, there are no other 2-leaves that can become 1-leaves. It is possible that some 1-leaves become 2-leaves by the addition of u and z. So we gain at least one 2-leaf.

Case 3.4: Suppose the following properties hold:

- v has a neighbor $v' \neq u$ that is a block-leaf. Let v'' be its CD-set neighbor.
- x' and v'' are part of the same block.

Note that v' cannot be equal to w, because in that case $v'' = y$ and is not part of a block. Consider $S_2 = S + u + v - v'' - x$ (See Figure 14).
Connected: $G'[S - x]$ and $G'[S - v'']$ are connected since v resp. v' are block leaves. Since x is not part of a block, $G'[S - x - v'']$ is also connected. u has a CD-set neighbor other than x or v'', and v is adjacent to u.
Dominating: In $S - x - v''$, only v and v' are not dominated, since x and v'' do not share a 2-leaf neighbor. $v \in S_2$, and v' is dominated by v.
2-CD-set: u and v have at least three non-2-leaf neighbors, so any 3-leaves adjacent to u or v can be prevented by deleting edges from G'. v'' and x have degree three and have at least one leaf neighbor in $\overline{S}_2$.
Minimum degree 3: u and v have degree at least three.
Improvement: $|S_2| = |S|$. We lose the 1-leaf v and the leaf u which can be a 1-leaf or a 2-leaf. v'' is part of a block so it becomes a 2-leaf. x becomes a 2-leaf since $v \in S_2$. z becomes a 1-leaf instead of a 2-leaf, v' remains a 1-leaf (with a new CD-set neighbor), and w becomes a 2-leaf instead of a 1-leaf. In S there are no 2-leaves adjacent to v'' since $d(v'') = 3$ and it is part of a block, so the removal from the CD-set of v'' cannot make 2-leaves into 1-leaves. Something similar is true for x. The addition of u and v can make some 1-leaves into 2-leaves. So we gain at least one 2-leaf.

Case 3.5: Suppose the following properties hold:

- v has a neighbor $v' \neq u$, $v' \neq w$ that is a block-leaf.

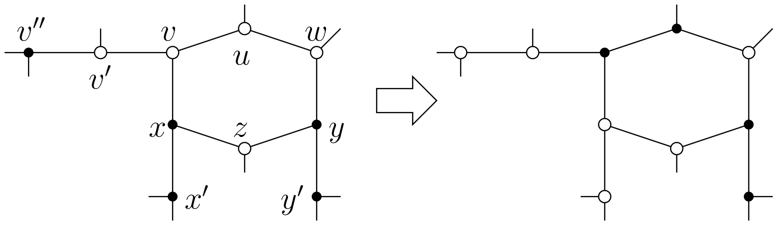

Figure 15: The change from case 3.5

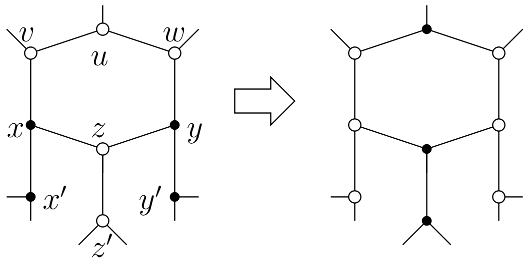

Figure 16: The change from case 3.6

- Let v'' be the unique CD-set neighbor of v'. v'' and x' are not part of the same block, and do not share a 2-leaf neighbor.

Consider $S_2 = S + u + v - x - x' - v''$ (See Figure 15).
Connected: Since v is a block-leaf, $G'[S - x - x']$ is connected. Since v' is a block-leaf, $G'[S - v'']$ is connected. In addition, since v'' and x' are not part of the same block, $G'[S - x - x' - v'']$ is connected. x' and v'' do not share a 2-leaf neighbor, so u has a CD-set neighbor other than x, x' or v''. v is adjacent to u. So $G'[S_2]$ is connected.
Dominating: The only vertices not dominated by $S - x - x' - v''$ are v, v' and x, since v and v' are the only 1-leaves adjacent to x resp. v'', x' is not a dominator, and x' and v'' do not share a 2-leaf neighbor. $v \in S_2$, and v' and x are dominated by v.
2-CD-set: x, x' and v'' all have degree three and have at least one leaf neighbor with respect to S_2. Both u and v have at least three neighbors that will not become 3-leaves.
Minimum degree 3: u and v have degree at least three.
Improvement: $|S_2| < |S|$.

Case 3.6: Suppose the following properties hold:

- x' and y' are not part of the same block and do not share a 2-leaf neighbor.
- z has a neighbor z' that is a 2-leaf.

Consider $S_2 = S + u + z + z' - x - x' - y - y'$ (See Figure 16).
Connected: Since v and w are block leaves, $G'[S-x-x']$ resp. $G'[S-y-y']$ are connected. Since x' and y' are not part of the same block, $G'[S-x-x'-y-y']$ is also connected. Since x' and y' do not share a 2-leaf neighbor, z' has a CD-set neighbor other than x', y', x and y. The same is true for u. z is adjacent to z'. So $G'[S_2]$ is connected.
Dominating: The only vertices not dominated by $S - x - y$ are v, w and z. x' and y' are not dominators and do not share a 2-leaf (also not with x and y), so the only additional vertices not dominated by $S - x - y - x' - y'$ are x and y. In S_2, v and w are dominated by u, z is part of S_2, and x and y are dominated by z.
2-CD-set: The new leaves have degree three and are adjacent to at least one leaf. u, z and z' have at least three neighbors that will not become 3-leaves.
Minimum degree 3: u, z and z' have degree at least three.
Improvement: $|S_2| < |S|$.

Unfortunately, the above cases do not cover all possibilities, so we cannot show that in S and G', every leaf has at most one block-leaf neighbor. However we can use the above cases to show that if a leaf has at least two block-leaf neighbors, it is part of a highly restricted structure. We will deduce a number of properties for this situation, and use this to conclude our proof of Property 7. For this we introduce some definitions and notations.

A cycle C in G' is called a *problem cycle* if its vertices can be labeled u, v, w, x, y and z as in Figure 10, such that u is a leaf with two block-leaf neighbors v and w, v and w have CD-set neighbors x and y, and x and y have the 2-leaf neighbor z in common. For any problem cycle C, we use the notation $u(C)$ to denote the vertex that is labeled u in such a labeling. We will also call $u(C)$ the *u-vertex of C*. For the other five vertices the notation is similar. Note that there is only one vertex in a problem cycle that can receive the label u, and the same holds for the label z, but the labels v and w are interchangeable, just like the labels x and y (though x and v should always be adjacent).

We will now prove that if a leaf u has at least two block-leaf neighbors v and w, then there is a problem cycle C with $u = u(C)$, $v = v(C)$ and $w = w(C)$. Then we will state a number of properties for problem cycles.

Consider a leaf u with at least two block-leaf neighbors v and w, which have CD-set neighbors x resp. y. If $S^* = S + u - x - y$ is again a CD-set, apply Case 1 to obtain a contradiction. If S^* is not connected, apply Case 2.1 or 2.2. If S^* is not dominating, then in Case 3 it is shown that these five vertices are part of a problem cycle C together with a vertex $z = z(C)$, with $d(z) \geq 3$. This covers all cases, so if a leaf u has block-leaf neighbors v and w with CD-set neighbors x resp. y, then $u = u(C)$, $v = v(C)$, $w = w(C)$, $x = x(C)$ and $y = y(C)$ for some problem cycle C. In this case we can define vertices x' and y' as it is done in Case 3, and use the subcases of Case 3 to deduce properties of C. These properties are stated in the following claims.

1. z has no 1-leaf neighbor, and has at least one 2-leaf neighbor.

Proof: If z has a 1-leaf neighbor, this cannot be u since then $S+u-x-y$ is a CD-set. If z is adjacent to w or v, we can apply Case 3.1 (using the symmetry of the cases). Otherwise z is adjacent to a 1-leaf $z' \notin V(C)$, and we can apply Case 3.2 or Case 3.3. Since $d(z) \geq 3$, z has a 2-leaf neighbor.

2. If v or w has a block leaf neighbor v', then $v' \in \{u, v, w\}$.
Proof: We prove the statement for v; for w it follows by symmetry. Suppose v has a block-leaf neighbor $v' \notin \{u, w\}$, with CD-set neighbor v''. If x' and v'' are part of the same block, we can apply Case 3.4. If x' and v'' are not part of the same block and do not share a 2-leaf neighbor, we can apply Case 3.5. So we now assume that x' and v'' share a 2-leaf neighbor. Then x' is not part of a block, and does not share a 2-leaf neighbor with y'. We know that z has a 2-leaf neighbor (Claim 1), so Case 3.6 can be applied.

In addition we state two claims for problem cycles that have vertices in common. A non-block-leaf is called *class 0* if it has no block-leaf neighbors, *class 1* if it has one block-leaf neighbor, and *class 2* if it has at least two block-leaf neighbors.

3. If C_1 and C_2 are problem cycles with $u(C_1) = u(C_2)$, then $C_1 = C_2$.
Proof: If $v(C_1) \in V(C_2)$ or $w(C_1) \in V(C_2)$, then w.l.o.g. $v(C_1) = v(C_2)$. Since $v(C_1)$ has exactly one CD-set neighbor, $x(C_1) = x(C_2)$. Since $v(C_1)$ is a block-leaf, $x(C_1)$ has only one other leaf neighbor, so $z(C_1) = z(C_2)$. $z(C_1)$ is a 2-leaf, so $y(C_1) = y(C_2)$. $y(C_1)$ is adjacent to only one block-leaf, so $w(C_1) = w(C_2)$. It follows that $C_1 = C_2$. We conclude that it is not possible that there are two different problem cycles C_1 and C_2 such that $u(C_1) = u(C_2)$ and $v(C_1)$ or $w(C_1)$ is equal to $v(C_2)$ or $w(C_2)$. Now if $u(C_1) = u(C_2)$ and the two cycles do not have any of these vertices in common, then we consider a new problem cycle C_3 that contains $u(C_1)$, $v(C_1)$ and $v(C_2)$ (since a leaf and two of its block-leaf neighbors are always part of a problem cycle). Now we have $u(C_3) = u(C_1)$ and $v(C_3) = v(C_1)$, but $C_1 \neq C_3$, a contradiction.

