

Recognition of Linear and Star Variants of Leaf Powers is in P

Benjamin Bergounoux ✉

University of Bergen, Norway

Svein Høgemo ✉

University of Bergen, Norway

Jan Arne Telle ✉

University of Bergen, Norway

Martin Vatshelle ✉

University of Bergen, Norway

Abstract

A k -leaf power of a tree T is a graph G whose vertices are the leaves of T and whose edges connect pairs of leaves whose distance in T is at most k . A graph is a leaf power if it is a k -leaf power for some k . Over 20 years ago, Nishimura et al. [J. Algorithms, 2002] asked if recognition of leaf powers was in P. Recently, Lafond [SODA 2022] showed an XP algorithm when parameterized by k , while leaving the main question open. In this paper, we explore this question from the perspective of two alternative models of leaf powers, showing that both a linear and a star variant of leaf powers can be recognized in polynomial-time.

2012 ACM Subject Classification Mathematics of computing → Graph algorithms

Keywords and phrases Leaf power Co-threshold tolerance graphs Interval graphs

1 Introduction

Leaf powers were introduced by Nishimura et al. in [23], and have enjoyed a steady stream of research. Leaf powers are related to the problem of reconstructing *phylogenetic trees*. For an integer k , a graph G is a k -leaf power if there exists a tree T – called a *leaf root* – with a one-to-one correspondence between $V(G)$ and the leaves of T , such that two vertices u and v are neighbors in G iff the distance between the two corresponding leaves in T is at most k . G is a leaf power if it is a k -leaf power for some k . The most important open problem in the field is whether leaf powers can be recognized in polynomial time.

Most of the results on leaf powers have followed two main lines, focusing either on the distance values k or on the relation of leaf powers to other graph classes, see e.g. the survey by Calamoneri et al. [7]. For the first approach, steady research for increasing values of k has shown that k -leaf powers for any $k \leq 6$ is recognizable in polytime [4, 5, 8, 11, 12, 23]. Moreover, the recognition of k -leaf powers is known to be FPT parameterized by k and the degeneracy of the graph [13]. Recently Lafond [19] gave a polynomial time algorithm to recognize k -leaf powers for any constant value of k . For the second approach, we can mention that interval graphs [5] and rooted directed path graphs [3] are leaf powers, and also that leaf powers have mim-width one [17] and are strongly chordal. Moreover, an infinite family of strongly chordal graphs that are not leaf powers has been identified [18]; see also Nevrís and Rosenke [22].

To decide if leaf powers are recognizable in polynomial time, it may be better not to focus on the distance values k . Firstly, the specialized algorithms for k -leaf powers for $k \leq 6$ do not seem to generalize. Secondly, the recent XP algorithm of Lafond [19] uses techniques that will not allow removing k from the exponent. In this paper we therefore take a different approach, and consider alternative models for leaf powers that do not rely on a distance bound. In order

to make progress towards settling the main question, we consider fundamental restrictions on the shape of the trees, in two distinct directions: subdivided caterpillars (linear) and subdivided stars, in both cases showing polynomial-time recognizability. We use two models: weighted leaf roots for the linear case and NeS models for the stars.

The first model uses rational edge weights between 0 and 1 in the tree T which allows to fix a bound of 1 for the tree distance. It is not hard to see that this coincides with the standard definition of leaf powers using an unweighted tree T and a bound k on distance. Given a solution of the latter type we simply set all edge weights to $1/k$, while in the other direction we let k be the least common denominator of all edge weights and then subdivide each edge a number of times equal to its weight times k .

The second model arises by combining the result of Brandstädt et al. that leaf powers are exactly the fixed tolerance NeST graphs [3, Theorem 4] with the result of Bibelnieks et al. [1, Theorem 3.3] that these latter graphs are exactly those that admit what they call a “neighborhood subtree intersection representation”, that we choose to call a NeS model. NeS models are a generalization of interval models: by considering intervals of the line as having a center that stretches uniformly in both directions, we can generalize the line to a tree embedded in the plane, and the intervals to embedded subtrees with a center, stretching uniformly in all directions from the center, along tree edges. Thus a NeS model of a graph G consists of an embedded tree and one such subtree for each vertex, such that two vertices are adjacent in G iff their subtrees have non-empty intersection. Precise definitions are given later. The leaf powers are exactly the graphs having a NeS model. Some results are much easier to prove using NeS models, to illustrate this, we show that leaf powers are closed under several operations such as the addition of a universal vertex (see Lemma 6).

We show that fundamental constraints on these models allow polynomial-time recognition. Using the first model, we restrict to edge-weighted caterpillars, i.e. trees with a path containing all nodes of degree 2 or more. We call linear leaf power a graph with such model. Brandstädt et al. [2] considered leaf roots restricted to caterpillars (see also [6]) in the unweighted setting, showing that unit interval graphs are exactly the k -leaf powers for some k with a leaf root being an unweighted caterpillar. In the unweighted setting, linear leaf powers are graphs with a leaf root that is a *subdivision* of a caterpillar. We show that linear leaf powers are exactly the *co-threshold tolerance graphs* [20], and combined with the algorithm of Golovach et al. [14] this implies that we can recognize linear leaf powers in $O(n^2)$ time. Our proof goes via the equivalent concept of blue-red interval graphs introduced by [15], see Figure 3.

The recognition of linear leaf powers in polynomial time could have practical applications for deciding whether the most-likely evolutionary tree associated with a set of organisms has a linear topology. Answering this question might find particular relevance inside the field of Tumor phylogenetics where, under certain model assumptions, linear topologies are considered more likely [9, 25].

For NeS models, we restrict to graphs having a NeS model where the embedding tree is a star, and show that they can be recognized in polynomial time. Note that allowing the embedding tree to be a subdivided star will result in the same class of graphs. Our algorithm uses the fact that the input graph must be a chordal graph, and for each maximal clique X we check if G admits a star NeS model where the set of vertices having a subtree containing the central vertex of the star is X . To check this we use a combinatorial characterization, that we call a “good partition”, of a star NeS model.

2 Preliminaries

For positive integer k , denote by $[k]$ the set $\{1, 2, \dots, k\}$. A partition of a set S is a collection of non-empty disjoint subsets B_1, \dots, B_t of S – called *blocks* – such that $S = B_1 \cup \dots \cup B_t$. Given two partitions \mathcal{A}, \mathcal{B} of S , we say $\mathcal{A} \sqsubseteq \mathcal{B}$ if every block of \mathcal{A} is included in a block of \mathcal{B} , i.e. \sqsubseteq is the *refinement* relation.

Graph. Our graph terminology is standard and we refer to [10]. The vertex set of a graph G is denoted by $V(G)$ and its edge set by $E(G)$. An edge between two vertices x and y is denoted by xy or yx . The set of vertices that is adjacent to x is denoted by $N(x)$. A vertex x is simplicial if $N(x)$ is a clique. Two vertices x, y are true twins if $xy \in E(G)$ and $N(x) \setminus \{y\} = N(y) \setminus \{x\}$. Given $X \subseteq V(G)$, we denote by $G[X]$ the graph induced by X . Given a vertex $v \in V(G)$, we denote the subgraph $G[V(G) \setminus \{v\}]$ by $G - v$. We denote by $\text{CC}(G)$ the partition of G into its connected components. Given a tree T and an edge-weight function $w : E(T) \rightarrow \mathbb{Q}$, the distance between two vertices x and y is denoted by $d_T(x, y)$ is $\sum_{e \in E(P)} w(e)$ with P is the unique path between x and y .

Leaf power. In the Introduction we have already given the standard definition of leaf powers and leaf roots, and also we argued the equivalence with the following. Given a graph G , a leaf root of G is a pair (T, w) of a tree T and a rational-valued weight function $w : E(T) \rightarrow [0, 1]$ such that the vertices of G are the leaves of T and for every $u, v \in V(G)$, u and v are adjacent iff $d_T(u, v) \leq 1$. Moreover, if T is a caterpillar we call (T, w) a *linear leaf root*. A graph is a *leaf power* if it admits a leaf root and it is a *linear leaf power* if it admits a linear leaf root. Since we manipulate both the graphs and the trees representing them, the vertices of trees will be called *nodes* to avoid confusion.

Interval graphs. A graph G is an *interval graph* if there exists a set of intervals in \mathbb{Q} , $\mathcal{I} = (I_v)_{v \in V(G)}$, such that for every pair of vertices $u, v \in V(G)$, the intervals I_u and I_v intersect iff $uv \in E(G)$. We call $(I_v)_{v \in V(G)}$ an *interval model* of G . For an interval $I = [\ell, r]$, we define the midpoint of I as $(\ell + r)/2$ and its length as $r - \ell$.

Clique tree. For a chordal graph G , a clique tree CT of G is a tree whose vertices are the maximal cliques of $V(G)$ and for every vertex $v \in V(G)$, the set of maximal cliques of G containing v induces a subtree of CT . Figure 4 gives an example of clique tree. Every chordal graph admits $O(n)$ maximal cliques and given a graph G , in time $O(n + m)$ we can construct a clique tree of G or confirm that G is not chordal [16, 24]. When a clique tree is a path, we call it a clique path. We denote by (K_1, \dots, K_k) the clique path whose vertices are K_1, \dots, K_k and where K_i is adjacent to K_{i+1} for every $i \in [k - 1]$.

3 Linear leaf powers

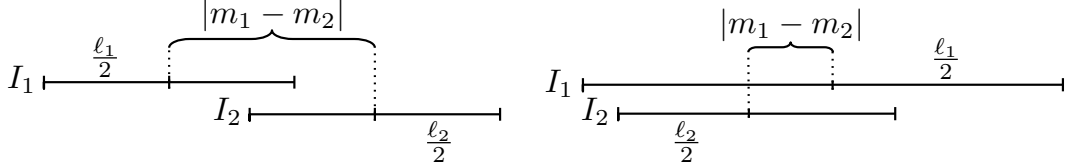
In this section we show that linear leaf powers are exactly the *co-threshold tolerance graphs* (co-TT graphs). Combined with the algorithm in [14], this implies that we can recognize linear leaf powers in $O(n^2)$ time.

Co-TT graphs were defined by Monma, Trotter and Reed in [20]; we will not define them here as we do not use this characterization. Rather, we work with the equivalent class of *blue-red interval graphs* [15, Proposition 3.3].

► **Definition 1** (Blue-red interval graph). *A graph G is a blue-red interval graph if there exists a bipartition (B, R) of $V(G)$ and an interval model $\mathcal{I} = (I_v)_{v \in V(G)}$ (with (B, R, \mathcal{I}) called a blue-red interval model) such that $E(G) = \{b_1 b_2 : b_1, b_2 \in B \text{ and } I_{b_1} \cap I_{b_2} \neq \emptyset\} \cup \{rb : r \in R, b \in B \text{ and } I_r \subseteq I_b\}$.*

The red vertices induce an independent set, $(I_b)_{b \in B}$ is an interval model of $G[B]$, and we have a blue-red edge for each red interval contained in a blue interval. The following fact can be easily deduced from Figure 1.

► **Fact 2.** Consider two intervals I_1, I_2 with lengths ℓ_1, ℓ_2 and midpoints m_1, m_2 respectively. We have $I_1 \cap I_2 \neq \emptyset$ iff $|m_1 - m_2| \leq \frac{\ell_1 + \ell_2}{2}$. Moreover, we have $I_2 \subseteq I_1$ iff $|m_1 - m_2| \leq \frac{\ell_1 - \ell_2}{2}$.



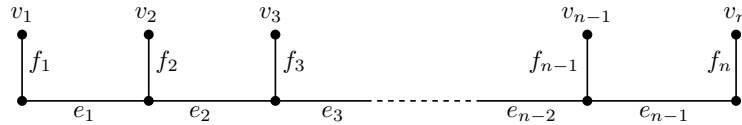
■ **Figure 1** Example of two intervals overlapping and one interval containing another one.

To prove that linear leaf powers are exactly blue-red interval graphs, we use a similar construction as the one used in [2, Theorem 6] to prove that every interval graph is a leaf power, but in our setting, we have to deal with red vertices and this complicates things quite a bit.

► **Theorem 3 (★).** G is a blue-red interval graph iff G is a linear leaf power.

Proof. (\Rightarrow) Let G be a blue-red interval graph with blue-red interval model $(B, R, (I_v)_{v \in V(G)})$. We assume w.l.o.g. that G is connected as otherwise we can obtain a leaf root of G from leaf roots of its connected components by creating a new node adjacent to an internal node of each leaf root via edges of weight 1. For every $v \in V(G)$, we denote by $\ell(v)$ and $m(v)$ the length and the midpoint of the interval I_v . We suppose w.l.o.g. that, for every $v \in V(G)$, we have $0 < \ell(v) \leq 1$ as we can always divide the endpoints of all intervals by $\max\{\ell(v) : v \in V(G)\}$ and add some $\epsilon > 0$ to the right endpoints of the intervals of length 0. We fix an ordering on $V(G)$, $\{v_1, \dots, v_n\}$, such that $m(v_i) \leq m(v_j)$ for every $i < j$.

