ELSEVIER

Theoretical Computer Science 175 (1997) 309–335

Theoretical Computer Science

On treewidth and minimum fill-in of asteroidal triple-free graphs

Ton Kloks [a,*], Dieter Kratsch [b,1], Jeremy Spinrad [c]

[a] *Department of Mathematics and Computing Science, Eindhoven University of Technology, P.O.Box 513, 5600 MB Eindhoven, The Netherlands*

[b] *Fakultät für Mathematik und Informatik, Friedrich-Schiller-Universität, 07740 Jena, Germany*

[c] *Department of Computer Science, Vanderbilt University, Nashville, TN 37235, USA*

Abstract

We present $O(n^5R + n^3R^3)$ time algorithms to compute the treewidth, pathwidth, minimum fill-in and minimum interval graph completion of asteroidal triple-free graphs, where n is the number of vertices and R is the number of minimal separators of the input graph. This yields polynomial time algorithms for the four NP-complete graph problems on any subclass of the asteroidal triple-free graphs that has a polynomially bounded number of minimal separators, as e.g. cocomparability graphs of bounded dimension and d-trapezoid graphs for any fixed $d \geqslant 1$.

1. Introduction

We present algorithms solving the problems TREEWIDTH, PATHWIDTH, MINIMUM FILL-IN and INTERVAL GRAPH COMPLETION, when they are restricted to the class of AT-free graphs. All four problems are NP-complete even when restricted to cobipartite graphs, a subclass of the AT-free graphs [1, 22]. Hence they remain NP-complete on AT-free graphs and there cannot be a polynomial time algorithm on AT-free graphs for any of these problems unless P = NP. Our algorithms have running time $O(n^5 R + n^3 R^3)$, where n is the number of vertices and R is the number of minimal separators of the input graph. This implies polynomial time algorithms for all the four problems on any subclass of the AT-free graphs that has a polynomially bounded number of minimal separators as e.g. permutation graphs, trapezoid graphs, cocomparability graphs of bounded dimension and d-trapezoid graphs for any fixed $d \geqslant 1$.

It has been shown in [4] that treewidth and pathwidth as well as minimum fill-in and minimum interval graph completion can be computed by polynomial time algorithms for d-trapezoid graphs, d a fixed positive integer, if the graph and a d-trapezoid diagram

* Corresponding author. E-mail: ton@win.tue.nl.

[1] This research was done while the author was with IRISA Rennes, France.

PII S0304-3975(96)00206-X

of it are given as input. (For earlier algorithms see [3, 17].) However, if we only allow the standard input, i.e., the graph is given as input, then this does not yield polynomial time algorithms, since we cannot compute the intersection model efficiently, if only the graph is given. Thus, it was left open whether the four problems become easy because the class of graphs, i.e., the d-trapezoid graphs for any fixed $d \geqslant 3$, is well behaved, or because having the d-trapezoid diagram (which is the solution to an NP-complete problem for any fixed $d \geqslant 3$ [23]) is such a powerful tool, that it gives the solution.

Now we are able to answer this question, since we obtain $O(n^{3d+3})$ algorithms for the four problems, when they are restricted to d-trapezoid graphs, for any fixed $d \geqslant 1$. Furthermore, these algorithms do not require a d-trapezoid diagram as part of the input. Of course, as one would expect, the running times of these algorithms are much worse than those of the best known algorithms computing treewidth and minimum fill-in of d-trapezoid graphs, which is $O(n\, tw(G)^{d-1})$ and $O(n^d)$ [4]. Notice that the latter algorithms heavily exploit the d-trapezoid diagram. Nevertheless, the algorithms given in our paper are polynomial time algorithms for d-trapezoid graphs, and hence also for cocomparability graphs of dimension at most d, for any fixed positive integer d, although they do not require an intersection model as part of the input.

This answers a tempting theoretical question, since it shows that small interval dimension and small dimension is capturing a property, which makes all these four problems simpler. Yannakakis has shown that the DIMENSION problem 'Given a partially ordered set P, is the dimension of P at most d' and the INTERVAL DIMENSION problem 'Given a partially ordered set P, is the interval dimension of P at most d' are both NP-complete for any fixed $d \geqslant 3$ [23]. Therefore we are not able to run a recognition algorithm to determine for fixed $d \geqslant 3$, whether our input graph is in the class of d-trapezoid graphs (respectively cocomparability graphs of dimension at most d).

Nevertheless the results of our algorithms can be used reliably, even if we do not know a priori that the input graph G is a d-trapezoid graph (respectively cocomparability graph of dimension at most d). This is possible since both algorithms work as follows. If the input graph G is not a cocomparability graph, then the algorithm reports this. If G is a cocomparability graph, then the algorithm solves the corresponding problem for G correctly, or reports, that G is not a d-trapezoid graph (respectively that G has dimension larger than d), since G has 'too many' minimal separators.

2. Preliminaries

All graphs in this paper are simple and undirected. We denote the number of vertices of a graph $G=(V,E)$ by n, and the size of a maximum clique in G by $\omega(G)$. $G[W]$ denotes the subgraph of $G=(V,E)$ induced by the vertices of $W \subseteq V$. For simplicity we abuse notation using the same notation for a connected component and its vertex set.

2.1. Graph classes

We list the definition of some graph classes that are relevant to the paper. For more information on graph classes we refer to [8] and for more information on partially ordered sets we refer to [21].

Definition 1. A graph G is said to be *chordal* if any cycle of length at least four has a chord.

Definition 2. A graph $G=(V,E)$ is an *interval graph* if the vertices of G can be put into one-to-one correspondence with intervals on the real line, such that two vertices are adjacent in G if and only if the corresponding intervals have a nonempty intersection.

Lemma 2.1. *$G=(V,E)$ is an interval graph if and only if the maximal cliques of G can be ordered $A_1, A_2, \dots, A_q$ such that for every vertex v the maximal cliques containing v occur consecutively.*

Such an ordering of the maximal cliques is said to be a *consecutive clique arrangement* of G.

Recently many interesting structural properties of AT-free graphs have been established by Corneil et al. [5, 6].

Definition 3. A set of three independent vertices x, y, z of a graph G is called an *asteroidal triple* (AT) if for any two of these vertices there exists a path joining them that avoids the (closed) neighbourhood of the third. A graph G is called an *asteroidal triple-free* (AT-free) graph if G does not contain an asteroidal triple.

Asteroidal triple-free graphs are a class of graphs containing well-known classes of perfect graphs as e.g. interval, permutation, trapezoid and cocomparability graphs.

Definition 4. A graph $G=(V,E)$ is said to be a *cocomparability graph* if there is a partially ordered set $P=(V, \prec_P)$ such that $\{u,v\} \in E$ if and only if u and v are incomparable in P (i.e. neither $u \prec_P v$ nor $v \prec_P u$).

The d-trapezoid graphs form a subclass of the cocomparability graphs that is of interest for our considerations.

Definition 5. Let d be a fixed positive integer. Then a *d-trapezoid diagram* $\mathfrak{D}(G)$ of a graph $G=(V,E)$ assigns to each vertex v of G a collection of d intervals

$$\mathfrak{I}(v) = \langle [l_v^i, r_v^i] : l_v^i, r_v^i \in \{1,2,\dots,2n\}, l_v^i < r_v^i, i \in \{1,2,\dots d\} \rangle$$

such that for each $i \in \{1, 2, \dots, d\}$ and any pair of vertices $v, w \in V$ the intervals $[l_v^i, r_v^i]$ and $[l_w^i, r_w^i]$ have no endpoint in common. Furthermore, $\{v,w\} \in E$ if and only if either there is an $i \in \{1,2,\ldots,d\}$ such that $[l_v^i, r_v^i]$ and $[l_w^i, r_w^i]$ have nonempty intersection or there are $i,j \in \{1,2,\ldots,d\}$ such that $l_v^i < r_v^i < l_w^i < r_w^i$ and $l_w^j < r_w^j < l_v^j < r_v^j$.

The following visualizing of a d-trapezoid diagram is useful. Draw d parallel horizontal lines labelled $D_1, D_2, \ldots, D_d$ from bottom to the top. Mark slots $1,2,\ldots,2n$ in unit distance from left to right on each of the horizontal lines. Then for any vertex $v \in V$ we obtain a polygon Q_v by drawing line segments between consecutive points in the chain $l_v^1, l_v^2, \ldots, l_v^d, r_v^d, r_v^{d-1}, \ldots, r_v^1, l_v^1$. The polygon Q_v is said to be a *d-trapezoid*. Consequently, $\{v,w\} \in E$ if and only if Q_v and Q_w have nonempty intersection.

Definition 6. A graph G is a *d-trapezoid graph* if it has a d-trapezoid diagram.

It is worth mentioning that for any positive integer d, the d-trapezoid graphs are exactly the cocomparability graphs of partially ordered sets of interval dimension at most d.

2.2. Minimal separators

Minimal separators are one of the fundamental concepts of this paper. They will play a central role in our algorithms.

Definition 7. Given a graph $G=(V,E)$ and two nonadjacent vertices a and b, a set $S \subseteq V$ is an *a, b-separator* if the removal of S separates a and b in distinct connected components. If no proper subset of an a,b-separator S is also an a,b-separator then S is a *minimal a, b-separator*. A *minimal separator* is a set of vertices S for which there exist nonadjacent vertices a and b such that S is a minimal a,b-separator.

The following lemma provides an easy algorithm to recognize minimal separators (cf. [8]).

Lemma 2.2. *Let S be a separator of the graph $G=(V,E)$. Then S is a minimal separator of G if and only if there are two connected components of $G[V \setminus S]$ such that every vertex of S has a neighbour in both of these components.*

Dirac was the first to discover a relation between minimal separators and chordal graphs [7].

Theorem 2.3. *A graph G is chordal if and only if every minimal separator is a clique.*

Any interval graph $G=(V,E)$ has a consecutive clique arrangement by Lemma 2.1. This allows us to characterize the minimal separators of an interval graph.

Theorem 2.4. *Let $A_1, A_2, \ldots, A_q$ be a consecutive clique arrangement of an interval graph G. Then the minimal separators of G are the sets $A_i \cap A_{i+1}$, $i \in \{1,\ldots,q-1\}$.*

Proof. Since each A_i is a maximal clique, we have that for each $1 \leqslant i < t$: $A_i \backslash A_{i+1} \neq \emptyset$ and $A_{i+1} \backslash A_i \neq \emptyset$. Let $x \in A_i \backslash A_{i+1}$ and $y \in A_{i+1} \backslash A_i$. Then clearly $A_i \cap A_{i+1}$ is a minimal x, y-separator.

