Discrete Mathematics 345 (2022) 112985

Contents lists available at ScienceDirect

Discrete Mathematics

www.elsevier.com/locate/disc

Well-partitioned chordal graphs ☆

Jungho Ahn [a,b], Lars Jaffke [c,*], O-joung Kwon [d,b], Paloma T. Lima [e]

[a] Department of Mathematical Sciences, KAIST, Daejeon, South Korea
[b] Discrete Mathematics Group, Institute for Basic Science (IBS), Daejeon, South Korea
[c] Department of Informatics, University of Bergen, Bergen, Norway
[d] Department of Mathematics, Hanyang University, Seoul, South Korea
[e] Computer Science Department, IT University of Copenhagen, Copenhagen, Denmark

A R T I C L E   I N F O

Article history:
Received 6 August 2021
Received in revised form 4 May 2022
Accepted 7 May 2022
Available online 19 May 2022

Keywords:
Well-partitioned chordal graph
Forbidden induced subgraphs
Graph class
Longest path transversal
Tree spanner
Geodetic set

A B S T R A C T

We introduce a new subclass of chordal graphs that generalizes the class of split graphs, which we call well-partitioned chordal graphs. A connected graph G is a *well-partitioned chordal graph* if there exist a partition $\mathcal{P}$ of the vertex set of G into cliques and a tree $\mathcal{T}$ having $\mathcal{P}$ as a vertex set such that for distinct $X, Y \in \mathcal{P}$, (1) the edges between X and Y in G form a complete bipartite subgraph whose parts are some subsets of X and Y, if X and Y are adjacent in $\mathcal{T}$, and (2) there are no edges between X and Y in G otherwise. A split graph with vertex partition (C, I) where C is a clique and I is an independent set is a well-partitioned chordal graph as witnessed by a star $\mathcal{T}$ having C as the center and each vertex in I as a leaf, viewed as a clique of size 1. We characterize well-partitioned chordal graphs by forbidden induced subgraphs, and give a polynomial-time algorithm that given a graph, either finds an obstruction, or outputs a partition of its vertex set that asserts that the graph is well-partitioned chordal.

We observe that there are problems, for instance Densest k-Subgraph and b-Coloring, that are polynomial-time solvable on split graphs but become NP-hard on well-partitioned chordal graphs. On the other hand, we show that the Geodetic Set problem, known to be NP-hard on chordal graphs, can be solved in polynomial time on well-partitioned chordal graphs.

We also answer two combinatorial questions on well-partitioned chordal graphs that are open on chordal graphs, namely that each well-partitioned chordal graph admits a polynomial-time constructible tree 3-spanner, and that each (2-connected) well-partitioned chordal graph has a vertex that intersects all its longest paths (cycles).

1. Introduction

A central methodology in the study of the complexity of computationally hard graph problems is to impose additional structure on the input graphs, and determine if the additional structure can be exploited in the design of an efficient algorithm. Typically, one restricts the input to be contained in a *graph class*, which is a set of graphs that share a common

☆ Extended abstracts of different parts of this work appeared in the proceedings of WG 2020 [1] and CIAC 2021 [2]. J.A. and O.K. are supported by the Institute for Basic Science (IBS-R029-C1). O.K. is also supported by the National Research Foundation of Korea (NRF) grant funded by the Ministry of Education (No. NRF-2018R1D1A1B07050294). L.J. acknowledges support from the Norwegian Research Council (No. 274526).

* Corresponding author.
E-mail addresses: junghoahn@kaist.ac.kr (J. Ahn), lars.jaffke@uib.no (L. Jaffke), ojoungkwon@hanyang.ac.kr (O. Kwon), palt@itu.dk (P.T. Lima).

https://doi.org/10.1016/j.disc.2022.112985

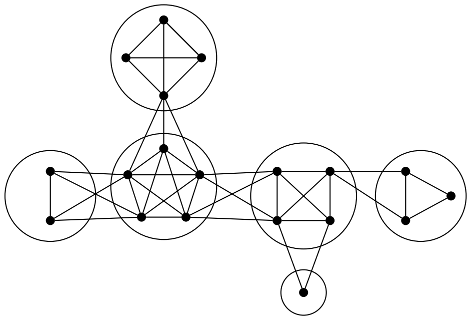


Fig. 1. A well-partitioned chordal graph.

structural property. For example, the class of *forests* is the class of graphs that do not contain a cycle. Following the establishment of the theory of NP-hardness, numerous problems were investigated in specific classes of graphs; either providing a polynomial-time algorithm for a problem Π on a specific graph class, while Π is NP-hard in a more general setting, or showing that Π remains NP-hard on a graph class. We refer to the textbooks [10,34] for a detailed introduction to the subject. A key question in this field is to find for a given problem Π that is hard on a graph class $\mathcal{A}$, a subclass $\mathcal{B} \subsetneq \mathcal{A}$ such that Π is efficiently solvable on $\mathcal{B}$. Naturally, the goal is to narrow down the gap $\mathcal{A} \setminus \mathcal{B}$ as much as possible, and several notions of hardness/efficiency can be applied. For instance, we can require our target problem to be NP-hard on $\mathcal{A}$ and polynomial-time solvable on $\mathcal{B}$; or, from the viewpoint of parameterized complexity [21,24], we require a target parameterized problem Π to be W[1]-hard on $\mathcal{A}$, while Π is in FPT on $\mathcal{B}$, or a separation in the kernelization complexity [28] of Π between $\mathcal{A}$ and $\mathcal{B}$.

Chordal graphs are arguably one of the main characters in the algorithmic study of graph classes. They find applications for instance in computational biology [55], optimization [57], and sparse matrix computations [32]. The class of split graphs is an important subclass of the class of chordal graphs. The complexities of computational problems on chordal and split graphs often coincide, see e.g., [6,7,27,48]; however, this is not always the case. For instance, several variants of graph (vertex) coloring problems are polynomial-time solvable on split graphs and NP-hard on chordal graphs, see the works of Havet et al. [37], and of Silva [56]. Also, the Sparsest k-subgraph [61] and Densest k-subgraph [20] problems are polynomial-time solvable on split graphs and NP-hard on chordal graphs. Other problems, for instance the Tree 3-Spanner problem [9], are easy on split graphs, while their complexity on chordal graphs is still unresolved.

In this work, we introduce the class of *well-partitioned chordal graphs*, a subclass of chordal graphs that generalizes split graphs, which can be used as a tool for narrowing down complexity gaps for problems that are hard on chordal graphs, and easy on split graphs. The definition of well-partitioned chordal graphs is mainly motivated by a property of split graphs: the vertex set of a split graph can be partitioned into sets that can be viewed as a central clique of arbitrary size and cliques of size one that have neighbors only in the central clique. Thus, this partition has the structure of a star. Well-partitioned chordal graphs relax these ideas in two ways: by allowing the parts of the partition to be arranged in a tree structure instead of a star, and by allowing the cliques in each part to have arbitrary size. The interaction between adjacent parts P and Q remains simple: it induces a complete bipartite graph between a subset of P and a subset of Q. Such a tree structure is called a partition tree, and we give an example of a well-partitioned chordal graph in Fig. 1. We formally define this class in Section 3.

The main structural contribution of this work is a characterization of well-partitioned chordal graphs by forbidden induced subgraphs. We also provide a polynomial-time recognition algorithm. We list the set $\mathbb{O}$ of obstructions in Fig. 2.

Theorem 1.1. *A graph is a well-partitioned chordal graph if and only if it has no induced subgraph isomorphic to a graph in $\mathbb{O}$. Furthermore, there is a polynomial-time algorithm that given a graph G, outputs either an induced subgraph of G isomorphic to a graph in $\mathbb{O}$, or a partition tree of each connected component which confirms that G is a well-partitioned chordal graph.*

Before we proceed with the discussion of the other results of this paper, we would like to briefly touch on the relationship of well-partitioned chordal graphs and width parameters. Each split graph is a well-partitioned chordal graph, and there are split graphs whose *maximum induced matching width* (mim-width) depends linearly on the number of vertices [46]. This rules out the applicability of any algorithmic meta-theorem based on one of the common width parameters such as tree-width or clique-width, to the class of well-partitioned chordal graphs. It is known that mim-width is a lower bound for them [58].

We now discuss the applicability and significance of well-partitioned chordal graphs by considering some algorithmic and combinatorial problems restricted to this graph class. First, we consider problems that are known to be polynomial-time solvable on split graphs and NP-hard on chordal graphs. It is not difficult to observe that the chordal graphs constructed in the NP-hardness proofs for vertex-coloring problems studied in the works [37,56], as well as the graphs in the NP-hardness proofs for Densest k-Subgraph [20] and Sparsest k-Subgraph [61] are in fact well-partitioned chordal graphs. We immediately narrowed down the complexity gaps of these problems from Chordal $\setminus$ Split to Well-partitioned chordal $\setminus$

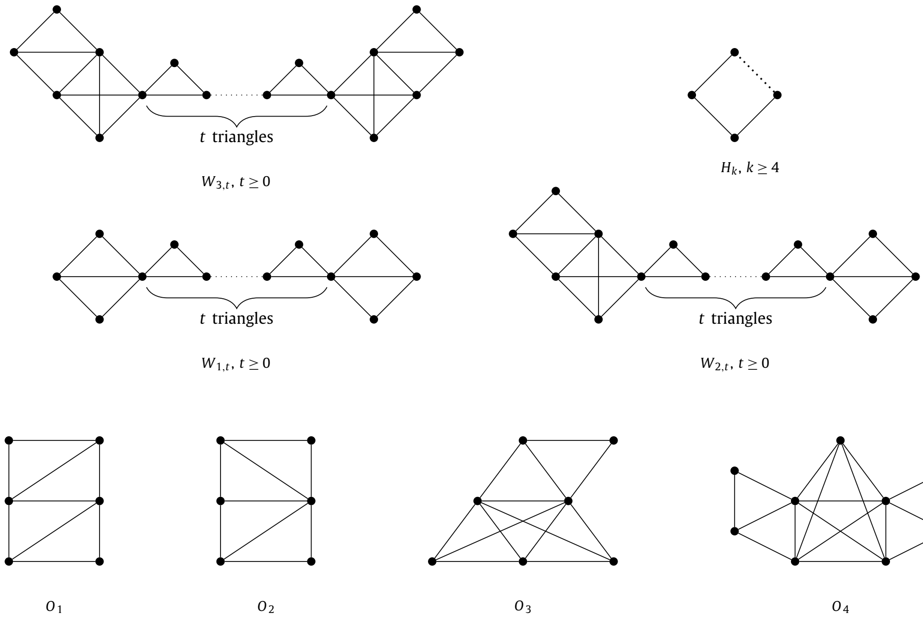


Fig. 2. The set of obstructions $\mathbb{O}$ for well-partitioned chordal graphs.

Split. In this work, we consider the Geodetic Set problem and show that the picture changes in this case. While the problem is known to be NP-hard on chordal graphs [22], we are able to devise a polynomial-time algorithm to solve it on well-partitioned chordal graphs. In this case, we narrowed down the complexity gap from Chordal \ Split to Chordal \ Well-partitioned chordal.

Besides narrowing the complexity gap between the classes of chordal and split graphs, the class of well-partitioned chordal graphs can also be useful as a step towards solving problems that are open for chordal graphs, but whose solution is known for split graphs thanks to their restricted structure. A natural path to a resolution of such questions on chordal graphs is to extend their solutions on split graphs to graph classes that are structurally closer to chordal graphs. Well-partitioned chordal graphs exhibit a tree structure, which makes them a natural target in such a scenario. We consider two such questions. We show that every (2-connected) well-partitioned chordal graph has a vertex that intersects all its longest paths (cycles), while the corresponding question on chordal graphs has remained an open problem [5]. We also show that every well-partitioned chordal graph has a polynomial-time constructible tree 3-spanner, while the complexity of the Tree 3-Spanner problem still remains unresolved on chordal graphs [9].

This paper is organized as follows. We give some preliminary definitions in Section 2, and introduce the class of well-partitioned chordal graphs in Section 3. In Section 4, we prove the characterization of well-partitioned chordal graphs in terms of forbidden induced subgraphs which gives a polynomial-time recognition algorithm for the class. We consider the Geodetic Set problem, the transversals of longest paths and cycles, and the Tree 3-Spanner problem on well-partitioned chordal graphs in Sections 5, 6, and 7, respectively. We conclude in Section 8.

2. Preliminaries

For a positive integer n, we let $[n] := \{i \in \mathbb{N} \mid 1 \leq i \leq n\}$. All graphs considered here are simple and finite. For a graph G we denote by $V(G)$ and $E(G)$ the vertex set and edge set of G, respectively. For an edge $uv \in E(G)$, we call u and v its *endpoints*. We say that G *is isomorphic to* H if there is a bijection $\phi\colon V(G) \to V(H)$ such that for all $u, v \in V(G)$, $uv \in E(G)$ if and only if $\phi(u)\phi(v) \in E(H)$. We say that H is a *subgraph* of G, denoted by $H \subseteq G$, if $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$.

For graphs G and H, let $G \cup H$ be the graph with vertex set $V(G) \cup V(H)$ and edge set $E(G) \cup E(H)$. For a vertex v of a graph G, $N_G(v) := \{w \in V(G) \mid vw \in E(G)\}$ is the set of *neighbors* of v in G, and we let $N_G[v] := N_G(v) \cup \{v\}$. The *degree* of v is $\deg_G(v) := |N_G(v)|$. Given a set $X \subseteq V(G)$, we let $N_G(X) := \bigcup_{v \in X} N_G(v) \setminus X$ and $N_G[X] := N_G(X) \cup X$. In all of the above, we may drop G as a subscript if it is clear from the context. The *subgraph induced by* X, denoted by $G[X]$, is the graph $(X, \{uv \in E(G) \mid u, v \in X\})$. We denote by $G - X$ the graph $G[V(G) \setminus X]$, and for a single vertex $x \in V(G)$, we use the shorthand '$G - x$' for '$G - \{x\}$'. For two sets $X, Y \subseteq V(G)$, we denote by $G[X, Y]$ the graph $(X \cup Y, \{xy \in E(G) \mid x \in X, y \in Y\})$. We say that X is *complete to* Y if $X \cap Y = \emptyset$ and each vertex in X is adjacent to every vertex in Y.

Let G be a graph. We say that G is *trivial* if $|V(G)| = 1$. A graph G is called *complete* if $uv \in E(G)$ for all distinct vertices $u, v \in V(G)$, and *empty* if $E(G) = \emptyset$. A set $X \subseteq V(G)$ is a *clique* if $G[X]$ is complete, and an *independent set* if $G[X]$ is empty. A graph G is called *bipartite* if there is a 2-partition (A, B) of $V(G)$, called a *bipartition of* G, such that A and B are independent sets in G. A bipartite graph G on bipartition (A, B) is called *complete bipartite* if A is complete to B. For positive integers n and m, we denote by $K_{n,m}$ a complete bipartite graph with bipartition (A, B) such that $|A| = n$ and $|B| = m$. A graph is a *star* if it is either trivial or isomorphic to $K_{1,n}$ for some positive integer n.

A graph G is *connected* if for each 2-partition (X, Y) of $V(G)$ with $X \neq \emptyset$ and $Y \neq \emptyset$, there is a pair $x \in X$, $y \in Y$ such that $xy \in E(G)$, and it is *disconnected* otherwise. A *connected component* of G is a maximal connected subgraph of G. A vertex $v \in V(G)$ is a *cut vertex* if $G - v$ has more connected components than G. A graph is *2-connected* if it does not contain a cut vertex. A *block* B of a graph G is a maximal 2-connected subgraph of G. A *cycle* is a connected graph all of whose vertices have degree 2. A graph that has no cycle as a subgraph is called a *forest*, a connected forest is a *tree*, and a tree of maximum degree at most 2 is a *path*. The vertices of degree one in a tree are called *leaves* and the leaves of a path are its *endpoints*. A connected subgraph of a tree is called a *subtree*.

A *hole* in a graph G is an induced cycle of G of length at least 4. A graph is *chordal* if it has no hole. A vertex v is *simplicial* if $N_G(v)$ is a clique. We say that a graph G has a *perfect elimination ordering* $v_1, \dots, v_n$ if v_i is simplicial in $G[\{v_i, v_{i+1}, \dots, v_n\}]$ for each $i \in [n-1]$. It is known that a graph is chordal if and only if it has a perfect elimination ordering [29]. We will use the following hole detecting algorithm and an algorithm to generate a perfect elimination ordering of a chordal graph.

Theorem 2.1 *(Nikolopoulos and Palios [49]). Given a graph G, one can detect a hole in G in time $\mathcal{O}(|V(G)| + |E(G)|^2)$, if one exists.*

Theorem 2.2 *(Rose et al. [54]). Given a graph G, one can generate a perfect elimination ordering of G in time $\mathcal{O}(|V(G)| + |E(G)|)$, if one exists.*

A graph G is a *split graph* if there is a 2-partition (C, I) of $V(G)$ such that C is a clique and I is an independent set. For a family $\mathcal{F}$ of graphs, the *intersection graph of* $\mathcal{F}$ is the graph with vertex set $\mathcal{F}$ and edge set $\{ST \mid S, T \in \mathcal{F}, S \neq T, \text{ and } V(S) \cap V(T) \neq \emptyset\}$. It is well-known that every chordal graph is the intersection graph of subtrees of some tree [31].

3. Well-partitioned chordal graphs

A connected graph G is a *well-partitioned chordal graph* if there exist a partition $\mathcal{P}$ of $V(G)$ and a tree $\mathcal{T}$ having $\mathcal{P}$ as a vertex set such that the following hold.

(i) Each part $X \in \mathcal{P}$ is a clique in G.
(ii) For each edge $XY \in E(\mathcal{T})$, there are subsets $X' \subseteq X$ and $Y' \subseteq Y$ such that

$$E(G[X, Y]) = \{xy \mid x \in X', y \in Y'\}.$$

(iii) For each pair of distinct $X, Y \in V(\mathcal{T})$ with $XY \notin E(\mathcal{T})$, $E(G[X, Y]) = \emptyset$.