We define T a caterpillar and its edges $f_1, \dots, f_n, e_1, \dots, e_{n-1}$ as depicted in Figure 2. Let $w : E(T) \rightarrow [0, 1]$ such that for every $i \in [n-1]$ we have $w(e_i) = m(v_{i+1}) - m(v_i)$ and for every $i \in [n]$ the weight of f_i is $\frac{1-\ell(v_i)}{2}$ if $v_i \in B$ and $\frac{1+\ell(v_i)}{2}$ if $v_i \in R$. Since we assume that G is connected and the length of the intervals are at most 1, for every $i \in [n-1]$, we have $w(e_i) = m(v_{i+1}) - m(v_i) \leq 1$ so the weights are well-defined.



■ **Figure 2** The linear branch-decomposition used to prove that every blue-red interval graph is a linear leaf power. Observe that the weight of an edge e_i can be 0 if I_{v_i} and $I_{v_{i+1}}$ have the same midpoint, in that case, e_i can be contracted.

We claim that (T, w) is a linear leaf root of G . Let $i, j \in [n]$ such that $i < j$. We have to prove that $v_i v_j \in E(G)$ iff $d_T(v_i, v_j) \leq 1$. The edges of the path between v_i and v_j in T are $f_i, e_i, e_{i+1}, \dots, e_{j-1}, f_j$. By construction, we have

$$w(e_i) + \dots + w(e_{j-1}) = m_{i+1} - m_i + m_{i+2} - m_{i+1} + \dots + m_{j-1} - m_{j-2} + m_j - m_{j-1} = m_j - m_i.$$

Hence $d_T(v_i, v_j) = w(f_i) + |m_j - m_i| + w(f_j)$. If v_i and v_j are red vertices, then $w(f_i), w(f_j) > \frac{1}{2}$ because we suppose that $\ell(x) > 0$ for every vertex x . Thus, $d_T(v_i, v_j) > 1$ for every pair of

red vertices (v_i, v_j) . So at least one vertex among v_i, v_j is blue. Suppose w.l.o.g. that v_i is blue. As $d_T(v_i, v_j) = w(f_i) + |m_j - m_i| + w(f_j)$, we deduce that

$$d_T(v_i, v_j) = \begin{cases} 1 + |m(v_i) - m(v_j)| - \left(\frac{\ell(v_i) + \ell(v_j)}{2}\right) & \text{if } v_j \in B \\ 1 + |m(v_i) - m(u_i)| - \left(\frac{\ell(v_i) - \ell(v_j)}{2}\right) & \text{if } v_j \in R. \end{cases} \quad (1)$$

If $v_j \in B$ then by Fact 2, we have $I_{v_i} \cap I_{v_j} \neq \emptyset$ iff $|m(v_i) - m(v_j)| \leq \frac{\ell(v_i) + \ell(v_j)}{2}$. We conclude that $v_i v_j \in E(G)$ iff $d_T(v_i, v_j) = |m(v_i) - m(v_j)| - \left(\frac{\ell(v_i) + \ell(v_j)}{2}\right) + 1 \leq 1$. On the other hand, if $v_j \in R$ then $I_{v_j} \subseteq I_{v_i}$ iff $|m(v_i) - m(v_j)| \leq \frac{\ell(v_i) - \ell(v_j)}{2}$. We conclude that $v_i v_j \in E(G)$ iff $d_T(v_i, v_j) = |m(v_i) - m(v_j)| - \left(\frac{\ell(v_i) - \ell(v_j)}{2}\right) + 1 \leq 1$. Hence, (T, w) is a linear leaf root of G . This proves that every blue-red interval graph is a linear leaf power.

(\Leftarrow) Let G be a linear leaf power and (T, w) a linear leaf root of G . Let (u_1, \dots, u_t) be the path induced by the internal vertices of T . We suppose w.l.o.g. that G does not contain isolated vertices as we can easily deal with such vertices by associating each of them with an interval that does not intersect the other intervals. Consequently, for every leaf a of T adjacent to some u_i , we have $d_T(a, u_i) \leq 1$. For every vertex $v \in V(G)$ whose neighbor in T is u_i , we associate v with an interval I_v and a color such that the midpoint of I_v is $m(v) = d_T(u_1, u_i)$ and

- if $w(u_i v) \leq \frac{1}{2}$, the length of I_v is $\ell(v) = 1 - 2w(u_i v)$ and the color of v is blue,
- otherwise ($\frac{1}{2} < w(u_i v) \leq 1$), the length of I_v is $\ell(v) = 2w(u_i v) - 1$ and the color of v is red.

Let $B, R \subseteq V(G)$ be the sets of blue and red vertices respectively. Observe that the red vertices induce an independent set since their distance to the inner path is strictly more than $1/2$. By construction, we have the following equation for every vertex v whose neighbor in T is u_i

$$w(u_i v) = \begin{cases} \frac{1 - \ell(v)}{2} & \text{if } v \in B, \\ \frac{1 + \ell(v)}{2} & \text{if } v \in R. \end{cases} \quad (2)$$

Moreover, from the definition of the midpoint's, we deduce that for every $v_i, v_j \in V(G)$ whose neighbors in T are respectively u_i and u_j , we have

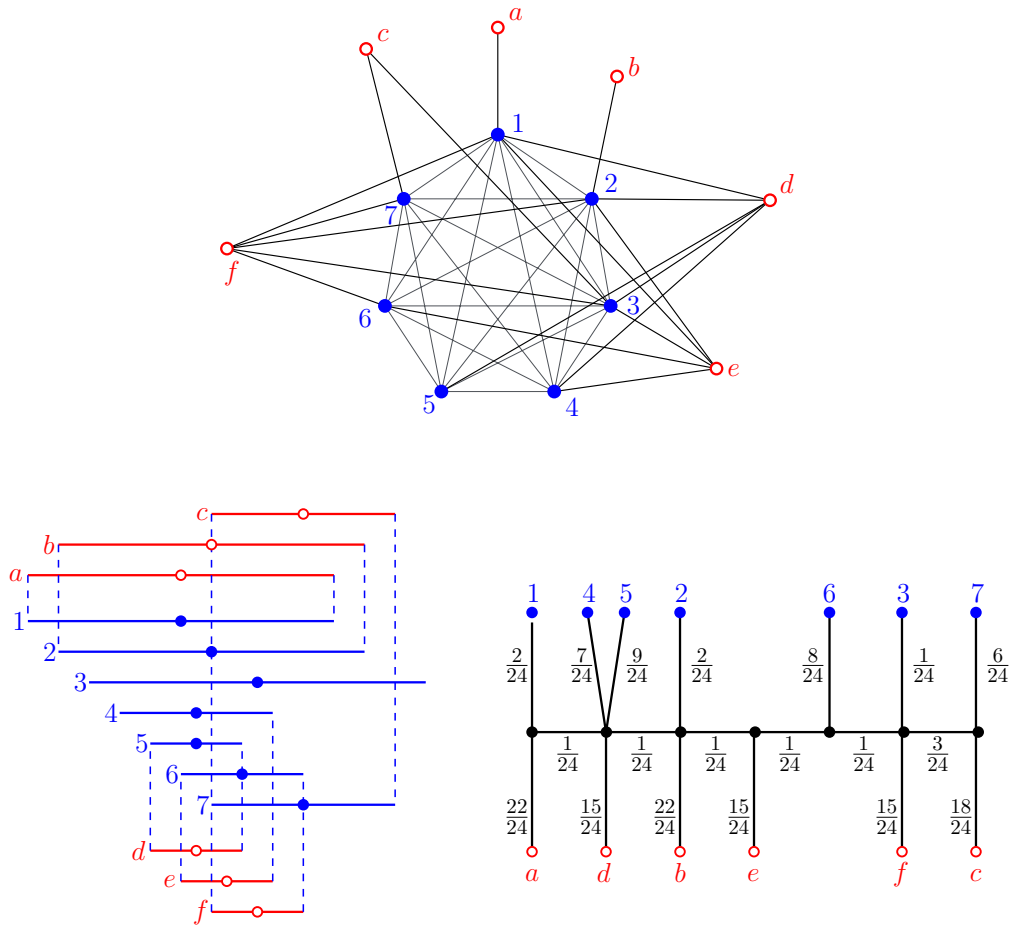
$$d_T(v_i, v_j) = |m(v_i) - m(v_j)| + w(u_i v_i) + w(u_j v_j). \quad (3)$$

From Equations 2 and 3, we deduce that Equation 1 holds also for this direction for every $v_i \in B$ and $v_j \in V(G)$. By Fact 2 and with symmetrical arguments to the ones of the previous direction, we conclude that $(B, R, (I_v)_{v \in V(G)})$ is a blue-red interval model of G . \blacktriangleleft

4 Star NeS model

In this section, we first present an alternative definition of leaf powers through the notion of NeS models. We then show that we can recognize in polynomial time graphs with a *star NeS model*: a NeS model whose embedding tree is a star (considering subdivided stars instead of stars does not make a difference).

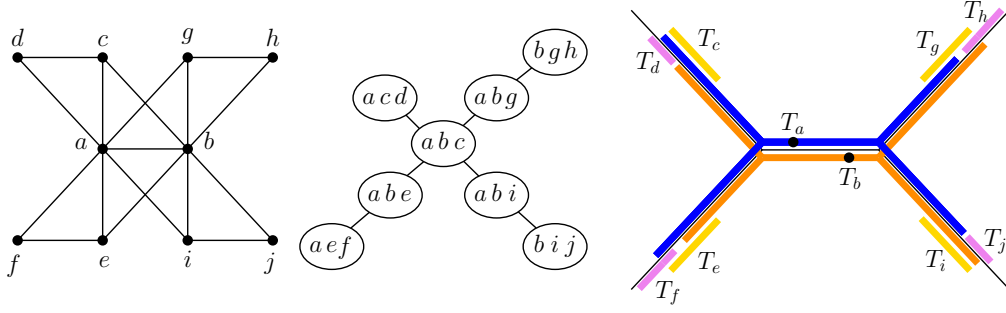
For each tree T , we consider a corresponding tree \mathcal{T} embedded in the Euclidean plane so that each edge of T corresponds to a line segment of \mathcal{T} , these lines segments can intersect



■ **Figure 3** A split graph G with a blue-red interval model of G and a linear leaf root of G obtained from the blue-red interval model by using the construction described in Theorem 3. Note the midpoint of each interval has been marked, to better understand where each vertex is placed in T . Weights on edges is calculated as explained in the proof of Theorem 3 and note every edge has a weight that is a multiple of $\frac{1}{24}$ because the length of I_3 is 12.

one another only at their endpoints, and the vertices of T correspond (one-to-one) to the endpoints of the lines. Each line segment of \mathcal{T} has a positive Euclidean length. These embedded trees allow us to consider \mathcal{T} as the infinite set of points on the line segments of \mathcal{T} . The notion of tree embedding used here is consistent with that found in [26]. The line segments of \mathcal{T} and their endpoints are called respectively the edges and the nodes of \mathcal{T} . The distance between two points x, y of \mathcal{T} denoted by $d_{\mathcal{T}}(x, y)$ is the length of the unique path in \mathcal{T} between x and y (the distance between two vertices of T and their corresponding endpoints in \mathcal{T} are the same).

► **Definition 4** (Neighborhood subtree, NeS-model). *Let \mathcal{T} be an embedding tree. For some point $c \in \mathcal{T}$ and non-negative rational w , we define the neighborhood subtree with center c and radius w as the set of points $\{p \in \mathcal{T} : d_{\mathcal{T}}(p, c) \leq w\}$. A NeS model $(\mathcal{T}, (T_v)_{v \in V(G)})$ of a graph G is a pair of an embedding tree \mathcal{T} and a collection of neighborhood subtrees of \mathcal{T} associated with each vertex of G such that for every $u, v \in V(G)$, we have $uv \in E(G)$ iff $T_u \cap T_v \neq \emptyset$.*



■ **Figure 4** A graph with a clique tree and NeS model. The dots are centers of T_a and T_b .

► **Theorem 5.** *A graph is a leaf power iff it admits a NeS model.*

Proof. Brandstädt et al. showed that leaf powers correspond to the graph class fixed tolerance NeST graph [3, Theorem 4]. Bidelnieks and Dearing showed that G is a fixed tolerance NeST graph iff G has a NeS model [1, Theorem 3.3]. ◀

See Figure 4 for a NeS model. Observe that every interval graph has a NeS model where the embedding tree is a single edge. Moreover, if a graph G admits a NeS model $(\mathcal{T}, (T_v)_{v \in V(G)})$, then for every embedding path \mathcal{L} of \mathcal{T} , $(\mathcal{L}, (T_v \cap \mathcal{L})_{v \in X})$ is an interval model of $G[X]$ with X the set of vertices v such that T_v intersects \mathcal{L} . As illustrated by the proofs of the following two lemmata, some results are easier to prove with NeS models than with other characterizations.

We use the following operation on NeS models several times in the following proofs:

- Given a NeS model $(\mathcal{T}, (T_v)_{v \in V(G)})$ of a graph G and $x \in V(G)$, if the center c_x of T_x is not an endpoint, we turn it into one by splitting the line containing c_x . We add a new line L with endpoint c_x that is sufficiently long so that L contains a point p that is not in any neighborhood subtree. We replace the center of T_x by p and increase the radius of T_x by $d_{\mathcal{T}}(p, c_x)$.