Now consider non-adjacent vertices a and b and let S be a minimal a, b-separator. Assume a appears before b in the consecutive clique arrangement. Let A_i be the last clique that contains a and let A_j be the first clique that contains b. If, for all $\ell \in \{i, \ldots, j-1\}$, there is a vertex in $A_\ell \cap A_{\ell+1}$ which is not in S, then there is a path from a to b in $G[V \backslash S]$. Hence there is an $\ell \in \{i, \ldots, j-1\}$ such that $A_\ell \cap A_{\ell+1} \subseteq S$. □

Lemma 2.5. *Let S be a minimal separator and a clique of G. Let C be a connected component of $G[V \backslash S]$ and let x and y be non-adjacent vertices of $G[S \cup C]$. Then every minimal x, y-separator S^* of G is a proper subset of $S \cup C$.*

Proof. Let S^* be a minimal x, y-separator of G and let C_x and C_y be the components of $G[V \backslash S^*]$ containing x and y, respectively.

C_x and C_y cannot both have non-empty intersection with S, since S is a clique. Without loss of generality we may assume that $C_x \cap S = \emptyset$. Hence x is a vertex of C and it is trivially also a vertex of C_x. This implies $C_x \subseteq C$. Any vertex $z \in V \backslash (S \cup C)$ belongs to a component of $G[V \backslash S]$ different from C and cannot have a neighbour in C_x. Hence $z \notin S^*$ by Lemma 2.2. Finally, S^* is a proper subset of $S \cup C$ since $x, y \notin S^*$. □

In the sequel we adopt the convention that if $S_1 \subseteq S_2$, then the vertex set of *every* connected component of $G[V \backslash (S_1 \cup S_2)]$ and $S_1 \backslash S_2$ are contained in one connected component of $G[V \backslash S_2]$ (since $S_1 \backslash S_2 = \emptyset$).

Definition 8. Two minimal separators S_1 and S_2 are said to be *non-crossing* if all vertices of $S_1 \backslash S_2$ are contained in one connected component of $G[V \backslash S_2]$ and all vertices of $S_2 \backslash S_1$ are contained in one connected component of $G[V \backslash S_1]$.

Lemma 2.6. *Let $G=(V,E)$ be a chordal graph. Then every pair of minimal separators in G is non-crossing.*

Proof. Let S_1 and S_2 be minimal separators of the graph G. Since G is chordal, S_1 and S_2 are cliques by Theorem 2.3. It follows that $S_1 \backslash S_2$ is contained in the one connected component of $G[V \backslash S_2]$, and that $S_2 \backslash S_1$ is contained in the one connected component of $G[V \backslash S_1]$. □

2.3. Triangulations

Triangulations and minimal triangulations of graphs have already been studied for about 30 years.

Definition 9. A *triangulation* of a graph G is a graph H with the same vertex set as G such that H is a chordal graph and G is a subgraph of H. In that case we say that G is *triangulated into H*.

Minimal triangulations have been considered in relation to Gaussian elimination of matrices, as well as minimal elimination orderings of graphs and the MINIMUM FILL-IN problem [16, 18, 19].

Definition 10. A triangulation H of a graph $G=(V,E)$ is a *minimal triangulation* of G if no proper subgraph of H is a triangulation of G.

The first characterization of minimal triangulations has been given in 1976. Rose et al. [19] have shown the following theorem and its corollary.

Theorem 2.7. *Let H be a triangulation of a graph G. Then H is a minimal triangulation of G if and only if, for all edges $e \in E(H)\setminus E(G)$, the graph $H-e$ is not chordal.*

Corollary 2.8. *Let H be a triangulation of a graph G. Then H is a minimal triangulation of G if and only if each edge $e \in E(H)\setminus E(G)$ is the unique chord of a cycle of length four in H.*

Another characterization of minimal triangulations is given by Parra and Scheffler in [17]. Now we give a new characterization of minimal triangulations related to the one by efficient triangulations in [12] (see also [10]).

Definition 11. For any subset $\mathfrak{C} \subseteq \mathfrak{P}(V)$ let $G_{\mathfrak{C}}$ be the graph obtained from G by adding edges between all pairs of nonadjacent vertices x and y of G for which an $S \in \mathfrak{C}$ with $\{x,y\} \subseteq S$ exists.

We denote the set of all minimal separators of a graph $G=(V,E)$ by $\mathfrak{Sep}(G)$.

Theorem 2.9. *Let H be a triangulation of the graph $G=(V,E)$. Then H is a minimal triangulation of G if and only if $H = G_{\mathfrak{Sep}(H)}$.*

Proof. Assume H is a minimal triangulation of G. By Theorem 2.3 every minimal separator of H is a clique. Thus $G_{\mathfrak{Sep}(H)}$ is a subgraph of H.

Now let $\{a,b\} \in E(H)\setminus E(G)$. Then $\{a,b\}$ is unique chord of a cycle of length four in H by Corollary 2.8. Hence removing the edge $\{a,b\}$ from H gives a square (a,p,b,q). Then clearly, every minimal p,q-separator of H contains a and b. Hence $\{a,b\} \subseteq S$ for some $S \in \mathfrak{Sep}(H)$, thus $\{a,b\}$ is an edge of $G_{\mathfrak{Sep}(H)}$. Consequently $H = G_{\mathfrak{Sep}(H)}$.

Assume H is not a minimal triangulation of G. By Theorem 2.7, there is an edge $\{a,b\} \in E(H)\setminus E(G)$ such that $H' = H - \{a,b\}$ is a triangulation of G.

We claim that there is no minimal separator S of H such that $\{a,b\} \subseteq S$. Suppose on the contrary that S is a minimal p,q-separator of H with $\{a,b\} \subseteq S$. Then S is a minimal p,q-separator of H' by Lemma 2.2. Since S is not a clique in H', H' is not chordal by Theorem 2.3, a contradiction. Hence $\{a,b\}$ is not an edge of $G_{\mathfrak{Sep}(H)}$, thus $H \neq G_{\mathfrak{Sep}(H)}$. $\square$

The following property of minimal triangulations has been given in [10, 12].

Theorem 2.10. *Let H be a minimal triangulation of a graph $G=(V,E)$. Then the following conditions are satisfied.*

1. *If a and b are nonadjacent vertices in H then every minimal a,b-separator in H is also a minimal a,b-separator in G.*
2. *If S is a minimal separator in H and C is a connected component of $H[V \setminus S]$ then $G[C]$ is a connected component of $G[V \setminus S]$.*

It is worth mentioning that the above theorem is crucial for our paper.

Lemma 2.11. *Let H be a minimal triangulation of G and let S_1, S_2 be minimal separators in H. Then S_1 and S_2 are non-crossing minimal separators in G.*

Proof. S_1 and S_2 are non-crossing in the chordal graph H by Lemma 2.6. H is a minimal triangulation of G, thus S_1 and S_2 are minimal separators in G by Theorem 2.10. Furthermore, the vertex sets of the connected components of $H[V \setminus S_i]$ are the same as those of $G[V \setminus S_i]$, $i \in \{1,2\}$. It follows that S_1 and S_2 are also non-crossing minimal separators in G. □

2.4. Treewidth, pathwidth, minimum fill-in and minimum interval graph completion

There are different ways to define the treewidth of a graph. The original definition uses the concept of a *tree-decomposition*. For more information on tree-decompositions the reader is referred to the survey paper [2]. Here we introduce the treewidth by means of triangulations. This turned out to be a fruitful approach for many of the recently designed efficient treewidth algorithms for special graph classes (see e.g. [10]).

Definition 12. The *treewidth* of a graph G, denoted by $tw(G)$, is the smallest value of $\omega(H)-1$, where the minimum is taken over all triangulations H of G.

The following lemma shows the equivalence of the above definition of treewidth and the original one by Robertson and Seymour. For a proof see, for example, [14].

Lemma 2.12. *A graph $G=(V,E)$ has a tree-decomposition of width at most k if and only if there is a triangulation H of G with $\omega(H) \leqslant k+1$.*

Observation 1. For any graph G there exists a minimal triangulation H with $tw(G) = \omega(H)-1$.

Hence solving the TREEWIDTH problem for an input graph $G=(V,E)$ is equivalent to determining the smallest size of a maximum clique of H over all minimal triangulations H of G. The pathwidth of a graph can be defined in terms of triangulations into an interval graph.

Definition 13. The *pathwidth* of a graph G, denoted by $pw(G)$, is the smallest value of $\omega(H) - 1$, where thec minimum is taken over all triangulations H of G for which H is an interval graph.

Observation 2. There are graphs G that do not have a minimal triangulation H such that $pw(G) = \omega(H) - 1$, as e.g. any tree that is not a caterpillar.

The problems MINIMUM FILL-IN and INTERVAL GRAPH COMPLETION are closely related to the problems TREEWIDTH and PATHWIDTH since they also consider finding minimal triangulations (into an interval graph) that optimize a certain graph parameter.

Definition 14. A *fill-in* of the graph $G = (V,E)$ is a set F of edges of $\overline{G}$ such that $H = (V, E \cup F)$ is chordal. The *minimum fill-in* of a graph G, denoted by $mfi(G)$, is the smallest value of $|E(H)| - |E(G)|$, where the minimum is taken over all triangulations H of G.

Observation 3. For any graph G there exists a minimal triangulation H such that $mfi(G) = |E(H)| - |E(G)|$.

Hence solving the MINIMUM FILL-IN problem on a graph G is equivalent to finding a minimal triangulation H of G, that has smallest number of edges among all minimal triangulations of G. Such a triangulation H of G is sometimes called a *minimum triangulation.*

Definition 15. An *interval graph completion* of the graph $G = (V,E)$ is a set F of edges of $\overline{G}$ such that $H = (V, E \cup F)$ is an interval graph. The *minimum interval graph completion* of a graph G, denoted by $mic(G)$, is the smallest value of $|E(H)| - |E(G)|$, where the minimum is taken over all triangulations H of G such that H is an interval graph.

Möhring has shown the following important theorem on minimal triangulations of AT-free graphs and its corollary in [15]. (For earlier related results see [3, 9].)

Theorem 2.13. *Any minimal triangulation H of an* AT-*free graph $G = (V,E)$ is an interval graph.*

Corollary 2.14. *$tw(G) = pw(G)$ and $mfi(G) = mic(G)$ for any* AT-*free graph $G = (V,E)$.*

Therefore we may concentrate on designing an algorithm solving the TREEWIDTH (and hence the PATHWIDTH) problem as well as an algorithm solving the MINIMUM FILL-IN (and hence the INTERVAL GRAPH COMPLETION) problem, as long as we consider the problems restricted to AT-free graphs.

By Observations 1 and 3, there are minimal triangulations H' and H'' of G such that $tw(G) = \omega(H') - 1$ and $mfi(G) = |E(H'')| - |E(G)|$. By Theorem 2.13, H' and

H'' are interval graphs, if the graph G is AT-free. It is important to keep in mind that one of the major goals of our algorithms is to find minimal triangulations with the properties of H' and H''.

Now let the input graph $G=(V,E)$ be an arbitrary graph. The idea of both algorithms is to break any (potential) minimal triangulation H of G for which H is an interval graph, into 'parts', so-called blocks, which allows the recursive computation of $\omega(H)$ and $|E(H)|$, respectively. Our algorithms guarantee that $\omega(H)$ and $|E(H)|$ are computed correctly (at least) for all minimal triangulations H of G into an interval graph. This leads to two algorithms.

