Treewidth and minimum fill-in on d-trapezoid graphs

Hans L. Bodlaender
Department of Computer Science, Utrecht University
P.O. Box 80.089, 3508 TB Utrecht, The Netherlands.

Ton Kloks
Department of Mathematics and Computing Science
Eindhoven University of Technology
P.O.Box 513, 5600 MB Eindhoven, The Netherlands.

Dieter Kratsch* Haiko Müller
Friedrich-Schiller-Universität, Fakultät für Mathematik und Informatik
07740 Jena, Germany.

Abstract

We show that the minimum fill-in and the minimum interval graph completion of a d-trapezoid graph can be computed in time $O(n^d)$. We also show that the treewidth and the pathwidth of a d-trapezoid graph can be computed by an $O(n\, tw(G)^{d-1})$ time algorithm. For both algorithms, d is supposed to be a fixed positive integer and it is required that a suitable intersection model of the given d-trapezoid graph is part of the input.

As a consequence, the minimum fill-in and the minimum interval graph completion as well as the treewidth and the pathwidth of a given trapezoid graph (or permutation graph) can be computed in time $O(n^2)$, even if no intersection model is part of the input.

1 Introduction

The notions of treewidth and pathwidth have come to play a central role in several recent investigations in algorithmic graph theory, due to several applications inside and outside graph theory. One reason for this interest is that many graph problems, including many well known and important problems, become polynomial time, and usually even linear time solvable (and become member of NC), when restricted to a class of graphs with bounded tree- or pathwidth [1, 3, 4, 5, 7, 27]. In general, such algorithms need to have a tree-decomposition or path-decomposition of suitable width given together with the input graph. Hence, an important problem is to find tree-decompositions (or path-decompositions) of minimum width. When the desired width of the tree-decomposition is bounded by a constant, then this problem can be solved in linear time [6]. However, the constant factor of this algorithm is exponential in the treewidth (of yes-instances), which limits its practicality. Thus, it is interesting for special classes of graphs to find algorithms, which are also polynomial in the treewidth.

*E-mail: `kratsch@minet.uni-jena.de`

A related graph problem is the minimum fill-in problem. In this problem, we want to add as little edges as possible to a given graph to make it chordal. The importance of this problem lies mainly in the fact that it is equivalent to finding an order of Gaussian elimination steps of a (usually sparse) symmetric matrix, minimizing the number of generated non-zero entries [32].

The problem `Given a graph G and a positive integer k, decide whether the treewidth (resp. pathwidth) of G is at most k' remains NP-complete on cobipartite graphs [2] and on bipartite graphs [21]. (The reader is refered to the preliminaries section for definitions.) For some special classes of graphs, it has been shown that the treewidth can be computed in polynomial time, as e.g. cographs [9], circular-arc graphs [35], chordal bipartite graphs [23], permutation graphs [10], circle graphs [19], cocomparability graphs of bounded dimension [25], cointerval graphs [16] and d-trapezoid graphs [31]. The algorithm for d-trapezoid graphs assumes that a d-trapezoid intersection model is part of the input.

In this paper we present an $O(n\, tw(G)^{d-1})$ algorithm finding optimal tree- and path-decompositions for d-trapezoid graphs, d a fixed positive integer, where a d-trapezoid diagram is part of the input. Note that the best timebound known up to now is $O(\max(n^{2.376\,d}, n^{2d+2}))$ [31].

On the other hand, there is an $O(n^5\, R + n^3\, R^3)$ algorithm computing the treewidth and pathwidth of a given asteroidal triple-free graph on n vertices with R minimal separators [25]. This implies that the treewidth and the pathwidth of a d-trapezoid graph can be computed by an $O(n^{3d+3})$ algorithm that does not require an intersection model as part of the input.

The knowledge on the algorithmic complexity of the minimum fill-in problem when restricted to special graph classes is relatively small compared to that of the treewidth and pathwidth problem, although this problem has very important applications in Gaussian matrix multiplication. Indeed, due to the lack of efficient algorithms for finding an optimal solution, in practice one usually has to work with certain heuristics for `approximating' a minimum fill-in.