4. If the problem cycles C_1 and C_2 both contain a class 0 leaf z, then w.l.o.g. $v(C_1) = v(C_2)$, $x(C_1) = x(C_2)$, $z(C_1) = z(C_2)$, $y(C_1) = y(C_2)$ and $w(C_1) = w(C_2)$ (the u-vertices may be different).
Proof: The only class 0 leaf on a problem cycle can be the z-vertex, so $z = z(C_1) = z(C_2)$, and it must be a 2-leaf. So it has only two CD-set neighbors, and w.l.o.g. $x := x(C_1) = x(C_2)$ and $y := y(C_1) = y(C_2)$. Since their other neighbors on the cycles are block-leaves, x and y are only adjacent to one block-leaf, and $v(C_1) = v(C_2)$ and $w(C_1) = w(C_2)$.

For the purpose of proving that at most half of the leaves are block-leaves, we define a function f that assigns non-block-leaves to block-leaves. Using the above claims, we will prove that f is injective. The assignment is done using the following method which contains three assignment steps.

Consider a block-leaf v to which no non-block-leaf has been assigned.

1. If v has a class 1 neighbor u, then assign $f(v) := u$.

2. Otherwise, if v has a class 2 neighbor u, which has another block-leaf neighbor w, then let C be the problem cycle with $v = v(C)$, $u = u(C)$ and $w = w(C)$. Assign $f(v) := u$, $f(w) := z(C)$.

3. Otherwise, v has only block-leaf neighbors. Since $d(v) \geq 3$ (Property 3), $v = u(C)$ for some problem cycle C. Assign $f(v) := z(C)$.

Repeat this until all block-leaves are assigned a non-block-leaf. Clearly, one of the above steps applies for every block-leaf, so f is a function. We will prove that f is injective, so every non-block-leaf is assigned at most once.

Consider a non-block-leaf z. If z is class 1, then it is assigned at most once in step 1. All leaves assigned in steps 2 and 3 are class 0 (Claim 1) or class 2, so z is not assigned in these steps.

If z is class 2, then $z = u(C)$ for some problem cycle C. z is only part of one such problem cycle (Claim 3), so z is assigned at most once in step 2, and not in step 3 (step 3 only assigns class 0 vertices).

If z is class 0, then it can be assigned in step 2 or step 3. For every problem cycle C that z is part of, $v(C)$ and $w(C)$ are the same (Claim 4), so z is assigned at most once in step 2 (even though z can be part of multiple problem cycles, once the v and w-vertices are assigned a non-block-leaf, problem cycles with the same v and w-vertices are not considered anymore in step 2). Suppose z is assigned once in step 2 and once in step 3, or at least twice in step 3. Then $z = z(C_1) = z(C_2)$ for two different problem cycles C_1 and C_2, and w.l.o.g. $u(C_2)$ is a block-leaf (C_2 corresponds to the cycle used in step 3). C_1 and C_2 must also have the v, w, x and y-vertices in common (Claim 4). This means that $v(C_1)$ has a block-leaf neighbor $u(C_2) \notin V(C_1)$, a contradiction with Claim 2. We conclude that also in this case, z is assigned at most once.

We have shown that an injective function from the block-leaves to the non-block-leaves can be constructed. This shows that at most half of the leaves are block-leaves, and completes the proof of Property 7.

We complete the proof of Theorem 18 by proving the following statement.

If $|S| = 2$, *then* $|\overline{S}| \geq 4$.
Proof: Suppose $|S| = 2$. Since there are no connectors, both vertices in S are dominators, and there are at least two 1-leaves u and v. Since the vertices in S have degree at least three (Property 1), there is at least one other leaf w. If in addition to w there is at least one other leaf, we are done. Otherwise, w is a 2-leaf. Since u and v have degree at least three (Property 3), both u and v are also adjacent to w. We conclude that $\{w\}$ is a standard CD-set, which is an improvement. □

Analogously to the use of Theorem 8 and Theorem 9, now that the existence of a CD-set with the properties stated in Theorem 18 has been established, we will use these properties to determine an upper bound on the size of such a CD-set. This will be done in the next section.

5.2 Upper bounds for the size of the constructed CD-set

We introduce a notion that enables us to explore the existence of block-leaves when considering a weight assignment to $G'[S]$.

Definition 19 *A subgraph P of a graph G is a* block path *of G if it is a path with end vertices l and b such that $d(l) = 1$, $d(b) = 3$, b is part of a block of G, and all internal vertices of P have degree two in G. For such a path P, l_P denotes the end vertex with degree one in G, and b_P denotes the end vertex with degree three in G.*

Note that block paths contain at least two vertices. The relation between block-leaves and block paths is as follows. For a CD-set S of graph G', every block-leaf v is adjacent to either a vertex in a block of $G'[S]$, or the end vertex l_P of a block path P in $G'[S]$. Every block path in $G'[S]$ is adjacent to at most one block-leaf.

The next lemma is a variant of Lemma 11 that can be used for $G'[S]$ when S and G' have the properties stated in Theorem 18, and when triangles are allowed. We can now only prove that there are roughly $\frac{2}{5}$ leaves for every CD-set vertex, instead of roughly $\frac{1}{2}$. The first three properties in Lemma 20 then correspond directly to the first three properties in Lemma 11, although the block property we need is more sophisticated. When combined with the weight assignment we use in Theorem 21, the additional path property states that every block path either ends in a block-leaf from which it receives extra weight, or if it does not end in a block-leaf, it receives extra weight in another way.

Lemma 20 *Let $G = (V, E)$ be a connected non-tree graph with non-negative weights w on the vertices such that:*

1. *If $d(v) = 2$ then $w(v) \geq \frac{2}{5}$.*
2. *If $d(v) = 1$ then $w(v) \geq \frac{6}{5}$.*
3. **(Block property)** *Consider a block $G[B]$ of G. Let C denote the set of cut vertices of G, and let L denote the vertices in B that have two neighbors in B, and $H = B \backslash L$. Then $w(B \backslash C) \geq \frac{2}{5}|B| - \frac{6}{5}|C \cap L| - \frac{4}{5}|C \cap H| + \frac{12}{5}$.*
4. **(Path property)** *If P is a block path in G, then $w(P) \geq \frac{2}{5}|V(P)| + \frac{4}{5}$.*

For this graph, $w(V) \geq \frac{2}{5}|V| + \frac{12}{5}$.

Proof: If G is 2-connected, then the statement follows immediately from the inequality for blocks. For the other cases we will construct a smaller graph G' with weight function w', and use induction on $|V|$. For this we need to prove the four properties for G' and w'.

First, a remark on proving the path property for G' and w'. Suppose Property 1 and 2 of the lemma hold for G' and w'. Let P be a block path in G'. Note these two properties already imply that $w'(P) \geq \frac{2}{5}(|V(P)| - 2) + \frac{6}{5} = \frac{2}{5}|V(P)| + \frac{2}{5}$. So in order to prove that the path property holds for P, it suffices to show that $w'(l_P) \geq \frac{8}{5}$, or $w'(b_P) \geq \frac{2}{5}$, or $w'(x) \geq \frac{4}{5}$ for some internal vertex x of P.

If G is not 2-connected, it has at least one leaf or at least one block with exactly one cut vertex (Lemma 5). We distinguish these two cases.

Let u be a leaf of G, with neighbor v. Consider the graph $G' = G - u$ with weight function $w'(v) = w(v) + w(u) - \frac{2}{5}$, and $w'(x) = w(x)$ for all other vertices. We prove that G' and w' satisfy the four properties of the lemma. Note that since $w(u) \geq \frac{6}{5}$, $w'(v) \geq w(v) + \frac{4}{5}$.

If v has degree two in G' then since $w'(v) \geq \frac{4}{5}$, the first property holds for v. If v has degree one in G', then $w(v) \geq \frac{2}{5}$ (since v has degree two in G), so $w'(v) \geq \frac{6}{5}$, and the second property holds for v. For all other vertices, the weights and degrees do not change so the first two properties hold for all vertices.

For a block $G'[B]$, it is obvious that the block property still holds, unless $v \in B$ and v is not a cut vertex anymore in G'. In this case, let C_G resp. $C_{G'}$ denote the set of cut vertices of G and G' ($C_G = C_{G'} + v$), and let L denote the set of vertices in B with two neighbors in B, and $H = B \backslash L$ (H and L are the same in G and G'). If $v \in H$, then

$$w'(B \backslash C_{G'}) \geq w(B \backslash C_G) + \frac{4}{5} \geq$$

$$\frac{2}{5}|B| - \frac{6}{5}|C_G \cap L| - \frac{4}{5}|C_G \cap H| + \frac{16}{5} =$$

$$\frac{2}{5}|B| - \frac{6}{5}|C_{G'} \cap L| - \frac{4}{5}|C_{G'} \cap H| + \frac{12}{5}.$$

If $v \in L$, then v has degree two in G' and degree three in G, so $v = b_P$ for a block path P in G on vertices u and v. Therefore $w'(v) \geq \frac{2}{5}|V(P)| + \frac{4}{5} - \frac{2}{5} = \frac{6}{5}$, thus $w'(B \backslash C_{G'}) \geq w(B \backslash C_G) + \frac{6}{5}$, and the same reasoning as above shows that the block inequality holds again.

Now suppose v is part of a block path P' in G'. If $v = P'_l$, then v is an internal vertex of a block path P of G, with $V(P) = V(P') + u$. In that case, $w'(P') = w(P) - \frac{2}{5} = \frac{2}{5}|V(P)| + \frac{4}{5} - \frac{2}{5} = \frac{2}{5}|V(P')| + \frac{4}{5}$, so the path property holds for P'. If $v = P'_b$ or v is an internal vertex of P', then the path property holds for P' since the first two properties hold for G', and $w'(v) \geq \frac{4}{5}$. All other block paths in G' correspond to block paths in G with the same weights.