The algorithm computing the treewidth of the input graph G is correct, if there is a minimal triangulation H of the input graph G such that H is an interval graph and $tw(G) = \omega(H) - 1$. Note that $tw(G) = pw(G)$ under these assumptions.

The algorithm computing the minimum fill-in of the input graph G is correct, if there is a minimal triangulation H of the input graph G such that H is an interval graph and $mfi(G) = |E(H)| - |E(G)|$. Note that $mfi(G) = mic(G)$ under these assumptions.

Indeed we shall see later that the condition that H is an interval graph can be slightly relaxed.

3. 1-Blocks

Blocks and realizations of blocks are useful concepts for designing treewidth and minimum fill-in algorithms.

Definition 16. A *1-block* of a graph $G=(V,E)$ is a pair $B=(S,C)$, where S is a minimal separator of G and C is a connected component of $G[V\setminus S]$. The graph obtained from $G[S \cup C]$ by adding edges such that S becomes a clique is said to be the *realization* of B and is denoted by $R(S,C)$.

The following lemma indicates how to exploit 1-blocks and their realizations.

Lemma 3.1. *Let $S \in \mathfrak{Sep}(G)$ and let $C_1, C_2, \ldots, C_r$ be the components of $G[V\setminus S]$. Suppose H_j is a minimal triangulation of $R(S,C_j)$ for any $j \in \{1,2,\ldots,r\}$. Then the graph $H = (V(H),E(H))$ with $V(H) = V(G)$ and $E(H) = \bigcup_{j=1}^{r} E(H_j)$ is a minimal triangulation of G.*

Conversely, let H be a minimal triangulation of G with $S \in \mathfrak{Sep}(H)$. Then $H[S\cup C]$ is a minimal triangulation of the realization $R(S,C)$ for each component C of $G[V\setminus S]$.

Proof. Let $S \in \mathfrak{Sep}(G)$ and let $C_1, C_2, \ldots, C_r$ be the components of $G[V\setminus S]$. For each $j \in \{1,2,\ldots, r\}$, let H_j be a minimal triangulation of $R(S,C_j)$, thus S is a clique in H_j.

We claim that the graph H with the same vertex set as G and with edge set $\bigcup_{j=1}^{r} E(H_j)$ is a chordal graph. Suppose not and let $\mathscr{C}$ be a chordless cycle in H

of length greater than three. Since H_j is chordal for each $j \in \{1,2,\ldots,r\}$, there are two vertices u and v of $\mathscr{X}$ belonging to different components C and C' of $G[V\backslash S]$. By the construction of H, S is a minimal separator of H and u, v belong to different components of $H[V\backslash S]$. Consequently there are two non consecutive vertices $s, s' \in S$ in $\mathscr{X}$ and $\{s,s'\} \in E(H)$ would be a chord in $\mathscr{X}$, a contradiction. Thus H is chordal and therefore a triangulation of G.

Suppose H is not a minimal triangulation of G. Thus by Corollary 2.8, there is an edge $e \in E(H)\backslash E(G)$ that is not the unique chord of a cycle of length four in H. Let $e = \{x,y\}$. Since H_j is a minimal triangulation of $R(S,C_j)$ for any $j \in \{1,2,\ldots,r\}$, x and y must belong to different components of $H[V \setminus S]$, a contradiction. Consequently H is a minimal triangulation of G.

Let H be a minimal triangulation of G. Let S be a minimal separator of the chordal graph H. Hence S is a clique in H. Since H is a minimal triangulation of G, Theorem 2.10 implies that S is also a minimal separator of G and that the vertex sets of the components of $G[V\backslash S]$ and of the components of $H[V\backslash S]$ are the same. Let $C_1, C_2, \ldots, C_r$ be the components of $G[V\backslash S]$ and let $C_1', C_2', \ldots, C_r'$ be the components of $H[V\backslash S]$ such that C_j and C_j' have the same vertex set for all $j \in \{1,2,\ldots,r\}$.

H is chordal, hence the graph $H[S \cup C_j]$ is chordal for each component C_j' of $H[V\backslash S]$. Furthermore S is a clique of $H[S \cup C_j]$, hence $H[S \cup C_j]$ is a triangulation of $R(S,C_j)$, for each $j \in \{1,2,\ldots,r\}$. Suppose $H[S \cup C_j]$ is not a minimal triangulation of $R(S,C_j)$ for some $j \in \{1,2,\ldots,r\}$. Then choosing minimal triangulations H_j' of $R(S,C_j)$ such that H_j' is a subgraph of $H[S \cup C_j]$ for all $j \in \{1,2,\ldots,r\}$, we obtain a minimal triangulation H' of G with $E(H') = \bigcup_{j=1}^{r} E(H_j')$, by the first part of the lemma. Therefore H' is a proper subgraph of H, a contradiction. □

Lemma 3.1 allows us to obtain the following theorem that gives an equation for computing the treewidth of a graph from the treewidth of the realizations of all 1-blocks of the graph. For a similar result dealing with separators of bounded size we refer to [1].

Theorem 3.2. *Let $G = (V,E)$ be a non-complete graph. Then*

$$tw(G) = \min_{S \in \mathfrak{Sep}(G)} \max_{C} tw(R(S,C)), \tag{1}$$

where the maximum is taken over all connected components C of $G[V\backslash S]$.

Proof. Let H be a minimal triangulation of G with $tw(G) = \omega(H) - 1$, that exists by Observation 2.4. Let S be any minimal separator of H. By Lemma 3.1, $H[S \cup C]$ is a minimal triangulation of $R(S,C)$ for each component C of $G[V\backslash S]$. Each maximal clique of H is a maximal clique of $H[S \cup C]$ for some component C, since a clique of H cannot contain vertices of different components of $H[V\backslash S]$. Therefore $tw(G) = \max_C tw(R(S,C))$.

Conversely, let $S \in \mathfrak{Sep}(G)$ and let $C_1, C_2, \ldots, C_r$ be the components of $G[V\backslash S]$. By Observation 1, there is a minimal triangulation H_j of $R(S,C_j)$ with $tw(R(S,C_j)) =$

$\omega(H_j) - 1$, for each $j \in \{1,2,\dots,r\}$. By Lemma 3.1, the graph $H = (V(H), E(H))$ with $V(H) = V(G)$ and $E(H) = \bigcup_{j=1}^{r} E(H_j)$ is a minimal triangulation of G. Thus $tw(G) \leqslant \max_{j=1,2,\dots,r} tw(R(S,C_j))$. $\square$

Corollary 3.3. *Let $G = (V,E)$ be a non-complete graph. Then*

$$tw(G) = \max_{C} tw(R(S,C)),$$

for each $S \in \mathfrak{Sep}(G)$ satisfying that S is a minimal separator of some minimal triangulation H of G with $tw(G)=\omega(H)-1$. The maximum is taken over all connected components C of $G[V\setminus S]$.

To facilitate the presentation of the equations concerning the computation of the treewidth in subsequent sections, we introduce the following abbreviation.

Definition 17. Let S be a minimal separator of G. Define

$$tw(G;S) := \max_{C} tw(R(S,C)), \tag{2}$$

where the maximum is taken over all connected components C of $G[V\setminus S]$.

Remark 1. Let S be a minimal separator of the graph G. Then $tw(G;S)$ is equal to the treewidth of the graph obtained from G by adding edges such that S becomes a clique.

Remark 2. Let G be a non-complete graph. Then

$$tw(G) = \min_{S\in\mathfrak{Sep}(G)} tw(G;S). \tag{3}$$

A statement similar to Theorem 3.2 can be obtained for the minimum fill-in of a graph.

Definition 18. For any graph $G = (V,E)$ and any $S \subseteq V$, $fill(S) := \binom{|S|}{2} - |E(G[S])|$ denotes the number of edges to be added to $G[S]$ such that S becomes a clique.

Theorem 3.4. *Let $G = (V,E)$ be a non-complete graph. Then*

$$mfi(G) = \min_{S\in\mathfrak{Sep}(G)} \Big(fill(S) + \sum_{C} mfi(R(S,C))\Big), \tag{4}$$

where the summation is over all connected components C of $G[V\setminus S]$.

Proof. Let H be a minimum triangulation of G, i.e., $mfi(G) = |E(H)| - |E(G)|$ and H is a minimal triangulation of G. Let S be any minimal separator of H. By Lemma 3.1, $H[S\cup C]$ is a minimal triangulation of $R(S,C)$ for each component C of $G[V\setminus S]$. Therefore $mfi(G) = |E(H)| - |E(G)| = fill(S) + \sum_C \big(|E(H[S\cup C])| - |E(R(S,C))|\big)$. Furthermore, there is a minimal triangulation H_j of $R(S,C_j)$ with $mfi(R(S,C_j)) = |E(H_j)| -$

$|E(R(S,C_j))|$, for any $j \in \{1,2,\dots,r\}$, where $C_1,C_2,\dots,C_r$ are the components of $G[V\setminus S]$. By Lemma 3.1, the graph with vertex set $V(G)$ and edge set $\bigcup_{j=1}^{r} E(H_j)$ is a minimal triangulation of G. Therefore, since H is a minimum triangulation, the induced subgraph $H[S\cup C_j]$ is a minimum triangulation of $R(S,C_j)$ for each component C_j of $G[V\setminus S]$, $j \in \{1,2,\dots,r\}$. Consequently $mfi(G) = |E(H)| - |E(G)| = fill(S) + \sum_{j=1}^{r} mfi(R(S,C_j))$.

On the other hand, consider any $S \in \mathfrak{Sep}(G)$ and let $C_1,C_2,\dots,C_r$ be the components of $G[V\setminus S]$. By Observation 3, there is a minimal triangulation H_j of $R(S,C_j)$ with $mfi(R(S,C_j)) = |E(H_j)| - |E(R(S,C_j))|$, for any $j \in \{1,2,\dots,r\}$. By Lemma 3.1, the graph $H = (V(H),E(H))$ with $V(H) = V(G)$ and $E(H) = \bigcup_{j=1}^{r} E(H_j)$ is a minimal triangulation of G. Thus $mfi(G) \leqslant fill(S) + \sum_{j=1}^{r} mfi(R(S,C_j)))$ for any $S \in \mathfrak{Sep}(G)$. □

Corollary 3.5. *Let $G=(V,E)$ be a graph. Then*

$$mfi(G) = fill(S) + \sum_{C} mfi(R(S,C))$$

for each $S \in \mathfrak{Sep}(G)$ that is a minimal separator of some minimum triangulation H of G. The summation is over all connected components C of $G[V\setminus S]$.

The important fact is, that the treewidth of a graph and the minimum fill-in of a graph can in principle be computed by recursive algorithms that inspect all minimal separators. In general, such an algorithm does not have a polynomially bounded running time. However for various graph classes refinements of this approach lead to efficient algorithms.

It turns out that the recursion solving the MINIMUM FILL-IN problem can be done more efficiently, if a related graph parameter is computed.