The problem `Given a graph G and a positive integer k, decide whether there is a fill-in of G with at most k edges' remains NP-complete on cobipartite graphs [38] and on bipartite graphs [36]. The only known graph classes for which the minimum fill-in can be computed by a polynomial time algorithm were for almost ten years the relatively small classes of cographs [13] and bipartite permutation graphs [34]. Now polynomial time algorithms for chordal bipartite graphs [20], multilolerance graphs [30] as well as circle and circular-arc graphs [26] are available.

We claimed in [22] that the scanline approach used for designing an efficient treewidth algorithm for permutation graphs in [10] can as well be used for the minimum fill-in problem on permutation graphs and cocomparability graphs of dimension at most d (with an intersection model as part of the input) leading to $O(n^2)$ and $O(n^d)$ algorithms, respectively. By now the best known algorithm for the minimum fill-in problem on d-trapezoid graphs with given intersection model has running time $O(\max(n^{2.376\,d}, n^{2d+2}))$ [31].

We are going to justify the above mentioned claim by presenting an $O(n^d)$ algorithm computing the minimum fill-in and the minimum interval graph completion for a given d-trapezoid graph in time $O(n^d)$, d a fixed positive integer, where a d-trapezoid diagram is supposed to be part of the input.

It is worth to mention that our algorithms are indeed natural extensions of the treewidth algorithm for permutation graphs in [10]. (Note that each permutation graph is a 2-trapezoid graph.) Finally, we emphasize that for trapezoid graphs we indeed obtain fast algorithms, namely an $O(n\, tw(G))$ algorithm computing the treewidth and pathwidth, and an $O(n^2)$ algorithm computing the minimum fill-in and the minimum interval graph completion. Here it is used that the known $O(n^2)$ recognition algorithm for trapezoid graphs, given in [28], also computes a trapezoid diagram, if the input is indeed a trapezoid graph.

2 Preliminaries

2.1 Preliminaries on treewidth, pathwidth and minimum fill-in

The concept of a chordal graph is fundamental for the treewidth and the minimum fill-in of graphs.

Definition 2.1 *A graph is* chordal *if it has no induced chordless cycle of length at least four.*

Chordal graphs (also called triangulated graphs) form a subclass of the perfect graphs. For detailed information on classes of perfect graphs the reader is refered to [12, 18].

There are different ways to define the treewidth of a graph. The original definition by Robertson and Seymour uses the concept of a *tree-decomposition*. For more information on tree-decompositions the reader is refered to the survey paper [7]. In this paper 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., [21]).

Definition 2.2 *A* triangulation *of a graph G is a graph H with the same vertex set as G, such that G is a subgraph of H and H is chordal.*

We denote the maximum cardinality of a clique in a graph G by $\omega(G)$.

Definition 2.3 *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 pathwidth can be defined in terms of triangulations of a special kind.

Definition 2.4 *An* interval graph *is a graph of which the vertices can be put into one-to-one correspondence with closed intervals on the real line, such that two vertices are adjacent if and only if the corresponding intervals have a nonempty intersection.*

Notice that the interval graphs form a proper subclass of the chordal graphs [18].

Definition 2.5 *The* pathwidth *of a graph G, denoted by $pw(G)$, is the smallest value of $\omega(H) - 1$ where the minimum is taken over all triangulations H of G for which H is an interval graph.*

Definition 2.6 *A* path-decomposition *of a graph $G = (V, E)$ is a sequence of subsets of V, $(X_1, \ldots, X_r)$, such that $\bigcup_{1 \leq i \leq r} X_i = V$, for all $\{v, w\} \in E$, there is an i, $1 \leq i \leq r$, $v, w \in X_i$, and for all $v \in V$, there are l_v, r_v, such that for all integers j, $1 \leq l_v \leq j \leq r_v \leq r \Leftrightarrow v \in X_j$. The* width *of path-decomposition $(X_1, \ldots, X_r)$ is $\max_{1 \leq i \leq r} |X_i| - 1$.*

The following lemma shows the equivalence of the above definition of pathwidth and the original one in terms of path-decompositions by Robertson and Seymour. For a proof see for example [8, Theorem 29] and [21, Lemma 2.2.8].

Lemma 2.1 *A graph G has a path-decomposition of width at most k if and only if there is a triangulation of G into an interval graph H such that $\omega(H) \leq k + 1$.*

The following characterization of interval graphs is due to Gilmore and Hoffman [17].