We conclude that if G has a leaf u, then we can construct a new graph $G' = G - u$ with weight function w' such that $w'(V(G')) = w(V(G)) - \frac{2}{5}$, for which the four properties hold. G' is not a tree since G is not a tree, so we can use induction. By induction, $w(V(G)) = w'(V(G') + \frac{2}{5} = \frac{2}{5}|V(G')| + \frac{12}{5} + \frac{2}{5} = \frac{2}{5}|V(G)| + \frac{12}{5}$, which proves the lemma for this case.

Now we consider the case that G has no leaves. Then G has a block B which contains only a single cut vertex u. Consider the graph $G' = G - (B - u)$ with weight function $w'(u) = \frac{8}{5}$ and $w'(x) = w(x)$ for all other vertices. Since G is not 2-connected and has no leaves, it has at least two blocks (Lemma 5), and therefore G' is not a tree. We will show that the four properties hold for G' and w', and use induction.

If $d_{G'}(u) = 1$ or $d_{G'}(u) = 2$, then w' clearly satisfies the corresponding properties. Since $w'(u) = \frac{8}{5}$, a block path in G' that contains u satisfies the path property. For a block $G'[B]$, the block property clearly holds when $u \notin B$ or u is still a cut vertex in G'. In the other case, let C_G resp. $C_{G'}$ denote the set of cut vertices of G and G', and let L denote the set of vertices in B with two neighbors in B, and $H = B \backslash L$. Now

$$w'(B \backslash C_{G'}) = w(B \backslash C_G) + \frac{8}{5} \geq$$

$$\frac{2}{5}|B| - \frac{6}{5}|C_G \cap L| - \frac{4}{5}|C_G \cap H| + \frac{20}{5} >$$
$$\frac{2}{5}|B| - \frac{6}{5}|C_{G'} \cap L| - \frac{4}{5}|C_{G'} \cap H| + \frac{12}{5}.$$

We conclude that G' and w' satisfy all properties of the lemma. Since G' is not a tree, we can use induction. Let $V = V(G)$ and $V' = V(G')$.

$$w(V) \geq w'(V') + w(B - u) - \frac{8}{5} \geq (\frac{2}{5}|V'| + \frac{12}{5}) + (\frac{2}{5}|B| + \frac{6}{5}) - \frac{8}{5} =$$

$$\frac{2}{5}|V'| + \frac{2}{5}|B| + \frac{10}{5} = \frac{2}{5}(|V| - |B| + 1) + \frac{2}{5}|B| + \frac{10}{5} = \frac{2}{5}|V| + \frac{12}{5}.$$

This proves the statement for the case that G has no leaves, which completes the proof of Lemma 20. □

Using the previous weight counting lemma, we can prove the following upper bound for the size of a minimal CD-set $S' \subseteq S$.

Theorem 21 *Let G' and S be a graph and a CD-set satisfying the properties from Theorem 18. If $S' \subseteq S$ is a minimal CD-set, then $|S'| \leq (5|V(G)| - 12 + D)/7$, where D is the number of cubic diamond blocks in $G'[S]$.*

Proof: In order to prove this theorem, we assign weights w to the vertices of S such that the total weight is at most $|V \backslash S| + \frac{7}{5}|S \backslash S'| + D/5$, and then show that $w(S) \geq \frac{2}{5}|S| + \frac{12}{5}$. Then we can combine the two inequalities to obtain the result.

The weight assignment is as follows: block-leaves assign a weight of $\frac{6}{5}$ to their CD-set neighbor, and all other leaves distribute a weight of $\frac{4}{5}$ equally among their neighbors in S' (vertices in $S \backslash S'$ receive no weight yet). Since at most half of the leaves are block-leaves (Property 7 of Theorem 18), we have assigned a total weight of not more than $|V \backslash S|$. Next, we assign a weight of $\frac{7}{5}$ to each vertex in $S \backslash S'$. Vertices in $S \backslash S'$ that are part of a cubic diamond block of $G'[S]$ receive an additional weight of $\frac{1}{5}$.

We prove that $G'[S]$ and this weight assignment w satisfy the four properties of Lemma 20.

1. A vertex v with degree two in $G'[S]$ is a dominator or connector (Property 4 of Theorem 18), so $v \in S'$. v has at least one leaf neighbor u (Property 1), and u has at most two neighbors in $G'[S]$, so it assigns at least $\frac{2}{5}$ to v. This shows that $w(v) \geq \frac{2}{5}$.

2. A vertex v with degree one in $G'[S]$ must be a dominator, since it cannot be a connector, and it must be a connector or dominator (Property 4). From the 1-leaf adjacent to it, it gets a weight of at least $\frac{4}{5}$. In addition, since v has degree at least three and leaves have at most two CD-set neighbors, it gains an additional weight of at least $\frac{2}{5}$. So $w(v) \geq \frac{6}{5}$.

3. If $G'[B]$ is a block of $G'[S]$, and L is the set of vertices in B with two neighbors in B and H is the set of vertices in B with at least three neighbors in B, then $|L| \geq |H|$ (Lemma 10). Let C denote the set of connectors. A vertex in L that is not a connector must be a dominator (Property 4). A dominator in L has weight at least $\frac{6}{5}$, since either its 1-leaf neighbor is a block-leaf, or it has degree at least four and receives weight at least $\frac{4}{5} + \frac{2}{5}$. We consider two cases:

   (a) Suppose $|(S \backslash S') \cap B| \geq 1$. Note that this implies that $|H| \geq 1$ (Property 4). Then one of the vertices in H has an additional weight of at least $\frac{7}{5}$. Now if $|B| \geq 5$ then $w(B \backslash C) \geq \frac{6}{5}|L \backslash C| + \frac{7}{5} \geq \frac{3}{5}|B| - \frac{6}{5}|C \cap L| + \frac{7}{5} \geq \frac{2}{5}|B| - \frac{6}{5}|C \cap L| + \frac{12}{5} \geq \frac{2}{5}|B| - \frac{6}{5}|C \cap L| - \frac{4}{5}|C \cap H| + \frac{12}{5}$. If $|B| < 5$, then $G'[B]$ is a diamond, since $|H| \geq 1$. In this case, let $v \in (S \backslash S') \cap B$. v is not a connector or dominator in S. Therefore, all its neighbors have degree three in G'. So there is one other vertex in B that is not a connector or dominator, and therefore v also has degree three. We conclude that B is a cubic diamond block, so the additional weight of v is $\frac{8}{5}$, and $w(B \backslash C) \geq \frac{6}{5}|L \backslash C| + \frac{8}{5} \geq \frac{3}{5}|B| - \frac{6}{5}|C \cap L| + \frac{8}{5} = \frac{2}{5}|B| - \frac{6}{5}|C \cap L| + \frac{12}{5} \geq \frac{2}{5}|B| - \frac{6}{5}|C \cap L| - \frac{4}{5}|C \cap H| + \frac{12}{5}$.

   (b) Suppose $(S \backslash S') \cap B = \emptyset$. Suppose B contains a vertex v that is not a dominator or connector with respect to S. Since $v \in S'$, it is a connector or dominator with respect to S'. Since $(S \backslash S') \cap B = \emptyset$, it is not a connector. So in S, v is adjacent to a 2-leaf u, and the other CD-set neighbor w of u is in $S \backslash S'$. In this case, v receives a weight of $\frac{4}{5}$. So vertices in $H \backslash C$ receive at least $\frac{4}{5}$, and vertices in $L \backslash C$ receive at least $\frac{6}{5}$.

   If $|B| = 3$, then $L = B$, and $w(B \backslash C) \geq \frac{6}{5}|L \backslash C| = \frac{6}{5}|B| - \frac{6}{5}|C \cap L| - \frac{4}{5}|C \cap H| = \frac{2}{5}|B| - \frac{6}{5}|C \cap L| - \frac{4}{5}|C \cap H| + \frac{12}{5}$. Now we assume $|B| \geq 4$, and also use $|L|+|H|=|B|$, $|L|\geq|B|/2$:

$$w(B\backslash C)\geq\frac{6}{5}|L\backslash C|+\frac{4}{5}|H\backslash C|=$$

$$\frac{6}{5}|L|+\frac{4}{5}|H|-\frac{6}{5}|C\cap L|-\frac{4}{5}|C\cap H|\geq\frac{2}{5}|L|+\frac{4}{5}|B|-\frac{6}{5}|C\cap L|-\frac{4}{5}|C\cap H|\geq$$

$$\frac{1}{5}|B|+\frac{4}{5}|B|-\frac{6}{5}|C\cap L|-\frac{4}{5}|C\cap H|\geq\frac{2}{5}|B|-\frac{6}{5}|C\cap L|-\frac{4}{5}|C\cap H|+\frac{12}{5}.$$

4. Let P be a block path in $G'[S]$. If l_P, b_P or an internal vertex x of P has degree at least four in G', then it can be checked that $w(l_P)\geq\frac{8}{5}$, $w(b_P)\geq\frac{2}{5}$ resp. $w(x)\geq\frac{4}{5}$. If l_P has two 1-leaf neighbors, then $w(l_P)\geq\frac{8}{5}$. If an internal vertex x is a dominator then $w(x)\geq\frac{4}{5}$. Otherwise, the 1-leaf neighbor of l_P is a block-leaf, and $w(l_P)\geq\frac{6}{5}+\frac{2}{5}$. Together with the first two properties above, this proves the path property.