Definition 19. Let $G=(V,E)$ be a graph. Then $met(G)$ denotes the minimum number of edges in H where the minimum is taken over all triangulations H of the graph G.

Consequently, $met(G) = mfi(G) + |E|$ for any graph $G=(V,E)$. Thus Theorem 3.4 and Corollary 3.5 imply

Corollary 3.6. *Let $G=(V,E)$ be a non-complete graph. Then*

$$met(G) = \min_{S\in\mathfrak{Sep}(G)} \left((1-t)\binom{|S|}{2} + \sum_{i=1}^{t} met(R(S,C_i)) \right), \tag{5}$$

where $C_1,C_2,\dots,C_t$ are the connected components of $G[V\setminus S]$.

Corollary 3.7. *Let $G=(V,E)$ be a graph. Then*

$$met(G) = (1-t)\binom{|S|}{2} + \sum_{i=1}^{t} met(R(S,C_i))$$

for each $S \in \mathfrak{Sep}(G)$ satisfying that S is a minimal separator of some minimum triangulation H, where $C_1, C_2, \ldots, C_t$ are the components of $G[V \setminus S]$.

Definition 20. Let S be a minimal separator of G and $C_1, C_2, \ldots, C_t$ the connected components of $G[V \setminus S]$. Define

$$met(G;S) : =(1-t)\binom{|S|}{2} + \sum_{i=1}^{t} met(R(S,C_i)). \tag{6}$$

Remark 3. For any minimal separator S of the graph G, $met(G;S)$ is the minimum number of edges in a triangulation H of G, for which S is a clique in H.

Remark 4. Let G be a non-complete graph. Then

$$met(G) = \min_{S \in \mathfrak{Sep}(G)} met(G;S). \tag{7}$$

We mention that our algorithms are based on the Corollaries 3.3 and 3.7 in the following sense. The algorithms correctly compute the treewidth (and minimum fill-in) of a given graph $G=(V,E)$ if G has a minimal triangulation H' (and H''), that satisfies $tw(G) = \omega(H') - 1$ (and $met(G) = |E(H'')|$), by looking for the minimal separators S of H' (and H''). Therefore we need a collection of technical lemmas saying how the components of 1-blocks or of certain subgraphs of a minimal triangulation of G look like when a minimal separator S^* has been removed. We consider this in detail in Sections 5 and 7.

4. 2-Blocks

Now we introduce another type of block.

Definition 21. Let S_1 and S_2 be two non-crossing minimal separators of the graph $G = (V,E)$. A connected component D of $G[V \setminus (S_1 \cup S_2)]$ is said to be *between* S_1 and S_2, if $S_2 \setminus S_1$ and the vertex set of D are contained in one connected component of $G[V \setminus S_1]$ and $S_1 \setminus S_2$ and the vertex set of D are contained in one connected component of $G[V \setminus S_2]$.

Recall the convention that if $S_1 \subseteq S_2$ then the vertex set of every connected component of $G[V \setminus (S_1 \cup S_2)]$ and $S_1 \setminus S_2$ are contained in one connected component of $G[V \setminus S_2]$.

Definition 22. Let S_1 and S_2 be non-crossing minimal separators of the graph $G = (V,E)$. The *2-block* $\mathfrak{B}(S_1,S_2)$ of G is the subset of V consisting of S_1, S_2 and the vertex sets of all connected components of $G[V \setminus (S_1 \cup S_2)]$ between S_1 and S_2.

Lemma 4.1. *Let S_1 and S_2 be non-crossing minimal separators in G. Let H be a minimal triangulation of G such that $S_1, S_2 \in \mathfrak{Sep}(H)$. Then the 2-block of S_1 and S_2 in G and the 2-block of S_1 and S_2 in H are equal.*

Proof. Let D be a connected component of $H[V\backslash(S_1 \cup S_2)]$ such that D is contained in the 2-block of S_1 and S_2 in H. Since H is a minimal triangulation of G, the vertex sets of the connected components of $H[V\backslash S_1]$ and $G[V\backslash S_1]$ are the same. Hence D is contained in the same connected component as $S_2\backslash S_1$ in $G[V\backslash S_1]$. Similarly, D and $S_1\backslash S_2$ are contained in one connected component of $G[V\backslash S_2]$. Thus the 2-block of S_1 and S_2 in H is a subset of the 2-block of S_1 and S_2 in G.

Analogously it can be shown that the 2-block of S_1 and S_2 in G is a subset of the 2-block of S_1 and S_2 in H. $\square$

Remark 5. We have shown that the 2-blocks of S_1 and S_2 in G and in any minimal triangulation H of G are equal, if $S_1, S_2 \in \mathfrak{Sep}(H)$. On the other hand, in general, it is not true that the vertex sets of the connected components of $G[V\backslash(S_1' \cup S_2')]$ and $H[V\backslash(S_1' \cup S_2')]$ are equal for all $S_1', S_2' \in \mathfrak{Sep}(G)$.

Definition 23. Let $\mathfrak{B}(S_1, S_2)$ be a 2-block of a graph $G = (V, E)$. The *realization* $\mathfrak{R}(S_1, S_2)$ is the graph obtained from $G[\mathfrak{B}(S_1, S_2)]$ by adding all edges between non-adjacent vertices of S_1 and all edges between non-adjacent vertices of S_2.

We distinguish between two types of 2-blocks.

Definition 24. A 2-block $\mathfrak{B}(S_1, S_2)$ of G is *degenerate* if $S_1 \subseteq S_2$ or $S_2 \subseteq S_1$. Otherwise the 2-block is *proper*.

Our major goal in the next three sections is twofold. On one hand we show that the 1-blocks of any realization of a 1-block or 2-block of the given graph $G = (V, E)$ is again a 1-block or a 2-block of G, assuming that certain conditions are fulfilled. On the other hand, we establish equations for the recursive computation of the treewidth and the minimum fill-in.

Recall the strong relation between a graph G and a minimal triangulation H of G with respect to their minimal separators and the vertex sets of components as described in Theorem 2.10.

5. Decomposing 1-blocks

Consider any minimal triangulation H of a given graph $G = (V, E)$. Let $S \in \mathfrak{Sep}(H)$ and let $B = (S, C)$ be a 1-block with realization $R(S, C)$. Let x and y be non-adjacent vertices in $H[S \cup C]$ and let S^* be a minimal x, y-separator in H. Then $S^* \subset S \cup C$ by Lemmas 2.5. Moreover S and S^* are non-crossing minimal separators of H and G by Lemmas 2.6 and 2.11.

We study how the removal of S^* can decompose the realization of a 1-block $B=(S,C)$. The next three lemmas describe all possible cases and give equations for calculating the treewidth and the minimum number of edges in a triangulation.

Lemma 5.1. *Let $B=(S,C)$ be a 1-block of the graph $G=(V,E)$. Let $S^* \subset S$. Then for any minimal triangulation H of G with $S^*, S \in \mathfrak{Sep}(H)$ and S^* minimal separator of $H[S \cup C]$, $H[(S \cup C)\backslash S^*]$ has exactly two connected components, namely, $H[S\backslash S^*]$ and C. Hence*

$$tw(R(S,C);S^*) = \max\left(|S|-1, tw(R(S^*,C))\right) \tag{8}$$

and

$$met(R(S,C);S^*) = met(R(S^*,C)) + \binom{|S|}{2} - \binom{|S^*|}{2}. \tag{9}$$

Proof. C and $H[S\backslash S^*]$ are each connected induced subgraphs of $H[V\backslash S^*]$. Thus an edge between a vertex of C and a vertex of $S\backslash S^*$ would indicate that S^* is not a separator of $H[S \cup C]$.

Thus for any minimal triangulation H of G with $S, S^* \in \mathfrak{Sep}(H)$, the components of $H[(S\cup C)\backslash S^*]$ are $H[S\backslash S^*]$ and C. Since $H[S\cup C]$ is a minimal triangulation of $R(S,C)$ by Lemma 3.1 and since S^* is a minimal separator of $H[S\cup C]$, Theorem 2.10 implies that S^* is also a minimal separator of $R(S,C)$ and the vertex sets of the connected components of $R(S,C)[(S\cup C)\backslash S^*]$ and $H[(S \cup C)\backslash S^*]$ are the same.

Thus the 1-blocks of the graph $R(S,C)$ with respect to the minimal separator S^* are $B_1 = (S^*, R(S,C)[S \setminus S^*])$ and $B_2 = (S^*, C)$. Notice that the realization $R_1 = (S^*, R(S,C)[S\backslash S^*])$ of B_1 is the complete graph on vertex set S.

The definition of $tw(G;S)$ implies $tw(R(S,C);S^*) = \max_{j=1,2,\ldots,r} tw(R(S^*,C_j))$, where $C_1, C_2, \ldots, C_r$ are the components of $R(S,C)[(S\cup C)\backslash S^*]$. Consequently, $tw(R(S,C);S^*) = \max(|S|-1, tw(R(S^*,C)))$.

The definition of $met(G;S)$ implies $met(R(S,C);S^*) = (1-t)\binom{|S^*|}{2} + \sum_{i=1}^{t} met(R(S^*,C_i))$, where $C_1, C_2, \ldots C_t$ are the components of $R(S,C)[(S \cup C)\backslash S^*]$. Consequently, we obtain $met(R(S,C);S^*) = met(R(S^*,C)) + \binom{|S|}{2} - \binom{|S^*|}{2}$. $\square$

We have demonstrated in this proof how the knowledge of the connected components of $H[(S \cup C)\backslash S^*]$ for all minimal triangulations H of G with $S, S^* \in \mathfrak{Sep}(H)$ and S^* a minimal separator of $H[S \cup C]$ can be used to compute $tw(R(S,C);S^*)$ and $met(R(S,C);S^*)$. Using the definitions of $tw(G;S)$ and $met(G;S)$ this only requires to know all the 1-blocks of $R(S,C)$ with respect to the minimal separator S^*. This part is omitted in the proofs of the Lemmas 5.2 and 5.3, since it can be done analogously.