The tree $\mathcal{T}$ is called a *partition tree of* G, and the elements of $\mathcal{P}$ are called its *bags*. A graph is a well-partitioned chordal graph if all of its connected components are well-partitioned chordal graphs. We remark that a connected well-partitioned chordal graph can have more than one partition tree. Also, observe that well-partitioned chordal graphs are closed under taking induced subgraphs.

We say that a bag B in a partition tree $\mathcal{T}$ of G is a *leaf bag* if $\deg_{\mathcal{T}}(B) = 1$, and it is an *internal bag* if $\deg_{\mathcal{T}}(B) > 1$.

A useful concept when considering partition trees of well-partitioned chordal graphs is that of a *boundary of a bag*. Let $\mathcal{T}$ be a partition tree of a connected well-partitioned chordal graph G and let $X, Y \in V(\mathcal{T})$ be two bags that are adjacent in $\mathcal{T}$. The *boundary of X with respect to Y*, denoted by $\mathrm{bd}(X, Y)$, is the set of vertices of X that have a neighbor in Y, i.e.,

$$\mathrm{bd}(X, Y) := \{x \in X \mid N_G(x) \cap Y \neq \emptyset\}.$$

By item (ii) of the definition of the class, we know that $\mathrm{bd}(X, Y)$ is complete to $\mathrm{bd}(Y, X)$.

We now consider the relation between well-partitioned chordal graphs and other well-studied classes of graphs. It is easy to see that every well-partitioned chordal graph G is a chordal graph because every leaf of the partition tree of a component of G contains a simplicial vertex of G, and after removing this vertex, the remaining graph is still a well-partitioned chordal graph. Thus, we may construct a perfect elimination ordering. We show that, in fact, well-partitioned chordal graphs constitute a subclass of substar graphs. A graph is a *substar graph* [18,39] if it is an intersection graph of substars of a tree.

Proposition 3.1. *Every well-partitioned chordal graph is a substar graph.*

Proof. Let G be a well-partitioned chordal graph with $V(G) = \{v_i \mid i \in [n]\}$ and a partition tree $\mathcal{T}$. We will exhibit a substar intersection model for G. That is, we will show that there exists a tree F and substars $S_1, \dots, S_n$ of F such that $v_iv_j \in E(G)$ if and only if $V(S_i) \cap V(S_j) \neq \emptyset$.

Let F be the tree obtained from $\mathcal{T}$ by the 1-subdivision of every edge. We denote by $v_{XY} \in V(F)$ the vertex originated from the 1-subdivision of the edge $XY \in E(\mathcal{T})$. Note that $N_F(v_{XY}) = \{X, Y\}$. For every $v_i \in V(G)$, we create a substar of F in the following way. Let $B \in V(\mathcal{T})$ be the bag containing v_i. Then S_i is a star with the center B and the leaf set $\{v_{BY} \mid v_i \in \mathrm{bd}(B, Y)$ for some $Y \in V(\mathcal{T})\}$.

To see that this is indeed an intersection model for G, let $v_iv_j \in E(G)$. If there exists $B \in V(\mathcal{T})$ such that $v_i, v_j \in B$, then $B \in V(S_i) \cap V(S_j)$. If v_i and v_j are not contained in the same bag, by item (ii), there exist $A, B \in V(\mathcal{T})$ such that $v_i \in A$, $v_j \in B$ and $AB \in E(\mathcal{T})$. Then, $v_{AB} \in V(S_i) \cap V(S_j)$. In both cases we have that $V(S_i) \cap V(S_j) \neq \emptyset$. Now suppose $V(S_i) \cap V(S_j) \neq \emptyset$. Note that, by construction, two stars that intersect either have the same center or they intersect in a vertex that is a leaf of both of them. If S_i and S_j have the same center B, then $v_i, v_j \in B$ and hence, by item (i), $v_iv_j \in E(G)$. If S_i and S_j have a common leaf, then this leaf is a vertex originated by the 1-subdivision of an edge. Then, there exist $A, B \in V(\mathcal{T})$ such that $v_i \in \mathrm{bd}(A, B)$ and $v_j \in \mathrm{bd}(B, A)$ and thus, by item (ii), $v_iv_j \in E(G)$. □

From the definition of well-partitioned chordal graphs, one can also see that every split graph is a well-partitioned chordal graph. Indeed, if G is a split graph with clique C and independent set I, the partition tree of G will be a star, with the clique C as its central bag and each vertex of I contained in a different leaf bag. We show that, in fact, every starlike graph is a well-partitioned chordal graph. A *starlike graph* [35] is an intersection graph of substars of a star.

Proposition 3.2. *Every starlike graph is a well-partitioned chordal graph.*

Proof. Let G be a starlike graph with $V(G) = \{v_i \mid i \in [n]\}$ and let S be the host star of the substar intersection model of G and S_i be the substar of S associated with vertex v_i. We may assume that G is connected and every vertex of S is contained in some substar of the intersection model.

We know that $v_iv_j \in E(G)$ if and only if $V(S_i) \cap V(S_j) \neq \emptyset$. To show that G is a well-partitioned chordal graph, we will construct a partition tree for G. Let c be the center of S and $f_1, \dots, f_k$ be its leaves. The partition tree $\mathcal{T}$ for G will be a star with center C and leaves $F_1, \dots, F_k$ such that $C = \{v_i \in V(G) \mid c \in S_i\}$ and $F_j = \{v_i \in V(G) \mid V(S_i) = \{f_j\}\}$. Note that this is indeed a partition of the vertex set of G, since each substar of S either contains the center or consists of a single leaf and every vertex of S is contained in some substar of the intersection model. Now we show this is indeed a partition tree for G. Note that, by construction, each bag is a clique, so item (i) holds. Also note that, for every i, if $v \in F_i$, then $N_G(v) \subseteq F_i \cup C$, thus item (iii) of the definition holds. Finally, note that the vertices of F_i are true twins in G, since the substars of S corresponding to those vertices consist of a single vertex, namely f_i. Hence, item (ii) also holds. We conclude that $\mathcal{T}$ is a partition tree for G. □

We will show that the graph O_1 in Fig. 2 is not a well-partitioned chordal graph. On the other hand, it is not difficult to see that O_1 is a substar graph. Also note that a path graph on 5 vertices is a well-partitioned chordal graph but not a starlike graph. These observations together with Propositions 3.1 and 3.2 show that we have the following hierarchy of graph classes between split graphs and chordal graphs:

$$\text{split graphs} \subsetneq \text{starlike graphs} \subsetneq \text{well-partitioned chordal graphs} \subsetneq \text{substar graphs} \subsetneq \text{chordal graphs}$$

4. Characterization by forbidden induced subgraphs

This section is entirely devoted to the proof of Theorem 1.1. That is, we show that the set $\mathbb{O}$ of graphs depicted in Fig. 2 is the set of all minimal forbidden induced subgraphs for well-partitioned chordal graphs, and give a polynomial-time recognition algorithm for this graph class. For convenience, we say that an induced subgraph of a graph that is isomorphic to a graph in $\mathbb{O}$ is an *obstruction* for well-partitioned chordal graphs, or simply an obstruction.

In Subsection 4.1, we show that the graphs in $\mathbb{O}$ are not well-partitioned chordal graphs (Proposition 4.2). In Subsection 4.2, we introduce the notion of a boundary-crossing path which is the main tool for devising the polynomial-time recognition algorithm. The resulting algorithm is in fact a *certifying* algorithm [45], meaning that it always outputs a certificate together with the Yes/No-answer on any input. In case of a Yes-instance the algorithm provides a partition tree and in case of a No-instance it outputs an obstruction. We present it in Subsection 4.3, which also concludes the proof of the characterization by forbidden induced subgraphs for well-partitioned chordal graphs.

It is not difficult to observe that no graph in $\mathbb{O}$ contains another graph in $\mathbb{O}$ as an induced subgraph. We remark that the results in Subsection 4.1 imply that graphs in $\mathbb{O}$ are minimal graphs with respect to the induced subgraph relation that are not well-partitioned chordal graphs.

The *diamond* graph is the graph obtained from K_4 by removing an edge. Note that for all $s \in [3]$, $t \geq 0$, the graph $W_{s,t}$ in $\mathbb{O}$ (see Fig. 2) contains two diamonds as induced subgraphs.

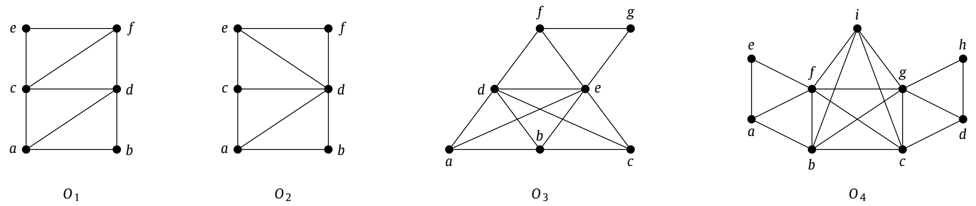


Fig. 3. Labellings of graphs O_1, O_2, O_3, and O_4.

4.1. Graphs in $\mathbb{O}$ are not well-partitioned chordal graphs

To argue that none of the graphs in $\mathbb{O}$ is a well-partitioned chordal graph, we make the following observation about cliques, which follows immediately from the definition of the partition tree.

Observation 4.1. *Let G be a connected well-partitioned chordal graph, and D be a clique in G. In any partition tree $\mathcal{T}$ of G, there are at most two bags whose intersection with D is non-empty.*

Given a connected well-partitioned chordal graph G and a clique D in G, we say that a partition tree of G *respects* D if it contains a bag having all the vertices of D. For a non-empty proper subset $D' \subset D$, we say that a partition tree *splits* D *into* $(D', D \setminus D')$ if it contains two distinct bags B_1 and B_2 such that $B_1 \cap D = D'$ and $B_2 \cap D = D \setminus D'$. If a partition tree splits D into $(D', D \setminus D')$ for some $D' \subset D$, then we may simply say that it *splits* D. By Observation 4.1, each partition tree either respects or splits each clique.

For $s \in [3]$ and $t \geq 0$, the vertex set of a block of $W_{s,t}$ having more than 3 vertices is called a *wing* of $W_{s,t}$.

Proposition 4.2. *The graphs in $\mathbb{O}$ are not well-partitioned chordal graphs.*

Proof. For $k \geq 4$, H_k is not a chordal graph, so it is not a well-partitioned chordal graph.

We prove an auxiliary claim that will be useful to show that the graphs O_1, O_2, O_3, and O_4 in $\mathbb{O}$ are not well-partitioned chordal graphs.

Claim 4.2.1. *Let H be a connected graph and $D = \{x, y, z\} \subseteq V(H)$ be a clique in H.*

(i) If there are adjacent vertices $u, v \in V(H) \setminus D$ such that $D \nsubseteq N_H(u)$ and $D \nsubseteq N_H(v)$, and $\emptyset \neq N_H(u) \cap D \neq N_H(v) \cap D \neq \emptyset$, then H has no partition tree respecting D.
(ii) If there exists a vertex $u \in V(H) \setminus D$ such that $N_H(u) \cap D = \{y, z\}$, then H has no partition tree splitting D into $(\{x, y\}, \{z\})$.
(iii) If there exist two non-adjacent vertices $u, v \in V(H) \setminus D$ such that $N_H(u) \cap D = D = N_H(v) \cap D$, then H has no partition tree splitting D.

Proof. In order to prove item (i), suppose there is a partition tree $\mathcal{T}$ of H respecting D, and let B be the bag containing D. First, since $D \nsubseteq N_H(u)$ and $D \nsubseteq N_H(v)$, we have that neither v nor u is contained in B as B is a clique in H. Furthermore, since $N_H(u) \cap D \neq \emptyset$ and $N_H(v) \cap D \neq \emptyset$, and since $uv \in E(G)$, it cannot be the case that u and v are in distinct bags, otherwise there would be a triangle in $\mathcal{T}$. However, since $N_H(u) \cap D \neq N_H(v) \cap D$, u and v cannot be in the same bag either.

Now we proceed to the proof of item (ii). Suppose there is a partition tree $\mathcal{T}$ of H that splits D into $(\{x, y\}, \{z\})$, and denote the two bags intersecting D by B_1 and B_2 with $B_1 \cap D = \{x, y\}$ and $B_2 \cap D = \{z\}$. Since u is not adjacent to x, $u \notin B_1$. Since $x \in N_H(z) \cap B_1$ and $x \notin N_H(u) \cap B_1$, u cannot be contained in B_2 either. However, since $uz, uy \in E(G)$, if u is in a bag other than B_1 and B_2, then $\{u, y, z\}$ is a clique that intersects three distinct bags of $\mathcal{T}$, a contradiction with Observation 4.1.

To conclude, we prove item (iii). Suppose there is a partition tree $\mathcal{T}$ of H that splits D, and again denote the two bags intersecting D by B_1 and B_2, with $B_1 \cap D = \{x, y\}$ and $B_2 \cap D = \{z\}$. First, since u and v are non-adjacent, they cannot be in the same bag. Furthermore, there cannot be a bag $B_3 \in V(\mathcal{T}) \setminus \{B_1, B_2\}$ such that $\{u, v\} \cap B_3 \neq \emptyset$: both u and v have neighbors in B_1 and in B_2, so this would imply the existence of a clique that intersects three distinct bags of $\mathcal{T}$ (B_1, B_2, and B_3). The last case that remains is when $u \in B_1$ and $v \in B_2$. However, in this case, B_1 contains a vertex that is adjacent to v, namely x, and a vertex that is not adjacent to v, namely u, a contradiction. ⌟

Now, let us consider the obstructions O_1, O_2, O_3 and O_4 and assume that their vertices are labeled as in Fig. 3.

By Observation 4.1, each partition tree either respects or splits every clique. First, consider the graph O_1 and the clique $D = \{a, c, d\}$. Because of the vertices e and f, we can observe that, by Claim 4.2.1(i), no partition tree of O_1 respects D. Furthermore, we obtain by Claim 4.2.1(ii) that no partition tree splits D into $(\{a, c\}, \{d\})$, $(\{a, d\}, \{c\})$, and $(\{c, d\}, \{a\})$ because of the vertices b, f, and b, respectively. Thus, no partition tree of O_1 splits D. Hence, O_1 does not admit a partition tree and therefore it is not a well-partitioned chordal graph.

For O_2, consider again the clique $\{a, c, d\}$. The arguments are similar to the previous ones, except that the vertex e should be used to show that no partition tree splits $\{a, c, d\}$ into $(\{a, d\}, \{c\})$.

For O_3, consider the clique $D = \{b, d, e\}$. Because of the vertices f and g, we observe that, by Claim 4.2.1(i), no partition tree of O_3 respects D. On the other hand, because of a and c, we observe that by Claim 4.2.1(iii), no partition tree of O_3 splits D. Hence, O_3 is not a well-partitioned chordal graph.

For O_4, consider the clique $D = \{b, c, g\}$. Because of the vertices d and h, we conclude by Claim 4.2.1(i) that no partition tree of O_4 respects D. Since $N_{O_4}(d) \cap D = \{c, g\} = D \setminus \{b\}$, by Claim 4.2.1(ii), no partition tree of O_4 splits D into $(\{b, c\}, \{g\})$ or $(\{b, g\}, \{c\})$. Thus, we may assume that each partition tree splits D into $(\{c, g\}, \{b\})$. Let B_1 and B_2 be the two bags such that $B_1 \cap D = \{c, g\}$ and $B_2 \cap D = \{b\}$.

Since $hc \notin E(G)$, we have that $h \notin B_1$. Also, since $N_{O_4}(h) \cap \{g, c\} \neq N_{O_4}(d) \cap \{g, c\}$, h and d cannot be in the same bag. Thus, we conclude that $d \in B_1$, otherwise $\{d, h, g\}$ would be a clique that intersects three distinct bags of $\mathcal{T}$. By considering the clique $\{b, c, f\}$, we conclude by symmetry that $\{a, b, f\}$ is contained in the same bag, which is B_2. As i is adjacent to neither a nor d, the bag containing i forms a clique with B_1 and B_2, a contradiction. We conclude that O_4 is not a well-partitioned chordal graph.

Next, we show that for all $s \in [3]$ and $t \geq 0$, $W_{s,t}$ is not a well-partitioned chordal graph. We first claim the following.

Claim 4.2.2. *For each $s \in [3]$ and $t \geq 0$, $W_{s,t}$ has no partition tree having a bag whose intersection with its wing consists of only the cut vertex contained in the wing.*

Proof. Suppose there are a partition tree $\mathcal{T}_1$ of $W_{1,t}$, a bag B of $\mathcal{T}_1$, and a wing $\{a, b, c, d\}$ of $W_{1,t}$ such that

- d is a cut vertex of $W_{1,t}$,
- $ac \notin E(W_{1,t})$, and
- $B \cap \{a, b, c, d\} = \{d\}$.

This implies that there exists a bag B_1 such that $\{a, b\} \subseteq B_1$, otherwise $\{a, b, d\}$ would be a clique that intersects three distinct bags of $\mathcal{T}_1$. Since a and c are non-adjacent, $c \notin B_1$ and, by assumption, $c \notin B$. Thus $\{b, c, d\}$ is a clique intersecting three bags of $\mathcal{T}_1$, a contradiction with Observation 4.1.

Next, suppose there is a partition tree $\mathcal{T}_2$ of $W_{2,t}$ that contains a bag B whose intersection with a wing of $W_{2,t}$ consists of the cut vertex alone. If the affected wing is isomorphic to the diamond, then the argument follows from the same argument given before. Now, assume that $\{a, b, c, d, e, f\}$ is the affected wing where

- d is a cut vertex of $W_{2,t}$,
- $\{a, b, c, d\}$ is a clique,
- $N_{W_{2,t}}(e) = \{b, c, f\}$ and $N_{W_{2,t}}(f) = \{c, e\}$.

First, there must exist a bag B_1 containing $\{a, b, c\}$, otherwise there is a clique violating Observation 4.1. Since neither e nor f is adjacent to a, $\{e, f\} \cap B_1 = \emptyset$. Since $N_{W_{2,0}}(e) \cap B_1 \neq N_{W_{2,0}}(f) \cap B_1$, by Claim 4.2.1(i), there is no partition tree respecting $\{a, b, c\}$, a contradiction.