If $G'[S]$ is not a tree, we can apply Lemma 20, which shows that $w(S)\geq\frac{2}{5}|S|+\frac{12}{5}$. It follows that

$$\frac{7}{5}|S\backslash S'|+D/5+|V\backslash S|=w(S)\geq\frac{2}{5}|S|+\frac{12}{5}\Leftrightarrow$$

$$\frac{7}{5}|S|-\frac{7}{5}|S'|+D/5+|V|-|S|\geq\frac{2}{5}|S|+\frac{12}{5}\Leftrightarrow$$

$$|V|+D/5-\frac{12}{5}\geq\frac{7}{5}|S'|\Leftrightarrow|S'|\leq(5|V|-12+D)/7.$$

If $G'[S]$ is a tree, we prove the last inequality in another way. Since there are no block-leaves, we assign the weights as we did in the proof of Theorem 9: every leaf distributes a weight of 1 equally among its CD-set neighbors. Using the same reasoning as in the proof of Theorem 9, we see that

- If v has degree two in $G'[S]$, then $w(v)\geq 1/2$.
- If v has degree one in $G'[S]$, then $w(v)\geq 3/2$.

It is a standard exercise to show that the number of leaves in a tree (on at least two vertices) is at least $h+2$, where h is the number of vertices in the tree with degree at least three. From this fact it follows that $w(S)\geq|S|/2+2$. If $|S|\geq 4$, then $|\overline{S}|=w(S)\geq|S|/2+2\geq\frac{2}{5}|S|+\frac{12}{5}$. If $|S|=3$, then $G'[S]=P_3$. There are at least two dominators in S, so there are at least two 1-leaves in $\overline{S}$. Since every vertex in S has degree at least three (Property 1), there are at least three edges between S and leaves other than the two aforementioned 1-leaves. Since there are no 3-leaves (Property 1), we conclude that there are at least four leaves. So $|\overline{S}|\geq 4>\frac{2}{5}|S|+\frac{12}{5}$. If $|S|=1$ then since the vertex in S has degree at least three, $|\overline{S}|\geq 3$, so $|\overline{S}|\geq 3>\frac{2}{5}|S|+\frac{12}{5}$. Finally, if $|S|=2$, then $|\overline{S}|\geq 4$ (Theorem 18), so $|\overline{S}|\geq 4>\frac{2}{5}|S|+\frac{12}{5}$. So in all cases:

$$|\overline{S}|\geq\frac{2}{5}|S|+\frac{12}{5}\Leftrightarrow$$

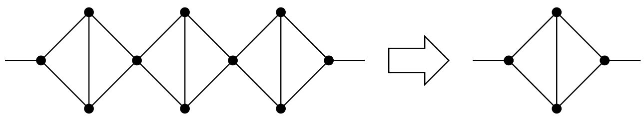

Figure 17: Reducing a necklace containing multiple diamonds

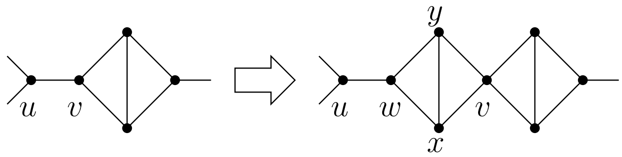

Figure 18: Reversing the necklace reduction

$$|V| - \frac{12}{5} \geq \frac{7}{5}|S| \Leftrightarrow |S| \leq \frac{5}{7}|V| - \frac{12}{7}.$$

Noting that $S' = S$ and $D = 0$ in this case, we obtain the desired inequality. □

In the bound from Theorem 21, the term D is equal to the number of cubic diamond blocks in $G'[S]$. Using Property 6 from Theorem 18, we know that for the S and G' we consider, all cubic diamond blocks are part of diamond necklaces in G, so the number of diamond necklaces is an upper bound for D. In the theorem below we show that diamond necklaces can be reduced when they consist of more than one diamond, and that therefore this term D can even be bounded by the number of cubic diamonds in G (diamond necklaces consisting of one diamond).

Theorem 22 *Let G be a connected graph on n vertices with $\delta \geq 3$. Then G has a CD-set S with $|S| \leq (5n - 12 + D)/7$, where D is the number of cubic diamonds in G that contain three vertices of S.*

Proof: First we reduce diamond necklaces that are not cubic diamonds in G: replace every diamond necklace containing at least two diamonds by a single cubic diamond as shown in Figure 17. These are called the new cubic diamonds, and the other cubic diamonds are called the original cubic diamonds. The resulting graph G_1 is again a simple connected graph with $\delta(G_1) \geq 3$.

Consider a CD-set S and subgraph G' of G_1 that satisfy the properties from Theorem 18. Let D and D' denote the number of original resp. new cubic diamonds of G_1 that are a block of $G'[S]$. Since all diamond necklaces are now cubic diamonds, the number of cubic diamond blocks in $G'[S]$ is $D + D'$ (Property 6). By Theorem 21, any minimal CD-set $S' \subseteq S$ has $|S'| \leq (5|V(G_1)| - 12 + D + D')/7$. Note that for any minimal CD-set $S' \subseteq S$, the cubic diamonds in G' that contain three vertices of S' are exactly those that are blocks in $G'[S]$.

From S' we can construct a CD-set for G: we can reconstruct G from G_1 by applying the operation illustrated in Figure 18 a number of times for every new cubic diamond. This operation introduces three new vertices. At the same time, we can maintain a CD-set by adding at most two new vertices to the CD-set every time we apply the operation: if u and v in Figure 18 are not in the previous CD-set, then we add v and x. If one of u and v is in the previous CD-set, then we add w and x. Let k be the number of times this operation has to be applied in order to reconstruct G from G_1, and construct the corresponding CD-set S''. Using $k \geq D'$, we can bound $|S''|$ as follows:

$$|S''| \leq |S'| + 2k \leq (5|V(G_1)| - 12 + D + D')/7 + 2k \leq$$

$$(5(|V(G)| - 3k) - 12 + D + k)/7 + 2k = (5|V(G)| - 12 + D)/7.$$

The reconstruction does not affect the original cubic diamonds, and also does not affect which vertices of these diamonds are in the CD-set. So D is the number of cubic diamonds in G that contain three vertices of S''. □

Note that Theorem 4 from the introduction follows immediately from Theorem 22.

6 Worst case examples for the CD-set size

In this section we show that the bounds in Theorem 13 and Theorem 22 are best possible (in the same sense as explained in Section 1). Throughout this section, n denotes the number of vertices of G.

Theorem 13 shows that every connected, triangle-free graph on n vertices with $\delta \geq 3$ has a CD-set S with $|S| \leq (2n-4)/3$. We use examples very similar to those mentioned in [8] to show that this bound is best possible: for $n = 6k$, take k copies of the graph shown in Figure 19(a), and connect them in a cyclic form as shown in Figure 19(b). The resulting graph is cubic, connected and contains no triangles. It can be checked that it has no CD-set with fewer than $\frac{2}{3}n - 2 = \lfloor (2n-4)/3 \rfloor$ vertices. This shows that for a linear bound $\alpha n - \beta$, $\alpha = \frac{2}{3}$ is best possible. When we consider the graph Q_3 on eight vertices which has no CD-set with fewer than four vertices, we see that $\beta = \frac{4}{3}$ cannot be decreased.

Theorem 22 shows that every connected graph G with $\delta \geq 3$ has a CD-set S with $|S| \leq (5n - 12 + D)/7$, where D is the number of cubic diamonds in G. To prove that this bound is best possible we use the graph N_0 in Figure 20 as a building block, together with diamonds. For any D and $n = 4D + 7k$, take D diamonds and k copies of N_0 and connect them in a cycle. The resulting graph is connected, has $\delta = 3$, and has D cubic diamonds. It can be checked that it has no CD-set with fewer than $5k + 3D - 2 = \frac{5}{7}n + \frac{1}{7}D - 2 = \lfloor (5n + D - 12)/7 \rfloor$ vertices. It follows that for a bound of the form $\alpha n + \gamma D - \beta$, only the choices $\alpha = \frac{5}{7}$ and $\gamma = \frac{1}{7}$ give an asymptotically sharp bound. Q_3 again shows that $\beta = \frac{12}{7}$ cannot be improved.

Just as for the bounds from Theorem 2 and 13, other than Q_3 we do not know any examples that show that β cannot be increased. However, for *every* $n \geq 4$ and D with $0 \leq 4D < n - 4$ we can construct examples with D cubic diamonds

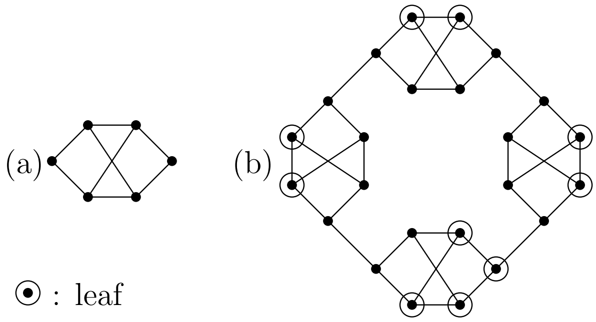

Figure 19: The bound of Theorem 13 is best possible

that do not have a CD-set with fewer than $\lfloor(5n + D)/7 - 2\rfloor$ vertices. For the construction of these examples, we take D diamonds, one copy of $N_{(n-4D) \bmod 7}$ (see Figure 20), and add copies of N_0 until we have exactly n vertices in total. These building blocks are again connected in a cycle. See Figure 21 for an example. The resulting graph has n vertices, is connected, has $\delta = 3$, has D cubic diamonds and has no CD-set with fewer than $\lfloor(5n + D)/7 - 2\rfloor$ vertices. Note that this difference of $\frac{2}{7}$ in the constant is the same difference that prevents Theorem 22 (Theorem 4) from being a direct generalization of Theorem 1 (see Section 1).

7 An algorithmic viewpoint

It can be verified that the proofs of Theorem 8 and Theorem 18 are constructive: if we start with any potential standard CD-set for the graph G (this can be simply $V(G)$), we can apply the steps mentioned in the proofs until a CD-set S is obtained that satisfies the desired properties. For Theorem 8, these steps are adding and removing edges, removing a single vertex from the CD-set and possibly one incident edge, or removing a pair of vertices from the CD-set. In fact, for both proofs it can be checked that recognizing violations of the properties and applying the corresponding improvement steps (according to the three resp. five priorities mentioned in the proofs) can be done in polynomial time.