Lemma 5.2. *Let $B=(S,C)$ be a 1-block of the graph $G=(V,E)$. Let $S \subset S^*$. Then for any minimal triangulation H of G with $S^*, S \in \mathfrak{Sep}(H)$ and S^* minimal separator* *of $H[S \cup C]$, the connected components of $H[(S \cup C) \backslash S^*]$ are exactly those connected components $C_1, C_2, \ldots, C_t$ of $H[V \backslash S^*]$ for which $C_j \subseteq C$, $j \in \{1,2,\ldots,t\}$. Hence*

$$tw(R(S,C);S^*) = \max_{i=1,2,\ldots,t} tw(R(S^*,C_i)) \tag{10}$$

and

$$met(R(S,C);S^*) = (1-t)\binom{|S^*|}{2} + \sum_{i=1}^{t} met(R(S^*,C_i)). \tag{11}$$

Proof. $S \subset S^*$ implies that the vertex set of any component of $H[V \backslash S^*]$ is either a subset of C or disjoint from $S \cup C$. □

Lemma 5.3. *Let $B = (S,C)$ be a 1-block of the graph $G = (V,E)$. Let $S \not\subseteq S^*$ and $S^* \not\subseteq S$. Then for any minimal triangulation H of G with $S^*, S \in \mathfrak{Sep}(H)$ and S^* minimal separator of $H[S \cup C]$, the connected components of $H[(S \cup C) \backslash S^*]$ are $H[\mathfrak{B}(S,S^*) \backslash S^*]$ and the components $C_1, C_2, \ldots, C_t$ of $H[V \backslash S^*]$ with $C_j \subseteq C$, $j \in \{1,2,\ldots,t\}$. Hence*

$$tw(R(S,C);S^*) = \max\left(tw(\mathfrak{R}(S,S^*)), \max_{i=1,2,\ldots,t} tw(R(S^*,C_i))\right) \tag{12}$$

and

$$met(R(S,C);S^*) = met(\mathfrak{R}(S,S^*)) - t\binom{|S^*|}{2} + \sum_{i=1}^{t} met(R(S^*,C_i)). \tag{13}$$

Proof. First we show that $\mathfrak{B}(S,S^*) \subset S \cup C$. Let $z \in \mathfrak{B}(S,S^*)$. Assume $z \notin S$. Then z and $S^* \backslash S$ are contained in one connected component of $H[V \backslash S]$. Since $S^* \backslash S \neq \emptyset$, and $S^* \subset S \cup C$, it follows that z is a vertex of C.

S^* is a minimal x,y-separator in H. Since S and S^* are non-crossing minimal separators of H, $S \backslash S^*$ is contained in one component of $H[V \backslash S^*]$. Therefore x and y cannot both belong to $\mathfrak{B}(S,S^*)$. It follows that $\mathfrak{B}(S,S^*) \neq S \cup C$. Hence $\mathfrak{B}(S,S^*) \subset S \cup C$.

Now we prove that $\mathfrak{B}(S,S^*) \backslash S^*$ is indeed the vertex set of a component of $H[(S \cup C) \backslash S^*]$. $S \not\subseteq S^*$ implies that there is one connected component A of $H[V \backslash S^*]$ containing $S \backslash S^*$. Moreover, $\mathfrak{B}(S,S^*) \backslash S^*$ is also contained in A.

Let X be an arbitrary component of $H[C \backslash S^*]$. Assume that the vertices of X are contained in a connected component Y of $H[V \backslash S^*]$. Clearly $X = Y$ implies that X is a component of $H[V \backslash S^*]$ with $X \subseteq C$. If $X \neq Y$, then Y must contain $S \backslash S^*$, hence $X \subseteq A$. Moreover there is exactly one component X of $H[C \backslash S^*]$ with $X \neq Y$. On the other hand, X is contained in C which is the component of $H[V \backslash S]$ containing $S^* \backslash S \neq \emptyset$. Therefore in this case we have $X \cup S \cup S^* = \mathfrak{B}(S,S^*)$. □

Notice that the above lemma requires the study of 2-blocks.

6. Degenerate 2-blocks

Degenerate 2-blocks are very easy to handle. Here we obtain simple equations for the computation of $tw(\mathfrak{R}(S_1,S_2))$ and $met(\mathfrak{R}(S_1,S_2))$. First we consider the case that S_1 and S_2 are not equal.

Lemma 6.1. *Let $\mathfrak{B}(S_1,S_2)$ be a degenerate 2-block of G with $S_1 \subset S_2$. Then*

$$tw(\mathfrak{R}(S_1,S_2)) = \max_{i=1,2,\dots,t} tw(R(S_2,C_i)) \tag{14}$$

and

$$met(\mathfrak{R}(S_1,S_2)) = (1-t)\binom{|S_2|}{2} + \sum_{i=1}^{t} met(R(S_2,C_i)), \tag{15}$$

where $C_1, C_2, \dots, C_t$ are the connected components of $G[V \setminus S_2]$, for which the vertex set is contained in the component of $G[V \setminus S_1]$, that contains $S_2 \setminus S_1$.

Proof. $\mathfrak{B}(S_1,S_2)$ consists of a minimal separator S_1 and the vertex set of the connected component A of $H[V \setminus S_1]$ that contains $S_2 \setminus S_1$. Hence $\mathfrak{B}(S_1,S_2) = S_1 \cup A$, where $B=(S_1,A)$ is a 1-block of H and G. However, the realizations of $\mathfrak{B}(S_1,S_2)$ and $B=(S_1,A)$ may not be equal, since S_2 is a clique in $\mathfrak{R}(S_1,S_2)$ but S_2 is not necessarily a clique in $R(S_1,A)$. Nevertheless, S_2 is a minimal separator of G with $S_2 \subset S_1 \cup A$. Hence $tw(\mathfrak{R}(S_1,S_2)) = tw(R(S_1,A);S_2)$ and $met(\mathfrak{R}(S_1,S_2)) = met(R(S_1,A);S_2)$. Lemma 5.2 implies Equation 14 and 15. □

For the case $S_1 = S_2$, the following lemma can be obtained.

Lemma 6.2. *Let $\mathfrak{B}(S,S)$ be a degenerate 2-block of the graph $G=(V,E)$. Then $\mathfrak{R}(S,S)$ is the graph obtained from G by making a clique of S. Hence*

$$tw(\mathfrak{R}(S,S)) = \max_{i=1,2,\dots,t} tw(R(S,C_i)) \tag{16}$$

and

$$met(\mathfrak{R}(S,S)) = (1-t)\binom{|S|}{2} + \sum_{i=1}^{t} met(R(S,C_i)), \tag{17}$$

where $C_1, C_2, \dots, C_t$ are the connected components of $G[V \setminus S]$.

Consequently, the treewidth and the minimum number of edges in any triangulation of a graph G can be computed as follows:

$$tw(G) = \min_{S \in \mathfrak{Sep}(G)} tw(\mathfrak{R}(S,S)), \tag{18}$$

$$met(G) = \min_{S \in \mathfrak{Sep}(G)} met(\mathfrak{R}(S,S)). \tag{19}$$

7. Decomposing proper 2-blocks

This section shows how realizations of proper 2-blocks are decomposed when a minimal separator (of a certain type) is removed. The lemmas give equations for calculating the treewidth and the minimum number of edges in a triangulation of the realization of a proper 2-block. Recall that $\mathfrak{B}(S_1,S_2)$ is a proper 2-block if $S_1 \not\subseteq S_2$ and $S_2 \not\subseteq S_1$.

Lemma 7.1. *Let $\mathfrak{B}(S_1,S_2)$ be a proper 2-block of the graph $G=(V,E)$ and let H be any minimal triangulation of G such that $S_1, S_2 \in \mathfrak{Sep}(H)$. Let x,y be non-adjacent vertices of $H[\mathfrak{B}(S_1,S_2)]$ and let S^* be a minimal x,y-separator of H. Then the following holds:*

1. $S^* \subset \mathfrak{B}(S_1,S_2)$.
2. S_1 *and* S^* *are non-crossing in* G *and also* S_2 *and* S^* *are non-crossing in* G.
3. $S^* \neq S_1$ *and* $S^* \neq S_2$.

Proof. By Lemma 4.1, the 2-blocks of S_1 and S_2 in G and in H are equal which enables us to analyse H instead of G. By assumption $S_2 \setminus S_1 \neq \emptyset$. Consider the connected component A of $H[V\setminus S_1]$ that contains $S_2 \setminus S_1$. Clearly x and y are both contained in $\mathfrak{B}(S_1,S_2) \subseteq S_1 \cup A$. It follows by Lemma 2.5 that S^* is also contained in $S_1 \cup A$. Hence $S^* \setminus (S_1 \cup S_2)$ and $S_2 \setminus S_1$ are both contained in the component A. One can show analogously that $S^* \setminus (S_1 \cup S_2)$ and $S_1 \setminus S_2$ are contained in one connected component of $H[V\setminus S_2]$. This implies $S^* \subset \mathfrak{B}(S_1,S_2)$.

Since S_1, S_2 and S^* are minimal separators in H they are pairwise non-crossing in G by Lemma 2.11.

Consider again the connected component A of $H[V\setminus S_1]$ which contains $S_2 \setminus S_1$. Since x and y are both contained in $S_1 \cup A$, S_1 cannot be a minimal x,y-separator. Hence $S_1 \neq S^*$. Analogously $S_2 \neq S^*$. □

The following assumptions of Lemma 7.1 are also made in all the subsequent lemmas of this section without mentioning all of them explicitly in each lemma. We assume that $\mathfrak{B}(S_1,S_2)$ is a proper 2-block of the graph $G=(V,E)$ and that H is any minimal triangulation of G such that $S_1,S_2 \in \mathfrak{Sep}(H)$. Furthermore we assume that x and y are non-adjacent vertices of $H[\mathfrak{B}(S_1,S_2)]$ and that S^* is a minimal x,y-separator of H.

Lemmas 7.2–7.4 consider the degenerate cases for the decomposition of the realization of a proper 2-block $\mathfrak{B}(S_1,S_2)$ by the removal of a minimal x,y-separator S^* of G with $x,y \in \mathfrak{B}(S_1,S_2)$.

Lemma 7.2. *Let $\mathfrak{B}(S_1,S_2)$ be a proper 2-block of the graph $G=(V,E)$. Let $S_1,S_2 \subset S^*$. Then for any minimal triangulation H of G with $S^*,S_1,S_2 \in \mathfrak{Sep}(H)$ and S^* a minimal separator of $H[\mathfrak{B}(S_1,S_2)]$, the connected components of $H[\mathfrak{B}(S_1,S_2)\setminus S^*]$ are those connected components $D_1,\dots,D_t$ of $H[V\setminus S^*]$ for which $D_j \subseteq \mathfrak{B}(S_1,S_2)$,*

$j \in \{1,2,\ldots,t\}$. *Hence*

$$tw(\mathfrak{R}(S_1,S_2);S^*) = \max_{i=1,2,\ldots,t} tw(R(S^*,D_i)) \tag{20}$$

and

$$met(\mathfrak{R}(S_1,S_2);S^*) = (1-t)\binom{|S^*|}{2} + \sum_{i=1}^{t} met(R(S^*,D_i)). \tag{21}$$

Proof. Let A be a connected component of $H[V\setminus S^*]$. Either $A \subset \mathfrak{B}(S_1,S_2)\setminus S^*$ or $A \cap (\mathfrak{B}(S_1,S_2)\setminus S^*) = \emptyset$, since A is a connected induced subgraph of $H[V\setminus(S_1 \cup S_2)]$. Thus the connected components of $H[\mathfrak{B}(S_1,S_2)\setminus S^*]$ are those components $D_1,\ldots,D_t$ of $H[V\setminus S^*]$ for which $D_j \subseteq \mathfrak{B}(S_1,S_2)$, $j \in \{1,2,\ldots,t\}$.