We call this operation *isolating* T_x in $(\mathcal{T}, (T_v)_{v \in V(G)})$. Observe that after this operation, $(\mathcal{T}, (T_v)_{v \in V(G)})$ will remain a NeS model of G since the intersection relations between T_x and the other neighborhood subtrees do not change. Moreover, after this operation, T_x is the only neighborhood subtree in $(T_v)_{v \in V(G)}$ that contains the center of T_x .

► **Lemma 6.** *For a graph G and $u \in V(G)$ such that either (1) u is universal, or (2) u has degree 1, or (3) $N(u)$ is a minimal separator in $G - u$, or (4) $N(u)$ is a maximal clique in $G - u$. Then G is a leaf power iff $G - u$ is a leaf power.*

Proof. The forward implication is trivial as being a leaf power is a hereditary property. For the backward implication, we assume we have a NeS model $(\mathcal{T}, (T_v)_{v \in V(G-u)})$ of $G - u$, and in each case we show how to obtain a NeS model for G :

- u is universal, we simply create T_u a neighborhood subtree with an arbitrary center and a radius sufficiently large so that T_u intersects all the other neighborhood subtrees.
- u has degree 1 with x as neighbor. We isolate T_x in $(\mathcal{T}, (T_v)_{v \in V(G-u)})$. Then, we set $T_u = \{c_x\}$ where c_x is the (new) center of T_x .
- $N(u)$ is a maximal clique in $G - u$. Then $T = \bigcap_{v \in N(u)} T_v$ is not empty and the only neighborhood subtrees intersecting T are those associated with the vertices in $N(u)$. We set $T_u = \{c_u\}$ for some arbitrary point $c_u \in T$.

- $N(u)$ is a minimal separator in $G - u$. Let $T = \bigcap_{v \in N(u)} T_v$. Since T is the intersection of neighborhood subtrees, by [1, Lemma 2.1], T is a neighborhood subtree and has a center c_T . We add a line L to \mathcal{T} with endpoint c_T . Let c_u be the point on $L \cap T$ such that $d_{\mathcal{T}}(c_T, c_u)$ is maximum. As $N(u)$ is a minimal separator, the only neighborhood subtrees containing T are those associated with the vertices in $N(u)$. By construction, we deduce that the only neighborhood subtrees containing c_u are those associated with the vertices in $N(u)$. We set $T_u = \{c_u\}$.

It is not hard to see that in each case, $(\mathcal{T}, (T_v)_{v \in V(G)})$ is a NeS model of G . ◀

► **Lemma 7.** *Let G be a graph and $u \in V(G)$ a cut vertex. Then G is a leaf power iff for every component C of $G - u$, $G[V(C) \cup \{u\}]$ is a leaf power.*

Proof. As with Lemma 6, the forward implication is trivial. For the backward implication, let C_1, \dots, C_k be the components of $G - u$, and for every $1 \leq i \leq k$ we let $C_i^+ = G[V(C_i) \cup \{u\}]$. By assumption, every C_i^+ is a leaf power and there exists a NeS model $(\mathcal{T}^i, (T_v^i)_{v \in V(C_i^+)})$ of C_i^+ . For each $1 \leq i \leq k$, we do the following, we isolate T_u^i in $(\mathcal{T}^i, (T_v^i)_{v \in V(C_i^+)})$ and we make the new center c_u^i of T_u^i an endpoint by subdividing the new line containing c_u^i . Then, we multiply the length of every line of \mathcal{T}^i and the radius of every $(T_v^i)_{v \in V(C_i^+)}$ by $\frac{1}{r_u^i}$ where r_u^i is the new radius of T_u^i . After these operations, for every $1 \leq i \leq k$, the radius of T_u^i is 1 and T_u^i is the only neighborhood subtree in $(T_v^i)_{v \in V(C_i^+)}$ that contains the center of T_u^i .

We make a new NeS model $(\mathcal{T}, (T_v)_{v \in V(G)})$ where \mathcal{T} is obtained from the union of $\mathcal{T}_1, \dots, \mathcal{T}_k$ by identifying the centers (which are endpoints) of the neighborhood subtrees T_u^1, \dots, T_u^k as a unique point c_u . We define T_u the neighborhood subtree with center c_u and radius 1. Finally, for every $v \in V(G - u)$, we define $T_v = T_v^i$ with $i \in [k]$ such that $v \in V(C_i)$. It is straightforward to check that $(\mathcal{T}, (T_v)_{v \in V(G)})$ is a NeS model of G . ◀

We now give the algorithm for recognizing graphs having a star NeS model. Our result is based on the purely combinatorial definition of *good partition*, and we show that a graph admits a star NeS model iff it admits a good partition. Given a good partition, we compute a star NeS model in polynomial time. Finally, we prove that our Algorithm 1 in polynomial time constructs a good partition of the input graph or confirms that it does not admit one.

Consider a star NeS model $(\mathcal{T}, (T_v)_{v \in V(G)})$ of a graph G . Observe that \mathcal{T} is the union of line segments L_1, \dots, L_β with a common endpoint c that is the center of \mathcal{T} . Let X be the set of vertices whose neighborhood subtrees contain c . For each $i \in [\beta]$, we let B_i be the set of all vertices in $V(G) \setminus X$ whose neighborhood subtrees are subsets of L_i . The family $\mathcal{B} = \{B_i : i \in [\beta]\}$ must then constitute a partition of $V(G) \setminus X$. We will show in Theorem 13 that the pair (X, \mathcal{B}) has the properties of a good partition.

► **Fact 8.** *Let G , $(\mathcal{T}, (T_v)_{v \in V(G)})$ and \mathcal{B} be as defined above. We then have:*

- *There is no edge between B_i and B_j for $i \neq j$ and thus $\text{CC}(G - X) \subseteq \mathcal{B}$.*
- *For every $i \in [\beta]$ the NeS model $(L_i, (T_v \cap L_i)_{v \in B_i \cup X})$ is an interval model of $G[X \cup B_i]$.*
- *For each $x \in X$ the neighborhood subtree T_x is the union of the β intervals $L_1 \cap T_x, \dots, L_\beta \cap T_x$ and there exist positive rationals ℓ_x and h_x with $\ell_x \leq h_x$ such that one interval among these intervals has length h_x and the other $\beta - 1$ intervals have length ℓ_x .*
- *If $\ell_x = h_x$, then the center of T_x is c .*

▷ **Claim 9.** *If G has a star NeS model, it has a one, $(\mathcal{T}, (T_v)_{v \in V(G)})$, where vertices whose neighborhood subtrees contain the center of \mathcal{T} is a maximal clique.*

Proof. Let $(\mathcal{T}, (T_v)_{v \in V(G)})$ be the star NeS model of a graph G with c the center of \mathcal{T} . Suppose that the set X of vertices x such that $c \in T_x$ is not a maximal clique. We deduce

that there exists at least one maximal clique Y containing X . For every such clique Y , $\cap_{y \in Y} T_y$ does not contain c since $X \subset Y$. We take a maximal clique Y such that $X \subset Y$ and $\min\{d_{\mathcal{T}}(c, p) : p \in \cap_{y \in Y} T_y\}$ is minimum. Let L be the line segment of \mathcal{T} containing $\cap_{y \in Y} T_y$. By Fact 8, the neighborhood subtrees of the vertices in $Y \setminus X$ are intervals of L . We modify the NeS model by stretching the neighborhood subtrees of the vertices in $Y \setminus X$ so that they admit c as an endpoint. After this operation, these subtrees remain intervals of L and consequently, they still are neighborhood subtrees. Moreover, the choice of Y implies that the only neighborhood subtrees that were intersecting the interval between c and $\cap_{y \in Y} T_y$ are those associated with the vertices of Y . Thus, after this operation, we obtain a star NeS model where the set of vertices x such that $c \in T_x$ is now the maximal clique Y . \triangleleft

Claim 9 follows since we can always stretch some intervals to make X a maximal clique. So far we have described a good partition as it arises from a star NeS model. Now we introduce the properties of a good partition that will allow to abstract away from geometrical aspects while still being equivalent, i.e. so that a graph has a good partition (X, \mathcal{B}) iff it has a star NeS model. The first property is $\text{CC}(G - X) \sqsubseteq \mathcal{B}$ and the second is that for every $B \in \mathcal{B}$ the graph $G[X \cup B]$ is an interval graph having a model where the intervals of X contain the last point used in the interval representation.

► **Definition 10** (*X-interval graph*). *Let X be a maximal clique of G . We say that G is an X-interval graph if G admits a clique path (K_1, \dots, K_k, X) .*

The third property is the existence of an elimination order for the vertices of X based on the lengths ℓ_x in the last item of Fact 8, namely the permutation (x_1, \dots, x_t) of X such that $\ell_{x_1} \leq \ell_{x_2} \leq \dots \leq \ell_{x_t}$. This permutation has the property that for any $i \in [t]$, among the vertices x_i, x_{i+1}, \dots, x_t the vertex x_i must have the minimal neighborhood in at least $\beta - 1$ of the blocks of \mathcal{B} ; we say that x_i is *removable* from $\{x_i, \dots, x_t\}$ for \mathcal{B} .

► **Definition 11** (*Removable vertex*). *Let $X \subseteq V(G)$, $Y \subseteq X$ and let \mathcal{B} be a partition of $V(G) \setminus X$. Given a block B of \mathcal{B} and $x \in Y$, we say $N(x)$ is minimal in B for Y if $N(x) \cap B \subseteq N(y)$ for every $y \in Y$. We say that a vertex $x \in Y$ is removable from Y for \mathcal{B} if $N(x)$ is minimal in at least $|\mathcal{B}| - 1$ blocks of \mathcal{B} for Y .*

► **Definition 12** (*Good partition*). *A good partition of a graph G is a pair (X, \mathcal{B}) where X is a maximal clique of G and \mathcal{B} a partition of $V(G) \setminus X$ satisfying:*

1. $\text{CC}(G - X) \sqsubseteq \mathcal{B}$, i.e. every $C \in \text{CC}(G - X)$ is contained in a block of \mathcal{B} .
2. For each block $B \in \mathcal{B}$, $G[X \cup B]$ is an X-interval graph.
3. There exists an elimination order (x_1, \dots, x_t) on X such that for every $i \in [t]$, x_i is removable from $\{x_i, \dots, x_t\}$ for \mathcal{B} .

X is the central clique of (X, \mathcal{B}) and (x_1, \dots, x_t) a good permutation of (X, \mathcal{B}) .

► **Theorem 13.** *A graph G admits a good partition iff it admits a star NeS model. Moreover, given the former we can compute the latter in polynomial time.*

Proof. (\Leftarrow) Let G be a graph with a star NeS model $(\mathcal{T}, (T_v)_{v \in V(G)})$. Let X be the set of vertices x such that T_x contains the center c of the star \mathcal{T} . By Claim 9, we can assume that X is a maximal clique. As \mathcal{T} is a star, \mathcal{T} is a union of line segments L_1, \dots, L_β with one common endpoint that is c . Let $\mathcal{B} = \{B_1, \dots, B_\beta\}$ such that, for every $i \in [\beta]$, B_i is the set of vertices $v \in V(G) \setminus X$ such that T_v intersects L_i . We claim that (X, \mathcal{B}) is a good partition of G . Fact 8 implies that Property 1 and Property 2 are satisfied. For every $x \in X$, let ℓ_x be the rational defined in Fact 8. Take a permutation (x_1, \dots, x_t) of X such that

$\ell_{x_1} \leq \ell_{x_2} \leq \dots \leq \ell_{x_t}$. Let $f(i) \in [\beta]$ such that the center of x_i lies in $L_{f(i)}$. From Fact 8, we have $|T_{x_i} \cap L_j| = \ell_{x_i}$ for every $j \neq f(i)$. Hence, for every $j \neq f(i)$ the interval $T_{x_i} \cap L_j$ are contained in the neighborhood subtree T_y for every $y \in \{x_i, \dots, x_t\}$. Consequently, $N(x_i)$ is minimal in B_j for $\{x_i, \dots, x_t\}$ for every $j \neq f(i)$. We conclude that (x_1, \dots, x_t) is a good permutation of (X, \mathcal{B}) , i.e. Property 3 is satisfied.

(\Rightarrow) Let (X, \mathcal{B}) be a good partition of a graph G with $\mathcal{B} = \{B_1, \dots, B_\beta\}$ and (x_1, \dots, x_t) be a good permutation of X . For every $i \in [t]$, we define $X_i = \{x_i, \dots, x_t\}$. Since x_i is removable from X_i , there exists an integer $f(i) \in [\beta]$ such that $N(x_i)$ is minimal for X_i in every block of \mathcal{B} different from $B_{f(i)}$ (note that $f(i)$ is not necessarily unique, as x_i could be minimal in every block of \mathcal{B} for X_i).

Take \mathcal{T} , the embedding of a star with center c that is the union of β line segments L_1, \dots, L_β of length $2t + 1$ whose intersection is $\{c\}$. We start by constructing the neighborhood subtree of the vertices in X . For doing so, we associate each $x_i \in X$ and each segment L_j with a rational $\ell(x_i, L_j)$ and define T_{x_i} as the union of over $j \in [\beta]$ of the points on L_j at distance at most $\ell(x_i, L_j)$ from c .