Such algorithms can be seen as local search algorithms, where the objective value is defined by the priorities stated in the proofs, and the neighborhood of a solution is defined by exactly those steps that are applied in the proofs.

However, implementing a local search algorithm using exactly these steps is not the most practical or most general way to program algorithms for this problem that guarantee the bounds from Theorem 9 and Theorem 21. In this section, we present short, practical, and stronger algorithms that generalize the methods from these proofs. These algorithms can again be seen as local search algorithms, with the same objective value as before, but this time the

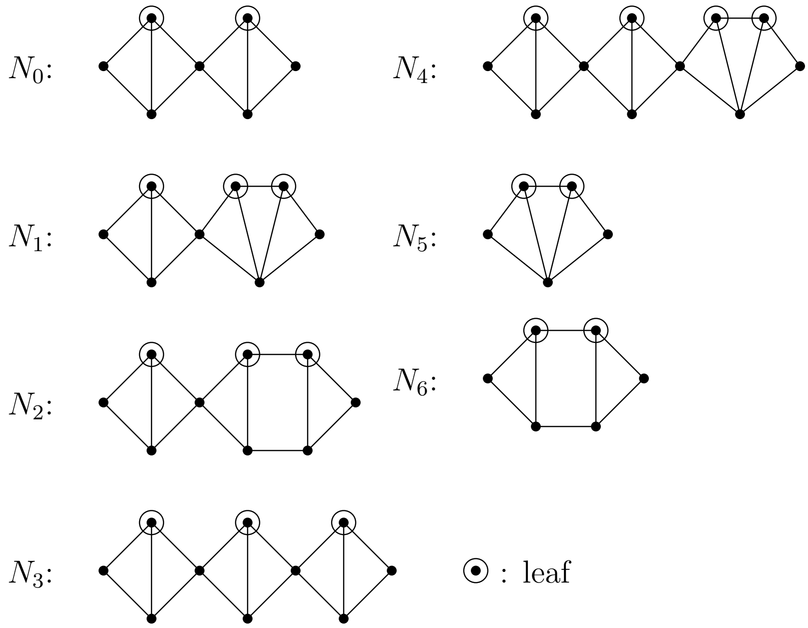

Figure 20: Building blocks for the worst case examples

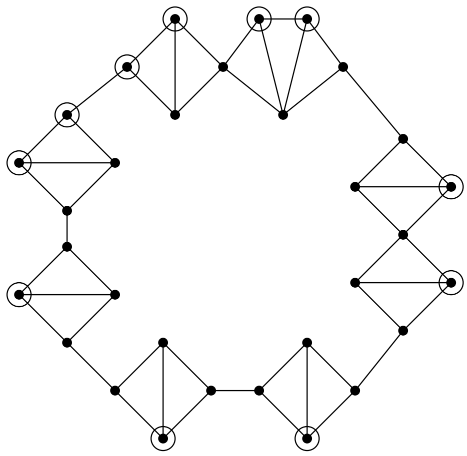

Figure 21: A worst case example with four cubic diamonds

neighborhood is defined more general (it contains the previous neighborhoods), and therefore the algorithms will give results that are at least as good, and probably better. Since the neighborhood is larger, the time complexity of the new algorithms is worse. For a practical implementation, the time complexity can be improved by making smarter choices of solutions to consider in the neighborhood, but we will not go into such details. We prefer to state the algorithms in a short and clear way, and leave the practical improvements to those that are interested.

We will consider potential standard CD-sets instead of standard CD-sets: for the first algorithm, this allows us to use a very 'linear' algorithm which only removes vertices from the CD-set until a minimal CD-set is obtained; for the second algorithm this enables us to ensure that the new neighborhood is of polynomial size. Before we can use potential standard CD-sets in the algorithm, we need the next lemma.

Lemma 23 *It can be checked in polynomial time whether $S \subseteq V(G)$ is a potential standard CD-set for G, and if so, a maximal realization G' of S can be found in polynomial time.*

Proof: We construct an auxiliary bipartite graph H with vertex set $A \cup B$: for every vertex $v \in S$, we add $d(v) - 3$ vertices to A (only vertices with degree at least four will correspond to vertices in A). For every i-leaf in $\overline{S}$ with $i > 2$, we add $i - 2$ vertices to B. We join vertices in A to vertices in B if in G edges exist between the corresponding vertices.

We show that a realization G' of S exists if and only if H has a matching M that saturates every vertex in B. If M is such a matching, delete every edge in G that corresponds to an edge of M, to obtain G'. For every former i-leaf (with $i > 2$), $i - 2$ incident edges are deleted since M saturates B, so only 1-leaves and 2-leaves remain. For every vertex in S with original degree d, at most $d - 3$ incident edges are deleted, so vertices in S still have degree at least three. Clearly, S is still a CD-set in G'.

Similarly, every maximal realization of S corresponds to a B-saturating matching in H.

The existence of polynomial time algorithms now follows from the well-known fact that polynomial time algorithms exist for finding a maximum matching (in fact, for bipartite graphs a specialized algorithm exists), and that if a B-saturating matching exists, every maximum matching saturates B. □

Algorithm 1 returns a CD-set that satisfies the bound from Theorem 13 when the input graph is triangle-free, but it is not required that the input graph is triangle-free. Using Lemma 23 we see that every step of the algorithm can be implemented in polynomial time. Since every step decreases the size of S or S', Algorithm 1 terminates in polynomial time. Note that for every graph G, $V(G)$ is a potential standard CD-set, so it is easy to find a correct input for the algorithm for every G. We prove that the output of this algorithm satisfies the bound from Theorem 13, regardless of the choice of input S.

Algorithm 1 The construction corresponding to Theorem 13

INPUT: A potential standard CD-set S for a connected graph G with $\delta(G) \geq 3$.

while there is a $U \subset S$ with $|U| \leq 2$ such that $S - U$ is a potential standard CD-set for G **do**
  $S := S - U$
end while
$S' := S$.
while there is a vertex $u \in S'$ such that $S' - u$ is a CD-set for G **do**
  $S' := S' - u$.
end while

Theorem 24 *For a connected, triangle-free graph G with $\delta(G) \geq 3$, and any potential standard CD-set S for G, Algorithm 1 returns a CD-set S' such that $|S'| \leq (2|V(G)| - 4)/3$.*

Proof: Let S and S' be the two CD-sets as they are after the algorithm has terminated. Choose a maximal realization G' for S, and in addition remove some edges from G': if $uv \in E(G')$, $\{u, v\} \subseteq S$ with $G'[S] - uv$ connected and $d_{G'}(u) \geq d_{G'}(v) \geq 4$, then delete uv from G'. Repeat this as long as such edges exist.

We prove that for these S and G', the properties from Theorem 8 are satisfied.

Property 2: If $u, v \in S$, $d(u) \geq d(v) \geq 4$ and $uv \in E(G')$, then uv is a bridge of $G'[S]$.
Proof: If uv is not a bridge, it would have been deleted from G' in the above step.

Property 3: Edges in $E(G) \backslash E(G')$ are between vertices in S, or between a vertex in S and a 2-leaf.
Proof: Let G'' be the maximal realization from which G' is obtained by deleting some edges between vertices in S. If an edge $e \in E(G) \backslash E(G')$ has no end vertices in S, or is incident with a 1-leaf, then also $e \notin E(G'')$. This is a contradiction with the fact that G'' is a maximal realization.

Property 4: Every vertex $v \in S$ that is neither a dominator nor a connector has at least three neighbors in S, and all of its neighbors in S have degree three.
Proof: Property 2 holds, so the same reasoning as in the proof of Theorem 8 applies. Note that in that proof, the only possible improvement to S that is considered consists of removing one vertex from S. This change is also considered in Algorithm 1.

Property 5: If $\{u, v\} \subseteq S$, $uv \in E(G')$ and u and v are both neither dominators nor connectors, then $G'[S] - u - v$ is not connected.

Proof: Since Property 4 holds, we can again apply the reasoning from the proof of Theorem 8. The improvement considered there consists of removing two vertices from S, which is also considered in Algorithm 1.

Since these properties hold and G contains no triangles, Theorem 9 shows that for any minimal CD-set $S^* \subseteq S$, S^* satisfies the bound. By the final step of the algorithm, S' is a minimal CD-set with $S' \subseteq S$. □

We next present an algorithm that can be used to find a CD-set S' that satisfies the bound from Theorem 22. We first point out a potential difficulty we have to cope with, and explain why we use potential standard CD-sets in the algorithm. In the proof of Property 6 from Theorem 18, the 'diamond necklace-like' structures can be arbitrarily long. So if we view a pair of a potential standard CD-set S and corresponding realization G' as a solution, there is no fixed upper bound on the number of edges that need to be changed in G' when going from one solution to the next. Therefore, without any additional rules on what edge sets to consider, we would have to consider an exponential size neighborhood. We want to avoid such additional rules since we want to state a simple, insightful formulation of the algorithm. Therefore we consider potential standard CD-sets instead. Another advantage is that this way, we consider a larger neighborhood, and therefore have a stronger algorithm. A disadvantage of using potential standard CD-sets is that for comparing two potential standard CD-sets S and S_2, we need to consider values that depend on (arbitrarily chosen) realizations. The following definition shows which values we take into account when evaluating such S and S_2.

Definition 25 *Let S and S_2 be potential standard CD-sets for graph G, and let G' resp. G'_2 be realizations for these CD-sets. We write $(S_2, G'_2) \prec (S, G')$ if*

- *$|S_2| < |S|$, or if*
- *$|S_2| = |S|$ and S_2 has fewer 1-leaves than S, or if*
- *$|S_2| = |S|$, S_2 and S have the same number of 1-leaves, and $G'_2[S_2]$ contains fewer cubic diamond blocks than $G'[S]$.*

Note that the number of 1-leaves is the same for every maximal realization. Unfortunately this is not true for the number of cubic diamond blocks. However, in the proof of Lemma 26 we show that for the final outcome, it does not matter that we choose arbitrary realizations in the algorithm.