The claim regarding $W_{3,t}$ follows as well by noting that the wings of $W_{3,t}$ are isomorphic to the one considered in the latter case. ⌟

Claim 4.2.3. *For each $s \in [3]$ and $t \geq 0$, $W_{s,t}$ is not a well-partitioned chordal graph.*

Proof. Suppose that there is a partition tree $\mathcal{T}$ of $W_{s,t}$. Let A_1 and A_2 be the wings of $W_{s,t}$, and for $i \in [2]$, let d_i be the cut vertex of $W_{s,t}$ contained in A_i. By Claim 4.2.2, we can assume that the bag B containing d_1 satisfies $|B \cap A_1| \geq 2$. If $d_1 = d_2$, then $|B \cap A_2| = 1$ and this contradicts Claim 4.2.2. So, we may assume that $d_1 \neq d_2$.

Now, let $C_1, C_2, \ldots, C_m$ be the set of distinct triangles in $W_{s,t}$ such that

- $d_1 \in V(C_1) \setminus V(C_2)$ and $d_2 \in V(C_m) \setminus V(C_{m-1})$, and
- for each $i \in [m-1]$, C_i and C_{i+1} intersect.

For convenience, let $C_0 = A_1$ and $C_{m+1} = A_2$. Let q be the largest integer j in $\{0\} \cup [m]$ such that the bag containing $V(C_j) \cap V(C_{j+1})$ has at least two vertices of C_j. Such an integer exists since $j = 0$ satisfies the condition.

We claim that $q = m$. Assume that $q < m$, and let B be the bag containing $V(C_q) \cap V(C_{q+1})$. Since $q < m$, C_{q+1} is a triangle. Because of Observation 4.1, there must exist a bag containing two other vertices of C_{q+1}. This implies that $q + 1$ also satisfies the condition.

Thus, $q = m$. But this contradicts Claim 4.2.2 for the wing A_2. We conclude that $W_{s,t}$ is not a well-partitioned chordal graph. ⌟

This concludes the proof of Proposition 4.2. □

4.2. Boundary-crossing paths

In the remaining part of this section, we present the certifying algorithm for well-partitioned chordal graphs. Here, we define the main concept of a boundary-crossing path and prove some useful lemmas.

Let G be a connected well-partitioned chordal graph with partition tree $\mathcal{T}$. For a bag X of $\mathcal{T}$ and $B \subseteq X$, a vertex $z \in V(G) \setminus X$ is said to *cross* B *in* X, if it has neighbors both in B and in $X \setminus B$. In this case, we also say that B has a *crossing vertex*. In the following definitions, a path $X_1X_2 \cdots X_\ell$ in $\mathcal{T}$ is considered to be ordered from X_1 to X_ℓ. Let $\ell \geq 3$ be an integer. A path $X_1X_2 \cdots X_\ell$ in $\mathcal{T}$ is called a *boundary-crossing path* if for each $i \in [\ell-2]$, there is a vertex in X_i that crosses $\mathrm{bd}(X_{i+1}, X_{i+2})$. A boundary-crossing path $X_1X_2 \cdots X_\ell$ in $\mathcal{T}$ is *exclusive* if

- for each $i \in [\ell-2]$, there is no bag $Y \in V(\mathcal{T}) \setminus \{X_i\}$ containing a vertex that crosses $\mathrm{bd}(X_{i+1}, X_{i+2})$,

and it is *complete* if

- for each $i \in [\ell-2]$, $\mathrm{bd}(X_i, X_{i+1})$ is complete to X_{i+1}.

If a boundary-crossing path is both complete and exclusive, then we call it *good*. For convenience, we say that any path in $\mathcal{T}$ with at most two bags is a boundary-crossing path.

The outline of the algorithm is as follows. First we may assume that a given graph G is chordal, as we can detect a hole in polynomial time using Theorem 2.1 if it exists. We may also assume that G is connected. So, it has a simplicial vertex v, and by an inductive argument, we can assume that $G - v$ is a well-partitioned chordal graph. As v is simplical, $G - v$ is also connected, and thus it admits a partition tree $\mathcal{T}$. If v has neighbors only in a single bag of $\mathcal{T}$, say B, then we can simply add one new bag B_v only containing v to $\mathcal{T}$, and add an edge between B_v and B. The resulting tree is a partition tree of G. Thus, we may assume that v has neighbors in two distinct bags, say C_1 and C_2. Then our algorithm is divided into three parts:

1. We find a maximal good boundary-crossing path ending in C_2C_1 (or C_1C_2). To do this, when we currently have a good boundary-crossing path $C_iC_{i-1} \cdots C_2C_1$, find a bag C_{i+1} containing a vertex crossing $\mathrm{bd}(C_i, C_{i-1})$. If there is no such bag, then this path is maximal. Otherwise, we argue that in polynomial time either we can find an obstruction, or verify that $C_{i+1}C_i \cdots C_2C_1$ is good.
2. Assume that $C_kC_{k-1} \cdots C_2C_1$ is the obtained maximal good boundary-crossing path. Then we can in polynomial time modify $\mathcal{T}$ so that no vertex crosses $\mathrm{bd}(C_2, C_1)$.
3. We show that if no vertex crosses $\mathrm{bd}(C_2, C_1)$ and no vertex crosses $\mathrm{bd}(C_1, C_2)$, then we can extend $\mathcal{T}$ to a partition tree of G.

Regarding Step 2, Lemma 4.3 shows that when a maximal good boundary-crossing path $C_kC_{k-1} \cdots C_2C_1$ is given, we can modify $\mathcal{T}$ to a partition tree $\mathcal{T}'$ such that no vertex crosses $\mathrm{bd}(C_2', C_1')$, where C_1' and C_2' are the bags in $\mathcal{T}'$ that correspond to C_1 and C_2 in $\mathcal{T}$, respectively – in particular, they are the bags containing the neighbors of v.

Lemma 4.3. *Let G be a graph, v be a simplicial vertex, and $G - v$ be a connected well-partitioned chordal graph with partition tree $\mathcal{T}$ such that v has neighbors in two distinct bags C_1 and C_2. Let $C_kC_{k-1} \cdots C_1$ be a good boundary-crossing path for some integer $k \geq 3$ such that no vertex crosses $\mathrm{bd}(C_k, C_{k-1})$. One can in polynomial time output a partition tree $\mathcal{T}'$ of $G - v$ that contains a good boundary-crossing path $C_{k-1}'C_{k-2} \cdots C_1$ such that no vertex in $G - v$ crosses $\mathrm{bd}(C_{k-1}', C_{k-2})$.*

Proof. Since no vertex crosses $\mathrm{bd}(C_k, C_{k-1})$, we can partition the neighbors of C_k in $\mathcal{T}$ into $\mathcal{S}_1$ and $\mathcal{S}_2$ such that for all $S_1 \in \mathcal{S}_1$, we have that $\mathrm{bd}(C_k, S_1) \subseteq C_k \setminus \mathrm{bd}(C_k, C_{k-1})$, and for all $S_2 \in \mathcal{S}_2$, $\mathrm{bd}(C_k, S_2) \subseteq \mathrm{bd}(C_k, C_{k-1})$. Let $C_k' := C_k \setminus \mathrm{bd}(C_k, C_{k-1})$ and $C_{k-1}' := C_{k-1} \cup \mathrm{bd}(C_k, C_{k-1})$. We obtain $\mathcal{T}'$ from $\mathcal{T}$ as follows.

- Remove C_k and C_{k-1}, and add C_k' and C_{k-1}'.
- Make all bags that have been adjacent to C_{k-1} in $\mathcal{T}$ adjacent to C_{k-1}'.
- Make all bags in $\mathcal{S}_1$ adjacent to C_k', and all bags in $\mathcal{S}_2$ adjacent to C_{k-1}'.

Since $\mathrm{bd}(C_k, C_{k-1})$ is complete to C_{k-1}, C_{k-1}' is indeed a clique in $G - v$, and thus we conclude that $\mathcal{T}'$ is a partition tree of $G - v$. Since C_{k-1}' contains C_{k-1} and there is no edge between $\mathrm{bd}(C_k, C_{k-1})$ and C_{k-2}, we know that $C_{k-1}'C_{k-2} \cdots C_1$ is a good boundary-crossing path. Clearly, $\mathcal{T}'$ can be obtained in polynomial time.

We claim that no vertex crosses $\mathrm{bd}(C_{k-1}', C_{k-2})$. Suppose for a contradiction that there exists a vertex $q \in V(G - v) \setminus C_{k-1}'$ that crosses $\mathrm{bd}(C_{k-1}', C_{k-2})$. We consider two cases. First, we assume q also crosses $\mathrm{bd}(C_{k-1}, C_{k-2})$. Since $C_k \cdots C_1$ is exclusive, any vertex crossing $\mathrm{bd}(C_{k-1}, C_{k-2})$ is in $\mathrm{bd}(C_k, C_{k-1})$. This means that $q \in C_{k-1}'$, a contradiction. Now assume

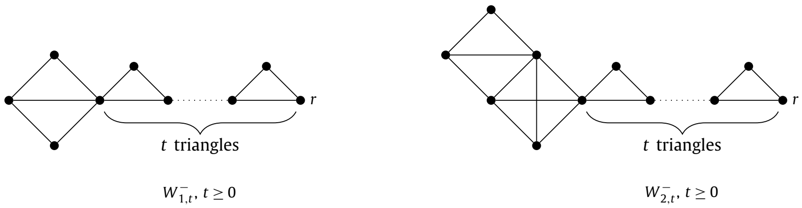


Fig. 4. The graphs $W^-_{1,t}$ and $W^-_{2,t}$.

that q is a vertex in $V(G-v) \setminus (C_k \cup C_{k-1})$ that is adjacent to a vertex in $\mathrm{bd}(C'_{k-1}, C_{k-2}) = \mathrm{bd}(C_{k-1}, C_{k-2})$ and a vertex in $C'_{k-1} \setminus C_{k-1} = \mathrm{bd}(C_k, C_{k-1})$. But this would mean that there is a triangle in $\mathcal{T}$, a contradiction.

We conclude that no vertex in $G-v$ crosses $\mathrm{bd}(C'_{k-1}, C_{k-2})$. □

With respect to Step 3, we prove the following lemma.

Lemma 4.4. *Let G be a graph, v be a simplicial vertex, and $G-v$ be a connected well-partitioned chordal graph with partition tree $\mathcal{T}$ such that v has neighbors in two distinct bags C_1 and C_2. If every vertex of $G-v$ crosses neither $\mathrm{bd}(C_1, C_2)$ nor $\mathrm{bd}(C_2, C_1)$, then one can output a partition tree for G in polynomial time.*

Proof. Assume that every vertex of $G-v$ crosses neither $\mathrm{bd}(C_1, C_2)$ nor $\mathrm{bd}(C_2, C_1)$. Let $\mathcal{S}_1$ denote all neighbors of C_1 in $\mathcal{T}$ such that for each $S_1 \in \mathcal{S}_1$, $\mathrm{bd}(C_1, S_1) \subseteq C_1 \setminus \mathrm{bd}(C_1, C_2)$; let $\mathcal{S}_2$ denote the set of all neighbors of C_2 in $\mathcal{T}$ such that for each $S_2 \in \mathcal{S}_2$, $\mathrm{bd}(C_2, S_2) \subseteq C_2 \setminus \mathrm{bd}(C_2, C_1)$; and let $\mathcal{S}_{12} := (N_{\mathcal{T}}(C_1) \cup N_{\mathcal{T}}(C_2)) \setminus \{C_1, C_2\} \setminus \mathcal{S}_1 \setminus \mathcal{S}_2$. Since every vertex of $G-v$ crosses neither $\mathrm{bd}(C_1, C_2)$ nor $\mathrm{bd}(C_2, C_1)$, for every $S \in \mathcal{S}_{12}$, we have $N_G(S) \cap (C_1 \cup C_2) \subseteq \mathrm{bd}(C_1, C_2) \cup \mathrm{bd}(C_2, C_1)$.

Now, let $C'_1 := C_1 \setminus \mathrm{bd}(C_1, C_2)$, $C'_2 := C_2 \setminus \mathrm{bd}(C_2, C_1)$, and $C'_{12} := \mathrm{bd}(C_1, C_2) \cup \mathrm{bd}(C_2, C_1)$. We obtain $\mathcal{T}'$ from $\mathcal{T}$ as follows.

- Remove C_1 and C_2; add C'_1, C'_2, and C'_{12}; make C'_1 and C'_2 adjacent to C'_{12}.
- Make all bags in $\mathcal{S}_1$ adjacent to C'_1, all bags in $\mathcal{S}_2$ adjacent to C'_2, and all bags in $\mathcal{S}_{12}$ adjacent to C'_{12}.
- Add a new bag $C_v := \{v\}$, and make it adjacent to C'_{12}.

This yields a partition tree of G. □

Considering Step 1, we present some lemmas useful to find an obstruction. To describe subparts of the long obstructions $W_{s,t}$, we use the graphs $W^-_{1,t}$ and $W^-_{2,t}$ as shown in Fig. 4. Note that each of them has a distinguished vertex r, that we call *terminal*.

The following lemma will be useful to find a wing at the beginning of a boundary-crossing path.

Lemma 4.5. *Let G be a connected well-partitioned chordal graph with partition tree $\mathcal{T}$. Let XYZ be a boundary-crossing path in $\mathcal{T}$ such that $\mathrm{bd}(Y, Z)$ is complete to Z, and B be a non-empty proper subset of Z. Suppose that one of the following conditions does not hold.*

(i) $\mathrm{bd}(X, Y)$ *is complete to Y.*
(ii) There is no bag $X' \in V(\mathcal{T}) \setminus \{X\}$ that contains vertices crossing $\mathrm{bd}(Y, Z)$.

Then one can in polynomial time output an induced subgraph H of $G[X \cup X' \cup Y \cup Z]$ for some neighbor X' of Y in $\mathcal{T}$ (X' can be X) that is isomorphic to $W^-_{1,1}$ or $W^-_{2,0}$, with the terminal vertex being mapped to a vertex in B, say r_H, such that $V(H) \cap B = \{r_H\}$.

Proof. Let $x \in \mathrm{bd}(X, Y)$ be a vertex that crosses $\mathrm{bd}(Y, Z)$. Choose a neighbor y in $\mathrm{bd}(Y, Z)$ and a neighbor y' in $Y \setminus \mathrm{bd}(Y, Z)$ of x. Since $\mathrm{bd}(Y, Z)$ is complete to Z by assumption, y has a neighbor in B and a neighbor in $Z \setminus B$. Let z and z' be these neighbors, respectively. We illustrate this situation and the following arguments in Fig. 5.

Suppose that (i) does not hold, i.e., that $\mathrm{bd}(X, Y)$, in particular the vertex x, is not complete to Y. Then, x has a non-neighbor, say y'' in Y. If $y'' \in Y \setminus \mathrm{bd}(Y, Z)$, then the set $\{x, y, y', y'', z, z'\}$ induces a $W^-_{1,1}$ with the terminal being mapped to z. See Fig. 5(a). On the other hand, if $y'' \in \mathrm{bd}(Y, Z)$, then $\{x, y, y', y'', z, z'\}$ induce a $W^-_{2,0}$ with the terminal vertex being mapped to z. See Fig. 5(b). So, we may assume that (i) holds.

Now suppose that (ii) does not hold, and let $x' \in X'$ be a vertex crossing $\mathrm{bd}(Y, Z)$. Then, x' has a neighbor $y \in \mathrm{bd}(Y, Z)$ and a neighbor $y' \in Y \setminus \mathrm{bd}(Y, Z)$. By (i), x is adjacent to y and y'. Then $G[X \cup X' \cup Y \cup Z]$ contains a $W^-_{1,1}$ with terminal z. To illustrate, one may think of x' in this case as y'' in Fig. 5(a), except the difference that now x' is in a new bag X', while y'' is in Y. □

We use the following lemmas to find an obstruction or extend a good boundary-crossing path.

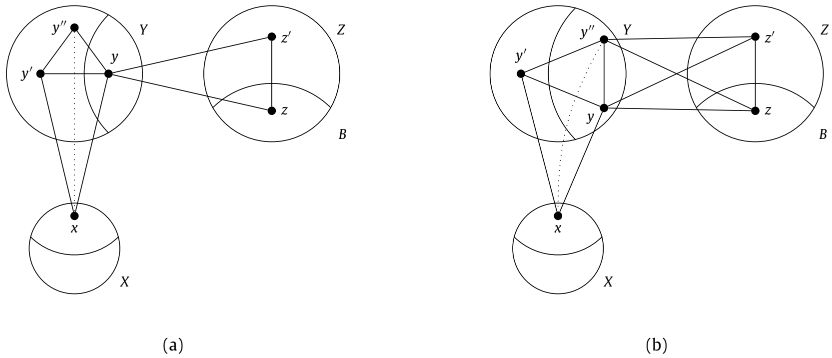


Fig. 5. Visual aides to the proof of Lemma 4.5.

Lemma 4.6. *Let G be a connected well-partitioned chordal graph with partition tree $\mathcal{T}$, and let B be a vertex set contained in some bag C_1. If $C_kC_{k-1}\cdots C_1$ is a boundary-crossing path for some $k \geq 2$ such that C_2 has a vertex that crosses B in C_1, then one can in polynomial time either*

1. *output an induced subgraph H isomorphic to $W^-_{s,t}$ for some $s \in [3]$ and $t \geq 0$ with terminal v such that $V(H) \cap B = \{v\}$,*
2. *output an induced subgraph H isomorphic to $W^-_{1,0}$ on $\{a, z_1, z_2, w\}$ such that both a and w have degree 2 in H, $a \in \mathrm{bd}(D, C_1)$ for some neighbor bag D of C_1, $z_2 \in C_1 \setminus B$, and $z_1, w \in B$, or*
3. *verify that it is a good boundary-crossing path such that $\mathrm{bd}(C_2, C_1)$ is complete to C_1 and no other bag contains a vertex crossing B in C_1.*

Proof. We prove the lemma by induction on k. Assume that $k = 2$. We check whether $\mathrm{bd}(C_2, C_1)$ is complete to C_1. Suppose not. Let $a \in \mathrm{bd}(C_2, C_1)$. Let z_1 be a neighbor of a in B, z_2 be a neighbor of a in $C_1 \setminus B$, and w be a non-neighbor of a in C_1. If $w \in C_1 \setminus B$, then $\{a, z_1, z_2, w\}$ induces $W^-_{1,0}$ with terminal z_1, so we have outcome 1. If $w \in B$, then $\{a, z_1, z_2, w\}$ induces a graph as in Case 2. Otherwise, we conclude that $\mathrm{bd}(C_2, C_1)$ is complete to C_1.