We claim that the chordal graph $H[\mathfrak{B}(S_1,S_2)]$ is a minimal triangulation of $\mathfrak{R}(S_1,S_2)$. Since H is a minimal triangulation of G, each edge $e \in E(H)\setminus E(G)$ is unique chord of a cycle of length four in H, by Corollary 2.8. Let $e=\{a,b\}$ be an edge of $H[\mathfrak{B}(S_1,S_2)]$ and suppose a vertex z of the cycle with unique chord e does not belong to $\mathfrak{B}(S_1,S_2)$. Without loss of generality z is not in the component of $G[V\setminus S_1]$ containing $S_2\setminus S_1$. Since a and b cannot both belong to the clique S_1 of $\mathfrak{R}(S_1,S_2)$, they are not both adjacent to z, a contradiction.

Since $H[\mathfrak{B}(S_1,S_2)]$ is a minimal triangulation of $\mathfrak{R}(S_1,S_2)$ and since S^* is a minimal separator of $H[\mathfrak{B}(S_1,S_2)]$, S^* is also a minimal separator of $\mathfrak{R}(S_1,S_2)$ and the components of $H[\mathfrak{B}(S_1,S_2)\setminus S^*]$ and $\mathfrak{R}(S_1,S_2)[\mathfrak{B}(S_1,S_2)\setminus S^*]$ are the same.

Therefore the 1-blocks of the graph $\mathfrak{R}(S_1,S_2)$ with respect to the minimal separator S^* are $B_1=(S^*,D_1), B_2=(S^*,D_2),\ldots, B_t=(S^*,D_t)$.

Now the Eqs. 20 and 21 follow immediately from the definitions of $tw(G;S)$ and $met(G;S)$, analogously to the proof of Lemma 5.1. □

We have demonstrated in this proof how the Eqs. (20) and (21) can be obtained easily, if the components of $H[\mathfrak{B}(S_1,S_2)\setminus S^*]$ are known. This part is omitted in all subsequent proofs, since it can be done analogously.

Lemma 7.3. *Let $\mathfrak{B}(S_1,S_2)$ be a proper 2-block of the graph $G=(V,E)$. Let $S_1 \subset S^*$ and $S_2\setminus S^* \neq \emptyset$. Then for any minimal triangulation H of G with $S^*,S_1,S_2 \in \mathfrak{Sep}(H)$ and S^* a minimal separator of $H[\mathfrak{B}(S_1,S_2)]$, the connected components of $H[\mathfrak{B}(S_1,S_2)\setminus S^*]$ are the connected components $D_1,\ldots,D_t$ of $H[V\setminus S^*]$ with $D_j \subseteq \mathfrak{B}(S_1,S_2)$, $j \in \{1,2,\ldots,t\}$, and $H[\mathfrak{B}(S_2,S^*)\setminus S^*]$. Hence*

$$tw(\mathfrak{R}(S_1,S_2);S^*) = \max\left(tw(\mathfrak{R}(S_2,S^*)), \max_{i=1,2,\ldots,t} tw(R(S^*,D_i))\right) \tag{22}$$

and

$$met(\mathfrak{R}(S_1,S_2);S^*) = -t\binom{|S^*|}{2} + met(\mathfrak{R}(S_2,S^*)) + \sum_{i=1}^{t} met(R(S^*,D_i)). \tag{23}$$

Proof. Consider the connected components of $H[V \setminus S^*]$. One of these, say A, contains $S_2 \setminus S^*$. The vertex set of any other component of $H[V \setminus S^*]$ is either completely contained in $\mathfrak{B}(S_1, S_2)$ or disjoint from it.

Let $z \in \mathfrak{B}(S_1, S_2) \cap A$. We show that $z \in \mathfrak{B}(S_2, S^*)$. If $z \in S_2 \setminus S^*$ this is clear, hence assume $z \in A \setminus S_2$. Since $z \in \mathfrak{B}(S_1, S_2)$, it is contained in the connected component of $H[V \setminus S_2]$, that contains $S_1 \setminus S_2$. By Lemma 7.1 $S^* \subset \mathfrak{B}(S_1, S_2)$, hence also $S^* \setminus S_2$ is in the component of $H[V \setminus S_2]$, that contains $S_1 \setminus S_2$. Hence z is in the component of $H[V \setminus S_2]$ that contains $S^* \setminus S_2$. Since z is also in the component of $H[V \setminus S^*]$ that contains $S_2 \setminus S^*$, it follows that $z \in \mathfrak{B}(S_2, S^*)$.

Finally, we have to show that $\mathfrak{B}(S^*, S_2) \subseteq \mathfrak{B}(S_1, S_2)$. Let $z \in \mathfrak{B}(S^*, S_2)$. If $z \in S^* \cup S_2$ then clearly $z \in \mathfrak{B}(S_1, S_2)$. Hence assume $z \notin S^* \cup S_2$. Then z is a vertex of A. Since $S_1 \subset S^*$, the vertex set of A is contained in the connected component of $H[V \setminus S_1]$ that contains $S_2 \setminus S_1$.

Furthermore, z is in the connected components of $H[V \setminus S_2]$ that contains $S^* \setminus S_2$. This component also contains $S_1 \setminus S_2$. Consequently, $z \in \mathfrak{B}(S_1, S_2)$. □

Lemma 7.4. *Let $\mathfrak{B}(S_1, S_2)$ be a proper 2-block of the graph $G = (V, E)$. Let $S^* \subset S_1$ and $S^* \subset S_2$. Then for any minimal triangulation H of G with $S^*, S_1, S_2 \in \mathfrak{Sep}(H)$ and S^* a minimal separator of $H[\mathfrak{B}(S_1, S_2)]$, $\mathfrak{B}(S_1, S_2) = S_1 \cup S_2$ and the graph $H[\mathfrak{B}(S_1, S_2)]$ has exactly two maximal cliques, namely, S_1 and S_2. Hence*

$$tw(\mathfrak{R}(S_1, S_2)) = \max(|S_1| - 1, |S_2| - 1) \tag{24}$$

and

$$met(\mathfrak{R}(S_1, S_2); S^*) = \binom{|S_1|}{2} + \binom{|S_2|}{2} - \binom{|S^*|}{2}. \tag{25}$$

Proof. For $i \in \{1, 2\}$, let A_i be the connected component of $H[V \setminus S^*]$, that contains $S_i \setminus S^*$. Notice that the connected component of $H[V \setminus S_1]$ that contains $S_2 \setminus S_1$ is exactly A_2, that the connected component of $H[V \setminus S_2]$ that contains $S_1 \setminus S_2$ is exactly A_1 and that $A_1 \neq A_2$, since otherwise S^* would not be a minimal separator of $H[\mathfrak{B}(S_1, S_2)]$. Hence $S_1 \cap S_2 = S^*$ and $\mathfrak{B}(S_1, S_2) \setminus (S_1 \cup S_2) = A_1 \cap A_2 = \emptyset$. □

Lemma 7.5. *Let $\mathfrak{B}(S_1, S_2)$ be a proper 2-block of the graph $G = (V, E)$. Let $S^* \subset S_1$, $S^* \not\subseteq S_2$ and $S_2 \not\subseteq S^*$. Then for any minimal triangulation H of G with $S^*, S_1, S_2 \in \mathfrak{Sep}(H)$ and S^* a minimal separator of $H[\mathfrak{B}(S_1, S_2)]$, the connected components of $H[\mathfrak{B}(S_1, S_2) \setminus S^*]$ are $H[S_1 \setminus S^*]$ and $H[\mathfrak{B}(S_2, S^*) \setminus S^*]$. Hence*

$$tw(\mathfrak{R}(S_1, S_2); S^*) = \max\left(|S_1| - 1, tw(\mathfrak{R}(S_2, S^*))\right) \tag{26}$$

and

$$met(\mathfrak{R}(S_1, S_2); S^*) = met(\mathfrak{R}(S_2, S^*)) + \binom{|S_1|}{2} - \binom{|S^*|}{2}. \tag{27}$$

Proof. For $i \in \{1,2\}$, let A_i be the connected component of $H[V \setminus S^*]$ that contains $S_i \setminus S^*$. The connected component of $H[V \setminus S_1]$ that contains $S_2 \setminus S_1$ is A_2. It follows that $\mathfrak{B}(S_1,S_2) \setminus S_1 \subseteq A_2$, hence $\mathfrak{B}(S_1,S_2) \subseteq A_2 \cup S_1$.

Let $z \in \mathfrak{B}(S_1,S_2) \setminus S_1$. We show that $z \in \mathfrak{B}(S^*,S_2)$. By definition, z is in the component of $H[V \setminus S_2]$ that contains $S_1 \setminus S_2$. This component also contains $S^* \setminus S_2$, since $S^* \subset \mathfrak{B}(S_1,S_2)$. Since z is a vertex of A_2, it follows that $z \in \mathfrak{B}(S^*,S_2)$.

It remains to show that $\mathfrak{B}(S^*,S_2) \subseteq \mathfrak{B}(S_1,S_2)$. Let $z \in \mathfrak{B}(S^*,S_2) \setminus (S^* \cup S_2)$. Then z is in A_2. Hence z is in the component of $H[V \setminus S_1]$ that contains $S_2 \setminus S_1$.

Furthermore, z belongs to the connected component of $H[V \setminus S_2]$ that contains $S^* \setminus S_2$. Since $S^* \setminus S_2 \neq \emptyset$, this component is uniquely determined and contains also $S_1 \setminus S_2$, since S_1 is a clique containing S^*. Consequently, $z \in \mathfrak{B}(S_1,S_2)$. □

Recall that any minimal triangulation H of an AT-free graph G is an interval graph by Theorem 2.13.

Lemma 7.6. *Let H be a minimal triangulation of G into an interval graph. Let $S_1, S_2 \in \mathfrak{Sep}(H)$, $S_1 \neq S_2$, S^* a minimal separator of $H[\mathfrak{B}(S_1,S_2)]$ and $S_1 \setminus S^* \neq \emptyset$, $S_2 \setminus S^* \neq \emptyset$. Then $S_1 \setminus S^*$ and $S_2 \setminus S^*$ are contained in different connected components of $H[V \setminus S^*]$.*

Proof. Since H is an interval graph, there is a consecutive clique arrangement $A_1, A_2, \ldots, A_q$ of H. By Theorem 2.4, there are indices i and j such that $S_1 = A_i \cap A_{i+1}$ and $S_2 = A_j \cap A_{j+1}$. Assume $i < j$. Then the 2-block of S_1 and S_2 is contained in $\bigcup_{k=i+1}^{j} A_k$. Let S^* be a minimal x,y-separator of $H[\mathfrak{B}(S_1,S_2)]$. Hence x and y belong to $\mathfrak{B}(S_1,S_2)$. Thus there is an index $i < k < j$ such that $S^* = A_k \cap A_{k+1}$. Consequently, $S_1 \not\subseteq S^*$ and $S_2 \not\subseteq S^*$ implies that S^* separates $S_1 \setminus S^*$ and $S_2 \setminus S^*$. □

The following lemma treats the last and the most important case for the decomposition of the realization of a proper 2-block by the removal of a suitable minimal separator S^* of G. It is the only lemma requiring an additional assumption on the minimal triangulations H to which the lemma can be applied. This forces indeed a significant restriction for possible applications of our approach.