Algorithm 2 is the main part of the algorithm we use to find a CD-set satisfying the properties from Theorem 18. Note again that input $S = V(G)$ and $G' = G$ can be chosen for any graph G. Formulated this way, the algorithm is not very fast because of the number of choices of U and W that are considered in each iteration. But we note that in every iteration, most of the possible choices of U and W that are considered are easily seen to be useless or redundant, so a large gain can be made here. See the proof of Theorem 18 for some ideas on

Algorithm 2 The construction of a CD-set corresponding to Theorem 18

INPUT: A potential standard CD-set S and a maximal realization G', for a connected graph G with $\delta(G) \geq 3$.

```
for all U ⊆ S̄ and W ⊆ S with |U| ≤ 3 and |U| ≤ |W| ≤ min{|U| + 2, 4} do
  S_2 := (S ∪ U)\W
  if S_2 is a potential standard CD-set then
    Find a maximal realization G'_2 of S_2.
    if (S_2, G'_2) ≺ (S, G') then
      Repeat the algorithm with input S_2, G'_2 and G.
    end if
  end if
end for
S' := S.
while there is a vertex u ∈ S' such that S' − u is a CD-set for G do
  S' := S' − u.
end while
```

useful choices of U and W. However, we preferred a short and clear formulation over speed, and therefore did not add additional rules for the choice of U and W.

Lemma 26 *For a connected graph G with $\delta(G) \geq 3$ and any potential standard CD-set and realization as input, Algorithm 2 finds in polynomial time a potential standard CD-set S, which has a realization G^* such that S and G^* satisfy the properties stated in Theorem 18.*

Proof: Let potential standard CD-set S and maximal realization G' be the S and G' as they are when the algorithm has terminated. In addition we consider another realization G^* for S that is obtained from G' by deleting some edges: start with $G^* = G'$. If $uv \in E(G^*)$, $\{u, v\} \subseteq S$ with $G^*[S] - uv$ connected and $d_{G^*}(u) \geq d_{G^*}(v) \geq 4$, then delete uv from G^*. Repeat this as long as such edges exist.

We prove that for the resulting S and G^*, the properties from Theorem 18 are satisfied. We omit the proof of the first properties, as they can be proved in exactly the same way as in the proof of Theorem 24.

Property 6: The number of cubic diamond blocks in $G^*[S]$ is at most the number of diamond necklaces in G.

Proof: Before we prove this property, we will prove two other statements. These are about the number of cubic diamond blocks in a realization of a given CD-set S. Observe that different (maximal) realizations of S can have a different number of cubic diamond blocks.

Claim 1: If a potential standard CD-set S has two maximal realizations G'

and G'_2 such that the number of cubic diamond blocks in $G'[S]$ and $G'_2[S]$ differs, then there is a vertex u such that $S-u$ is again a potential standard CD-set.

Since G' and G'_2 are maximal, $G'[S] = G'_2[S]$. Let D be a cubic diamond block in $G'[S]$ that is not a cubic diamond block in $G'_2[S]$. Since D is a diamond and a block in both, there is a $v \in V(D)$ with $d_{G'_2}(v) \geq 4$. Let $u \in V(D) - v$ be adjacent to all other vertices in D. Now $S - u$ is a standard CD-set for $G'_2 - uv$ (u becomes a 2-leaf, and v still has degree at least three. u is not a connector or dominator in any maximal realization).

Claim 2: Let G' and G'_2 be two realizations for S such that $G'_2 = G' - e$ for some edge e. If $G'[S]$ and $G'_2[S]$ have a different number of cubic diamond blocks, then there are vertices u or u and v such that $S - u$ resp. $S - u - v$ is again a potential standard CD-set.

First observe that since G' and G'_2 are both realizations of S, any cubic diamond block in $G'[S]$ is a cubic diamond block in $G'_2[S]$. So if the number of cubic diamond blocks differs, then there is a cubic diamond block D in $G'_2[S]$ that is not a cubic diamond block in $G'[S]$. If vertices of D are incident with e, then there is a vertex $v \in V(D)$ with $d_{G'}(v) \geq 4$. Let $u \in V(D) - v$ be adjacent to all other vertices in D. In this case, $S - u$ is a standard CD-set for $G' - uv$ (see Claim 1). Now suppose no vertices of D are incident with e. In this case, D is a cubic diamond in $G'[S]$, but not a block. Let u and v be the two vertices in D that are adjacent to all other vertices of D. $S - u - v$ is a standard CD-set for G' (u and v become 2-leaves, and since D was not a block and $d_{G'}(u) = d_{G'}(v) = 3$, $S - u - v$ is a CD-set).

Using these two claims we prove Property 6. Let S and G' be the S and G' as the are when Algorithm 2 has terminated (G' is a maximal realization for S). Let G^* again be a realization for S obtained from G' by deleting edges until Property 2 is satisfied. Suppose that in $G^*[S]$, cubic diamond blocks exist that are not part of a diamond necklace in G, or multiple cubic diamond blocks are part of the same cubic diamond necklace. We show that this leads to a contradiction with the fact that S is the output from algorithm 2.

In the proof of Theorem 18 it is shown that in this case a new potential standard CD-set $S_2 = S - x + y$ and realization G^*_2 exist such that $G^*_2[S_2]$ has fewer cubic diamonds than $G^*[S]$, and S_2 has the same number of 1-leaves. Note that even though this change from S to S_2 is described as a series of changes, the end result differs only in one vertex. In Algorithm 2, S_2 is considered. Let G'_2 be the maximal realization for S_2 that is considered in the algorithm. Finally, let G''_2 be a maximal realization of S_2 of which G^*_2 is a subgraph. (G''_2 and G'_2 do not have to be the same.) We will show that an improvement exists that is considered in the algorithm, regardless of the arbitrary choices of G' and G'_2, which is a contradiction.

Realization G^*_2 for S_2 is a (spanning) subgraph of maximal realization G''_2. Therefore if $G^*_2[S_2]$ and $G''_2[S_2]$ have a different number of cubic diamond blocks, Claim 2 shows that an improvement $S_2 - u$ or $S_2 - u - v$ exists. These can be written as $S - x + y - u$ resp. $S - x + y - u - v$, so these improvements are considered in the algorithm, a contradiction. So now we may assume that $G_2^*[S_2]$ and $G_2''[S_2]$ have the same number of cubic diamond blocks. G_2' and G_2'' are both maximal realizations of S_2. If $G_2'[S_2]$ and $G_2''[S_2]$ have a different number of cubic diamond blocks, Claim 1 shows that an improvement $S_2 - u$ exists. The set $S_2 - u = S - x + y - u$ is considered in the algorithm, a contradiction. We conclude that $G_2'[S_2]$ also has the same number of cubic diamond blocks as $G_2^*[S_2]$. Realization G^* of S is a subgraph of maximal realization G', so similar reasoning as previously shows that $G'[S]$ and $G^*[S]$ have the same number of cubic diamond blocks. Since $G_2^*[S_2]$ has fewer cubic diamonds than $G^*[S]$, $G_2'[S_2]$ has fewer cubic diamonds than $G'[S]$, and therefore the algorithm finds an improvement when S_2 and G_2' are compared with S and G', a contradiction.

In every case an improvement is found. We conclude that every cubic diamond block of $G^*[S]$ is part of a diamond necklace of G, and that every diamond necklace of G contains at most one cubic diamond block of $G^*[S]$, so Property 6 holds for S and G^*.

Property 7: The number of block-leaves is at most half the total number of leaves.
The improvement steps used in cases 1, 2.1, 2.2, 3.1-3.6 of the proof of Property 7 (Theorem 18) are all considered in the algorithm. In all of these cases, $|S|$ decreases or the number of 2-leaves increases. The number of 2-leaves in a maximal realization does not depend on the choice of the realization, so all of these changes are always accepted as an improvement by Algorithm 2. So for S and G', these cases are excluded, and the same reasoning as in the rest of the proof of this property applies.

We now use a very rough analysis to show that the algorithm can be implemented in polynomial time. The number of sets U and W that satisfy the first step is bounded by a polynomial in $|V(G)|$. Every step of the algorithm can be implemented in polynomial time (Lemma 23). Every time the algorithm is repeated, the input is 'smaller' with regard to the relation '$\prec$'. The number of possible combinations of values that are considered for evaluating this relation is bounded by a polynomial in $|V(G)|$. □

It follows that for finding the CD-set described in Theorem 22, also a simple polynomial time algorithm exists.

Theorem 27 *Let G be a connected graph with $\delta(G) \geq 3$. We can find in polynomial time a CD-set S for G with $|S| \leq (5|V(G)| - 12 + D)/7$, where D is the number of cubic diamonds in G that contain three vertices of S.*

Proof: The algorithm is as follows. First replace diamond necklaces by cubic diamonds, as described in the proof of Theorem 22. For the resulting graph G_1, use Algorithm 2 to find a potential standard CD-set S and realization G' that satisfy the properties stated in Theorem 18 (Lemma 26), and a CD-set $S' \subseteq S$. By adding vertices in the diamond necklaces of G, S' can be made into a CD-set S'' of G. See the proof of Theorem 22 for details.

We see that S' is a minimal CD-set, by the last step of Algorithm 2. So S'' satisfies the above bound (see again the proof of Theorem 22 for details). □

8 An improved FPT algorithm for MLST

A decision version of the problem of finding spanning trees with many leaves is the following, which is called the Max-Leaf Spanning Tree (MLST) problem:

INPUT: A connected graph G, and an integer k.
QUESTION: Does G have a spanning tree with at least k leaves?

(G, k) will denote an instance for this problem. This is a YES-instance if G has a spanning tree with at least k leaves, and a NO-instance otherwise. Two instances (G, k) and (G', k') are called *equivalent* if both are YES-instances or both are NO-instances.