For every $i \in [t]$ and $j \in [\beta]$ such that $j \neq f(i)$, we define $\ell(x_i, L_j) = i$. We define $\ell(x_t, L_{f(t)}) = t$ and for every i from $t - 1$ to 1 we define $\ell(x_i, L_{f(i)})$ as follows:

- If $N(x_i)$ is minimal in $B_{f(i)}$ for X_i then we define $\ell(x_i, L_{f(i)}) = i$.
- If $N(x_j) \cap B_{f(i)} \subset N(x_i)$ for every $x_j \in X_i$, then we define $\ell(x_i, L_{f(i)}) = 1 + \max\{\ell(y, L_{f(i)}) : y \in X_{i+1}\}$.
- Otherwise, we take x_{\min} and $x_{\max} \in X_{i+1}$ such that

$$N(x_{\min}) \cap B_{f(i)} \subseteq N(x_i) \cap B_{f(i)} \subseteq N(x_{\max}) \cap B_{f(i)}$$

and $N(x_{\min}) \cap B_{f(i)}$ is maximal and $N(x_{\max}) \cap B_{f(i)}$ is minimal. We define $\ell(x_i, L_{f(i)}) = (\ell(x_{\min}, L_{f(i)}) + \ell(x_{\max}, L_{f(i)}))/2$. Observe that the vertices x_{\min} and x_{\max} exist because $G[X \cup B_{f(i)}]$ is an X -interval graph and thus the neighborhoods of the vertices in X in $B_{f(i)}$ are pairwise comparable for the inclusion.

By construction, we deduce the following properties on the lengths $\ell(x_i, L_j)$.

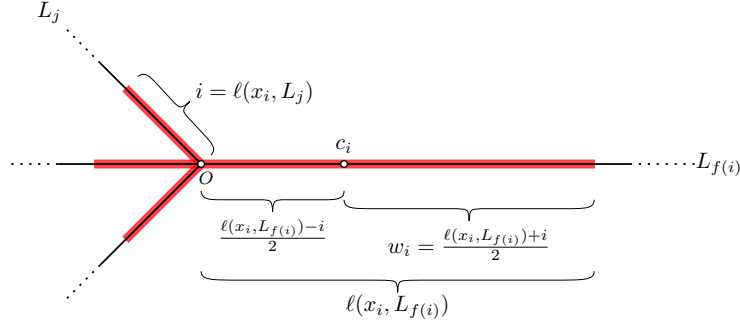
\triangleright **Claim 14.** The following conditions hold for every $j \in [\beta]$:

1. For every $i \in [t]$, we have $\ell(x_i, L_j) \geq i$.
2. For every $x, y \in X$, if $N(x) \cap B_j \subset N(y)$ then $\ell(x, L_j) < \ell(y, L_j)$.

Proof. We prove by induction on i from t to 1 that Condition (1) holds for x_i and that Condition (2) holds for every $x, y \in X_i$. That is obviously the case when $i = t$. Let $i \in [t - 1]$ and suppose that Condition (1) holds for every x_{i+1}, \dots, x_t and Condition (2) holds for every $x, y \in X_{i+1}$. Let $j \in [\beta]$ such that $B_j \neq B_{f(i)}$. By construction, we have $\ell(x_i, L_j) = i$ so Condition (1) holds for x_i and B_j . Moreover, we have $N(x_i) \setminus B_{x_i} \subseteq N(y)$ for every $y \in X_i$. As $B_j \neq B_{f(i)}$, it follows that $N(x_i) \cap B_j \subseteq N(y)$ for every $y \in X_i$. The induction hypothesis implies that $\ell(y, L_j) \geq i + 1$ for every $y \in X_i$. Hence, $\ell(x_i, L_j) < \ell(y, L_j)$ for every $y \in X_i$ and Condition (2) holds for X_i and every $B_j \neq B_{f(i)}$.

It remains to prove that both Conditions holds for $B_{f(i)}$. We define $\ell(x_i, L_{f(i)})$ either to i , $a + 1$ or $(b + c)/2$ where a, b and c belong to $\{\ell(y, L_{f(i)}) : y \in X_{i+1}\}$. The induction assumption implies that $\ell(y, L_{f(i)}) \geq i + 1$ for every $y \in X_{i+1}$. Thus a, b and c greater than or equal to $i + 1$. We deduce that $\ell(x_i, L_{f(i)}) \geq i$. This proves that Condition (1) holds for x_i .

If $N(x_i)$ is minimal in $B_{f(i)}$ for X_i , then $\ell(x_i, L_j) < \ell(y, L_j)$ for every $y \in X_i$ and Condition (2) is satisfied. Suppose that $N(x_i)$ is maximal in $B_{f(i)}$ for X_i , i.e. $N(y) \cap B_{f(i)} \subseteq$



■ **Figure 5** Illustration of the neighborhood subtree T_{x_i} where the number β of edges of the star NeS model is 4.

$N(x_i)$ for every $y \in X_i$. In this case, we set $\ell(x_i, L_{f(i)})$ is define as $1 + \max\{\ell(y, L_{f(i)}) : y \in X_{i+1}\}$. Thus, we have $\ell(x_i, L_{f(i)}) > \ell(y, L_{f(i)})$ for every $y \in X_{i+1}$ and Condition (2) holds for $\leq B_{f(i)}$.

Finally, assume that $N(x_i)$ is neither minimal nor maximal in $B_{f(i)}$ for X_i . If $x^{\max} = x^{\min}$, then we have $\ell(x_i, L_{f(i)}) = \ell(x^{\max}, L_{f(i)})$ and also $N(x_i) \cap B_{f(i)} = N(x^{\min}) \cap B_{f(i)}$. We deduce that Condition (2) holds for every $x, y \in X_i$ because by induction hypothesis we have for every $x, y \in X_{i+1}$ and $x^{\min} \in X_{i+1}$.

Now, assume that $x^{\min} \neq x^{\max}$. The way we choose x^{\min} and x^{\max} implies that

$$N(x^{\min}) \cap B_{f(i)} \subset N(x_i) \cap B_{f(i)} \subset N(x^{\max}) \cap B_{f(i)}.$$

As $N(x^{\min}) \cap B_{f(i)} \subset N(x^{\max}) \cap B_{f(i)}$, by induction hypothesis, we deduce that $\ell(x^{\min}, L_{f(i)}) < \ell(x^{\max}, L_{f(i)})$ and thus we have

$$\ell(x^{\min}, L_{f(i)}) < \ell(x_i, L_{f(i)}) < \ell(x^{\max}, L_{f(i)})$$

So Condition (2) holds for every $x, y \in \{x_i, x^{\min}, x^{\max}\}$.

Let $y \in X_{i+1}$ such that $N(x_i) \cap B_{f(i)} \subseteq N(y)$. The minimality of $N(x^{\max}) \cap B_{f(i)}$ implies that $N(x^{\max}) \cap B_{f(i)} \subseteq N(y)$. By induction hypothesis, we have $\ell(x^{\max}, L_{f(i)}) \leq \ell(y, L_{f(i)})$. Thus, for every $y \in X_{i+1}$ such that $N(x_i) \cap B_{f(i)} \subseteq N(y)$, we have $\ell(x_i, L_{f(i)}) < \ell(y, L_{f(i)})$.

Symmetrically, the maximality of x^{\min} implies that for every $y \in X_{i+1}$ such that $N(y) \cap B_{f(i)} \subseteq N(x_i)$, we have $\ell(y, L_{f(i)}) < \ell(x_i, L_{f(i)})$. We conclude that Condition (2) holds for every $x, y \in X_i$. By induction, we conclude that Condition (1) holds for every $i \in [t]$ and Condition (2) holds for every $x, y \in X$. This concludes the proof of Claim 14. \triangleleft

Observe that each T_{x_i} is a neighborhood subtree as by construction and Condition (1) of Claim 14 the lengths $\ell(x_i, L_j)$ satisfy the last item of Fact 8. See Figure 5 for an illustration of this neighborhood subtree.

It remains to construct the neighborhood subtrees of the vertices in $V(G) \setminus X$. Let us explain how we do it for the vertices in B_j for some $j \in [\beta]$. For every $b \in B_j$, we define T_b as an interval on the line segment L_j . As (X, \mathcal{B}) is a good partition, $G[X \cup B_j]$ admits a clique path (X, K_1, \dots, K_k) . Clique path properties implies that $X \supseteq K_1 \cap X \supseteq \dots \supseteq K_k \cap X$. For every $i \in [k]$, $x \in X \setminus K_i$ and $y \in X \cap K_i$, we have $N(x) \cap B_j \subset N(y)$ and thus $\ell(x, L_j) < \ell(y, L_j)$ thanks to Condition (2) of Claim 14. As $\ell(x, L_j)$ corresponds to the length of the interval $T_x \cap L_j$ for every $x \in X$, we conclude that there exist p_1, \dots, p_k points on L_j such that

- for every $i \in [k]$, $K_i \cap X$ is exactly the set of vertices x in X such that $p_i \in T_x$ and

■ we have $0 < d_{\mathcal{T}}(p_1, c) < d_{\mathcal{T}}(p_2, c) < \dots < d_{\mathcal{T}}(p_k, c)$.

For every $b \in B_j$ such that b is contained in every clique K_i for i between s and t , we define T_b as the interval of L_j between p_s and p_t .

By construction, $(\mathcal{T}, (T_v)_{v \in V(G)})$ fulfill every property of Fact 8. We deduce that $(\mathcal{T}, (T_v)_{v \in V(G)})$ is a NeS model of G . Obviously, the construction of $(\mathcal{T}, (T_v)_{v \in V(G)})$ can be done in polynomial time. ◀

It is easy to see that every graph that admits a star NeS model has a clique-tree that is a subdivided star. The converse is not true. In fact, for every graph G with a clique tree that is a subdivided star with center X , the pair $(X, \text{CC}(G - X))$ satisfies Properties 1 and 2 of Definition 12 but Property 3 might not be satisfied. See for example the graph in Figure 4 and note that the pair $(\{a, b, c\}, \text{CC}(G - \{a, b, c\}))$ does not satisfy Property 3, as after removing the vertex c neither a nor b is removable from $\{a, b\}$.

We now describe Algorithm 1 that decides whether a graph G admits a good partition. Clearly G must be chordal, so we start by checking this. A chordal graph has $O(n)$ maximal cliques, and for each maximal clique X we try to construct a good partition (X, \mathcal{A}) of G . We start with $\mathcal{A} \leftarrow \text{CC}(G - X)$ and note that (X, \mathcal{A}) trivially satisfies Property 1 of Definition 12. Moreover, if G admits a good partition with central clique X , then (X, \mathcal{A}) must satisfy Property 2 of Definition 12, and we check this in Line 3. Then, Algorithm 1 iteratively in a while loop tries to construct a good permutation $(w_1, \dots, w_{|X|})$ of X , while possibly merging some blocks of \mathcal{A} along the way so that it satisfies Property 3, or discover that there is no good partition with central clique X .

For doing so, at every iteration of the while loop, Algorithm 1 searches for a vertex w in W (the set of unprocessed vertices) such that $\text{notmin}(w, W, \mathcal{A})$ – the union of the blocks of \mathcal{A} where $N(w)$ is not minimal for W (see Definition 15) – induces an X -interval graph with X . If such a vertex w exists, then Algorithm 1 sets w_r to w , increments r and merges the blocks of \mathcal{A} contained in $\text{notmin}(w_r, W, \mathcal{A})$ (Line 8) to make w_r removable in \mathcal{A} for W . Otherwise, when no such vertex w exists, Algorithm 1 stops the while loop (Line 7) and tries another candidate for X . For the graph in Figure 4, with $X = \{a, b, c\}$ the first iteration of the while loop will succeed and set $w_1 = c$, but in the second iteration neither a nor b satisfy the condition of Line 6.

At the start of an iteration of the while loop, the algorithm has already chosen the vertices w_1, w_2, \dots, w_{r-1} and $W = X \setminus \{w_1, \dots, w_{r-1}\}$. For every $i \in [r-1]$, w_i is removable from $X \setminus \{w_1, \dots, w_{i-1}\}$ for \mathcal{A} . According to Definition 11 the next vertex w_r to be removed should have $N(w_r)$ non-minimal for W in at most one block of the good partition we want to construct. However, the neighborhood $N(w_r)$ may be non-minimal for W in several blocks of the current partition \mathcal{A} , since these blocks may be (unions of) separate components of $G \setminus X$ that should live on the same line segment of a star NeS model and thus actually be a single block which together with X induces an X -interval graph. An example of this merging is given in Figure 6. The following definition captures, for each $w \in W$, the union of the blocks of \mathcal{A} where $N(w)$ is not minimal for W .

► **Definition 15** (*notmin*). For $W \subseteq X$, $x \in W$ and partition \mathcal{A} of $V(G) \setminus X$, we denote by $\text{notmin}(x, W, \mathcal{A})$ the union of the blocks $A \in \mathcal{A}$ where $N(x)$ is not minimal in A for W .

As we already argued, when Algorithm 1 starts a while loop, (X, \mathcal{A}) satisfies Properties 1 and 2, and it is not hard to argue that in each iteration, for every $i \in [r-1]$, w_i is removable from $X \setminus \{w_1, \dots, w_{i-1}\}$ for \mathcal{A} . Hence, when $W = \emptyset$, then $(w_1, \dots, w_{|X|})$ is a good permutation of (X, \mathcal{A}) and Property 3 is satisfied.