Lemma 7.7. *Let $\mathfrak{B}(S_1,S_2)$ be a proper 2-block of the graph $G=(V,E)$. Let $S_i \not\subseteq S^*$ and $S^* \not\subseteq S_i$ for $i \in \{1,2\}$. Then for any minimal triangulation H of G with $S^*, S_1, S_2 \in \mathfrak{Sep}(H)$ and S^* a minimal separator of $H[\mathfrak{B}(S_1,S_2)]$, that separates $S_1 \setminus S^*$ and $S_2 \setminus S^*$ into different components of $H[V \setminus S^*]$, the connected components of $H[\mathfrak{B}(S_1,S_2) \setminus S^*]$ are $H[\mathfrak{B}(S_1,S^*) \setminus S^*]$, $H[\mathfrak{B}(S_2,S^*) \setminus S^*]$ and the connected components $D_1, \ldots, D_t$ of $H[V \setminus S^*]$ with $D_j \subseteq \mathfrak{B}(S_1,S_2)$, $j \in \{1,2,\ldots,t\}$, that are contained neither in the component of $H[V \setminus S^*]$ containing $S_1 \setminus S^*$ nor in the component of $H[V \setminus S^*]$ containing $S_2 \setminus S^*$. Then*

$$tw(\mathfrak{R}(S_1,S_2); S^*) = \max\left(tw(\mathfrak{R}(S_1,S^*)), tw(\mathfrak{R}(S_2,S^*)), \max_{i=1,2,\ldots,t} tw(R(S^*,D_i))\right) \tag{28}$$

and

$$met(\mathfrak{R}(S_1,S_2);S^*) = (-t-1)\binom{|S^*|}{2} + met(\mathfrak{R}(S_1,S^*))$$
$$+met(\mathfrak{R}(S_2,S^*)) + \sum_{i=1}^{t} met(R(S^*,D_i)). \qquad (29)$$

Proof. For $i \in \{1,2\}$, let A_i be the connected component of $H[V \setminus S^*]$ which contains $S_i \setminus S^*$. By our assumptions, $S_2 \setminus S^*$ and $S_1 \setminus S^*$ are contained in different connected components of $H[V \setminus S^*]$. Hence A_1 and A_2 are indeed two different components of $H[V \setminus S^*]$. Then the vertex set of any connected component of $H[V \setminus S^*]$ different from A_1 and A_2 is either a subset of $\mathfrak{B}(S_1,S_2)$ or disjoint from $\mathfrak{B}(S_1,S_2)$. Notice that $\mathfrak{B}(S_1,S^*) \setminus S^* \subseteq A_1$ and $\mathfrak{B}(S_2,S^*) \setminus S^* \subseteq A_2$.

Now let $z \in A_1 \cap \mathfrak{B}(S_1,S_2)$. We show that $z \in \mathfrak{B}(S_1,S^*)$. Since z and $S^* \setminus S_1$ are both in $\mathfrak{B}(S_1,S_2)$, and since $S^* \setminus S_1 \neq \emptyset$, it follows that z and $S^* \setminus S_1$ are contained in one connected component of $H[V \setminus S_1]$. Since z is a vertex of A_1, it follows that $z \in \mathfrak{B}(S_1,S^*)$.

We now show that $\mathfrak{B}(S_1,S^*) \subset \mathfrak{B}(S_1,S_2)$. Let $z \in \mathfrak{B}(S_1,S^*)$. Since $S_1 \setminus S^* \neq \emptyset$ it follows that $z \in S^* \cup A_1$. If $z \in S^* \cup S_1$ then $z \in \mathfrak{B}(S_1,S_2)$. Hence we may assume that $z \in A_1 \setminus S_1$.

Now z and $S^* \setminus S_1$ are in one connected component of $H[V \setminus S_1]$ since $z \in \mathfrak{B}(S_1,S^*)$. Since $S^* \subset \mathfrak{B}(S_1,S_2)$ and $S^* \setminus S_1 \neq \emptyset$, this component of $H[V \setminus S_1]$ also contains $S_2 \setminus S_1$. It follows that z and $S_2 \setminus S_1$ are in the same connected component of $H[V \setminus S_1]$. $z \in A_1$ implies that z is a vertex of the component of $H[V \setminus S_2]$ that contains S_1. Thus $z \in \mathfrak{B}(S_1,S_2)$. Consequently, $A_1 \cap \mathfrak{B}(S_1,S_2) = \mathfrak{B}(S^*,S_1) \setminus S^*$.

$A_2 \cap \mathfrak{B}(S_1,S_2) = \mathfrak{B}(S_2,S^*) \setminus S^*$ can be shown analogously. □

Note that the assumption of the lemma is fulfilled for all minimal triangulations of an AT-free graph G by Lemma 7.6 and since any minimal triangulation of an AT-free graph is an interval graph.

8. The algorithms

We present algorithms for computing the treewidth and the minimum fill-in of the input graph. The algorithm computing the treewidth works correctly, if there is a minimal triangulation of the input graph G into an interval graph H with $tw(G) = \omega(H) - 1$. The algorithm computing the minimum fill-in of the input graph G works correctly, if there is a minimal triangulation of the input graph G into an interval graph H with $met(G) = |E(H)|$. Hence both algorithms work correctly for AT-free graphs.

Both algorithms are very similar. They only differ in the equations, that have to be used in the recursion phase (step 5). First we describe the algorithm computing the treewidth.

Step 1: List all minimal separators of the input graph $G = (V,E)$ using the algorithm of [11].

Step 2: For every minimal separator S of G and every connected component C of $G[V \setminus S]$, compute the 1-block $B = (S, C)$ and $|S \cup C|$, i.e., the number of vertices of $R(S, C)$.

Step 3: For each pair of minimal separators S_1 and S_2 of G, check whether S_1 and S_2 are non-crossing. If so, compute the 2-block $\mathfrak{B}(S_1, S_2)$ and $|\mathfrak{B}(S_1, S_2)|$, i.e., the number of vertices of $\mathfrak{R}(S_1, S_2)$.

Step 4: Order the blocks by increasing number of vertices (no matter whether they are 1-blocks or 2-blocks).

Step 5: In the order of step 4, compute for each block the treewidth of its realization. If the realization of the block is a complete graph then its treewidth is the number of vertices of the block decreased by one. Otherwise, there are three possible cases.

Step 5.1: The current block is the 1-block $B = (S, C)$.

For each minimal separator S^* of G, check whether S^* is also a minimal separator of $G[S \cup C]$. If so, compute $tw(R(S, C); S^*)$ using the suitable equation of Section 5, i.e., Eqs. (8), (10) or (12). Finally, compute $tw(R(S, C)) = \min_{S^*} tw(R(S, C); S^*)$, where the minimization is over all minimal separators S^* of G that are also minimal separators of $G[S \cup C]$.

Step 5.2: The current block is the degenerate 2-block $\mathfrak{B}(S_1, S_2)$.

Compute $tw(\mathfrak{R}(S_1, S_2))$ using the suitable equation of Section 6, i.e., Eq. (14) or (16).

Step 5.3: The current block is the proper 2-block $\mathfrak{B}(S_1, S_2)$.

For every minimal separator S^* of G, check whether S^* is a minimal separator of $G[\mathfrak{B}(S_1, S_2)]$. If $S_1 \setminus S^* \neq \emptyset$, $S_2 \setminus S^* \neq \emptyset$, $S^* \setminus S_1 \neq \emptyset$ and $S^* \setminus S_2 \neq \emptyset$, then check whether $S_1 \setminus S^*$ and $S_2 \setminus S^*$ are in different components of $G[V \setminus S^*]$. For each S^* that fulfills the conditions, compute $tw(\mathfrak{R}(S_1, S_2); S^*)$ using the suitable equation of Section 7, i.e., Eqs. (20), (22), (24), (26) or (28). Finally, compute $tw(\mathfrak{R}(S_1, S_2)) = \min_{S^*} tw(\mathfrak{R}(S_1, S_2); S^*)$, where the minimization is over all minimal separators S^* of G that fulfill all the conditions (checked at the beginning of step 5.3).

Step 6: Output $\min_{S \in \mathfrak{Sep}(G)} tw(\mathfrak{R}(S, S))$.

The algorithm computing the minimum fill-in of the input graph G is quite similar. Therefore we only mention the differences to the treewidth algorithm. Clearly, in step 5 of the algorithm the equations concerning the maximum number of edges in any triangulation of the realization of the block have to be used. Finally, the algorithm computes $met(G) = \min_{S \in \mathfrak{Sep}(G)} met(\mathfrak{R}(S, S))$ by Eq. 19 and outputs $mfi(G) = met(G) - |E(G)|$.

Theorem 8.1. *There is an* $O(n^5 R + n^3 R^3)$ *algorithm computing the treewidth and the pathwidth of any given graph G, that has a minimal triangulation H such that H is an interval graph and* $tw(G) = \omega(H) - 1$. *There is an* $O(n^5 R + n^3 R^3)$ *algorithm computing minimum fill-in and minimum interval graph completion of any given graph G, that has a minimal triangulation H such that H is an interval graph and* $\text{met}(G) = |E(H)|$. *Here R is the number of minimal separators of the input graph G.*

Proof. Let $G = (V, E)$ be the input graph. Using the algorithm listing all minimal separators of any given graph, presented in [11], all minimal separators of G can be computed in time $O(n^5 R)$. G has at most nR different 1-blocks and at most R^2 different 2-blocks. Clearly, steps 2–4 of the algorithm can be done within the stated timebound.

The most time consuming step of the algorithm is step 5.3, that we consider now. For each of the at most R^2 different proper 2-blocks $\mathfrak{B}(S_1, S_2)$, there are R minimal separators S^* of G to check. The algorithm computes the components of $G[\mathfrak{B}(S_1, S_2) \setminus S^*]$ and checks whether S^* is a minimal separator of $G[\mathfrak{B}(S_1, S_2)]$ in time $O(n + m)$. $G[\mathfrak{B}(S_1, S_2) \setminus S^*]$ has at most n smaller components. For any component D_i either $B = (S^*, D_i)$ is a 1-block of G or D_i is subset of the 2-block $\mathfrak{B}(S_1, S^*)$ and $\mathfrak{B}(S_2, S^*)$, respectively. This depends on the inclusion relation of the three sets S_1, S_2 and S^* and is specified in the corresponding lemma of Section 7. Hence the corresponding 1-blocks and 2-blocks can be computed in time $O(n^2)$.

For each of these smaller blocks, we can look up the treewidth of its realization in $O(n)$ time using a suitable data structure. Thus we can find the treewidth of a proper 2-block in $O(Rn^3)$ time. Consequently, step 5.3 can be done in time $O(n^3 R^3)$.