The MLST problem is $\mathcal{NP}$-complete, but if we view k as the parameter, *fixed parameter tractable* (FPT) algorithms exist for this problem (See [5] for more information on FPT algorithms). These are algorithms that have a time complexity of $f(k)\text{poly}(n)$, where k is the chosen parameter and n is a measure of the input size, in a reasonable encoding. $\text{poly}(n)$ denotes an arbitrary polynomial in n. $f(k)$ is called the *parameter function* of the FPT algorithm, and can be any function.

There are two important ways of comparing the strength of different FPT algorithms: one is by comparing kernel sizes (see Estivill-Castro et al. [6] for an explanation), and the other is by comparing the parameter functions. Though these are often closely related, the result is not always the same: in this section we will present a new FPT algorithm that is the new best FPT algorithm when judging by the parameter function, but the FPT algorithm with the smallest kernel size is still the one presented in [6].

The MLST problem has a long history of FPT algorithms, each improving the previous parameter function. See [3] for an overview. One of the previous best FPT algorithms for the MLST problem is by Bonsma, Brueggemann and Woeginger [3], which has parameter function $f(k) \in O(9.49^k)$. We give an overview of their algorithm, and show how our new results can be used to improve this algorithm to yield the new best FPT algorithm for this problem.

A previous FPT algorithm for MLST For any graph G, $\textsc{shave}(G)$ is defined as the graph obtained by repeatedly deleting leaves from G until it is not possible anymore. Clearly, if G is a tree then $\textsc{shave}(G)$ is the empty graph and otherwise $\delta(\textsc{shave}(G)) \geq 2$. Now the vertices of G can be partitioned into 3 categories:

- The *branch vertices* are the vertices that do not appear in $\textsc{shave}(G)$.

- The *path vertices* are the vertices with degree 2 in $\textsc{shave}(G)$.
- The *junction vertices* are the vertices with degree at least 3 in $\textsc{shave}(G)$.

Using this categorization of vertices, we can define a *backbone graph* $\textsc{backbone}(G) = (V', E')$ for every graph G that has a non-empty set of junction vertices: V' is the set of junction vertices of G. For every path in G between two junction vertices u and v, in which all other vertices are path vertices, an edge is added between u and v in $\textsc{backbone}(G)$. Note that $\textsc{backbone}(G)$ can contain multi-edges. For every cycle containing exactly one junction vertex u, we add a loop to u in $\textsc{backbone}(G)$. It follows that the vertex degrees in $\textsc{backbone}(G)$ are equal to the degrees in G.

In [3] a set of *reduction rules* is given that can be used to transform an instance (G, k) into an equivalent instance (G', k'), such that either G' has no junction vertices, or $\textsc{backbone}(G')$ is a simple graph. In addition, $k' \leq k$ and some other properties are true which are needed for the next phase of their algorithm. This transformation can be done in polynomial time.

It is easy to see that if $\textsc{backbone}(G')$ has a spanning tree with l leaves, then G' has a spanning tree with at least l leaves. Since $\textsc{backbone}(G')$ is simple and has $\delta \geq 3$, Theorem 1 shows that G' has a spanning tree with at least $n/4 + 2$ leaves, where n is the number of junction vertices in G'. If $n/4 + 2 \geq k'$, then we know (G', k') is a YES-instance, and therefore (G, k) is a YES-instance.

Otherwise, in [3] an *enumerative procedure* is described that that decide in $O(9.49^k)$ time whether (G', k') is a YES-instance. This procedure enumerates all subsets of size at most k' of $V(\textsc{backbone}(G'))$ and computes in polynomial time the maximum number of leaves in a spanning tree that contains these vertices as leaves. Since $n < 4k'$ and $k' \leq k$, the number of these subsets can be bounded by a function of k. See the proof of Theorem 28 for details on how this leads to $O(9.49^k)$, since the time complexity of the new algorithm is deduced in a very similar manner. The complexity of the other part of the algorithm is independent of k (and polynomial in the input size), so it follows that the parameter function of the algorithm is $O(9.49^k)$.

An improved FPT algorithm We can combine our extremal result with the techniques from [3] to improve the parameter function of the FPT algorithm. The only addition to the algorithm is a preprocessing step that removes cubic diamonds. A very similar preprocessing step was previously mentioned in [6]. Algorithm 3 is the new FPT algorithm.

In step 4, if G' has no junction vertices, then either G' is a tree or $\textsc{shave}(G')$ is a cycle. In both cases it can easily be checked if (G', k') is a YES-instance.

Theorem 28 *Algorithm 2 is an FPT algorithm for the MLST problem with parameter function* $f(k) \in O(8.12^k)$.

Proof: We prove that the output of the algorithm is correct. First we show that the operation used in step 3 yields an instance equivalent with the previous instance.

Algorithm 3 the new FPT algorithm

INPUT: a connected graph G and parameter k.

1. $G' := G$, $k' = k$.
2. While this is possible, apply the reduction rules from [3] to the instance (G', k').
3. If G' contains a cubic diamond D between u and v, then delete the other two vertices of D and add edge uv (see the top of Figure 22). $k' := k' - 1$. Go to step 2.
4. If G' has no junction vertices, then use a simple polynomial time algorithm to find the answer, stop. (See below for details.)
5. If $k' \leq (2|V(\text{BACKBONE}(G'))| + 12)/7$, then output 'YES', stop.
6. Otherwise, use the enumerative procedure from [3] to decide whether G' has a spanning tree with at least k' leaves.

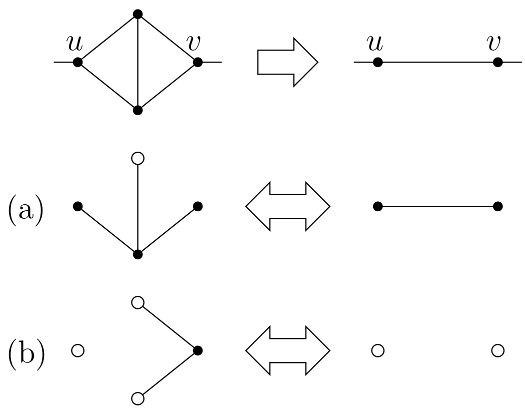

Figure 22: Removing a cubic diamond, and the corresponding spanning trees

Let G_2 be obtained from G_1 by replacing a cubic diamond D between u and v with edge uv. Let T_1 be a spanning tree of G_1 with maximum number of leaves. k is the number of leaves of T_1. We may assume w.l.o.g. that $E(T_1) \cap E(D)$ contains the either the edges shown in Figure 22(a) or those shown in Figure 22(b) (we can always replace $E(T_1) \cap E(D)$ by one of these two sets without decreasing the number of leaves, introducing cycles or destroying the connectivity). In the first case, we replace these edges of T_1 by a single edge uv, to obtain a spanning tree T_2 of G_2 with $k-1$ leaves. (If u or v previously was a leaf, it is again a leaf.) In the second case, we remove these edges to obtain a spanning tree T_2 of G_2 with $k-1$ leaves. Similarly, if a spanning tree of G_2 contains uv, we replace this by the edge set in Figure 22(a), otherwise we add the edge set in Figure 22(b). In both cases this yields a spanning tree of G_1 with one more leaf. This shows that (G_1, k) and $(G_2, k-1)$ are equivalent instances. Observe also that this operation yields again a simple, connected graph.

Now we will show that if in step 5 a 'YES' answer is given, this is correct. Since step 2 is only completed when either no junction vertices remain or $\textsc{backbone}(G')$ is simple and has $\delta \geq 3$, step 5 is only entered when $\textsc{backbone}(G')$ is simple, has $\delta \geq 3$ and G' contains no cubic diamonds. In addition, at this point the instance (G', k') is still equivalent with the original instance (G, k). Let (G', k') be the instance as it is when the algorithm is finished. Then we know that every cubic diamond in $\textsc{backbone}(G')$ does not correspond to a cubic diamond in G'. More formally this can be described as follows: G' can be constructed from $\textsc{backbone}(G')$ by applying edge subdivisions and adding leaves (introducing a vertex and connecting it to an existing vertex). If D is a cubic diamond in $\textsc{backbone}(G')$, then in this construction of G', at least one edge of D is subdivided, or at least one leaf is added adjacent to a vertex of D.

Let T_1 be a spanning tree of G_1, and let D be a cubic diamond of G_1 that contains exactly one leaf of T_1. Apply an edge subdivision on an edge of D, and call the resulting graph G_2, and the subgraph corresponding to D is called D'. Now we can gain one leaf: remove the edges of D from $E(T_1)$, and add the appropriate edge set from Figure 23 or a symmetric edge set to $E(T_1)$. Note that since D is cubic, we can choose to use a different leaf than the original leaf in D. The resulting graph T_2 is is a spanning tree for G_2 with one more leaf. Now let G_2 be obtained from G_1 by adding a leaf adjacent to one of the vertices of D. In this case, we can also gain a leaf: in T_1, replace the edges of D with one of the edge sets shown in Figure 24 (or a symmetric edge set)[1].

This allows us to prove that if $k' \leq (2|V(\textsc{backbone}(G'))| + 12)/7$, the instance is a YES-instance. Let $n = |V(\textsc{backbone}(G'))|$. $\textsc{backbone}(G')$ has a spanning tree T with at least $n - (5n - 12 + d)/7 = (2n + 12 - d)/7$ leaves (Theorem 22), where d is the number of cubic diamonds that contain only one leaf of T. We construct G' from $\textsc{backbone}(G')$ by applying edge subdivisions and leaf additions. At every step we maintain a spanning tree T: if we subdivide an edge that is part of a cubic diamond D, or add a leaf adjacent to a vertex in a

[1] We need to assume that D contains only one leaf of T: otherwise there are cases where no leaf can be gained after an edge subdivision or leaf addition. This is why we chose to formulate Theorem 22 this way.