► **Lemma 16.** If Algorithm 1 returns (X, \mathcal{B}) , then (X, \mathcal{B}) is a good partition.

Algorithm 1

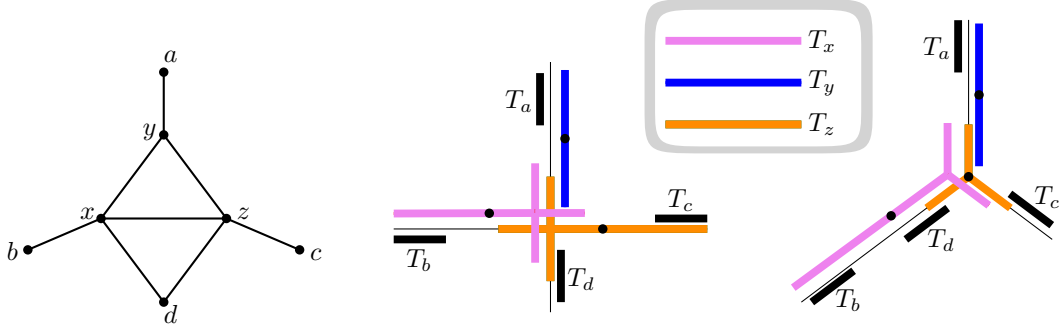
Input: A graph G .
Output: A good partition of G or “no”.

- 1 Check if G is chordal and if so, compute its set of maximal cliques \mathcal{X} , otherwise **return no**;
- 2 **for every** $X \in \mathcal{X}$ **do**
- 3 **if** there exists $C \in \text{CC}(G - X)$ such that $G[X \cup C]$ is not an X -interval graph **then continue**;
- 4 $\mathcal{A} \leftarrow \text{CC}(G - X)$, $W \leftarrow X$ and $r \leftarrow 1$;
- 5 **while** $W \neq \emptyset$ **do**
- 6 **if** there exists $w \in W$ such that $G[X \cup \text{notmin}(w_r, W, \mathcal{A})]$ is an X -interval graph **then** $w_r \leftarrow w$;
- 7 **else break**;
- 8 Replace the blocks of \mathcal{A} contained in $\text{notmin}(w_r, W, \mathcal{A})$ by $\text{notmin}(w_r, W, \mathcal{A})$;
- 9 $W \leftarrow W \setminus \{w_r\}$ and $r \leftarrow r + 1$;
- 10 **end**
- 11 **if** $W = \emptyset$ **then return** (X, \mathcal{A}) ;
- 12 **end**
- 13 **return no**;

Proof. Assume that Algorithm 1 returns a pair (X, \mathcal{B}) . We show that (X, \mathcal{B}) satisfies Definition 12. As the algorithm starts with $\mathcal{A} = \text{CC}(G - X)$ and only merge blocks, Property 1 holds. Line 7 guarantees that every block of $\text{CC}(G - X)$ induces with X an X -interval graph. Moreover, when merging blocks, Line 6 checks that the merged block induces with X an X -interval graph. Thus Property 2 holds.

Observe that at every step of the while loop, we have $\mathcal{A} \sqsubseteq \mathcal{B}$ since we only merge blocks of \mathcal{A} and \mathcal{B} is the last value of \mathcal{A} . We show that (w_1, \dots, w_t) is a good permutation of (X, \mathcal{B}) where, for each $i \in [t]$, w_i is the vertex chosen by the algorithm at the i -th iteration. Let $i \in [t]$. At the start of the i -th iteration of the while loop, the value of W is $X_i = \{w_i, \dots, w_t\}$. During this iteration, we merge the blocks of \mathcal{A} contained in $\text{notmin}(w_i, X_i, \mathcal{A})$. Consequently, after the merging, there exists a block A_{w_i} of \mathcal{A} such that $N(w_i)$ is minimum for X_i in every block of \mathcal{A} different from A_{w_i} . As $\mathcal{A} \sqsubseteq \mathcal{B}$, there exists a block B_{w_i} of \mathcal{B} containing A_{w_i} . We deduce that, $N(w_i)$ is minimal for X_i in every block of \mathcal{B} different from B_{w_i} , i.e. w_i is removable from X_i for \mathcal{B} . Thus, (w_1, \dots, w_t) is a good permutation of (X, \mathcal{B}) and we conclude that (X, \mathcal{B}) is a good partition of G . ◀

To prove the opposite direction, namely that if G has a good partition (X, \mathcal{B}) associated with a good permutation (x_1, \dots, x_t) , then Algorithm 1 finds a good partition, we need two lemmata. The easy case is when Algorithm 1 chooses consecutively $w_1 = x_1, \dots, w_t = x_t$, and we can use Lemma 17 to prove that it will not return no. However, Algorithm 1 does not have this permutation as input and at some iteration with $w_1 = x_1, \dots, w_{r-1} = x_{r-1}$, the algorithm might stop to follow the permutation (x_1, \dots, x_t) and choose a vertex $w_r = x_i$ with $r < i$ because x_r may not be the only vertex satisfying the condition of Line 6. In Lemma 18 we show that choosing $w_r = x_i$ is then not a mistake as it implies the existence of another good partition and another good permutation that starts with $(x_1, \dots, x_{r-1}, w_r = x_i)$. See Figure 6 for an example of a very simple graph with several good permutations leading to quite distinct star NeS models.



■ **Figure 6** A graph G and two star NeS models. The dots are the centers of the subtrees. $X = \{x, y, z\}$ is a maximal clique with four components in $G - X$. The left NeS model corresponds to permutations (y, x, z) or (y, z, x) and the right to permutation (x, y, z) . If Algorithm 1 chooses $w_1 = x$, then the components $\{b\}$ and $\{d\}$ will be merged in Line 8. There is a third star NeS model, corresponding to permutation (z, y, x) , similar to the one on the right. The last two permutations of X are not good permutations.

We need some definitions. Given permutation $P = (x_1, \dots, x_\ell)$ of a subset of X and $i \in [\ell]$, define $\mathcal{A}_0^P = \text{CC}(G - X)$ and \mathcal{A}_i^P the partition of $V(G) \setminus X$ obtained from \mathcal{A}_{i-1}^P by merging the blocks contained in $\text{notmin}(x_i, X \setminus \{x_1, \dots, x_{i-1}\}, \mathcal{A}_{i-1}^P)$. Observe that when Algorithm 1 treats X and we have $w_1 = x_1, \dots, w_\ell = x_\ell$, then the values of \mathcal{A} are successively $\mathcal{A}_0^P, \dots, \mathcal{A}_\ell^P$. The following lemma proves that if there exists a good permutation $P = (x_1, \dots, x_t)$ and at some iteration we have $w_1 = x_1, \dots, w_{r-1} = x_{r-1}$, then the vertex x_r satisfies the condition of Line 6 and Algorithm 1 does not return no during this iteration. Thus, as long as Algorithm 1 follows a good permutation, it will not return no.

► **Lemma 17.** *Let G be a graph with good partition (X, \mathcal{B}) and $P = (x_1, \dots, x_t)$ be a good permutation of (X, \mathcal{B}) . For every $i \in [t]$, we have $\mathcal{A}_i^P \subseteq \mathcal{B}$ and the graph $G[X \cup \text{notmin}(x_i, \{x_i, \dots, x_t\}, \mathcal{A}_{i-1}^P)]$ is an X -interval graph.*

Proof. For every $i \in [t]$, we denote $\{x_i, \dots, x_t\}$ by X_i . We start by proving by induction that, for every $i \in [t]$, we have $\mathcal{A}_i^P \subseteq \mathcal{B}$. This is true for $\mathcal{A}_0^P = \text{CC}(G - X)$ by Property 1 of Definition 12. Let $i \in [t]$ and suppose that $\mathcal{A}_{i-1}^P \subseteq \mathcal{B}$. Every block of \mathcal{A}_i^P different from $\text{notmin}(x_i, X_i, \mathcal{A}_{i-1}^P)$ is a block of \mathcal{A}_{i-1}^P and is included in a block of \mathcal{B} by the induction hypothesis. As x_i is removable from X_i for \mathcal{B} , there exists $B_{x_i} \in \mathcal{B}$ such that $N(x_i)$ is minimal for X_i in every block of \mathcal{B} different from B_{x_i} . Since $\mathcal{A}_{i-1}^P \subseteq \mathcal{B}$, for every block $\hat{A} \in \mathcal{A}_{i-1}^P$ such that $N(x_i)$ is not minimal in \hat{A} for X_i , we have $\hat{A} \subseteq B_{x_i}$. As the union of these blocks \hat{A} 's is $\text{notmin}(x_i, X_i, \mathcal{A}_{i-1}^P)$, we deduce that $\text{notmin}(x_i, X_i, \mathcal{A}_{i-1}^P) \subseteq B_{x_i}$. Hence, every block of \mathcal{A}_i^P is included in a block of \mathcal{B} , thus by induction $\mathcal{A}_i^P \subseteq \mathcal{B}$ for every $i \in [t]$.

For every $i \in [t]$, we have proved that $\text{notmin}(x_i, X_i, \mathcal{A}_{i-1}^P) \subseteq B_{x_i}$. From Property 2 of Definition 12, $G[X \cup B_{x_i}]$ is an X -interval graph. We conclude that $G[X \cup \text{notmin}(x_i, X_i, \mathcal{A}_{i-1}^P)]$ is also an X -interval graph. ◀

► **Lemma 18.** *Let $P = (x_1, \dots, x_t)$ be a good permutation of X and $i \in [t]$. For every $w \in \{x_i, \dots, x_t\}$ such that $G[X \cup \text{notmin}(w, \{x_i, \dots, x_t\}, \mathcal{A}_{i-1}^P)]$ is an X -interval graph, there exists a good permutation of X starting with (x_1, \dots, x_{i-1}, w) .*

Proof. This lemma relies on the following relation that will be applied to vertex subsets corresponding to unions of connected components of $G - X$.

► **Definition 19.** For every $A \subseteq V(G) \setminus X$, we define $X\text{-max}_\cap(A) = N(A) \cap X$ and $X\text{-min}_\cap(A) = \bigcap_{a \in A} N(a) \cap X$. Given $A_1, A_2 \subseteq V(G) \setminus X$, we say that $A_1 \leq_X A_2$ if $X\text{-max}_\cap(A_1) \subseteq X\text{-min}_\cap(A_2)$.

As an example of the use of this, note that in Figure 6, we have $\{b\} \leq_{\{x,y,z\}} \{d\}$ and in the rightmost star NeS model we see this ordering reflected by d being placed closer to the center of the star than b .

For every $A \subseteq V(G) \setminus X$, we have $X\text{-min}_\cap(A) \subseteq X\text{-max}_\cap(A)$ and thus \leq_X is transitive. The following claim reveals the connection between Definition 19 and Property 2 of Definition 12.

▷ **Claim 20.** Let $P = (x_1, \dots, x_t)$ be a good permutation of X and $\ell \in \{0, \dots, t\}$. For all distinct blocks $A, A^* \in \mathcal{A}_\ell^P$, if $G[X \cup A \cup A^*]$ is an X -interval graph, then A and A^* are comparable for \leq_X .

Proof. We prove this claim by induction on ℓ . By definition, we have $\mathcal{A}_0^P = \text{CC}(G - X)$. Let C, C^* be two distinct components of $G - X$ such that $G[X \cup C \cup C^*]$ is an X -interval graph. Let (K_1, \dots, K_k, X) be a clique path of $G[X \cup C \cup C^*]$. Since C and C^* are distinct connected components, for every $j \in [k]$, K_j is either a maximal clique of $G[X \cup C]$ or $G[X \cup C^*]$. Assume w.l.o.g. that K_1 is a maximal clique of $G[X \cup C]$. Moreover, there exists $j \in [k-1]$ such that X, K_1, K_2, \dots, K_j are all the maximal cliques of $G[X \cup C]$ and X, K_{j+1}, \dots, K_k are all the maximal cliques of $G[X \cup C^*]$. Since (K_1, \dots, K_k, X) is a clique path, we have $K_1 \cap X \subseteq \dots \subseteq K_k \cap X \subseteq X$. From Definition 19, we deduce that $X\text{-max}_\cap(C) = K_j \cap X$ and $X\text{-min}_\cap(C^*) = K_{j+1} \cap X$. As $K_j \cap X \subseteq K_{j+1} \cap X$, we have $X\text{-max}_\cap(C) \subseteq X\text{-min}_\cap(C^*)$ and thus $C \leq_X C^*$ and the claim holds for \mathcal{A}_0^P .

Let $\ell \in [t]$ and suppose that the claim holds for $\mathcal{A}_{\ell-1}^P$. In the rest of this proof, we use the shorthand $\text{notmin}(x_\ell)$ to denote $\text{notmin}(x_\ell, \{x_\ell, \dots, x_t\}, \mathcal{A}_{\ell-1}^P)$. We obtain \mathcal{A}_ℓ^P from $\mathcal{A}_{\ell-1}^P$ by merging the blocks of $\mathcal{A}_{\ell-1}^P$ contained in $\text{notmin}(x_\ell)$. So every block of \mathcal{A}_ℓ^P different from $\text{notmin}(x_\ell)$ is also a block of $\mathcal{A}_{\ell-1}^P$. By assumption, the claim holds for every pair of distinct blocks of $\mathcal{A}_\ell^P \setminus \{\text{notmin}(x_\ell)\}$. To prove that it holds for \mathcal{A}_ℓ^P , it is enough to prove that it holds for every pair $(A, \text{notmin}(x_\ell))$ with $A \neq \text{notmin}(x_\ell)$.