We find a bag $D \neq C_2$ in $\mathcal{T}$ containing a vertex d crossing B in C_1. If such a vertex d exists, then by the above procedure, we may assume that d is complete to C_1. Then similarly to the previous case when $w \in C_1 \setminus B$, again we can find an induced subgraph isomorphic to the diamond on $\{a, d, z_1, z_2\}$. If such a vertex does not exist, then we conclude that no other bag contains a vertex crossing B in C_1.

Now, we assume that $k \geq 3$. By the induction hypothesis, the claim holds for the path $C_{k-1}C_{k-2}\cdots C_1$. We can assume that it is good.

We check whether $\mathrm{bd}(C_k, C_{k-1})$ is not complete to C_{k-1}, and there is a bag $D \in V(\mathcal{T}) \setminus \{C_k\}$ that has a vertex crossing $\mathrm{bd}(C_{k-1}, C_{k-2})$. If neither of them holds, then we verified that $C_kC_{k-1}\cdots C_1$ is good. Assume one of two statements holds.

Let $X := \mathrm{bd}(C_{k-2}, C_{k-3})$ if $k \geq 4$ and $X = B$ if $k = 3$. Now, by applying Lemma 4.5 to the pair $(C_kC_{k-1}C_{k-2}, X)$, we can find an induced subgraph H isomorphic to $W^-_{1,1}$ or $W^-_{2,0}$ in $G[C_k \cup D \cup C_{k-1} \cup C_{k-2}]$ for some neighbor D of C_{k-1} in $\mathcal{T}$ so that its terminal r is mapped to some vertex in X and $V(H) \cap X = \{r\}$. If $k = 3$, then we have outcome 1 as $X = B$.

Assume $k \geq 4$. Let $x_{k-2} := r$. We recursively choose pairs of vertices (x_i, y_i) for $i \in [k-3]$ as follows. First assume $i > 1$ and x_{i+1} is defined but x_i is not defined yet. Then choose a neighbor x_i of x_{i+1} in $\mathrm{bd}(C_i, C_{i-1})$ and a neighbor y_i of x_{i+1} in $C_i \setminus \mathrm{bd}(C_i, C_{i-1})$. Such neighbors exist since x_{i+1} crosses $\mathrm{bd}(C_i, C_{i-1})$. When $i = 1$, choose a neighbor x_1 of x_2 in B and y_1 of x_2 in $C_1 \setminus B$. Then it is clear that $G[\{x_1, y_1, x_2, y_2, \ldots, x_{k-3}, y_{k-3}\} \cup V(H)]$ is isomorphic to $W^-_{s',t'}$ for some $s' \in \{1, 2\}$ and $t' \geq 0$ with terminal x_1 such that its intersection on B is exactly x_1. This concludes the lemma. $\square$

Lemma 4.7. *Let G_1 and G_2 be two connected graphs with non-empty sets $A \subseteq V(G_1)$ and $B \subseteq V(G_2)$, and G be the graph obtained from the disjoint union of G_1 and G_2 by adding all edges between A and B such that*

- *for every $v \in B$, $G[V(G_1) \cup \{v\}]$ is isomorphic to $W^-_{s,t}$ with terminal v for some $s \in \{1, 2\}$ and $t \geq 0$,*
- *G_2 is a well-partitioned chordal graph with partition tree $\mathcal{T}$ such that B is contained in some bag C_1.*

Then the following two statements hold.

(1) If $C_kC_{k-1}\cdots C_1$ is a boundary-crossing path in $\mathcal{T}$ for some $k \geq 2$ such that C_2 has a vertex that crosses B in C_1, then one can in polynomial time either output an obstruction in G, or verify that it is a good boundary-crossing path such that $\mathrm{bd}(C_2, C_1)$ is complete to C_1 and no other bag contains a vertex crossing B in C_1.

(2) If C_2C_1 is a boundary-crossing path in $\mathcal{T}$, that is, an edge in $\mathcal{T}$, then one can in polynomial time either output an obstruction in G, or find a maximal good boundary-crossing path ending in C_2C_1 such that $\text{bd}(C_2, C_1)$ is complete to C_1 and no other bag contains a vertex crossing B in C_1.

Proof. We prove (1). Applying Lemma 4.6 to G_2 and B, we conclude that in polynomial time, we can either

1. output an induced subgraph H isomorphic to $W^-_{s,t}$ for some $s \in [3]$ and $t \geq 0$ with terminal v such that $V(H) \cap B = \{v\}$,
2. output an induced subgraph H isomorphic to the diamond on $\{a, z_1, z_2, w\}$ such that a, w have degree 2 in H, $a \in \text{bd}(D, C_1)$ for some neighbor bag D of C_1, $z_2 \in C_1 \setminus B$, and $z_1, w \in B$, or
3. verify that it is a good boundary-crossing path such that $\text{bd}(C_2, C_1)$ is complete to C_1 and no other bag contains a vertex crossing B in C_1.

For case (i) it is clear that together with an obstruction $W^-_{s,t}$ in $G[V(G_1) \cup \{v\}]$ given by the assumption, $G[V(H) \cup V(G_1)]$ is isomorphic to $W_{s,t}$ for some $s \in [3]$ and $t \geq 0$. For case (ii), we can observe that $G[V(H) \cup V(G_1)]$ is an obstruction as follows.

- If $G[V(G_1) \cup \{z_1\}]$ is isomorphic to $W^-_{1,0}$, then $G[V(G_1) \cup \{a, w, z_1, z_2\}]$ is isomorphic to O_3.
- If $G[V(G_1) \cup \{z_1\}]$ is isomorphic to $W^-_{2,0}$, then $G[V(G_1) \cup \{a, w, z_1, z_2\}]$ is isomorphic to O_4.
- If $G[V(G_1) \cup \{z_1\}]$ is isomorphic to $W^-_{s,t}$ for some $s \in \{1, 2\}$ and $t \geq 1$, then $G[V(G_1) \cup \{a, w, z_1, z_2\}]$ is isomorphic to $W_{s',t-1}$ for some $s' \in \{2, 3\}$.

It shows the statement (1).

Now, we show (2). By (1), we can in polynomial time either output an obstruction, or verify that $\text{bd}(C_2, C_1)$ is complete to C_1 and no other bag crosses $\text{bd}(C_2, C_1)$. For $i \geq 3$, we recursively find a neighbor bag C_i of C_{i-1} that has a vertex crossing $\text{bd}(C_{i-1}, C_{i-2})$. If there is such a bag C_i, then by applying (1), one can in polynomial time find an obstruction or guarantee that $C_iC_{i-1} \cdots C_1$ is good. As the graph is finite, this procedure terminates with some path $C_kC_{k-1} \cdots C_1$ such that it is good and no vertex crosses $\text{bd}(C_k, C_{k-1})$, unless we found an obstruction. □

4.3. A certifying algorithm

In this subsection, we prove the following.

Proposition 4.8. *Given a connected graph G, one can in polynomial time either output an obstruction in G or output a partition tree of G confirming that G is a well-partitioned chordal graph.*

As explained in Subsection 4.2, we mainly consider the case when G is a connected chordal graph, v is a simplicial vertex of G and $G - v$ is a connected well-partitioned chordal graph with partition tree $\mathcal{T}$, and v has neighbors in two distinct bags C_1 and C_2. Throughout the following, we assume these conditions on G, v, $\mathcal{T}$, C_1, and C_2, so we omit them in the statements of lemmas and the like. We deal with the following three cases:

- Case 1: $C_1 \subseteq N_G(v)$.
- Case 2: $\text{bd}(C_1, C_2) \setminus N_G(v) \neq \emptyset$ and $C_2 \setminus N_G(v) \neq \emptyset$.
- Case 3: $C_1 \setminus N_G(v) \neq \emptyset$, $C_2 \setminus N_G(v) \neq \emptyset$ and $N_G(v) = \text{bd}(C_1, C_2) \cup \text{bd}(C_2, C_1)$.

In each case, we show that one can in polynomial time either find an obstruction or output a partition tree of G. This is proved in Lemmas 4.9, 4.10, and 4.11, respectively. We give a proof of Proposition 4.8 assuming that these lemmas hold.

Proof of Proposition 4.8. We apply Theorem 2.1 to find a hole in G if one exists. We may assume that G is chordal. Since a graph is a well-partitioned chordal graph if and only if its connected components are well-partitioned chordal graphs, it is sufficient to show it for each connected component. From now on, we assume that G is connected. Using the algorithm in Theorem 2.2, we can find a perfect elimination ordering $(v_1, v_2, \ldots, v_n)$ of G in polynomial time.

For each $i \in [n]$, let $G_i := G[\{v_i, v_{i+1}, \ldots, v_n\}]$. Observe that since G is connected and v_i is simplicial in G_i for all $1 \leq i \leq n-1$, each G_i is connected. From $i = n$ to 1, we recursively find either an obstruction or a partition tree of G_i. Clearly, G_n admits a partition tree. Let $1 \leq i \leq n-1$, and assume that we obtained a partition tree $\mathcal{T}$ of G_{i+1}.

Since v_i is simplicial in G_i, $N_{G_i}(v_i)$ is a clique. This implies that there are at most two bags in $V(\mathcal{T})$ that have a non-empty intersection with $N_{G_i}(v_i)$. If there is only one such bag in $V(\mathcal{T})$, say C, we can construct a partition tree of G_i by simply adding a bag consisting of v_i and making it adjacent to C.

Hence, from now on, we can assume that there are precisely two distinct adjacent bags $C_1, C_2 \in V(\mathcal{T})$ that have a non-empty intersection with $N_{G_i}(v_i)$. As $N_{G_i}(v_i)$ is a clique, we can observe that $N_{G_i}(v_i) \subseteq \text{bd}(C_1, C_2) \cup \text{bd}(C_2, C_1)$.

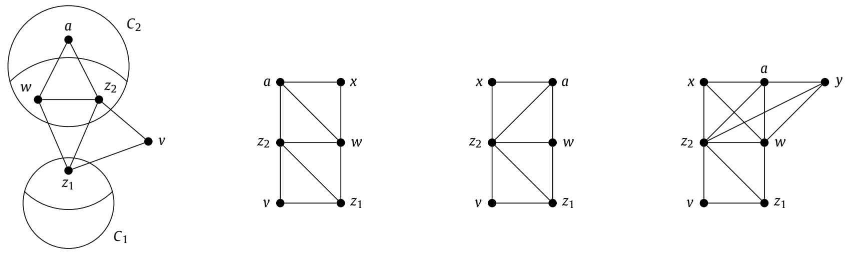


Fig. 6. Proof of Claim 4.9.1.

If $C_1 \subseteq N_{G_i}(v_i)$ or $C_2 \subseteq N_{G_i}(v_i)$, then by Lemma 4.9, we can in polynomial time either output an obstruction or output a partition tree of G_i. Thus, we may assume that $C_1 \setminus N_{G_i}(v_i) \neq \emptyset$ and $C_2 \setminus N_{G_i}(v_i) \neq \emptyset$. If $\mathrm{bd}(C_1, C_2) \setminus N_{G_i}(v_i) \neq \emptyset$ or $\mathrm{bd}(C_2, C_1) \setminus N_{G_i}(v_i) \neq \emptyset$, then by Lemma 4.10, we can in polynomial time either output an obstruction or output a partition tree of G_i. Thus, we may further assume that $\mathrm{bd}(C_1, C_2) \setminus N_{G_i}(v_i) = \emptyset$ and $\mathrm{bd}(C_2, C_1) \setminus N_{G_i}(v_i) = \emptyset$. Then by Lemma 4.11, we can in polynomial time either output an obstruction or output a partition tree of G_i, and this concludes the proposition. □

Now, we focus on proving the three lemmas.

Lemma 4.9. *If $C_1 \subseteq N_G(v)$, then one can in polynomial time either output an obstruction in G or output a partition tree of G confirming that G is a well-partitioned chordal graph.*

Proof. Since v is a simplicial vertex, we have that $\mathrm{bd}(C_1, C_2) = C_1$. If $N_G(v) \cap C_2 = \mathrm{bd}(C_2, C_1)$, then we can obtain a partition tree of G by adding v to C_1. Thus, we may assume that $N_G(v) \cap C_2 \neq \mathrm{bd}(C_2, C_1)$.

Assume that $C_2 = \mathrm{bd}(C_2, C_1)$. Since $\mathrm{bd}(C_2, C_1)$ is complete to C_1, we have that $C_1 \cup C_2$ is a clique. Hence, we can obtain a partition tree $\mathcal{T}'$ of G from $\mathcal{T}$ by removing C_1 and C_2, adding a new bag $C^* = C_1 \cup C_2$, making all neighbors of C_1 and C_2 in $\mathcal{T}$ adjacent to C^*, and adding a new bag $C_v := \{v\}$ and making C_v adjacent to C^*. Thus, we may assume that $C_2 \setminus \mathrm{bd}(C_2, C_1) \neq \emptyset$.

Since $C_1 = \mathrm{bd}(C_1, C_2)$, no vertex of $G - v$ crosses $\mathrm{bd}(C_1, C_2)$. If no vertex of $G - v$ crosses $\mathrm{bd}(C_2, C_1)$, then by Lemma 4.4, we can obtain a partition tree of G in polynomial time. Thus, we may assume that there is a bag C_3 having a vertex that crosses $\mathrm{bd}(C_2, C_1)$. So, $C_3C_2C_1$ is a boundary-crossing path. We will find either an obstruction or a maximal good boundary-crossing path ending in $C_3C_2C_1$. We first check that $C_3C_2C_1$ is good, unless some obstruction from $\mathbb{O}$ appears.

Claim 4.9.1. *Let $z_1 \in N_G(v) \cap C_1$, $z_2 \in N_G(v) \cap C_2$, $w \in \mathrm{bd}(C_2, C_1) \setminus N_G(v)$, and $a \in C_2 \setminus \mathrm{bd}(C_2, C_1)$.*

(i) If there is a vertex $x \in V(G) \setminus \{v, a, w, z_1, z_2\}$ such that $N_G(x) \cap \{v, a, w, z_1, z_2\} = \{a, w\}$, then $G[\{v, a, w, z_1, z_2, x\}]$ is isomorphic to O_1.
(ii) If there is a vertex $x \in V(G) \setminus \{v, a, w, z_1, z_2\}$ such that $N_G(x) \cap \{v, a, w, z_1, z_2\} = \{a, z_2\}$, then $G[\{v, a, w, z_1, z_2, x\}]$ is isomorphic to O_2.
(iii) If there is a pair of distinct non-adjacent vertices $x, y \in V(G) \setminus \{v, a, w, z_1, z_2\}$ such that $N_G(x) \cap \{v, a, w, z_1, z_2\} = N_G(y) \cap \{v, a, w, z_1, z_2\} = \{a, w, z_2\}$, then $G[\{v, a, w, z_1, z_2, x, y\}]$ is isomorphic to O_3.

Proof. It is straightforward to check it; see Fig. 6. ⌟

Claim 4.9.2. *One can in polynomial time output an obstruction or verify that $C_3C_2C_1$ is good.*

Proof. We consider the bag C_3, and first check whether $\mathrm{bd}(C_3, C_2)$ is complete to C_2. If so, then we are done. Otherwise, choose a vertex $p \in \mathrm{bd}(C_3, C_2)$, and a non-neighbor q of p in C_2. As p crosses $\mathrm{bd}(C_2, C_1)$, p has a neighbor a in $C_2 \setminus \mathrm{bd}(C_2, C_1)$ and a neighbor b in $\mathrm{bd}(C_2, C_1)$. There are three possibilities; q is contained in one of $N_G(v) \cap C_2$, $\mathrm{bd}(C_2, C_1) \setminus N_G(v)$, or $C_2 \setminus \mathrm{bd}(C_2, C_1)$. Let $z_1 \in N_G(v) \cap C_1$.

If q and b are in distinct parts of $N_G(v) \cap C_1$ and $\mathrm{bd}(C_2, C_1) \setminus N_G(v)$, then $G[\{p, q, z_1, a, b, v\}]$ is isomorphic to O_1 or O_2 by Claims 4.9.1(i) and (ii). Assume q and b are in the same part of $N_G(v) \cap C_1$ or $\mathrm{bd}(C_2, C_1) \setminus N_G(v)$. Then by the previous argument, we may assume that p is complete to one of the sets $N_G(v) \cap C_1$ or $\mathrm{bd}(C_2, C_1) \setminus N_G(v)$ that does not contain q. Then by choosing a vertex in this set, we can again output O_1 or O_2. Thus, we may assume that q is contained in $C_2 \setminus \mathrm{bd}(C_2, C_1)$ and p is complete to $\mathrm{bd}(C_2, C_1)$. Then $q \neq a$ and by using vertices from $N_G(v) \cap C_1$ and $\mathrm{bd}(C_2, C_1) \setminus N_G(v)$ together with $\{a, p, q, v, z_1\}$, we can output O_3 by Claim 4.9.1(iii).