It is not hard to see that steps 5.1 and 5.2 can be done in time $O(n^3 R^3)$ in a similar fashion. Hence the algorithm computes the treewidth of G in time $O(n^3 R^3)$ from the list of all minimal separators of G.

The time analysis of the algorithm computing the minimum fill-in can be done analogously, since the two algorithms differ only in the equations, which does not influence the running time.

Consider the correctness of the treewidth algorithm. Except in the case of Lemma 7.7, the algorithm computes the correct value of the treewidth of the realization of the current block, if the treewidth of the realizations of all the smaller blocks is correct. This is shown for a 1-block in Lemmas 5.1–5.3 and for a proper 2-block in Lemmas 7.2–7.5, and 7.7.

Unfortunately, to make sure that $tw(\mathfrak{R}(S_1, S_2); S^*)$ is computed correctly, if S_1, S_2 and S^* are pairwise incomparable by inclusion, we must require that S^* separates $S_1 \setminus S^*$ and $S_2 \setminus S^*$. If H is a minimal triangulation into an interval graph, then such a minimal separator S^* exists by Lemma 7.6. If there is a minimal triangulation of G into an interval graph H with $tw(G) = \omega(H) - 1$, then the algorithm computes $tw(\mathfrak{R}(S_1, S_2))$ correctly, since a minimal separator S^* with $tw(\mathfrak{R}(S_1, S_2)) = tw(\mathfrak{R}(S_1, S_2); S^*)$ exists for all non-crossing separators $S_1, S_2 \in \mathfrak{Sep}(G)$, as long as $\mathfrak{B}(S_1, S_2)$ is not a clique of G. Therefore the algorithm computes the treewidth of the realization of any block correctly and by Eq. (18) the output is indeed $tw(G)$.

Analogously, the correctness of the minimum fill-in algorithm can be shown.

Finally, note that the two different assumptions on the input graph G imply $tw(G) = pw(G)$ and $mfi(G) = mic(G)$, respectively. □

Theorem 2.13 implies that the conditions of Theorem 8.1 are fulfilled for all AT-free graphs.

Corollary 8.2. *There are $O(n^5R + n^3R^3)$ algorithms computing the treewidth, pathwidth, minimum fill-in and minimum interval graph completion of a given* AT*-free graph, where R is the number of minimal separators of the input graph.*

9. Conclusions

Certainly, the $O(n^5R+n^3R^3)$ algorithms to compute the treewidth and the pathwidth as well as the minimum fill-in and the minimum interval graph completion of a given AT-free graph are polynomial time algorithms if the class of input graphs is restricted to any subclass of the AT-free graphs that has a polynomially bounded number of minimal separators. Of course this does not lead to competitive timebounds for graph classes such as permutation graphs or trapezoid graphs.

The situation is more interesting for the d-trapezoid graphs and their proper subclass, the cocomparability graphs of dimension at most d, for any fixed positive integer $d \geqslant 3$. Any d-trapezoid graph (with $n \geqslant 2$) has at most $(2n-3)^d$ minimal separators [17] (see also [10, 13]). Hence the running time of the two algorithms is $O(n^{3d+3})$, if the input graph is a d-trapezoid graph, $d \geqslant 3$ a fixed integer.

But there is still a problem. Unfortunately, the recognition problem for d-trapezoid graphs, and also the one for cocomparability graphs of dimension at most d, is NP-complete for any fixed $d \geqslant 3$ [23]. Even worse, no good approximation algorithms are known for the dimension and the interval dimension problem of partially ordered sets. Consequently, for each fixed $d \geqslant 3$, we cannot check whether the input graph is indeed a d-trapezoid graph in polynomial time. However exactly this assumption guarantees, that R is bounded by a polynomial in n and that we obtain polynomial time algorithms to compute the treewidth and the minimum fill-in.

Nevertheless there is an easy solution to this dilemma. We emphasize that the corresponding polynomial time algorithms do not require an intersection model as part of the input and that they will also not compute an intersection model. We describe how to modify the algorithm for d-trapezoid graphs, for any fixed $d \geqslant 3$. The approach is similar for cocomparability graphs of dimension at most d.

The input is a graph $G=(V,E)$ and no intersection model is required as part of the input. If the input graph G is not a d-trapezoid graph, there are three possible outcomes. If G is not a cocomparability graph, which can be checked in time $O(n^{\alpha})$ [20], then the algorithm rejects the input. (Here $O(n^{\alpha})$ is the time for multiplying two binary $n \times n$ matrices.) If G is a cocomparability graph, the algorithm lists t minimal separators of G in time $O(n^5t)$, if G has at least t minimal separators, using the algorithm of [11]. We set $t=(2n-3)^d+1$. If the number of minimal separators is larger than $(2n-3)^d$, then the algorithm rejects the input, after finding $t=(2n-3)^d+1$ minimal separators of G. If the number of minimal separators is at most $(2n-3)^d$ and G is a cocomparability graph, then the algorithm, computing the treewidth and pathwidth (or the minimum fill-in and the minimum interval graph completion) of AT-free graphs, presented in the previous section, is applied to G and computes the parameter correctly, even if G is not a d-trapezoid graph.

Corollary 9.1. *For each $d \geqslant 3$, there exist $O(n^{3d+3})$ time bounded algorithms to compute the treewidth, pathwidth, minimum fill-in and minimum interval graph completion of a given d-trapezoid graph, and hence also of any given cocomparability graph of dimension at most d. The algorithms do not require an intersection model as part of the input.*

Proof. The listing algorithm is used up to at most $(2n-3)^d + 1$ minimal separators, thus its running time is $O(n^{d+5})$. Suppose the input graph is a cocomparability graph and has at most $(2n-3)^d$ minimal separators. Then the corresponding algorithm of Theorem 8.1 is applied to G. This algorithm has running time $O(n^5R + n^3R^3)$. Since the input graph has at most $(2n-3)^d$ minimal separators, the algorithm works in time $O(n^{3d+3})$. □

Finally we would like to emphasize the relation of our results to the dimension of partially ordered sets. The dimension is one of the most carefully studied parameters of a partially ordered set [21]. Yannakakis showed that determining whether a partially ordered set has dimension at most d is NP-complete for any fixed $d \geqslant 3$ [23]. Many problems have been shown to be efficiently solvable on partially ordered sets of dimension two. In fact the initial motivation for our research was that no NP-complete partially ordered set problem had been known, that is solvable by a polynomial time algorithm for partially ordered sets of some fixed dimension greater than two.

We have shown that the problems TREEWIDTH, PATHWIDTH, MINIMUM FILL-IN and INTERVAL GRAPH COMPLETION are problems, for which restricted dimension helps. On the other hand, the problems are NP-complete (for cobipartite graphs and hence) for cocomparability graphs, when the dimension is unbounded [1, 22].

References

[1] S. Arnborg, D.G. Corneil and A. Proskurowski, Complexity of finding embeddings in a k-tree, *SIAM J. Algeb. Discrete Methods* **8** (1987) 277–284.

[2] H. Bodlaender, A tourist guide through treewidth, *Acta Cybernet.* **11** (1993) 1–23.

[3] H. Bodlaender, T. Kloks and D. Kratsch, Treewidth and pathwidth of permutation graphs, *SIAM J. Discrete Math.* **8** (1995) 606–616.

[4] H. Bodlaender, T. Kloks, D. Kratsch and H. Müller, Treewidth and minimum fill-in on d-trapezoid graphs, manuscript, 1996.

[5] D.G. Corneil, S. Olariu and L. Stewart, Asteroidal triple-free graphs, in: *Proc. 19th Internat. Workshop on Graph-Theoretic Concepts in Computer Science*, Lecture Notes in Computer Science, Vol. 790 (Springer, Berlin, 1994) 211–224.

[6] D.G. Corneil, S. Olariu and L. Stewart, A linear time algorithm to compute dominating pairs in asteroidal triple-free graphs, in: *Proc. 22nd Internat. Coll. on Automata, Languages and Programming*, Lecture Notes in Computer Science, Vol. 944 (Springer, Berlin, 1995) 292–302.

[7] G.A. Dirac, On rigid circuit graphs, *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg* **25** (1961) 71–76.

[8] M.C. Golumbic, *Algorithmic Graph Theory and Perfect Graphs* (Academic Press, New York, 1980).
[9] M. Habib and R.H. Möhring, Treewidth of cocomparability graphs and a new order-theoretic parameter, *Order* **11** (1994) 47–60.
[10] T. Kloks, *Treewidth – Computations and Approximations*, Lecture Notes in Computer Science, Vol. 842 (Springer, Berlin, 1994).
[11] T. Kloks and D. Kratsch, Finding all minimal separators of a graph, in: *Proc. 11th Ann. Symp. on Theoretical Aspects of Computer Science*, Lecture Notes in Computer Science, Vol. 775 (Springer, Berlin, 1994) 759–768; also to appear in *SIAM J. Comput.*
[12] T. Kloks, D. Kratsch and H. Müller, Approximating the bandwidth of AT-free graphs, in: *Proc. 3rd European Symp. on Algorithms*, Lecture Notes in Computer Science, Vol. 979 (Springer, Berlin, 1995), 434–447.
[13] D. Kratsch, The structure of graphs and the design of efficient algorithms, Habilitation, Thesis, Friedrich-Schiller-Universität Jena, 1996.
[14] J. van Leeuwen, Graph algorithms, in: *Handbook of Theoretical Computer Science, A: Algorithms and Complexity Theory* (North-Holland, Amsterdam, 1990) 527–631.
[15] R.H. Möhring, Triangulating graphs without asteroidal triples, *Discrete Appl. Math.* **64** (1996) 281–287.
[16] T. Ohtsuki, A fast algorithm for finding an optimal ordering for vertex elimination on a graph, *SIAM J. Comput.* **5** (1976) 133–145.
[17] A. Parra and P. Scheffler, How to use the minimal separators of a graph for its chordal triangulation, in: *Proc. 22nd Internat. Coll. on Automata, Languages and Programming*, Lecture Notes in Computer Science, Vol. 944 (Springer, Berlin, 1995) 123–134.
[18] D.J. Rose, Triangulated graphs and the elimination process, *J. Math. Anal. Appl.* **32** (1970) 597–609.
[19] D.J. Rose, R.E. Tarjan and G.S. Lueker, Algorithmic aspects of vertex elimination on graphs, *SIAM J. Comput.* **5** (1976) 266–283.
[20] J. Spinrad, On comparability and permutation graphs, *SIAM J. Comput.* **14** (1985) 658–670.
[21] W.T. Trotter, *Combinatorics and Partially Ordered Sets: Dimension Theory* (John Hopkins University Press, Baltimore, MD, 1992).
[22] M. Yannakakis, Computing the minimum fill-in is NP-complete, *SIAM J. Algebraic Discrete Methods* **2** (1981) 77–79.
[23] M. Yannakakis, The complexity of the partial order dimension problem, *SIAM J. Algebraic Discrete Methods* **3** (1982) 351–358.