Let $A \in \mathcal{A}_\ell^P$ such that $A \neq \text{notmin}(x_\ell)$ and $G[X \cup A \cup \text{notmin}(x_\ell)]$ is an X -interval graph. We need to prove that $A \leq_X \text{notmin}(x_\ell)$ or $\text{notmin}(x_\ell) \leq_X A$. Observe that $\text{notmin}(x_\ell)$ is a union of blocks A_1, \dots, A_k of $\mathcal{A}_{\ell-1}^P$. As $G[X \cup A \cup \text{notmin}(x_\ell)]$ is an X -interval graph, also for any two blocks B, C out of A, A_1, \dots, A_k the graph $G[X \cup B \cup C]$ is an X -interval graph, and thus by induction hypothesis, we deduce that every pair of blocks among A, A_1, \dots, A_k is comparable for \leq_X . Suppose w.l.o.g. that $A_1 \leq_X \dots \leq_X A_k$.

Assume towards a contradiction that there exists $j \in [k-1]$ such that $A_j \leq_X A \leq_X A_{j+1}$. By Definition 15, we have

$$X\text{-max}_\cap(A_j) \subseteq X\text{-min}_\cap(A) \subseteq X\text{-max}_\cap(A) \subseteq X\text{-min}_\cap(A_{j+1}).$$

Since $N(w)$ is not minimal in A_j for X_ℓ , we have $N(w) \cap A_j \neq \emptyset$ and thus $w \in X\text{-max}_\cap(A_j)$. We deduce that w belongs to $X\text{-min}_\cap(A)$ and $X\text{-min}_\cap(A_{j+1})$ and thus $A, A_{j+1} \subseteq N(w)$. As $N(w)$ is not minimal in A_{j+1} and $A_{j+1} \subseteq N(w)$, we deduce that there exists $y \in X_\ell$ such that $A_{j+1} \not\subseteq N(y)$ which means that $y \notin X\text{-min}_\cap(A_{j+1})$. Since $X\text{-max}_\cap(A) \subseteq X\text{-min}_\cap(A_{j+1})$, we have $y \notin X\text{-max}_\cap(A)$. Hence, $A \subseteq N(w)$ and $N(y) \cap A = \emptyset$. We conclude that $N(w)$ is not minimal in A for X_ℓ , a contradiction with $A \neq \text{notmin}(x_\ell)$ because $N(w)$ is minimal for X_ℓ in every block of \mathcal{A}_ℓ^P different from $\text{notmin}(x_\ell)$.

It follows that $A \leq_X A_1$ or $A_k \leq_X A$. Since $\text{notmin}(x_\ell) = A_1 \cup \dots \cup A_k$, we deduce that $A \leq_X \text{notmin}(x_\ell)$ or $\text{notmin}(x_\ell) \leq_X A$, that is the claim holds for \mathcal{A}_ℓ^P . By induction, we conclude that the claim is true for every $\ell \in \{0, \dots, t\}$. \triangleleft

\triangleright **Claim 21.** Let $X \subseteq V(G)$ and \mathcal{A} a partition of $V(G) \setminus X$ such that $\text{CC}(G - X) \sqsubseteq \mathcal{A}$. For any $A_1, \dots, A_k \in \mathcal{A}$ that are pairwise comparable for \leq_X , if for each $j \in [k]$ $G[X \cup A_j]$ is an X -interval graph, then $G[X \cup A_1 \cup \dots \cup A_k]$ is an X -interval graph.

Proof. Let A_1, \dots, A_k be k blocks of \mathcal{A} such that $A_1 \leq_X A_2 \leq_X \dots \leq_X A_k$. Assume that $G[X \cup A_j]$ is an X -interval graph for each $j \in [k]$.

Since $G[X \cup A_j]$ is an X -interval graph for every $j \in [k]$, $G[X \cup A_j]$ admits a clique path $(K_1^j, \dots, K_{f(j)}^j, X)$, where $f(j)$ is the number of maximal cliques minus one in $G[X \cup A_j]$. For every $j \in [k]$, we have $K_1^j \cap X \subseteq K_2^j \cap X \subseteq \dots \subseteq K_{f(j)}^j \cap X$. Moreover, by definition, we have $X\text{-min}_\cap(A_i) = K_1^i$ and $X\text{-max}_\cap(A_i) = K_{f(i)}^i$. For every $j \in [k-1]$, since $A_j \leq_X A_{j+1}$, we have $X\text{-max}_\cap(A_j) \subseteq X\text{-min}_\cap(A_{j+1})$, that is $K_{f(j)}^j \cap X \subseteq K_1^{j+1} \cap X$. Because $\text{CC}(G - X) \sqsubseteq \mathcal{A}$, every maximal clique of $G[X \cup A_1 \cup \dots \cup A_k]$ different from X is a maximal clique of a unique $G[X \cup A_j]$. We conclude that $(K_1^1, \dots, K_{f(1)}^1, \dots, K_1^k, \dots, K_{f(k)}^k, X)$ is a clique path of $G[X \cup A_1 \cup \dots \cup A_k]$. \triangleleft

For every $\ell \in [t]$, we denote by X_ℓ the set $\{x_\ell, \dots, x_t\}$. Let $w \in X_i$ such that $G[X \cup \text{notmin}(w, X_i, \mathcal{A}_{i-1}^P)]$ is an X -interval graph. Let (X, \mathcal{B}) be a good partition of G such that (x_1, \dots, x_t) is a good permutation of (X, \mathcal{B}) . If $w = x_p$ is removable from \mathcal{B} for X_i , then $(x_1, \dots, x_{i-1}, w, x_i, \dots, x_{p-1}, x_{p+1}, \dots, x_t)$ is a good permutation of (X, \mathcal{B}) and we are done. In particular, w is removable from \mathcal{B} if $|\text{notmin}(w, X_i, \mathcal{A}_{i-1}^P)| \leq 1$. In the following, we assume that $|\text{notmin}(w, X_i, \mathcal{A}_{i-1}^P)| \geq 2$.

We construct a good partition $(X, \mathcal{B}_{\text{new}})$ that admits good permutation starting with (x_1, \dots, x_{i-1}, w) . Let $A_1, \dots, A_k \in \mathcal{A}_{i-1}^P$ such that $\text{notmin}(w, X_i, \mathcal{A}_{i-1}^P) = A_1 \cup A_2 \cup \dots \cup A_k$. As $G[X \cup \text{notmin}(w, X_i, \mathcal{A}_{i-1}^P)]$ is an X -interval graph, by Claim 20, the blocks A_1, \dots, A_k are pairwise comparable for \leq_X . Suppose w.l.o.g. that $A_1 \leq_X \dots \leq_X A_k$. In Figure 6 with $\text{notmin}(x, \{x, y, z\}, \{\{a\}, \{b\}, \{c\}, \{d\}\}) = \{b\} \cup \{d\}$ we have $A_1 = \{b\}$ and $A_2 = \{d\}$.

By Lemma 17, we have $\mathcal{A}_{i-1}^P \sqsubseteq \mathcal{B}$. Thus, there exists a block B_1 of \mathcal{B} containing A_1 and B_1 is a union of blocks of \mathcal{A}_{i-1}^P . Let B_1^{max} be the union of all the blocks A of \mathcal{A}_{i-1}^P included in B_1 such that A is not contained in $\text{notmin}(w, X_i, \mathcal{A}_{i-1}^P)$ and $A_1 \leq_X A$. Note that for every $A \in B_1^{\text{max}}$, we have $A_k \leq_X A$ because otherwise we have $A_1 \leq_X A \leq_X A_k$ and that implies $A \subseteq \text{notmin}(w, X_i, \mathcal{A}_{i-1}^P)$ (see the arguments used in the proof of Claim 20).

\triangleright **Claim 22.** We can assume that B_1^{max} is empty.

Proof. Suppose that $B_1^{\text{max}} \neq \emptyset$. Let $B_1^* = B_1 \setminus B_1^{\text{max}}$ and \mathcal{B}^* the partition obtained from \mathcal{B} by replacing the block B_1 with the blocks B_1^{max} and B_1^* . As B_1^{max} is a union of blocks of \mathcal{A}_{i-1}^P and $\mathcal{A}_{i-1}^P \sqsubseteq \mathcal{B}$, by construction, we have $\mathcal{A}_{i-1}^P \sqsubseteq \mathcal{B}^*$. We claim that (X, \mathcal{B}^*) is a good partition that admits (x_1, \dots, x_t) as a good permutation, this is sufficient to prove the claim.

As $\text{CC}(G - X) \sqsubseteq \mathcal{A}_{i-1}^P$, by definition we have $\text{CC}(G - X) \sqsubseteq \mathcal{B}^*$. So Property 1 of Definition 12 is satisfied. As B_1^* and B_1^{max} are subsets of B_1 , we deduce that $G[X \cup B_1^*]$ and $G[X \cup B_1^{\text{max}}]$ are both X -interval graphs. Thus, Property 2 is satisfied.

It remains to prove that (x_1, \dots, x_t) is a good permutation of (X, \mathcal{B}^*) . First, observe that for every $j \in [i-1]$, by definition, x_j is removable from X_j for \mathcal{A}_j^P . Since $\mathcal{A}_j^P \sqsubseteq \mathcal{A}_{i-1}^P \sqsubseteq \mathcal{B}^*$, we deduce that x_j is removable from X_j for \mathcal{B}^* .

Let $j \in \{i, \dots, t\}$. The vertex x_j is not minimal in at most one block of \mathcal{B} for X_j . We claim that $N(x_j)$ is minimal in B_1^{max} for X_j . Since the partition \mathcal{B}^* is obtained from \mathcal{B} by

splitting the block B_1 into B_1^* and B_1^{\max} , this implies that x_j is removable from X_j for B_1^* . We prove that $N(x_j)$ is minimal in B_1^{\max} for X_j by showing that, for every $x_\ell \in X_i$, we have $B_1^{\max} \subseteq N(x_\ell)$.

Since $A_1 \subseteq \text{notmin}(w, X_i, \mathcal{A}_{i-1}^P)$, we have $N(w) \cap A_1 \neq \emptyset$ and thus $w \in X\text{-max}_\cap(A_1)$. Let A be a block of \mathcal{A}_{i-1}^P contained in B_1^{\max} . As $A_1 \leq_X A$, we deduce that $w \in X\text{-min}_\cap(A)$, that is $A \subseteq N(w)$. By definition, A is not contained in $\text{notmin}(w, X_i, \mathcal{A}_{i-1}^P)$ and thus $N(w)$ is minimal in A for X_i . Consequently, for every $x_\ell \in X_i$, we have $A \subseteq N(x_\ell)$. As this holds for every $A \subseteq B_1^{\max}$, we deduce that $B_1^{\max} \subseteq N(x_\ell)$ for every $x_\ell \in X_i$. This ends the proof of Claim 22. \triangleleft

From now, based on Claim 22, we assume that $B_1^{\max} = \emptyset$. We construct \mathcal{B}_{new} as follows. Recall that $A_1 \subseteq B_1$. We create a new block $\widehat{B}_1 = B_1 \cup A_2 \cup \dots \cup A_k$, and for every block $B \in \mathcal{B}$ such that $B \neq B_1$, we create a new block $\widehat{B} = B \setminus (A_2 \cup \dots \cup A_k)$. We define $\mathcal{B}_{\text{new}} = \{\widehat{B}_1\} \cup \{\widehat{B} : B \in \mathcal{B} \setminus \{B_1\} \text{ and } \widehat{B} \neq \emptyset\}$. The construction of \mathcal{B}_{new} is illustrated in Figure 7.

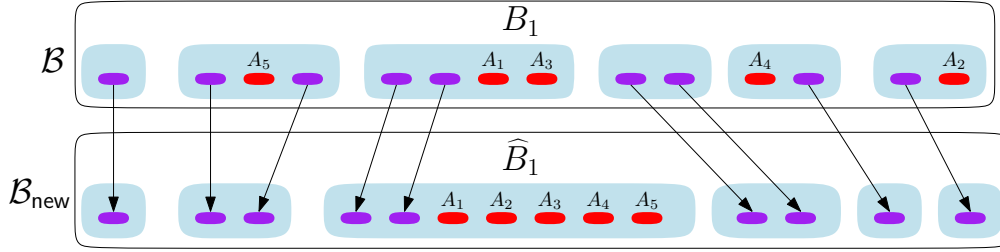


Figure 7 Construction of \mathcal{B}_{new} from \mathcal{B} with $k = 5$. The blocks of \mathcal{B} and \mathcal{B}_{new} are in blue, the blocks A_1, \dots, A_5 of \mathcal{A}_{i-1}^P contained in $\text{notmin}(w, X_i, \mathcal{A}_{i-1}^P)$ are in red, the other blocks of \mathcal{A}_{i-1}^P are in purple. In each block of \mathcal{B} or \mathcal{B}_{new} , the blocks of \mathcal{A}_{i-1}^P are ordered w.r.t. \leq_X from left to right.