To verify whether $C_3C_2C_1$ is exclusive, we check if there exists another neighbor bag $D \neq C_3$ of C_2 having a vertex q that crosses $\mathrm{bd}(C_2, C_1)$. If there is such a vertex q, then by applying the previous procedure, we may assume that q is complete to C_2. Thus, by using vertices from each of $N_G(v) \cap C_2$, $\mathrm{bd}(C_2, C_1) \setminus N_G(v)$, and $C_2 \setminus \mathrm{bd}(C_2, C_1)$ together with $\{p, q, z_1, v\}$, we can output O_3 by Claim 4.9.1(iii). Otherwise, $C_3C_2C_1$ is a good boundary-crossing path. ⌟

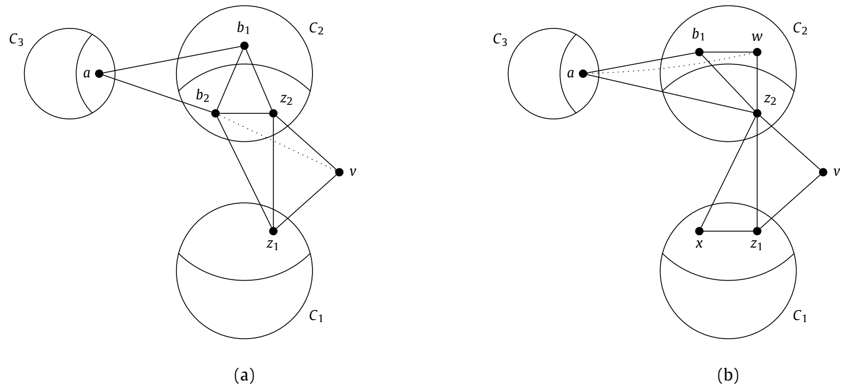


Fig. 7. Illustration of some obstructions appearing in the proof of Claim 4.10.1.

By Claim 4.9.2, we may assume that $C_3C_2C_1$ is good. If no bag contains a vertex crossing $\mathrm{bd}(C_3, C_2)$, then $C_3C_2C_1$ is a maximal good boundary-crossing path. So, we may assume that there is a bag C_4 containing a vertex crossing $\mathrm{bd}(C_3, C_2)$.

We choose $z_1 \in N_G(v) \cap C_1$, $z_2 \in N_G(v) \cap C_2$, $w \in \mathrm{bd}(C_2, C_1) \setminus N_G(v)$, and $a \in C_2 \setminus \mathrm{bd}(C_2, C_1)$. To apply Lemma 4.7, let $G_1 = G[\{v, z_1, z_2, w, a\}]$ and G_2 be the component of $G - V(C_2)$ that contains C_3 and $G' = G[V(G_1) \cup V(G_2)]$. It is clear that G' can be obtained from the disjoint union of G_1 and G_2 by adding edges between $\mathrm{bd}(C_3, C_2)$ and $\{w, a, z_2\}$. Also, for each vertex $p \in \mathrm{bd}(C_3, C_2)$, $\{p\} \cup V(G_1)$ is a wing of $W_{2,0}$ with terminal p.

Thus, by (2) of Lemma 4.7, we can in polynomial time either output an obstruction, or find a maximal good boundary-crossing path ending in C_4C_3 in G_2 such that $\mathrm{bd}(C_4, C_3)$ is complete to C_3 and no other bag contains a vertex crossing C_3. Thus, in the latter case, we obtain a maximal boundary-crossing path ending in C_2C_1 in $G - v$. We now repeatedly apply Lemma 4.3 to modify $\mathcal{T}$ along this path and obtain a partition tree $\mathcal{T}'$ of $G - v$ such that no vertex crosses $\mathrm{bd}(C_2, C_1)$. Note that, for simplicity, we call again C_1 and C_2 the bags of $\mathcal{T}'$ containing the neighbors of v. We can now apply Lemma 4.4 to obtain a partition tree of the entire graph G in polynomial time. □

Lemma 4.10. *If $\mathrm{bd}(C_1, C_2) \setminus N_G(v) \neq \emptyset$ and $C_2 \setminus N_G(v) \neq \emptyset$, then one can in polynomial time either output an obstruction in G or output a partition tree of G confirming that G is a well-partitioned chordal graph.*

Proof. We choose a neighbor z_1 of v in $\mathrm{bd}(C_1, C_2)$, a neighbor z_2 of v in $\mathrm{bd}(C_2, C_1)$ and a non-neighbor x of v in $\mathrm{bd}(C_1, C_2)$. We first consider the case when $\mathrm{bd}(C_1, C_2) = C_1$.

Case 1 *($\mathrm{bd}(C_1, C_2) = C_1$).* Note that no vertex in $G - v$ crosses $\mathrm{bd}(C_1, C_2)$. If no vertex in $G - v$ crosses $\mathrm{bd}(C_2, C_1)$, then we can obtain a partition tree of G from $\mathcal{T}$ by Lemma 4.4. We may assume that there is a bag C_3 containing a vertex a that crosses $\mathrm{bd}(C_2, C_1)$.

Claim 4.10.1. *One can in polynomial time output an obstruction or verify that $C_3C_2C_1$ is good.*

Proof. As a crosses $\mathrm{bd}(C_2, C_1)$, a has a neighbor both in $C_2 \setminus \mathrm{bd}(C_2, C_1)$ and in $\mathrm{bd}(C_2, C_1)$. Let b_1 and b_2 be neighbors of a in $C_2 \setminus \mathrm{bd}(C_2, C_1)$ and $\mathrm{bd}(C_2, C_1)$, respectively.

Assume that a and v have no common neighbors. Then b_2 is not adjacent to v and z_2 is not adjacent to a. So, $G[\{a, b_1, b_2, z_2, z_1, v\}]$ is isomorphic to O_1, see Fig. 7(a). Thus, we may assume that a and v have the common neighbor in $\mathrm{bd}(C_2, C_1)$. We assume that z_2 is a common neighbor.

Now suppose that there is a vertex $w \in C_2 \setminus \mathrm{bd}(C_2, C_1)$ that is not adjacent to a. Recall that since $N_G(v) \cap C_2 \subseteq \mathrm{bd}(C_2, C_1)$, we have that v is not adjacent to w. Thus, we can output a $W_{1,0}$ on $\{v, x, z_1, z_2, a, b_1, w\}$, see Fig. 7(b). So, we may assume that a is complete to $C_2 \setminus \mathrm{bd}(C_2, C_1)$.

Assume that there is a vertex $w \in \mathrm{bd}(C_2, C_1)$ that is not adjacent to a. Note that v may or may not be adjacent to w. If v is adjacent to w, then G contains O_3 as an induced subgraph, and if v is not adjacent to w, then G contains O_2 as an induced subgraph; both these cases are illustrated in Fig. 8. Otherwise, we conclude that a is complete to C_2.

To check whether $C_3C_2C_1$ is exclusive, we find a bag $D \neq C_3$ containing a vertex w crossing $\mathrm{bd}(C_2, C_1)$. If there is no such a vertex, then it is exclusive. Assume that such a vertex w exists. By repeating the above argument, we may assume that w is complete to C_2. Then, $G[\{v, z_1, z_2, x, a, b_1, w\}]$ is isomorphic to $W_{1,0}$ (see Fig. 7(b), but note that in this case $w \notin C_2$). ⌟

By Claim 4.10.1, we may assume that $C_3C_2C_1$ is good. Let $a \in C_2 \setminus \mathrm{bd}(C_2, C_1)$. If no bag contains a vertex crossing $\mathrm{bd}(C_3, C_2)$, then $C_3C_2C_1$ is a maximal good boundary-crossing path. So, we may assume that there is a bag C_4 containing a vertex crossing $\mathrm{bd}(C_3, C_2)$.

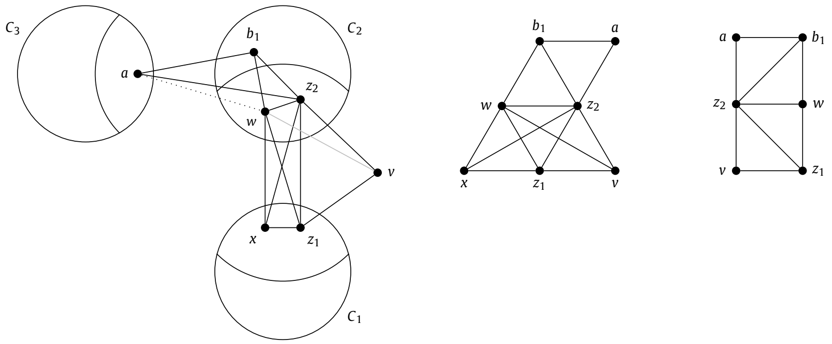


Fig. 8. Illustration of some more obstructions appearing in the proof of Claim 4.10.1. Note that the edge between v and w may or may not be present, depending on which we either have an O_3 or an O_2 as an induced subgraph in G.

To apply Lemma 4.7, let $G_1 = G[\{v, x, z_1, z_2, a\}]$ and G_2 be the component of $G - V(C_2)$ containing C_3 and $G' = G[V(G_1) \cup V(G_2)]$. It is clear that G' can be obtained from the disjoint union of G_1 and G_2 by adding edges between $\mathrm{bd}(C_3, C_2)$ and $\{a, z_2\}$. Observe that for each vertex $p \in \mathrm{bd}(C_3, C_2)$, $G[\{p, a, v, z_1, z_2, z\}]$ is isomorphic to $W^{-}_{1,1}$ with terminal p.

By (2) of Lemma 4.7, we can in polynomial time either output an obstruction, or find a maximal good boundary-crossing path ending in C_4C_3 in G_2 such that $\mathrm{bd}(C_4, C_3)$ is complete to C_3 and no other bag contains a vertex crossing C_3. Thus, in the latter case, we obtain a maximal boundary-crossing path ending in C_2C_1 in $G - v$. We can now repeatedly apply Lemma 4.3 to modify $\mathcal{T}$ along this path and obtain a partition tree $\mathcal{T}'$ for $G - v$ such that no vertex crosses $\mathrm{bd}(C_2, C_1)$. Note that, for simplicity, we call again C_1 and C_2 the bags of $\mathcal{T}'$ containing the neighbors of v. We can now apply Lemma 4.4 to obtain a partition tree for the entire graph G in polynomial time.

Case 2 *($C_1 \setminus \mathrm{bd}(C_1, C_2) \neq \emptyset$).* If there is no vertex crossing $\mathrm{bd}(C_1, C_2)$ and no vertex crossing $\mathrm{bd}(C_2, C_1)$ in $G - v$, then by Lemma 4.4, one can output a partition tree of G from $\mathcal{T}$ in polynomial time. Recall that we have neighbors of v, namely $z_1 \in \mathrm{bd}(C_1, C_2)$ and $z_2 \in \mathrm{bd}(C_2, C_1)$, and a non-neighbor of v, namely $x \in \mathrm{bd}(C_1, C_2)$.

Claim 4.10.2. *If there is a vertex crossing* $\mathrm{bd}(C_1, C_2)$ *or* $\mathrm{bd}(C_2, C_1)$*, then one can in polynomial time output an obstruction or output a partition tree of G from $\mathcal{T}$ confirming that G is a well-partitioned chordal graph.*

Proof. First we consider the case in which only $\mathrm{bd}(C_1, C_2)$ has a crossing vertex. Let a be a vertex in a bag $C_3 \in V(\mathcal{T}) \setminus \{C_1, C_2\}$ that crosses $\mathrm{bd}(C_1, C_2)$. Let $b \in C_1 \setminus \mathrm{bd}(C_1, C_2)$ be a neighbor of a. Note that a neighbor of a in $\mathrm{bd}(C_1, C_2)$ is either adjacent to v, as z_1, or non-adjacent to v, as x. As in Claim 4.9.1, we can restrict the way $N_G(a)$ intersects $\{x, b, z_1\}$, and as we did in Claim 4.9.2, we can deduce that $\mathrm{bd}(C_3, C_1)$ is complete to C_1 and that there is no bag other than C_3 containing a vertex that crosses $\mathrm{bd}(C_1, C_2)$.

Observe that $\{v, z_1, z_2, x, a, b\}$ induces a $W^{-}_{2,0}$ with terminal vertex a. By applying Lemma 4.7 similarly to Case 1, one can in polynomial time find an obstruction or find a maximal good boundary-crossing path ending in $C_3C_1C_2$. In the latter case, we apply Lemma 4.3 to modify $\mathcal{T}$ along this path and obtain a partition tree $\mathcal{T}'$ of $G - v$ such that no vertex crosses $\mathrm{bd}(C_1, C_2)$. Then, since both $\mathrm{bd}(C_1, C_2)$ and $\mathrm{bd}(C_2, C_1)$ have no crossing vertices, we can apply Lemma 4.4 to obtain a partition tree of G.

Now we consider the case in which only $\mathrm{bd}(C_2, C_1)$ has a crossing vertex. Let a be a vertex in a bag C_3 that crosses $\mathrm{bd}(C_2, C_1)$. Note that $\{v, z_1, z_2, x\}$ is a wing of $W_{1,0}$ with terminal z_2, as in Case 1 (see (b) of Fig. 7). As in Claim 4.10.1 and Lemma 4.7, we can find a maximal good boundary-crossing path ending in C_2C_1. We apply Lemma 4.3 to modify $\mathcal{T}$ along this path and obtain a partition tree $\mathcal{T}'$ of $G - v$ such that no vertex crosses $\mathrm{bd}(C_2, C_1)$. Then, since both $\mathrm{bd}(C_1, C_2)$ and $\mathrm{bd}(C_2, C_1)$ have no crossing vertices, we can apply Lemma 4.4 to obtain a partition tree of G.

To conclude, in the case in which both $\mathrm{bd}(C_1, C_2)$ and $\mathrm{bd}(C_2, C_1)$ have crossing vertices, we can first modify $\mathcal{T}$ along a maximal boundary-crossing path ending in C_2C_1, then along a maximal boundary-crossing path ending in C_1C_2. In this way we obtain a partition tree of $G - v$ in which, again, both $\mathrm{bd}(C_1, C_2)$ and $\mathrm{bd}(C_2, C_1)$ have no crossing vertices and we proceed with Lemma 4.4. ⌟

This concludes the lemma. □

Lemma 4.11. *If $C_1 \setminus N_G(v) \neq \emptyset$, $C_2 \setminus N_G(v) \neq \emptyset$ and $N_G(v) = \mathrm{bd}(C_1, C_2) \cup \mathrm{bd}(C_2, C_1)$, then one can in polynomial time either output an obstruction in G or output a partition tree of G confirming that G is a well-partitioned chordal graph.*

Proof. We first show that if at least one of $\mathrm{bd}(C_1, C_2)$ and $\mathrm{bd}(C_2, C_1)$ has no crossing vertex, then we can obtain a partition tree of G.

Claim 4.11.1. *If there is no vertex crossing* $\mathrm{bd}(C_1, C_2)$*, then one can obtain a partition tree of* G *from* $\mathcal{T}$ *in polynomial time. The same holds for* $\mathrm{bd}(C_2, C_1)$.

Proof. We prove the claim for $\mathrm{bd}(C_1, C_2)$ and note that the argument for $\mathrm{bd}(C_2, C_1)$ is symmetric. Let $C_1' := C_1 \setminus \mathrm{bd}(C_1, C_2)$, and $C_{12}' := \mathrm{bd}(C_1, C_2) \cup \{v\}$. Let $\mathcal{S}_1 \subseteq N_{\mathcal{T}}(C_1)$ be such that for all $S_1 \in \mathcal{S}_1$, $\mathrm{bd}(C_1, S_1) \subseteq C_1 \setminus \mathrm{bd}(C_1, C_2)$, and let $\mathcal{S}_2 \subseteq N_{\mathcal{T}}(C_1)$ be such that for all $S_2 \in \mathcal{S}_2$, $\mathrm{bd}(C_1, S_2) \subseteq \mathrm{bd}(C_1, C_2)$. We obtain a partition tree $\mathcal{T}'$ of G from $\mathcal{T}$ as follows.

- Remove C_1; add C_1' and C_{12}'; make C_1' adjacent to C_{12}', and C_{12}' adjacent to C_2.
- Make each bag in $\mathcal{S}_1$ adjacent to C_1', and each bag in $\mathcal{S}_2$ adjacent to C_{12}'.

This yields a partition tree of G. ⌟

From now on, we assume that both $\mathrm{bd}(C_1, C_2)$ and $\mathrm{bd}(C_2, C_1)$ have crossing vertices. Let C_2' be a bag containing a vertex crossing $\mathrm{bd}(C_1, C_2)$, and let C_3 be a bag containing a vertex crossing $\mathrm{bd}(C_2, C_1)$. For convenience, let $C_1' := C_1$.

Using Lemma 4.6 with $B = \mathrm{bd}(C_1', C_2)$, we recursively find a longer good boundary-crossing path or a partial obstruction. Starting from $C_2'C_1'$, for a path $C_i'C_{i-1}' \cdots C_1'$, we find a neighbor bag C_{i+1}' of C_i' that contains a vertex crossing $\mathrm{bd}(C_i', C_{i-1}')$. At the end, either we can find one of first two outcomes in Lemma 4.6, or we can find a maximal good boundary-crossing path ending in $C_2'C_1'C_2$. In the latter case, we can repeatedly apply Lemma 4.3 to modify $\mathcal{T}$ along this path and obtain a partition tree $\mathcal{T}'$ of $G - v$ such that no vertex crosses $\mathrm{bd}(C_1', C_2')$. We can now apply Claim 4.11.1 to obtain a partition tree of the entire graph G. Thus, we may assume that we have an induced subgraph H_1 which is one of two outcomes in Lemma 4.6. Let v_1 be the terminal of H_1 in $\mathrm{bd}(C_1', C_2)$.

By applying the same argument for C_2C_3, we may assume that we have an induced subgraph H_2 which is one of two outcomes in Lemma 4.6. Let v_2 be the terminal of H_2 in $\mathrm{bd}(C_2, C_1)$.

If both H_1 and H_2 are the first outcome in Lemma 4.6, then $G[V(H_1) \cup V(H_2) \cup \{v\}]$ is isomorphic to $W_{s,t}$ for some $s \in [3]$ and $t \geq 0$. If H_1 is the first outcome and H_2 is the second outcome of Lemma 4.6, then $G[V(H_1) \cup V(H_2) \cup \{v\}]$ is isomorphic to $W_{s,t}$ for some $s \in \{2, 3\}$ and $t \geq 0$, where $G[V(H_2) \cup \{v, v_1\}]$ is isomorphic to $W_{2,0}^-$. If both are the second outcomes in Lemma 4.6, then $G[V(H_1) \cup V(H_2) \cup \{v\}]$ is isomorphic to O_4, and this concludes the lemma. □

5. Geodetic sets

For two vertices v and w in a graph G, we denote by $I[v, w]$ the set of all vertices lying on a shortest path between v and w. For a vertex set $S \subseteq V(G)$, we denote by $I[S] := \bigcup_{v,w \in S} I[v, w]$. The set $I[S]$ is called a *geodetic closure of* S [36]. Such a set S is called a *geodetic set* if $I[S] = V(G)$. The Geodetic Set problem asks, given a graph G, for the smallest size of any geodetic set in G.