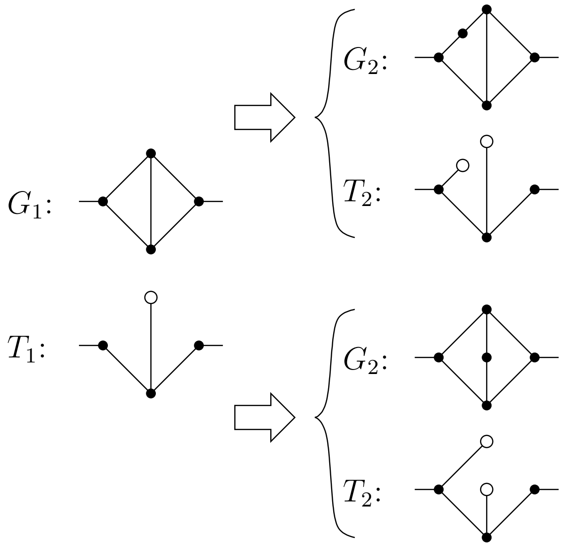

Figure 23: Changing the spanning tree when a diamond edge is subdivided

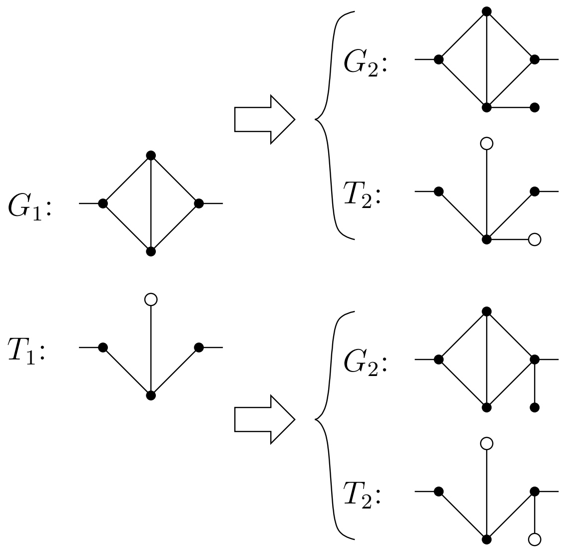

Figure 24: Changing the spanning tree when a leaf is added to a diamond

cubic diamond D, and D contains one leaf of T, we change T as described above. We gain one leaf in this case. Otherwise, we extend T in the straightforward way, which does not decrease the number of leaves. We observed that on every cubic diamond of $\textsc{backbone}(G')$, at least one of these operations is applied. Therefore when G' is obtained, at least d leaves are gained. It follows that the constructed spanning tree of G' has at least $(2n+12-d)/7+d \geq (2n+12)/7$ leaves, so the instance is a YES-instance if $k' \leq (2n+12)/7$.

Once step 6 of the algorithm is entered, none of the reduction rules from [3] (those that are applied in step 2) can be applied to the instance (G', k'). This guarantees that the other properties that are necessary for the correct execution of step 6 (that we will not mention here) are true for (G', k'). This concludes the proof of correctness of Algorithm 2.

Now we consider the time complexity of the algorithm. Step 1–5 of the algorithm can be implemented in polynomial time (polynomial in the input size): our new reduction rule reduces the number of edges in G', so this rule fits the reduction rule criteria from [3], thus their proof of polynomial time also holds when this rule is included.

If the last step of the algorithm is used, then $k' \geq (2n+12)/7$, so $n \leq \frac{7}{2}k'$, where $n = |V(\textsc{backbone}(G'))|$ again. For this step we need to use the expensive enumerative algorithm from [3]. This algorithm checks all subsets of size at most k' of the vertices of $\textsc{backbone}(G')$ (these are the leaf candidates). For every such subset, a polynomial time algorithm is applied (polynomial in n, so also in k'). The number of subsets is $\binom{n}{k'} + \binom{n}{k'-1} + \ldots + \binom{n}{0} \leq \binom{\frac{7}{2}k'}{k'} + \binom{\frac{7}{2}k'}{k'-1} + \ldots + \binom{\frac{7}{2}k'}{0} \leq k'\binom{\frac{7}{2}k'}{k'} + 1$. Therefore there is a polynomial $g(k')$ such that the complexity of the last step is $O(g(k')\binom{\frac{7}{2}k'}{k'})$. In addition, $k' \leq k$ (our reduction rule and the reduction rules from [3] do not increase k'). Now we use Stirling's approximation $x! \approx x^x e^{-x}\sqrt{2\pi x}$, or alternatively, $x! \in \Theta(x^x e^{-x}\sqrt{x})$. This gives

$$\binom{\frac{7}{2}k'}{k'} \leq \binom{\frac{7}{2}k}{k} = \frac{(3.5k)!}{(2.5k)!k!} \in$$

$$O\left(\frac{(3.5k)^{3.5k}\sqrt{3.5k}}{e^{3.5k}} \cdot \frac{e^{2.5k}}{(2.5k)^{2.5k}\sqrt{2.5k}} \cdot \frac{e^k}{k^k\sqrt{k}}\right) = O\left(\frac{3.5^{3.5k}}{2.5^{2.5k}\sqrt{k}}\right) \subset$$

$$O\left(\left(\frac{80.2118}{9.8821}\right)^k\right) \subset O(8.117^k).$$

For any polynomial $g(k)$, we have $g(k)8.117^k \in O(8.12^k)$, so we conclude that the parameter function of this algorithm is $f(k) \in O(8.12^k)$. □

9 Discussion

In this paper, we introduced new techniques to find small CD-sets/spanning trees with many leaves. We feel that our methods show that considering small

CD-sets is more practical for this purpose than looking at leafy spanning trees, which was previously done to deduce bounds of this kind. For instance, the concept of minimality of a CD-set proves to be very useful, but no similar concept exists for spanning trees. We have also shown that there are simple but powerful ways of doing local search on minimal CD-sets, which again have no easy and equally powerful equivalent in the world of leafy trees. To put it differently, we feel that spanning trees contain too much information for our purposes: it does not matter which leaves are connected to which CD-set vertices, or which tree is chosen to connect the CD-set vertices.

These techniques may be useful to prove other similar results: for instance Linial [11] conjectured that every graph on n vertices with minimum degree δ has a spanning tree with at least $n-3n/(\delta+1)+c_\delta$ leaves, for some appropriate constant c_δ. For sufficiently large δ, this conjecture has been disproved by Alon [1], but for small values of δ the conjecture has been shown to be true. For $\delta = 3$, Theorem 1 proves the conjecture. For $\delta = 4$, a proof appears in [10]. For $\delta = 5$, Griggs and Wu [9] gave a proof. For $\delta \geq 6$, little is known. For more information, see [4].

The worst case examples for the bound from Theorem 22 presented in Section 6 still contain many diamonds: we expect that if we forbid all diamonds (not just cubic diamonds), every graph in this class has a CD-set with at most $\frac{2}{3}n - \frac{4}{3}$ vertices. So we expect that Theorem 2 can be generalized to all graphs with $\delta \geq 3$. Even though proving this will be hard, we think that the local search algorithm presented in Section 7, or a similar algorithm that considers a slightly larger neighborhood, might attain this bound. If this generalization can be proved, this would allow a further improvement of the parameter function of the FPT algorithm for the MLST problem to $O(6.75^k)$.

We mentioned that for our two bounds (Theorem 13 and 22) and the bound in Theorem 2, Q_3 is the only known example that prevents us to find a stronger linear bound. If we exclude this graph and a few other small graphs (in [8], $K_2 \times C_5$ and another graph on ten vertices are mentioned), it may be possible to show that the bounds become $|S'| \leq \frac{2}{3}|V| - 2$ resp. $|S'| \leq \frac{5}{7}|V| + \frac{1}{7}D - 2$. We have shown in Section 6 that the second bound cannot be improved further unless we exclude infinitely many graphs. For the other bound, similar examples can be constructed that show the same. Again we do not expect an easy proof for these improved bounds.

We remark that Algorithm 1 mentioned in Section 7 does not attain this improved bound: if we consider $K_2 \times K_3$, we see that Algorithm 1 may output a CD-set on three vertices (this is a minimal CD-set). Observe that if we add a local search step as used in Algorithm 2, we do find a good CD-set in this case (on two vertices). We expect that Algorithm 2 always finds CD-sets that satisfy the improved bounds.

References

[1] N. Alon. Transversal numbers of uniform hypergraphs. *Graphs Combin.*, 6(1):1–4, 1990.

[2] J. A. Bondy and U. S. R. Murty. *Graph theory with applications.* American Elsevier Publishing Co., Inc., New York, 1976.

[3] P. S. Bonsma, T. Brueggemann, and G. J. Woeginger. A faster FPT algorithm for finding spanning trees with many leaves. In *Mathematical foundations of computer science 2003*, volume 2747 of *Lecture Notes in Comput. Sci.*, pages 259–268. Springer, Berlin, 2003.

[4] Y. Caro, D. B. West, and R. Yuster. Connected domination and spanning trees with many leaves. *SIAM J. Discrete Math.*, 13(2):202–211 (electronic), 2000.

[5] R. G. Downey and M. R. Fellows. *Parameterized complexity.* Monographs in Computer Science. Springer-Verlag, New York, 1999.

[6] V. Estivill-Castro, M. R. Fellows, M. A. Langston, and F. A. Rosamond. FPT is P-time extremal structure I. In *Algorithms and complexity in Durham 2005*, volume 4 of *Texts in algorithmics*, pages 1–41, London, 2005. King's College Publications.

[7] M. R. Garey and D. S. Johnson. *Computers and intractability.* W. H. Freeman and Co., San Francisco, Calif., 1979.

[8] J. R. Griggs, D. J. Kleitman, and A. Shastri. Spanning trees with many leaves in cubic graphs. *J. Graph Theory*, 13(6):669–695, 1989.

[9] J. R. Griggs and M. Wu. Spanning trees in graphs of minimum degree 4 or 5. *Discrete Math.*, 104(2):167–183, 1992.

[10] D. J. Kleitman and D. B. West. Spanning trees with many leaves. *SIAM J. Discrete Math.*, 4(1):99–106, 1991.

[11] N. Linial and D. Sturtevant. unpublished, 1987.