We claim that $(X, \mathcal{B}_{\text{new}})$ is a good partition that admits a good permutation starting with (x_1, \dots, x_{i-1}, w) . By construction, \mathcal{B}_{new} is a partition of $V(G) \setminus X$ and $\mathcal{A}_{i-1}^P \sqsubseteq \mathcal{B}_{\text{new}}$. As $\text{CC}(G - X) \sqsubseteq \mathcal{A}_{i-1}^P$ by definition, we have $\text{CC}(G - X) \sqsubseteq \mathcal{B}_{\text{new}}$. So $(X, \mathcal{B}_{\text{new}})$ satisfies Property 1 of Definition 12.

Observe that for every block \widehat{B} of \mathcal{B}_{new} such that $\widehat{B} \neq \widehat{B}_1$, \widehat{B} is a subset of the block B of \mathcal{B} . Since (X, \mathcal{B}) is a good partition, $G[X \cup B]$ is an X -interval graph and hence $G[X \cup \widehat{B}]$ is also an X -interval graph. To prove that Property 2 is satisfied, it remains to show that $G[X \cup \widehat{B}_1]$ is an X -interval graph, for which we will use Claim 21.

Since $G[X \cup B_1]$ is an X -interval graph we know by Claim 20 that the blocks of \mathcal{A}_{i-1}^P contained in B_1 are pairwise comparable for \leq_X . Since $B_1^{\max} = \emptyset$, for every block $A \in \mathcal{A}_{i-1}^P$ such that $A \subseteq B_1$ and A is not contained in $\text{notmin}(w, X_i, \mathcal{A}_{i-1}^P)$, we have $A \leq_X A_1$. Moreover, we have $A_1 \leq_X \dots \leq_X A_k$. We deduce that the blocks of \mathcal{A}_{i-1}^P contained in \widehat{B}_1 are pairwise comparable for \leq_X because \leq_X is transitive. By Claim 21, this implies that $G[X \cup \widehat{B}_1]$ is an X -interval graph. We conclude that Property 2 is satisfied.

It remains to prove that $(X, \mathcal{B}_{\text{new}})$ admits a good permutation starting with (x_1, \dots, x_{i-1}, w) . We prove it with the following four claims.

\triangleright **Claim 23.** For every $j \in [i-1]$, x_j is removable from X_j for \mathcal{B}_{new} and w is removable from X_i for \mathcal{B}_{new} . Moreover, $N(w)$ is minimal for X_i in every block of \mathcal{B}_{new} different from \widehat{B}_1 .

Proof. For every $j \in [i-1]$, we have $\mathcal{A}_j^P \sqsubseteq \mathcal{A}_{i-1}^P$ and thus x_j is removable from X_j for \mathcal{A}_j^P . By construction of \mathcal{B}_{new} , we have $\mathcal{A}_{i-1}^P \sqsubseteq \mathcal{B}_{\text{new}}$. We deduce that x_j is removable from X_j

for \mathcal{B}_{new} for every $j \in [i-1]$. By construction of \mathcal{B}_{new} , we have $\text{notmin}(w, X_i, \mathcal{A}_{i-1}^P) \subseteq \widehat{B}_1$. Hence, $N(w)$ is minimal for X_i in every block of \mathcal{B}_{new} different from \widehat{B}_1 . It follows that w is removable from X_i for \mathcal{B}_{new} . \triangleleft

Let $p \in \{i, \dots, t\}$ such that $x_p = w$.

\triangleright **Claim 24.** For every $j, \ell \in [t]$ such that $i \leq j < \ell$ and $p < \ell$, if $N(x_j)$ is minimal in B_1 for X_j , then $N(x_j) \cap \widehat{B}_1 \subseteq N(x_\ell)$.

Proof. Let $j, \ell \in [t]$ such that $i \leq j < \ell$ and $p < \ell$ and $N(x_j)$ is minimal in B_1 for X_j . Assume towards a contradiction that $N(x_j) \cap \widehat{B}_1 \not\subseteq N(x_\ell)$. It follows that $N(x_\ell) \cap \widehat{B}_1 \subset N(x_j)$ because $G[X \cup \widehat{B}_1]$ is an X -interval graph (these neighborhoods in \widehat{B}_1 are comparable). As $N(x_j)$ is minimal in B_1 for X_j and $\ell \in X_i$, we have¹ $N(x_j) \cap B_1 \subseteq N(x_\ell)$. So by construction of \widehat{B}_1 , there exists a block $A \in \mathcal{A}_{i-1}^P$ such that $A \subseteq \widehat{B}_1 \setminus B_1$ and $N(x_\ell) \cap A \subseteq N(x_j) \cap A$. This implies that $x_\ell \notin X\text{-min}_\cap(A)$. Since $A \subseteq \widehat{B}_1 \setminus B_1 \subseteq \text{notmin}(w, X_i, \mathcal{A}_{i-1}^P)$ and $A \not\subseteq B_1$, we have $A_1 \leq_X A$ and thus $X\text{-max}_\cap(A_1) \subseteq X\text{-min}_\cap(A)$. We deduce that $x_\ell \notin X\text{-max}_\cap(A_1)$.

By definition $A_1 \subseteq \text{notmin}(w, X_i, \mathcal{A}_{i-1}^P)$ and thus $w \in X\text{-max}_\cap(A_1) \subseteq X\text{-min}_\cap(A)$. Hence, we have $N(x_\ell) \cap A \subset N(w) \cap A$ and $N(x_\ell) \cap A_1 \subset N(w) \cap A_1$. Since $p < \ell$ and $w = x_p$, we have $x_\ell \in X_p$. Consequently, $N(w)$ is not minimal in A and A_1 for X_p . Let B_A be the block of \mathcal{B} containing A . Since $A_1 \subseteq B_1$ and $A \subseteq B_A$, $N(w)$ is not minimal in B_1 and B_A for X_p . But B_1 and B_A are two distinct blocks of \mathcal{B} because $A \not\subseteq B_1$, this is a contradiction with w being removable from X_p for \mathcal{B} . \triangleleft

\triangleright **Claim 25.** For every $j \in \{p+1, \dots, t\}$, the vertex x_j is removable from X_j for \mathcal{B}_{new} .

Proof. Let $j \in \{p+1, \dots, t\}$. The vertex x_j is removable from X_j from \mathcal{B} . So there exists a block $B_{x_j} \in \mathcal{B}$ such that $N(x_j)$ is minimal for X_j in every block of \mathcal{B} different from B_{x_j} . By construction, every block \widehat{B} of \mathcal{B}_{new} is associated with a block B of \mathcal{B} and if $B \neq B_1$, we have $B \subseteq \widehat{B}$. Thus, for every block $\widehat{B} \in \mathcal{B}_{\text{new}}$ different from B_1 and B_{x_j} , $N(x_j)$ is minimal in \widehat{B} for X_j because $\widehat{B} \subseteq B$ and $N(x_j)$ is minimal in B for X_j . If $B_{x_j} = B_1$, then x_j is removable from X_j for \mathcal{B}_{new} .

Now suppose that $B_{x_j} \neq B_1$. It follows that $N(x_j)$ is minimal for X_j in B_1 . As $p < j$, by Claim 24, for every $\ell > j$, $N(x_j) \cap \widehat{B}_1 \subseteq N(x_\ell)$. Thus, $N(x_j)$ is minimal in \widehat{B}_1 for X_j . Consequently, $N(x_j)$ is minimal in every block of \mathcal{B}_{new} different from \widehat{B}_{x_j} . In both cases, x_j is removable from X_j for \mathcal{B}_{new} . \triangleleft

It remains to deal with the vertices of (x_1, \dots, x_t) between x_i and x_{p-1} .

\triangleright **Claim 26.** There exists a permutation $(x_{i+1}^*, \dots, x_p^*)$ of $\{x_i, \dots, x_{p-1}\}$ such that, for every $j \in \{i+1, \dots, p\}$, x_j^* is removable from $\{x_j^*, \dots, x_p^*\} \cup X_{p+1}$ for \mathcal{B}_{new} .

Proof. Since (x_1, \dots, x_t) is a good permutation of (X, \mathcal{B}) , for every $j \in \{i, \dots, p-1\}$, there exists a block B_{x_j} of \mathcal{B} such that $N(x_j)$ is minimal for X_j in every block of \mathcal{B} different from B_{x_j} . Let $j \in \{i, \dots, p-1\}$. We start by proving that (\spadesuit) $N(x_j)$ is minimal for X_i in every block of \mathcal{B}_{new} different from \widehat{B}_{x_j} and \widehat{B}_1 . Take $\widehat{B} \in \mathcal{B}_{\text{new}}$ different from \widehat{B}_{x_j} and \widehat{B}_1 . By construction, \widehat{B} is associated with a block B of \mathcal{B} . Since $\widehat{B} \neq \widehat{B}_{x_j}$, $N(x_j)$ is minimal in B for X_j . As $j < p$ and $w = x_p$, we have $w \in X_j$ and thus $N(x_j) \cap B \subseteq N(w)$. By Claim 23, $N(w)$ is minimal for X_i in every block of \mathcal{B}_{new} different from \widehat{B}_1 . In particular, $N(w)$ is minimal in

¹ In fact, we have $N(x_j) \cap B_1 = N(x_\ell) \cap B_1$ since $B_1 \subseteq \widehat{B}_1$ and $N(x_\ell) \cap \widehat{B}_1 \subset N(x_i)$.

\widehat{B} for X_i . As $N(x_j) \cap B \subseteq N(w)$ and $\widehat{B} \subseteq B$, we deduce that $N(x_j)$ is also minimal in \widehat{B} for X_i . This proves \spadesuit .

Since $G[X \cup \widehat{B}_1]$ is an X -interval graph, the neighborhoods of X in \widehat{B}_1 are pairwise comparable for the inclusion. Thus, there exists a permutation $(x_{i+1}^*, \dots, x_p^*)$ of $\{x_i, \dots, x_{p-1}\}$ such that $N(x_{i+1}^*) \cap \widehat{B}_1 \subseteq N(x_{i+2}^*) \cap \widehat{B}_1 \subseteq \dots \subseteq N(x_p^*) \cap \widehat{B}_1$.

Let $j \in \{i+1, \dots, p\}$ and $X_j^* = \{x_j^*, \dots, x_p^*\} \cup X_{p+1}$. We need to show that x_j^* is removable from X_j^* for \mathcal{B}_{new} . As X_j^* is a subset of X_i , from \spadesuit , we know that $N(x_j^*)$ is minimal for X_j^* in every block of \mathcal{B}_{new} different from \widehat{B}_1 and $\widehat{B}_{x_j^*}$. Thus, if $\widehat{B}_1 = \widehat{B}_{x_j^*}$, then x_j^* is removable from X_j^* for \mathcal{B}_{new} .

Suppose that $\widehat{B}_1 \neq \widehat{B}_{x_j^*}$. Then $N(x_j^*)$ is minimal in \widehat{B}_1 for X_q with $q \in [t]$ such that $x_j^* = x_q$. By Claim 24, for every $x_\ell \in X_{p+1}$, we have $N(x_j^*) \cap \widehat{B}_1 \subseteq N(x_\ell)$. Since $N(x_j^*) \cap \widehat{B}_1 \subseteq N(x_{j+1}^*) \cap \widehat{B}_1 \subseteq \dots \subseteq N(x_p^*) \cap \widehat{B}_1$ and $X_j^* = \{x_j^*, \dots, x_p^*\} \cup X_{p+1}$, $N(x_j^*)$ is minimal for X_j^* in \widehat{B}_1 . We deduce that $N(x_j^*)$ is minimal for X_j^* in every block of \mathcal{B}_{new} different from $\widehat{B}_{x_j^*}$. Thus, x_j^* is removable from X_j^* for \mathcal{B}_{new} . \triangleleft

Let $(x_{i+1}^*, \dots, x_p^*)$ be the permutation of $\{x_i, \dots, x_{p-1}\}$ satisfying the condition of Claim 26. Take $P_{\text{new}} = (x_1, \dots, x_{i-1}, w, x_{i+1}^*, \dots, x_p^*, x_{p+1}, \dots, x_t)$, from Claims 23, 25 and 26, we deduce that P_{new} is a good permutation of $(X, \mathcal{B}_{\text{new}})$. Hence, $(X, \mathcal{B}_{\text{new}})$ satisfies Property 3 of Definition 12 and $(X, \mathcal{B}_{\text{new}})$ is a good partition of G . As P_{new} starts with (x_1, \dots, x_{i-1}, w) , this concludes the proof of Lemma 18. \blacktriangleleft

► **Lemma 27.** *If G has a good partition then Algorithm 1 returns one.*

Proof. Suppose G admits a good partition with central clique X . We prove the following invariant holds at the end of the i -th iteration of the while loop for X .

Invariant. G admits a good permutation starting with w_1, \dots, w_i .