The study of geodetic sets was initiated by Harary et al. [36] in 1986, and is related to convexity measures in graphs; we refer to [51] for an overview. Harary et al. [36] showed that the Geodetic Set problem is NP-hard on general graphs, see also [4]. Dourado et al. [22] showed that Geodetic Set remains NP-hard on chordal graphs, and that it is polynomial-time solvable on split graphs. We extend their ideas to give a polynomial-time algorithm for well-partitioned chordal graphs, the main result of this section.

Theorem 5.1. *There is a polynomial-time algorithm that given a well-partitioned chordal graph* G*, computes a minimum-size geodetic set of* G.

Before we proceed to the proof of Theorem 5.1, it is worth mentioning that the complexity of Geodetic Set has also been deeply studied on other graph classes. Besides the above mentioned results, it was shown to be NP-hard on chordal bipartite [22] and bipartite [23] graphs, as well as co-bipartite [25], subcubic [11], and planar graphs [17]. Very recently, Chakraborty et al. [16] showed NP-hardness on subcubic partial grids, which unifies hardness on subcubic, planar, and bipartite graphs. Interestingly, they showed that Geodetic Set is NP-hard even on interval graphs, while a polynomial-time algorithm for *proper* interval graphs is known due to Ekim et al. [25]. Other graph classes that are known to admit polynomial-time algorithms are cographs [22], outerplanar graphs [47], block-cactus graphs [25], and solid grid graphs [16]. Kellerhals and Koana [41] recently assessed the parameterized complexity of Geodetic Set, and proved it to be W[1]-hard parameterized by solution size plus pathwidth plus feedback vertex set, while devising FPT-algorithms for the parameter feedback edge set as well as for tree-depth.

We first observe that any geodetic set of a graph contains all its simplicial vertices. Since the neighborhood of a simplicial vertex v is a clique, v is never an internal vertex of any shortest path: Suppose v is an internal vertex of a path P, and let u_1 and u_2 be the two neighbors of v in P. Since u_1 and u_2 are adjacent, we can obtain a shorter path P' from P by replacing u_1vu_2 with u_1u_2 such that P' has the same endpoints as P.

Observation 5.2. *Let* G *be a graph and let* $v \in V(G)$ *be a simplicial vertex in* G*. Then, every geodetic set of* G *contains* v.

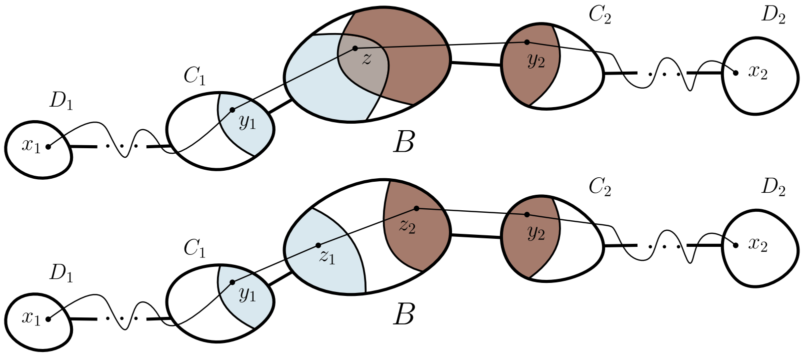


Fig. 9. Illustration of the proof of Lemma 5.4. The top drawing shows item 1 and the bottom one item 2.

From now on we assume that we are given a connected well-partitioned chordal graph G with partition tree $\mathcal{T}$, such that $\mathcal{T}$ has at least two nodes (otherwise, G is simply a complete graph). If G is not connected, we can apply the procedure described below to each of its connected components. As a consequence of Observation 5.2, we have that each leaf bag of $\mathcal{T}$ has a vertex that is contained in every geodetic set of G. Let $B \in V(\mathcal{T})$ be a leaf bag with neighbor C. If $\mathrm{bd}(B, C) \neq B$, then each vertex in $B \setminus \mathrm{bd}(B, C)$ is simplicial. If $\mathrm{bd}(B, C) = B$, then each vertex in B is simplicial. This also immediately implies that each non-simplicial vertex in a leaf bag is on some shortest path between two simplicial vertices: if we have a non-simplicial vertex in B, then $\mathrm{bd}(B, C) \neq B$ and the non-simplicial vertices are precisely the ones in $\mathrm{bd}(B, C)$. Since $\mathcal{T}$ has at least two nodes, there is some other leaf bag in $\mathcal{T}$ which again has some simplicial vertex, say x. Now, each shortest path from a simplicial vertex in B to x uses some vertex from $\mathrm{bd}(B, C)$, and since the vertices in $\mathrm{bd}(B, C)$ are twins in $G[B \cup C]$, each of them is on such a shortest path.

Observation 5.3. *Let G be a connected well-partitioned chordal graph with partition tree $\mathcal{T}$, and let S be the set of simplicial vertices of G. Each leaf bag B of $\mathcal{T}$ contains a simplicial vertex, and $B \subseteq I[S]$.*

In the following, we adapt the idea of Dourado et al. [22] about split graphs to the case of internal bags in a partition tree of a well-partitioned chordal graph. First, we prove a small auxiliary lemma; for an illustration of its arguments see Fig. 9.

Lemma 5.4. *Let G be a connected well-partitioned chordal graph with partition tree $\mathcal{T}$, let S denote the set of simplicial vertices of G, and let $B \in V(\mathcal{T})$ be an internal bag.*

1. *For distinct $C_1, C_2 \in N_{\mathcal{T}}(B)$, $\mathrm{bd}(B, C_1) \cap \mathrm{bd}(B, C_2) \subseteq I[S]$.*
2. *For all $C_1, C_2 \in N_{\mathcal{T}}(B)$ with $\mathrm{bd}(B, C_1) \cap \mathrm{bd}(B, C_2) = \emptyset$, we have that $\mathrm{bd}(B, C_1) \cup \mathrm{bd}(B, C_2) \subseteq I[S]$.*

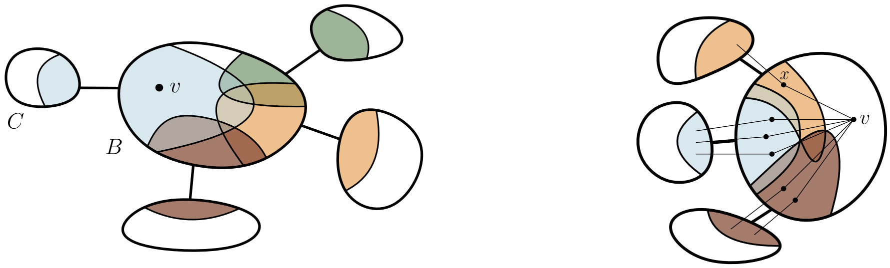
Proof. 1. Let $u \in \mathrm{bd}(B, C_1) \cap \mathrm{bd}(B, C_2)$. There are leaves D_1, D_2 in $\mathcal{T}$ such that $C_1 B C_2$ is on the path from D_1 to D_2 in $\mathcal{T}$. By Observation 5.3, for all $i \in [2]$, D_i contains a simplicial vertex, say x_i. Each shortest path from x_1 to x_2 is of the form $x_1 \cdots y_1 z y_2 \cdots x_2$, where $y_1 \in C_1$, $y_2 \in C_2$, and $z \in \mathrm{bd}(B, C_1) \cap \mathrm{bd}(B, C_2)$. Since the vertices in $\mathrm{bd}(B, C_1) \cap \mathrm{bd}(B, C_2)$ are twins in $G[B \cup C_1 \cup C_2]$, $x_1 \cdots y_1 u y_2 \cdots x_2$ is also a shortest path from x_1 to x_2, and therefore $u \in I[x_1, x_2] \subseteq I[S]$.

2. The proof is similar to (i), with the difference that each shortest path between the (corresponding) vertices x_1 and x_2 uses both a vertex from $\mathrm{bd}(B, C_1)$ and one from $\mathrm{bd}(B, C_2)$. □

Lemma 5.5. *Let G be a connected well-partitioned chordal graph with partition tree $\mathcal{T}$, let S denote the set of simplicial vertices of G, and let $B \in V(\mathcal{T})$ be an internal bag. If B contains a simplicial vertex, then $B \subseteq I[S]$.*

Proof. Let X be the set of vertices in B that are not contained in any boundary. Note that $X \subseteq S$. Then, we obtain $\mathcal{T}'$ from $\mathcal{T}$ by removing B, adding a bag $B' := B \setminus X$ and a bag X. We make all bags in $N_{\mathcal{T}}(B) \cup \{X\}$ adjacent to B' in $\mathcal{T}'$. Since B is a clique, X is a clique and complete to B', satisfying the requirements of the definition of a partition tree. Since no vertex in X was in any boundary, the boundaries from the other neighbors of B' in $\mathcal{T}'$ remain the same as the ones in $\mathcal{T}$ to B. We can conclude that $\mathcal{T}'$ is a partition tree of G. Moreover, each vertex $v \in B'$ is in $\mathrm{bd}(B', X)$ and at least one more boundary, since $v \notin X$. By Lemma 5.4(1), $B' \subseteq I[S]$, so $B' \cup X = B \subseteq I[S]$.

We may thus assume that each simplicial vertex v of B is contained in a boundary. Clearly, a simplicial vertex can be contained in at most one boundary; let $C \in N_{\mathcal{T}}(B)$ be such that $v \in \mathrm{bd}(B, C)$. Since v is simplicial, we have that $\mathrm{bd}(B, C) = B$. Therefore, for each vertex $u \in B$ such that there is some neighbor $C' \neq C$ of B with $u \in \mathrm{bd}(B, C')$, we have by Lemma 5.4(1) that $u \in I[S]$. On the other hand, each vertex in $B \setminus \bigcup_{C' \in N_{\mathcal{T}}(B) \setminus \{C\}} \mathrm{bd}(B, C')$ is simplicial as well, so we can conclude that $B \subseteq I[S]$. □



(a) Illustration of a problematic vertex v. The only boundary v is contained in is $\mathrm{bd}(B, C)$, and every other boundary in B intersects $\mathrm{bd}(B, C)$.

(b) Illustration of a problem solver v. Note that v may be in $I[S]$, and that x is a problem solver as well.

Fig. 10. Problematic vertices and problem solvers.

In the remainder, we show how to deal with vertices that are not on shortest paths between simplicial vertices. We call such vertices *problematic*, and they are the ones that are contained in internal bags without simplicial vertices and do not fall under one of the cases of Lemma 5.4. For an illustration of a problematic vertex, see Fig. 10a.

Definition 5.6. Let G be a connected well-partitioned chordal graph with partition tree $\mathcal{T}$, and let $B \in V(\mathcal{T})$ be an internal bag that does not contain any simplicial vertex. A vertex $v \in B$ is called *problematic* if

1. there is a unique $C \in N_{\mathcal{T}}(B)$ such that $v \in \mathrm{bd}(B, C)$, and
2. for each $C' \in N_{\mathcal{T}}(B) \setminus \{C\}$, $\mathrm{bd}(B, C) \cap \mathrm{bd}(B, C') \neq \emptyset$.

In this case we call C a *problematic neighbor bag*.

Suppose that some bag B has no simplicial vertex. Then each shortest path in G between two simplicial vertices that uses a vertex from B passes through two neighbors of B. If a vertex is problematic, then it cannot be on any such shortest path, and if it is not problematic, then it falls under one of the cases of Lemma 5.4, which leads to the following observation.

Observation 5.7. *Let G be a connected well-partitioned chordal graph with partition tree $\mathcal{T}$, let S denote the set of simplicial vertices of G, and let $B \in V(\mathcal{T})$ be an internal bag with $B \cap S = \emptyset$. Let P be the set of problematic vertices of B, then $P = B \setminus I[S]$.*

By similar reasoning, we observe that if a problematic vertex in B is on some shortest path, then this shortest path has to have an endpoint in B.

Observation 5.8. *Let G be a connected well-partitioned chordal graph with partition tree $\mathcal{T}$, and let $B \in V(\mathcal{T})$ be an internal bag. Let $v \in B$ be a problematic vertex. Any shortest path that has v as an internal vertex has one endpoint in B.*

By Observations 5.7 and 5.8, we know that if a bag B has no simplicial vertex and it has at least one problematic vertex, then we need at least one more vertex from B in any geodetic set. The following notion captures in which situation a single additional vertex suffices. We illustrate the following definition in Fig. 10b.

Definition 5.9. Let G be a connected well-partitioned chordal graph with partition tree $\mathcal{T}$ and let $B \in V(\mathcal{T})$. Let $P \subseteq B$ denote the set of problematic vertices in B and $C_1, \ldots, C_\ell$ the problematic neighbor bags. A vertex $v \in B$ is called a *problem solver* if for each $i \in [\ell]$, either $v \notin \mathrm{bd}(B, C_i)$ or $\mathrm{bd}(B, C_i) \cap P = \{v\}$.

Lemma 5.10. *Let G be a connected well-partitioned chordal graph with partition tree $\mathcal{T}$ and let S denote the set of simplicial vertices of G. Let $B \in V(\mathcal{T})$ be an internal bag containing a problematic vertex and let $W \subseteq V(G)$ with $S \subseteq W \subseteq V(G) \setminus B$. For each $v \in B$, $B \subseteq I[W \cup \{v\}]$ if and only if v is a problem solver.*

Proof. Throughout the proof, we denote by P the set of problematic vertices of B and by $C_1, \ldots, C_\ell$ the problematic neighbor bags. Let $v \in B$.

Suppose that v is a problem solver. By Observation 5.7, each vertex in $B \setminus I[S]$ is problematic, so we have to argue that each problematic vertex is on a shortest path from v to a vertex in W. Let $u \in P$ with problematic neighbor bag C. If $\mathrm{bd}(B, C) \cap P = \{u\}$ and $u = v$, then clearly $u \in I[W \cup \{v\}]$. Otherwise we have that $v \notin \mathrm{bd}(B, C)$, so each shortest path from v that goes through C contains a vertex from $\mathrm{bd}(B, C)$. Moreover, there is a leaf $D \in V(\mathcal{T})$ such that C is on the path from D to B in $\mathcal{T}$. By Observation 5.3, D has a simplicial vertex so the first direction follows.

For the other direction, suppose for a contradiction that $B \subseteq I[W \cup \{v\}]$, while v is not a problem solver. Since v is not a problem solver, for some $i \in [\ell]$, $v \in \mathrm{bd}(B, C_i)$ and there is some $u \in (\mathrm{bd}(B, C_i) \cap P) \setminus \{v\}$. Since $u \in I[W \cup \{v\}]$, u is on a shortest path between v and some vertex in W, denote that path by Q. Since u is problematic, it is not on a shortest path between two vertices in W. Moreover, by Observation 5.8, one of the endpoints of Q has to be in B. Since $B \cap W = \emptyset$, we know that one of the endpoints of the path is v.

If Q uses a vertex from C_i, in particular from $\mathrm{bd}(C_i, B)$, then we can remove u from Q and go from the vertex in $\mathrm{bd}(C_i, B)$ directly to v and obtain a shorter path, a contradiction. If Q does not use a vertex from C_i, then it must use a vertex from some other neighbor of B, say $D \in N_{\mathcal{T}}(B) \setminus \{C_i\}$. This is because the other endpoint of Q, distinct than v, is not contained in B. Now, since u is problematic, we have that $u \notin \mathrm{bd}(B, D)$. However, Q contains a vertex in $\mathrm{bd}(B, D)$, so we can remove u from Q and obtain a shorter path with the same endpoints, again a contradiction. □

Next we show that if there are at least two distinct problematic neighbor bags, then two additional vertices always suffice.

Lemma 5.11. *Let G be a connected well-partitioned chordal graph with partition tree $\mathcal{T}$, let S denote the set of simplicial vertices of G, let $B \in V(\mathcal{T})$ be an internal bag containing a problematic vertex and let $W \subseteq V(G)$ with $S \subseteq W \subseteq V(G) \setminus B$. If there are two distinct problematic neighbor bags of B, then there are two vertices $v_1, v_2 \in B$ such that $B \subseteq I[W \cup \{v_1, v_2\}]$.*

Proof. Let $C_1, C_2 \in N_{\mathcal{T}}(B)$ be two distinct problematic neighbor bags of B, and for all $i \in [2]$, let v_i be a problematic vertex in $\mathrm{bd}(B, C_i)$.

We claim that $B \subseteq I[W \cup \{v_1, v_2\}]$. By Observation 5.7, we have to argue that each problematic vertex in B except v_1, v_2 is on a shortest path from v_1 or v_2 to a vertex in W.

Let u be a problematic vertex other than v_1, v_2 and let $C \in N_{\mathcal{T}}(B)$ be the corresponding problematic neighbor bag. Suppose $C = C_1$. There is a leaf bag D (containing a simplicial vertex by Observation 5.3) such that C is on the path from D to B in $\mathcal{T}$, and each shortest path from a vertex in D to a vertex in $B \setminus \mathrm{bd}(B, C)$ uses a vertex from $\mathrm{bd}(B, C)$. Since $v_2 \notin \mathrm{bd}(B, C)$ by the definition of a problematic vertex, it follows that $u \in I[W \cup \{v_2\}]$. On the other hand, if $C \neq C_1$, then we have that $u \in I[W \cup \{v_1\}]$. We can conclude that $B \subseteq I[W \cup \{v_1, v_2\}]$. □

Finally we show that in the remaining case when there is only one problematic neighbor bag and no problem solver, then any geodetic set of G has to include all problematic vertices.

Lemma 5.12. *Let G be a connected well-partitioned chordal graph with partition tree $\mathcal{T}$, let S denote the set of simplicial vertices of G, let $B \in V(\mathcal{T})$ be an internal bag containing a problematic vertex and let $W \subseteq V(G)$ with $S \subseteq W \subseteq V(G) \setminus B$. Let $P \subseteq B$ be the set of problematic vertices of B. If there is a neighbor $C \in N_{\mathcal{T}}(B)$ such that $P \subseteq \mathrm{bd}(B, C)$ and there is no problem solver, then every geodetic set of G contains P.*

Proof. Note that the condition that there is no problem solver is equivalent to the condition that $\mathrm{bd}(B, C) = B$; any vertex outside of $\mathrm{bd}(B, C)$ would be a problem solver. Suppose that some $v \in P$ is on a shortest path between two vertices x_1 and x_2. Since v is a problematic vertex, we may assume by Observation 5.8 that $x_1 \in B$ and $x_2 \notin B$. Let $D \in V(\mathcal{T})$ denote the bag containing x_2. Let $C^* \in N_{\mathcal{T}}(B)$ denote the neighbor of B that is on the path from D to B in $\mathcal{T}$. If $C^* \neq C$, then v cannot be in an internal vertex of a shortest path from x_1 to x_2: since v is problematic, $v \notin \mathrm{bd}(B, C^*)$. We may assume that $C^* = C$. Since $x_1 \in B = \mathrm{bd}(B, C)$, v cannot be an internal vertex on a shortest path from x_1 to x_2. We can conclude that every geodetic set of G must contain v. □

Now that we have covered all the cases, we can derive the algorithm to compute a minimum geodetic set of a well-partitioned chordal graph by properly prioritizing the cases. We describe the procedure in Algorithm 1.

We now argue the correctness of the algorithm. In line 1, it adds all simplicial vertices to the set it produces. This is safe by Observation 5.2. By Observation 5.3, any vertex contained in any leaf bag of the partition tree is contained in the geodetic closure of the simplicial vertices.

Let $\overline{S}$ be the set S obtained in the final loop. Let B be an internal bag. In line 3, the algorithm asserts that if B contains a simplicial vertex, then no additional vertex of B has to be added. Correctness of this decision is argued in Lemma 5.5. Also, if B has no problematic vertex, then no additional vertex of B has to be added. Now, we can assume that B has no simplicial vertex but has a problematic vertex. Then based on Lemmas 5.10 to 5.12, we add a vertex set in the algorithm so that the geodetic closure of the resulting set contains B. Thus, $\overline{S}$ is a geodetic set.

We claim that $\overline{S}$ is a minimum geodetic set. Suppose that this is not a minimum geodetic set, and let $\widehat{S}$ be a minimum geodetic set. For every leaf bag B, $B \cap \overline{S} = B \cap \widehat{S} = B \cap S$, which implies that there is an internal bag B such that $|\widehat{S} \cap B| < |\overline{S} \cap B|$. As $|\overline{S} \cap B| > 0$, by lines 3 and 4 of Algorithm 1, B contains no simplicial vertices and contains a problematic vertex. We consider three cases corresponding to Lemmas 5.10 to 5.12.

Input : A connected well-partitioned chordal graph G with partition tree $\mathcal{T}$.
Output: A minimum-size geodetic set of G.
1 Find the set S of simplicial vertices of G;
2 **foreach** *internal bag* $B \in V(\mathcal{T})$ **do**
3   **if** *B contains a simplicial vertex* **then** do nothing;
4   **else if** *B contains no problematic vertex* **then** do nothing ;
5   **else if** *there is a problem solver $v \in B$* **then** $S \leftarrow S \cup \{v\}$;
6   **else if** *B has two distinct problematic neighbor bags C_1 and C_2* **then**
7   Let $v_1 \in \mathrm{bd}(B, C_1)$ and $v_2 \in \mathrm{bd}(B, C_2)$ be problematic;
8         $S \leftarrow S \cup \{v_1, v_2\}$;
9   **else** $S \leftarrow S \cup P$, where P is the set of problematic vertices in B;
10 **return** S;

Algorithm 1: A polynomial-time algorithm for finding a minimum-size geodetic set of a well-partitioned chordal graph.

- (Case 1. B has a problem solver.) In this case, $|\overline{S} \cap B| = 1$ by line 5, and by our assumption, $|\widehat{S} \cap B| = 0$. However, since B contains a problematic vertex, by Lemma 5.10, $\widehat{S}$ is not a geodetic set, a contradiction. Thus, we may assume that B has no problem solver.
- (Case 2. There are at least two distinct problematic neighbor bags.) By line 6, $|\overline{S} \cap B| = 2$ and by our assumption, $|\widehat{S} \cap B| < 2$. However, Lemma 5.10 says that if B has no problem solver, then at least two vertices are necessary to contain B as a geodetic closure, and we deduce that $\widehat{S}$ is not a geodetic set, a contradiction. Finally, we may assume that there are no two distinct problematic neighbor bags.
- (Case 3. There is only one problematic neighbor bag.) In this case, all the problematic vertices in B are contained in any geodetic set by Lemma 5.12 and $\overline{S} \cap B$ is exactly the set of such vertices by line 9. So, it is not possible that $|\widehat{S} \cap B| < |\overline{S} \cap B|$ and we have a contradiction.

We conclude that $\overline{S}$ is a minimum geodetic set.

It is easy to verify that each line in Algorithm 1 takes polynomial time, and that the main loop has a polynomial number of iterations. Since well-partitioned chordal graphs can be recognized in polynomial time by an algorithm that produces a partition tree if one exists, see Proposition 4.8, this proves Theorem 5.1.

6. Transversals of longest paths and cycles

It is well-known that in a connected graph, every two longest paths always share a common vertex. In 1966, Gallai [30] asked whether every graph contains a vertex that belongs to all of its longest paths. This question, whose answer is already known to be negative in general [60,62], was shown to have a positive answer on several well-known graph classes. It is not difficult to see that it holds for trees, and it has been shown for outerplanar graphs and 2-trees [53], which has later been generalized to series-parallel graphs, or equivalently, graphs of treewidth at most 2 [19]. (Interestingly, the couterexample for general graphs [60] has treewidth 3.) Besides that, Gallai's question has a positive answer on circular arc graphs [5, 40], P_4-sparse (which includes cographs) and $(P_5, K_{1,3})$-free graphs [15], dually chordal graphs [38], bipartite permutation graphs [14] and $2K_2$-free graphs [33]. As alluded to above, it has a positive answer on split graphs [42], and this result has been generalized to starlike graphs [15].

Both split graphs and starlike graphs are subclasses of well-partitioned chordal graphs. It remains a challenging open problem to determine whether all chordal graphs admit a longest path transversal of size one. As a step towards answering this question for chordal graphs, we show that well-partitioned chordal graphs admit such a transversal.

A closely related question is whether a 2-connected graph has a vertex that intersects all its longest cycles. This question has also been studied extensively on various graph classes, and several of the above mentioned references contain positive answers to this question on the corresponding graph classes. In some cases the results are not stated explicitly, but it is not too difficult to adapt the proofs for the case of longest paths to the case of longest cycles. In this section, we answer this question positively on 2-connected well-partitioned chordal graphs as well.

We start with the following lemma, the proof of which exploits the Helly property[1] of subtrees of a tree to show the existence of a bag of the partition tree that intersects all longest paths of a well-partitioned chordal graph. The same proof strategy has been used by Rautenbach and Sereni [52] to show that for any graph G, there exists a set of size $\mathbf{tw}(G) + 1$ that intersects all the longest paths of G.

Lemma 6.1. *Let G be a connected well-partitioned chordal graph with partition tree $\mathcal{T}$. Then there exists $X \in V(\mathcal{T})$ such that every longest path of G contains a vertex of X.*

[1] The Helly property of trees states that in every tree, every collection of pairwise intersecting subtrees has a common nonempty intersection.

Proof. Let $P_1, \dots, P_\ell$ be the longest paths of G. For each $i \in [\ell]$, the set of bags of $\mathcal{T}$ containing at least one vertex from P_i forms a subtree of $\mathcal{T}$. Let T_i be such a subtree. Since in any connected graph every two longest paths have a vertex in common, we have that $V(T_i) \cap V(T_j) \neq \emptyset$ for every $i \neq j$. By the Helly property of subtrees of a tree, there exists $X \in V(\mathcal{T})$ such that $X \in V(T_i)$ for every $i \in [\ell]$. That is, X is a bag of $\mathcal{T}$ that intersects every longest path of G. □

We prove a similar lemma for longest cycles of 2-connected well-partitioned chordal graphs. The proof of this lemma follows the same lines as the one presented above, hence we omit it here.

Lemma 6.2. *Let G be a 2-connected well-partitioned chordal graph with partition tree $\mathcal{T}$. Then there exists $X \in V(\mathcal{T})$ such that every longest cycle of G contains a vertex of X.*

We now proceed to prove the main results of this section.

Theorem 6.3. *Every connected well-partitioned chordal graph contains a vertex that intersects all its longest paths.*

Proof. Let G be a connected well-partitioned chordal graph. If G is a complete graph, then the result is trivial. Thus, we may assume that G is not a complete graph, and it implies that any partition tree of G consists of at least two bags.

By Lemma 6.1, there exists a bag $B \in V(\mathcal{T})$ such that every longest path of G contains a vertex of B. Let $B_1, \dots, B_k$ be the neighbors of B in $\mathcal{T}$. We define $\mathcal{T}_i$ to be the connected component of $\mathcal{T} - B$ containing B_i and G_i to be the subgraph of G induced by the vertices contained in the bags of $\mathcal{T}_i$. Let p_i be the length of a longest path in G_i with one endpoint in $\mathrm{bd}(B_i, B)$. We may assume without loss of generality that $p_1 \geq p_i$ for every $i > 1$.

We will now show that every longest path of G contains all the vertices of $\mathrm{bd}(B, B_1)$. Let P be a longest path of G and suppose for a contradiction that there exists $v \in \mathrm{bd}(B, B_1)$ such that $v \notin V(P)$. Recall that $V(P) \cap B \neq \emptyset$. If there exist $x, y \in B$ such that $xy \in E(P)$, then we can obtain a path longer than P by inserting v between x and y in P, a contradiction with the fact that P is a longest path of G. Similarly, no endpoint of P belongs to B, otherwise we would also find a path longer than P in G. The same holds also if there exists $x \in \mathrm{bd}(B, B_1)$ and $y \in \mathrm{bd}(B_1, B)$ such that $xy \in E(P)$. Indeed, since $\mathrm{bd}(B, B_1) \cup \mathrm{bd}(B_1, B)$ is a clique, we would again find a path longer than P by inserting v between x and y in P. Therefore P contains no edge crossing from B to B_1, which implies that $V(P) \cap V(G_1) = \emptyset$. Let $P = x_1 x_2 \cdots x_t$ and let x_j be a vertex of $V(P) \cap B$ such that for every $i \geq 1$ we have $x_{j+i} \notin B$. Such a vertex exists since $x_t \notin B$.

Assume that $x_{j+1} \in \mathrm{bd}(B_a, B)$ for some $a \in [k]$. Note that $x_{j+1} x_{j+2} \cdots x_t$ is a path in G_a with an endpoint in $\mathrm{bd}(B_a, B)$. Hence the length of this path is at most p_1. Let $y_1 y_2 \cdots y_{p_1+1}$ be a longest path in G_1 with an endpoint $y_1 \in \mathrm{bd}(B_1, B)$. Then $x_1 x_2 \cdots x_j v y_1 y_2 \cdots y_{p_1+1}$ is a path in G that is longer than P, a contradiction. □

With a more careful argument, we can prove the analogous result for longest cycles.

Theorem 6.4. *Every 2-connected well-partitioned chordal graph contains a vertex that intersects all its longest cycles.*

Proof. Let G be a 2-connected well-partitioned chordal graph. If G is a complete graph, then the result is trivial. Thus, we may assume that G is not a complete graph, and it implies that any partition tree of G consists of at least two bags.

We start as in the proof of Theorem 6.3. By Lemma 6.2, there exists a bag $B \in V(\mathcal{T})$ such that every longest cycle of G contains a vertex of B. Note that we can assume B is not a leaf of $\mathcal{T}$, since if all the longest cycles intersect a bag that is a leaf, they also intersect the bag that is the neighbor of such a leaf. Let $B_1, \dots, B_k$ be the neighbors of B in $\mathcal{T}$. We define $\mathcal{T}_i$ to be the connected component of $\mathcal{T} - B$ containing B_i and let G_i to be the subgraph of G induced by the vertices contained in the bags of $\mathcal{T}_i$.

Now, let p_i be the length of a longest path in G_i with *both* endpoints in $\mathrm{bd}(B_i, B)$. Note that this is well-defined, since $|\mathrm{bd}(B_i, B)| \geq 2$ for every i, as G is a 2-connected graph. We may assume without loss of generality that $p_1 \geq p_i$ for every $i > 1$.

We will now show that every longest cycle of G contains all the vertices of $\mathrm{bd}(B, B_1)$. Let C be a longest cycle of G and suppose for a contradiction that there exists $v \in \mathrm{bd}(B, B_1)$ such that $v \notin V(C)$. We first point out the following.

Claim 6.4.1. $|V(C) \cap B| \geq 2$.

Proof. We already know that $|V(C) \cap B| \geq 1$. Suppose for a contradiction that $|V(C) \cap B| = 1$. Then there exist $x_1, x_2, x_3 \in V(C)$ such that x_1, x_2, and x_3 appear consecutively in the cycle, and $x_2 \in B$ and $x_1, x_3 \notin B$. In particular, x_2 belongs to the boundary between B and some neighboring bag B_i, and $x_1, x_3 \in \mathrm{bd}(B_i, B)$. Since G is 2-connected, there exists $u \in \mathrm{bd}(B, B_i)$, with $u \neq x_2$, such that $u \notin V(C)$. Thus, we can add u between x_2 and x_3 in C and obtain a cycle longer than C, a contradiction. ⌟

If there exist $x, y \in B$ such that $xy \in E(C)$, then we can obtain a cycle longer than C by inserting v between x and y in C, a contradiction with the fact that C is a longest cycle of G. The same holds if there exist $x \in \mathrm{bd}(B, B_1)$ and $y \in \mathrm{bd}(B_1, B)$

such that $xy \in E(C)$. Indeed, since $\mathrm{bd}(B, B_1) \cup \mathrm{bd}(B_1, B)$ is a clique, we would again find a cycle longer than C by inserting v between x and y in C. Therefore C contains no edge crossing from B to B_1, which implies that $V(C) \cap V(G_1) = \emptyset$. Consider $u \in \mathrm{bd}(B, B_1)$ such that $u \neq v$. We consider two cases.

First assume that $u \in V(C)$. Since C cannot have two consecutive vertices in B, there exists $i \neq 1$ such that $u \in \mathrm{bd}(B, B_i)$, and there exists $u' \in \mathrm{bd}(B_i, B)$ such that $uu' \in E(C)$. Moreover, by the above claim, there exists $u'' \in V(C) \cap \mathrm{bd}(B, B_i)$ such that if P is the subpath of C starting in u, ending in u'', and containing u', then $(V(P) \setminus \{u, u''\}) \subseteq V(G_i)$. Note also that $|E(P)| \le p_i + 2$, since the neighbors of u and u'' in P belong to $\mathrm{bd}(B_i, B)$. Let $y_1 y_2 \cdots y_q$ be a longest path of G_1 with both endpoints in $\mathrm{bd}(B_1, B)$ and let $P' = u y_1 y_2 \cdots y_q v u''$. Let C' be the cycle obtained from C by replacing P by P'. Since $|E(P')| = p_1 + 3$ and $p_1 \ge p_i$, we have that C' is a cycle longer than C, a contradiction.

Now we consider the case in which $u \notin V(C)$. Recall that C cannot have two consecutive vertices in B. By Claim 6.4.1, there exists $i \neq 1$ such that $V(C) \cap V(G_i) \neq \emptyset$. Let $x, x', y, y' \in V(C)$ be such that $x, y \in \mathrm{bd}(B, B_i)$, $x', y' \in \mathrm{bd}(B_i, B)$, $xx', yy' \in E(C)$ and the subpath P of C starting in x, ending in y, and containing x' and y' is such that $(V(P) \setminus \{x, y\}) \subseteq V(G_i)$. Note that it can be the case that $x' = y'$. Moreover, $|E(P)| \le p_i + 2$. Let $y_1 y_2 \cdots y_q$ be a longest path of G_1 with both endpoints in $\mathrm{bd}(B_1, B)$ and let $P' = x u y_1 y_2 \cdots y_q v y$. Let C' be the cycle obtained from C by replacing P by P'. Since $|E(P')| = p_1 + 4$ and $p_1 \ge p_i$, we have that C' is a cycle longer than C, a contradiction. This concludes the proof that all the vertices of $\mathrm{bd}(B, B_1)$ are contained in all longest cycles of G. □

7. Tree 3-spanners

For a connected graph G and a positive integer t, a spanning tree T of G is a *tree t-spanner* of G if for every pair (v, w) of vertices in G, $\mathrm{dist}_T(v, w) \le t \cdot \mathrm{dist}_G(v, w)$, where $\mathrm{dist}_G(v, w)$ (resp., $\mathrm{dist}_T(v, w)$) denotes the length of shortest path in G (resp., T) from v to w. The Tree t-Spanner problem asks whether a given graph G has a tree t-spanner. Tree t-spanners are motivated from applications including network research and computational geometry [3,43]. Cai and Corneil [13] showed that Tree t-Spanner is linear-time solvable if $t \le 2$, and is NP-complete if $t \ge 4$. For $t = 3$, the complexity of Tree 3-Spanner is not yet unveiled. Brandstädt et al. [9] investigated the complexity of Tree t-Spanner on chordal graphs of small diameter. They showed that for even $t \ge 4$ (resp., odd $t \ge 5$) it is NP-complete to decide if a chordal graph of diameter at most $t + 1$ (resp., $t + 2$) has a tree t-spanner. On the other hand, for any even t (resp., odd t), every chordal graph of diameter at most $t - 1$ (resp., $t - 2$) admits a tree t-spanner which can be found in linear time. Brandstädt et al. [9] also showed that Tree 3-Spanner is polynomial-time solvable on chordal graphs of diameter at most 2. On general chordal graphs, the complexity of Tree 3-Spanner is still open. Several subclasses of chordal graphs, such as split [59], very strongly chordal [9], and interval [44] graphs were shown to be *tree 3-spanner admissible*, meaning that each of its members admits a tree 3-spanner. In the above mentioned cases, such tree 3-spanners can always be computed in polynomial time. We show that the same holds for well-partitioned chordal graphs, generalizing the result for split graphs [59].