By assumption, X admits a good permutation and thus the invariant holds before the algorithm starts the first iteration of the while loop. By induction, assume the invariant holds when Algorithm 1 starts the i -th iteration of the while loop. Let $L = (w_1, \dots, w_{i-1})$ be the consecutive vertices chosen at Line 6 before the start of iteration i (observe that L is empty when $i = 1$). The invariant implies that there exists a good permutation $P = (w_1, \dots, w_{i-1}, x_i, x_{i+1}, \dots, x_t)$ of G starting with L . By Lemma 17, the graph $G[X \cup \text{notmin}(x_i, \{x_i, \dots, x_t\}, \mathcal{A}_{i-1}^P)]$ is an X -interval graph. Observe that \mathcal{A}_{i-1}^P and $\{x_i, \dots, x_t\}$ are the values of the variables \mathcal{A} and W when Algorithm 1 starts the i -th iteration. Thus, at the start of the i -th iteration, there exists a vertex w_i such that $G[X \cup \text{notmin}(w_i, W, \mathcal{A})]$ is an X -interval graph. Consequently, the algorithm does not return no at the i -th iteration and chooses a vertex $w_i \in \{x_i, \dots, x_t\}$ such that the graph $G[X \cup \text{notmin}(w_i, \{x_i, \dots, x_t\}, \mathcal{A}_i^P)]$ is an X -interval graph. By Lemma 18, G admits a good permutation starting with $(w_1, \dots, w_{i-1}, w_i)$. Thus, the invariant holds at the end of the i -th iteration. If at the end of the i -th iteration W is empty, then the while loop stops and the algorithm returns a pair (X, \mathcal{A}) . Otherwise, the algorithm starts an $i+1$ -st iteration and the invariant holds at the start of this new iteration. By induction, the invariant holds at every step and Algorithm 1 returns a pair (X, \mathcal{A}) which is a good partition by Lemma 16. \blacktriangleleft

► **Theorem 28.** *Algo. 1 decides in polynomial time if G admits a star NeS model.*

Proof. By Theorem 13 G admits a star NeS model iff G admits a good partition, and by Lemmata 16 and 27 Algorithm 1 finds a good partition iff the input graph has a good partition. Let us argue for the runtime. Checking that G is chordal and finding the $O(n)$

maximal cliques can be done in polynomial time [16, 24]. Given $X, Y \subseteq V(G)$ we check whether $G[X \cup Y]$ is an X -interval graph, as follows. Take G' the graph obtained from $G[X \cup Y]$ by adding u and v such that $N(u) = \{v\}$ and $N(v) = \{u\} \cup X$. It is easy to see that $G[X \cup Y]$ is an X -interval graph iff G' is an interval graph, which can be checked in polynomial time. \blacktriangleleft

5 Conclusion

The question if leaf powers can be recognized in polynomial time was raised over 20 years ago [23], and this problem related to phylogenetic trees still remains wide open. We have shown that polynomial-time recognition can be achieved if the weighted leaf root is required to be a caterpillar (using the connection to blue-red interval graphs and co-TT graphs) or if the NeS model has a single large-degree branching.

We strongly believe our results could be combined to recognize in polynomial time the graphs with a leaf root whose internal vertices induce a subdivided star, that is a subdivided star with some leaves added to it (this generalizes linear leaf powers as the internal vertices of a caterpillar induce a path). Let us instead sketch how the techniques in this paper could be generalized to handle the general case of leaf powers. Given a NeS model $\mathcal{N} = (\mathcal{T}, (T_v)_{v \in V(G)})$ of a graph G with \mathcal{T} the embedding of a tree T , we define the *topology* of \mathcal{N} as the pair $(T', (X_u)_{u \in V(T')})$ where T' is the subtree induced by the internal nodes of T and for each node u of T' , we define $X_u = \{v \in V(G) : u \in T_v\}$, which will be a clique of G . For a star NeS model, T' is a single node and we proved that (1) if a graph admits a star NeS model, then it admits one where the only clique in the topology is maximal and (2) given a topology with T' a single node, we can decide in polynomial time whether the graph admits a star NeS model with this topology. For the general case it would suffice to solve the following problems:

- Problem 1: given a graph G compute in polynomial time a polynomial-sized family \mathcal{F} of topologies such that G is a leaf power if and only if G admits a NeS model with topology in \mathcal{F} . A first step towards solving this problem would be to generalize Claim 9 to show that we can always modify a NeS model so that all the cliques of its topology become maximal cliques (or minimal separators, as these are also manageable in chordal graphs). A second step is to study the structural properties of strongly chordal graphs with the aim of showing a connection between these properties and a family \mathcal{F} of topologies of NeS models. One candidate among these structural properties is the notion of *clique arrangements* introduced by Nevries and Rosenke [21, 22], which describes the intersections between the maximal cliques of a chordal graph G more precisely than the clique trees.
- Problem 2: given a graph and a topology, construct a NeS model with this topology or confirm that no such NeS model exists, in polynomial time. For star NeS models, this is handled inside the for loop of Algorithm 1 and relies on Definition 12. A step towards solving this problem would be to generalize the combinatorial definition of a good partition to handle any topology.

We suggest to first employ this two-problems approach to graphs admitting NeS models with a very simple topology. Natural first cases are topologies $(T, (X_u)_{u \in V(T)})$ where either T is an edge or where $\bigcup_{u \in V(T)} X_u$ is a clique. Both these restrictions generalize the topologies of star NeS models and positive results should produce tools useful to attack the general case.

Let us end with a subproblem whose solution would simplify this approach towards settling the general case. Inspired by blue-red interval graphs, we can define blue-red NeS models where the red vertices induce an independent set and for every red vertex v its

neighbors are the blue vertices u such that $T_v \subseteq T_u$. We can adapt the proof of Theorem 3 to show that G admits a NeS model iff G has a blue-red NeS model where the blue vertices are simplicial vertices without true twins, so these have the same modeling power. However, for blue-red NeS models the necessary topologies can be significantly simpler. For example, NeS models of linear leaf powers are caterpillars, whose topologies are paths, while its blue-red NeS models can be restricted to an edge, whose topology is empty. Can we generalize the concept of blue-red NeS models so that we can recursively color more vertices blue (by allowing some of them to be adjacent under some conditions) and thereby simplify the topologies we need to consider to solve Problems 1 and 2?

References

- 1 Eric Bibelnicks and Perino M. Dearing. Neighborhood subtree tolerance graphs. *Discret. Appl. Math.*, 43(1):13–26, 1993. doi:10.1016/0166-218X(93)90165-K.
- 2 Andreas Brandstädt and Christian Hundt. Ptolemaic graphs and interval graphs are leaf powers. In *LATIN 2008: Theoretical Informatics, 8th Latin American Symposium, Búzios, Brazil, April 7-11, 2008, Proceedings*, volume 4957 of *Lecture Notes in Computer Science*, pages 479–491. Springer, 2008. doi:10.1007/978-3-540-78773-0_42.
- 3 Andreas Brandstädt, Christian Hundt, Federico Mancini, and Peter Wagner. Rooted directed path graphs are leaf powers. *Discret. Math.*, 310(4):897–910, 2010. doi:10.1016/j.disc.2009.10.006.
- 4 Andreas Brandstädt and Van Bang Le. Structure and linear time recognition of 3-leaf powers. *Inf. Process. Lett.*, 98(4):133–138, 2006. doi:10.1016/j.ipl.2006.01.004.
- 5 Andreas Brandstädt, Van Bang Le, and R. Sritharan. Structure and linear-time recognition of 4-leaf powers. *ACM Trans. Algorithms*, 5(1):11:1–11:22, 2008. doi:10.1145/1435375.1435386.
- 6 Tiziana Calamoneri, Antonio Frangioni, and Blerina Sinimeri. Pairwise compatibility graphs of caterpillars. *Comput. J.*, 57(11):1616–1623, 2014. doi:10.1093/comjnl/bxt068.
- 7 Tiziana Calamoneri and Blerina Sinimeri. Pairwise compatibility graphs: A survey. *SIAM Rev.*, 58(3):445–460, 2016. doi:10.1137/140978053.
- 8 Maw-Shang Chang and Ming-Tat Ko. The 3-steiner root problem. In Andreas Brandstädt, Dieter Kratsch, and Haiko Müller, editors, *Graph-Theoretic Concepts in Computer Science, 33rd International Workshop, WG 2007, Dornburg, Germany, June 21-23, 2007. Revised Papers*, volume 4769 of *Lecture Notes in Computer Science*, pages 109–120. Springer, 2007. doi:10.1007/978-3-540-74839-7_11.
- 9 Alexander Davis, Ruli Gao, and Nicholas Navin. Tumor evolution: Linear, branching, neutral or punctuated? *Biochimica et Biophysica Acta (BBA) - Reviews on Cancer*, 1867(2):151–161, 2017. Evolutionary principles - heterogeneity in cancer? doi:https://doi.org/10.1016/j.bbcan.2017.01.003.
- 10 Reinhard Diestel. *Graph Theory, 4th Edition*, volume 173 of *Graduate texts in mathematics*. Springer, 2012.
- 11 Michael Dom, Jiong Guo, Falk Hüffner, and Rolf Niedermeier. Extending the tractability border for closest leaf powers. In Dieter Kratsch, editor, *Graph-Theoretic Concepts in Computer Science, 31st International Workshop, WG 2005, Metz, France, June 23-25, 2005, Revised Selected Papers*, volume 3787 of *Lecture Notes in Computer Science*, pages 397–408. Springer, 2005. doi:10.1007/11604686_35.
- 12 Guillaume Ducoffe. The 4-steiner root problem. In Ignasi Sau and Dimitrios M. Thilikos, editors, *Graph-Theoretic Concepts in Computer Science - 45th International Workshop, WG 2019, Vall de Núria, Spain, June 19-21, 2019, Revised Papers*, volume 11789 of *Lecture Notes in Computer Science*, pages 14–26. Springer, 2019. doi:10.1007/978-3-030-30786-8_2.
- 13 David Eppstein and Elham Havvaei. Parameterized leaf power recognition via embedding into graph products. *Algorithmica*, 82(8):2337–2359, 2020. doi:10.1007/s00453-020-00720-8.

- 14 Petr A. Golovach, Pinar Heggernes, Nathan Lindzey, Ross M. McConnell, Vinícius Fernandes dos Santos, Jeremy P. Spinrad, and Jayme Luiz Szwarcfiter. On recognition of threshold tolerance graphs and their complements. *Discret. Appl. Math.*, 216:171–180, 2017. doi:10.1016/j.dam.2015.01.034.
- 15 Martin Charles Golumbic, Nirit Lefel Weingarten, and Vincent Limouzy. Co-tt graphs and a characterization of split co-tt graphs. *Discret. Appl. Math.*, 165:168–174, 2014. doi:10.1016/j.dam.2012.11.014.
- 16 Michel Habib, Ross M. McConnell, Christophe Paul, and Laurent Viennot. Lex-bfs and partition refinement, with applications to transitive orientation, interval graph recognition and consecutive ones testing. *Theor. Comput. Sci.*, 234(1-2):59–84, 2000. doi:10.1016/S0304-3975(97)00241-7.
- 17 Lars Jaffke, O-joung Kwon, Torstein J. F. Strømme, and Jan Arne Telle. Mim-width III. graph powers and generalized distance domination problems. *Theor. Comput. Sci.*, 796:216–236, 2019. doi:10.1016/j.tcs.2019.09.012.
- 18 Manuel Lafond. On strongly chordal graphs that are not leaf powers. In *Graph-Theoretic Concepts in Computer Science - 43rd International Workshop, WG 2017, Eindhoven, The Netherlands, June 21-23, 2017, Revised Selected Papers*, pages 386–398, 2017. doi:10.1007/978-3-319-68705-6_29.
- 19 Manuel Lafond. Recognizing k-leaf powers in polynomial time, for constant k. In *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1384–1410. SIAM, 2022. URL: <https://epubs.siam.org/doi/abs/10.1137/1.9781611977073.58>, arXiv:<https://epubs.siam.org/doi/pdf/10.1137/1.9781611977073.58>, doi:10.1137/1.9781611977073.58.
- 20 Clyde L. Monma, Bruce A. Reed, and William T. Trotter. Threshold tolerance graphs. *J. Graph Theory*, 12(3):343–362, 1988. doi:10.1002/jgt.3190120307.
- 21 Ragnar Nevries and Christian Rosenke. Characterizing and computing the structure of clique intersections in strongly chordal graphs. *Discrete Applied Mathematics*, 181:221–234, 2015. URL: <https://www.sciencedirect.com/science/article/pii/S0166218X14003941>, doi:<https://doi.org/10.1016/j.dam.2014.09.003>.
- 22 Ragnar Nevries and Christian Rosenke. Towards a characterization of leaf powers by clique arrangements. *Graphs Comb.*, 32(5):2053–2077, 2016. doi:10.1007/s00373-016-1707-x.
- 23 Naomi Nishimura, Prabhakar Ragde, and Dimitrios M. Thilikos. On graph powers for leaf-labeled trees. *J. Algorithms*, 42(1):69–108, 2002. doi:10.1006/jagm.2001.1195.
- 24 Donald J. Rose, Robert Endre Tarjan, and George S. Lueker. Algorithmic aspects of vertex elimination on graphs. *SIAM J. Comput.*, 5(2):266–283, 1976. doi:10.1137/0205021.
- 25 Erfan Sadeqi Azer, Mohammad Haghiri Ebrahimabadi, Salem Malikić, Roni Khardon, and S. Cenk Sahinalp. Tumor phylogeny topology inference via deep learning. *iScience*, 23(11):101655, 2020. doi:<https://doi.org/10.1016/j.isci.2020.101655>.
- 26 A. Tamir. A class of balanced matrices arising from location problems. *Siam Journal on Algebraic and Discrete Methods*, 4:363–370, 1983.