Before we proceed to the proof of this result, we point out that a subclass of chordal graphs that is not tree 3-spanner admissible and yet has a polynomial-time algorithm for Tree 3-Spanner is that of 2-sep chordal graphs, as shown by Das and Panda [50]. Other (non-chordal) graph classes that are known to be tree 3-spanner admissible are bipartite ATE-free graphs [8] (which include convex graphs) and permutation graphs [44]; and there are polynomial-time algorithms for Tree 3-Spanner on cographs and co-bipartite graphs [12], as well as planar graphs [26].

We now proceed to the proof of the main result of this section. More specifically, we show that given a connected well-partitioned chordal graph, one can always find a tree 3-spanner in polynomial time.

Theorem 7.1. *Every connected well-partitioned chordal graph admits a tree 3-spanner, which one can find in polynomial time.*

Proof. Let G be a connected well-partitioned chordal graph with partition tree $\mathcal{T}$. We choose a bag R of $\mathcal{T}$ and consider it as a root bag. For each non-root bag B, let $P(B)$ denote the parent bag of B. For each non-root bag B,

- let S^*_B be a star whose center is in $\mathrm{bd}(B, P(B))$ and all leaves are exactly the vertices in $V(B) \setminus \mathrm{bd}(B, P(B))$,
- let S^{**}_B be a star whose center is in $\mathrm{bd}(P(B), B)$ and all leaves are exactly the vertices in $\mathrm{bd}(B, P(B))$, and
- let $S_B := S^*_B \cup S^{**}_B$.

Observe that the vertex set of S_B consists of all vertices of B and one vertex in $\mathrm{bd}(P(B), B)$. Moreover, S_B is a tree. For the root bag R, let S_R be a star in $G[R]$. We claim that $U := \bigcup_{B \in V(\mathcal{T})} S_B$ is a tree 3-spanner of G. It is sufficient to show that U is a spanning tree, and for every edge vw in G, $\mathrm{dist}_U(v, w) \le 3$.

We first verify that U is a spanning tree. Note that for each non-root bag B, S_B is a tree containing all vertices of B and at least one edge between B and $P(B)$, and furthermore, S_R is a spanning tree of $G[R]$. Therefore, U is a connected subgraph containing all vertices of G. Suppose that U contains a cycle C.

Observe that for each non-root bag B of $\mathcal{T}$, the center of S^{**}_B separates $V(B)$ and $V(P(B))$ in U. Let B' be the bag containing a vertex of C such that $\mathrm{dist}_{\mathcal{T}}(R, B')$ is minimum. Since $U[V(B')]$ has no cycle, there is a child bag B'' of B' containing a vertex of C. By the above observation, $V(B') \cap V(C)$ has only one vertex that is the center of $S^{**}_{B''}$. As $|V(B') \cap V(C)| = 1$, there is no other child bag of B' containing a vertex of C.

We can observe that there is no child bag of B'' containing a vertex of C. If such a bag exists, then by the same argument, we derive that $|V(B'') \cap V(C)| = 1$, a contradiction. Therefore, C is contained in $S_{B''}$, but by the construction, $S_{B''}$ has no cycle. We conclude that U is a spanning tree.

Now, we claim that for every edge vw in G, $\text{dist}_U(v, w) \le 3$. Choose an edge vw of G. If vw is an edge in a bag B, then $\text{dist}_U(v, w) = \text{dist}_{S_B}(v, w) \le 3$. Assume that vw is an edge between a bag B and its parent $P(B)$ so that $v \in V(B)$ and $w \in V(P(B))$. If $vw \in E(S_B)$, then it is trivial. Assume that $w \notin V(S_B)$. Let z be the vertex of S_B contained in $P(B)$. Then $\text{dist}_U(v, w) = \text{dist}_{S_B}(v, z) + \text{dist}_{S_{P(B)}}(z, w) \le 3$.

Our construction of a tree 3-spanner for G immediately follows the partition tree $\mathcal{T}$ of G. By Proposition 4.8, a partition tree of a well-partitioned chordal graph can be obtained in polynomial time, and therefore one can find a tree 3-spanner for G in polynomial time. □

8. Conclusions

In this paper, we introduced the class of *well-partitioned chordal graphs*, a subclass of chordal graphs that generalizes the class of split graphs. We provided a characterization by a set of forbidden induced subgraphs which also gave a polynomial-time recognition algorithm. We showed that well-partitioned chordal graphs can be used to narrow down complexity gaps for problems that are NP-hard on chordal graphs and polynomial-time solvable on split graphs. In particular, we showed that Geodesic Set is an example of such a problem that becomes polynomial-time solvable on well-partitioned chordal graphs. On the other hand, we observed that there are problems that are NP-hard on chordal graphs and remain NP-hard on well-partitioned chordal graphs, even though they are polynomial-time solvable on split graphs. It would be interesting to see other problems for which well-partitioned chordal graphs can be used to better understand the complexity difference between split graphs and chordal graphs.

Another typical characterization of (subclasses of) chordal graphs is via vertex orderings. For instance, chordal graphs are famously characterized as the graphs admitting perfect elimination orderings [29]. It would be interesting to see if well-partitioned chordal graphs admit a concise characterization in terms of vertex orderings as well. While the degree of the polynomial in the runtime of our recognition algorithm is moderate, our algorithm does not run in linear time. We therefore ask if it is possible to recognize well-partitioned chordal graphs in linear time; and note that a characterization in terms of vertex orderings can be a promising step in this direction.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] J. Ahn, L. Jaffke, O. Kwon, P.T. Lima, Well-partitioned chordal graphs: obstruction set and disjoint paths, in: I. Adler, H. Müller (Eds.), Proceedings of the 46th International Workshop on Graph-Theoretic Concepts in Computer Science, WG 2020, Springer, Cham, 2020, pp. 148–160.
[2] J. Ahn, L. Jaffke, O. Kwon, P.T. Lima, Three problems on well-partitioned chordal graphs, in: T. Calamoneri, F. Corò (Eds.), Proceedings of the 12th International Conference on Algorithms and Complexity, CIAC 2021, Springer, 2021, pp. 23–36.
[3] I. Althöfer, G. Das, D. Dobkin, D. Joseph, J. Soares, On sparse spanners of weighted graphs, Discrete Comput. Geom. 9 (1993) 81–100, https://doi.org/10.1007/BF02189308.
[4] M. Atici, Computational complexity of geodetic set, Int. J. Comput. Math. 79 (2002) 587–591, https://doi.org/10.1080/00207160210954.
[5] P.N. Balister, E. Györi, J. Lehel, R.H. Schelp, Longest paths in circular arc graphs, Comb. Probab. Comput. 13 (2004) 311–317, https://doi.org/10.1017/S0963548304006145.
[6] A.A. Bertossi, Dominating sets for split and bipartite graphs, Inf. Process. Lett. 19 (1984) 37–40.
[7] H.L. Bodlaender, K. Jansen, On the complexity of the maximum cut problem, Nord. J. Comput. 7 (2000) 14–31.
[8] A. Brandstädt, F.F. Dragan, H. Le, V.B. Le, R. Uehara, Tree spanners for bipartite graphs and probe interval graphs, Algorithmica 47 (2007) 27–51, https://doi.org/10.1007/s00453-006-1209-y.
[9] A. Brandstädt, F.F. Dragan, H.O. Le, V.B. Le, Tree spanners on chordal graphs: complexity and algorithms, Theor. Comput. Sci. 310 (2004) 329–354, https://doi.org/10.1016/S0304-3975(03)00424-9.
[10] A. Brandstädt, V.B. Le, J.P. Spinrad, Graph Classes: A Survey, SIAM Monographs in Discrete Mathematics and Applications, 1999.
[11] L.R. Bueno, L.D. Penso, F. Protti, V.R. Ramos, D. Rautenbach, U.S. Souza, On the hardness of finding the geodetic number of a subcubic graph, Inf. Process. Lett. 135 (2018) 22–27, https://doi.org/10.1016/j.ipl.2018.02.012.
[12] L. Cai, Tree spanners: spanning trees that approximate distances, Ph.D. thesis, University of Toronto, 1992.
[13] L. Cai, D.G. Corneil, Tree spanners, SIAM J. Discrete Math. 8 (1995) 359–387, https://doi.org/10.1137/S0895480192237403.
[14] M.R. Cerioli, C.G. Fernandes, R. Gómez, J. Gutiérrez, P.T. Lima, Transversals of longest paths, Discrete Math. 343 (2020) 111717.
[15] M.R. Cerioli, P.T. Lima, Intersection of longest paths in graph classes, Discrete Appl. Math. 281 (2020) 96–105, https://doi.org/10.1016/j.dam.2019.03.022.
[16] D. Chakraborty, S. Das, F. Foucaud, H. Gahlawat, D. Lajou, B. Roy, Algorithms and complexity for geodetic sets on planar and chordal graphs, https://doi.org/10.4230/LIPIcs.ISAAC.2020.7, 2020.
[17] D. Chakraborty, F. Foucaud, H. Gahlawat, S.K. Ghosh, B. Roy, Hardness and approximation for the geodetic set problem in some graph classes, in: M. Changat, S. Das (Eds.), Proceedings of the 6th International Conference on Algorithms and Discrete Applied Mathematics, CALDAM 2020, Springer, 2020, pp. 102–115.
[18] Y.W. Chang, M.S. Jacobson, C.L. Monma, D.B. West, Subtree and substar intersection numbers, Discrete Appl. Math. 44 (1993) 205–220, https://doi.org/10.1016/0166-218X(93)90232-D.
[19] G. Chen, J. Ehrenmüller, C.G. Fernandes, C.G. Heise, S. Shan, P. Yang, A.N. Yates, Nonempty intersection of longest paths in series-parallel graphs, Discrete Math. 340 (2017) 287–304, https://doi.org/10.1016/j.disc.2016.07.023.

[20] D.G. Corneil, Y. Perl, Clustering and domination in perfect graphs, Discrete Appl. Math. 9 (1984) 27–39, https://doi.org/10.1016/0166-218X(84)90088-X.
[21] M. Cygan, F.V. Fomin, L. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, S. Saurabh, Parameterized Algorithms, Springer, 2015.
[22] M.C. Dourado, F. Protti, D. Rautenbach, J.L. Szwarcfiter, Some remarks on the geodetic number of a graph, Discrete Math. 310 (2010) 832–837.
[23] M.C. Dourado, F. Protti, J.L. Szwarcfiter, On the complexity of the geodetic and convexity numbers of a graph, in: Lecture Notes of the Ramanujan Mathematical Society, vol. 7, 2006, pp. 497–500.
[24] R.G. Downey, M.R. Fellows, Fundamentals of Parameterized Complexity, Texts in Computer Science, Springer, 2013.
[25] T. Ekim, A. Erey, P. Heggernes, P. van 't Hof, D. Meister, Computing minimum geodetic sets of proper interval graphs, in: D. Fernández-Baca (Ed.), Proceedings of the 10th Latin American Symposium on Theoretical Informatics, LATIN 2012, Springer, 2012, pp. 279–290.
[26] S.P. Fekete, J. Kremer, Tree spanners in planar graphs, Discrete Appl. Math. 108 (2001) 85–103, https://doi.org/10.1016/S0166-218X(00)00226-2.
[27] P. Festa, P.M. Pardalos, M.G.C. Resende, Feedback set problems, in: Handbook of Combinatorial Optimization, Supplement Vol. A, Kluwer Acad. Publ., Dordrecht, 1999, pp. 209–258.
[28] F.V. Fomin, D. Lokshtanov, S. Saurabh, M. Zehavi, Kernelization, Cambridge University Press, 2019.
[29] D. Fulkerson, O. Gross, Incidence matrices and interval graphs, Pac. J. Math. 15 (1965) 835–855.
[30] T. Gallai, Problem 4, in: P. Erdős, G.O.H. Katona (Eds.), Proceedings of the Colloquium on Theory of Graphs Held in Tihany, Hungary, 1966, 1968, p. 362.
[31] F. Gavril, The intersection graphs of subtrees in trees are exactly the chordal graphs, J. Comb. Theory, Ser. B 16 (1974) 47–56, https://doi.org/10.1016/0095-8956(74)90094-x.
[32] A. George, J.R. Gilbert, J.W. Liu, Graph Theory and Sparse Matrix Computation, vol. 56, Springer Science & Business Media, 2012.
[33] G. Golan, S. Shan, Nonempty intersection of longest paths in $2K_2$-free graphs, Electron. J. Comb. 25 (2018) P2.37.
[34] M.C. Golumbic, Algorithmic Graph Theory and Perfect Graphs, Annals of Discrete Mathematics, vol. 57, North-Holland Publishing Co., 2004.
[35] J. Gustedt, On the pathwidth of chordal graphs, Discrete Appl. Math. 45 (1993) 233–248.
[36] F. Harary, E. Loukakis, C. Tsouros, The geodetic number of a graph, Math. Comput. Model. 17 (1993) 89–95.
[37] F. Havet, C.L. Sales, L. Sampaio, b-Coloring of tight graphs, Discrete Appl. Math. 160 (2012) 2709–2715.
[38] A.S. Jobson, A.E. Kézdy, J. Lehel, S.C. White, Detour trees, Discrete Appl. Math. 206 (2016) 73–80, https://doi.org/10.1016/j.dam.2016.02.002.
[39] F. Joos, A characterization of substar graphs, Discrete Appl. Math. 175 (2014) 115–118, https://doi.org/10.1016/j.dam.2014.05.033.
[40] F. Joos, A note on longest paths in circular arc graphs, Discuss. Math., Graph Theory 35 (2015) 419–426.
[41] L. Kellerhals, T. Koana, Parameterized complexity of geodetic set, https://doi.org/10.4230/LIPIcs.IPEC.2020.20, 2020.
[42] S. Klavžar, M. Petkovšek, Graphs with nonempty intersection of longest paths, Ars Comb. 29 (1990) 43–52.
[43] M.C. Loui, D.R. Luginbuhl, Optimal on-line simulations of tree machines by random access machines, SIAM J. Comput. 21 (1992) 959–971, https://doi.org/10.1137/0221056.
[44] M.S. Madanlal, G. Venkatesan, C.P. Rangan, Tree 3-spanners on interval, permutation and regular bipartite graphs, Inf. Process. Lett. 59 (1996) 97–102, https://doi.org/10.1016/0020-0190(96)00078-6.
[45] R.M. McConnell, K. Mehlhorn, S. Näher, P. Schweitzer, Certifying algorithms, Comput. Sci. Rev. 5 (2011) 119–161, https://doi.org/10.1016/j.cosrev.2010.09.009.
[46] S. Mengel, Lower bounds on the mim-width of some graph classes, Discrete Appl. Math. 248 (2018) 28–32.
[47] M. Mezzini, Polynomial time algorithm for computing a minimum geodetic set in outerplanar graphs, Theor. Comput. Sci. 745 (2018) 63–74, https://doi.org/10.1016/j.tcs.2018.05.032.
[48] H. Müller, Hamiltonian circuits in chordal bipartite graphs, Discrete Math. 156 (1996) 291–298.
[49] S.D. Nikolopoulos, L. Palios, Detecting holes and antiholes in graphs, Algorithmica 47 (2007) 119–138, https://doi.org/10.1007/s00453-006-1225-y.
[50] B.S. Panda, A. Das, Tree 3-spanners in 2-sep chordal graphs: characterization and algorithms, Discrete Appl. Math. 158 (2010) 1913–1935, https://doi.org/10.1016/j.dam.2010.08.015.
[51] I.M. Pelayo, Geodesic Convexity in Graphs, Springer, 2013.
[52] D. Rautenbach, J. Sereni, Transversals of longest paths and cycles, SIAM J. Discrete Math. 28 (2014) 335–341, https://doi.org/10.1137/130910658.
[53] S.F. de Rezende, C.G. Fernandes, D.M. Martin, Y. Wakabayashi, Intersecting longest paths, Discrete Math. 313 (2013) 1401–1408, https://doi.org/10.1016/j.disc.2013.02.016.
[54] D.J. Rose, R.E. Tarjan, G.S. Lueker, Algorithmic aspects of vertex elimination on graphs, SIAM J. Comput. 5 (1976) 266–283, https://doi.org/10.1137/0205021.
[55] C. Semple, M. Steel, Phylogenetics, Oxford Lecture Series in Mathematics and Its Applications, vol. 24, Oxford University Press, Oxford, 2003.
[56] A. Silva, Graphs with small fall-spectrum, Discrete Appl. Math. 254 (2019) 183–188.
[57] L. Vandenberghe, M.S. Andersen, Chordal graphs and semidefinite optimization, Found. Trends Optim. 1 (2015) 241–433, https://doi.org/10.1561/2400000006.
[58] M. Vatshelle, New width parameters of graphs, Ph.D. thesis, University of Bergen, Norway, 2012.
[59] G. Venkatesan, U. Rotics, M.S. Madanlal, J.A. Makowsky, C.P. Rangan, Restrictions of minimum spanner problems, Inf. Comput. 136 (1997) 143–164, https://doi.org/10.1006/inco.1997.2641.
[60] H. Walther, H.J. Voss, Über Kreise in Graphen, Deutscher Verlag der Wissenschaften, 1974.
[61] R. Watrigant, M. Bougeret, R. Giroudeau, Approximating the sparsest k-subgraph in chordal graphs, Theory Comput. Syst. 58 (2016) 111–132, https://doi.org/10.1007/s00224-014-9568-2.
[62] T. Zamfirescu, On longest paths and circuits in graphs, Math. Scand. 38 (1976) 211–239.