Lemma 2.2 *G is an interval graph if and only if the maximal cliques of G can be ordered so that for every vertex the maximal cliques containing it occur consecutively.*

Such an ordering of the maximal cliques is said to be a *consecutive clique arrangement* (abbr. CCA) of G. By assigning to each vertex $v \in V$ the interval $[\min\{i \mid v \in X_i\}, \max\{i \mid v \in X_i\}]$, we directly get the following result.

Lemma 2.3 *Let $(X_1, \ldots, X_r)$ be a path-decomposition of $G = (V, E)$. The graph $H = (V, F)$, obtained by making each set X_i, $1 \leq i \leq r$ a clique, (i.e., for all $v, w \in V, v \neq w$: $\{v, w\} \in F \Leftrightarrow \exists i : v, w \in X_i$), is an interval graph that contains G as a subgraph.*

The decision problems TREEWIDTH and PATHWIDTH are NP-complete [2]. However, for *constant* k, graphs with treewidth at most k are recognizable in $O(n)$ time [6]. The large constants involved in these algorithms make them usually not very practical. It is therefore of importance to find fully polynomial algorithms for treewidth and pathwidth for special classes of graphs which are as large as possible. The aim of this paper is to present fast algorithms for computing treewidth and pathwidth as well as the minimum fill-in and the minimum interval graph completion on a relatively large parameterized class of graphs.

Definition 2.7 *A* fill-in *of the graph $G = (V, E)$ is a set F of edges of the complement of 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.*

Hence solving the minimum fill-in problem on a graph G is equivalent to finding a triangulation H of G that has smallest number of edges among all triangulations of G.

Definition 2.8 *An* interval graph completion *of the graph $G = (V, E)$ is a set F of edges of the complement of 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.*

2.2 Preliminaries on minimal separators and triangulations

One of the main reasons why there exist fast algorithms for many problems when restricted to graphs with bounded treewidth, is the existence of vertex separators of bounded size. For designing efficient treewidth algorithms on special graph classes that do not have bounded treewidth, vertex separators of bounded size have been replaced by minimal separators (see, e.g., [10, 21]).

Definition 2.9 *Let $G = (V, E)$ be a graph. A subset $S \subset V$ is an a, b-separator for nonadjacent vertices a and b, if the removal of S separates a and b in distinct connected components. If no proper subset of the a, b-separator S is itself an a, b-separator then S is a minimal a, b-separator. A* minimal separator *S is a subset $S \subset V$ such that S is a minimal a, b-separator for some nonadjacent vertices a and b.*

The following lemmas must have been rediscovered many times (see, e.g., [18]).

Lemma 2.4 *Let S be a minimal a, b-separator of $G = (V, E)$, and let C_a and C_b be the connected components of $G[V \setminus S]$, containing a and b respectively. Then every vertex of S has a neighbor in C_a and a neighbor in C_b.*

Lemma 2.5 *$S \subset V$ is a minimal separator of the graph $G = (V, E)$ if and only if $G[V \setminus S]$ has at least two components for which every vertex of S has a neighbor in the component.*

Using the characterization in Lemma 2.2, one can easily identify the minimal separators of an interval graph which has been shown in [25].

Lemma 2.6 *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 $A_i \cap A_{i+1}$, $i \in \{1, 2, \ldots, q-1\}$.*

For designing efficient treewidth algorithms on special graph classes, the restriction to certain types of triangulations has been used by different authors (see, e.g., [10, 21, 25, 31]).

Definition 2.10 *A triangulation H of a graph G is a* minimal triangulation *of G if no proper subgraph of H is a triangulation of G.*

Minimal triangulations have already been studied in [33]. Among others, the authors give the following characterization of minimal triangulations.

Theorem 2.1 *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.*

The following theorem of Möhring, given in [29], is important for us.

Theorem 2.2 *Any minimal triangulation of an asteroidal triple-free graph is an interval graph. Hence $pw(G) = tw(G)$ and $mfi(G) = mic(G)$ for each asteroidal triple-free graph.*

(See [14] for more informations on asteroidal triple-free graphs.) The d-trapezoid graphs are a subclass of the asteroidal triple-free graphs for any fixed d. Hence 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.

The following characterization of minimal triangulations has been shown in [24].

Theorem 2.3 *Let H be a triangulation of $G = (V, E)$ and let $\Delta(H)$ be the set of minimal separators of H. Then H is a minimal triangulation of the graph G if and only if the following three 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 the vertex set of a connected component of $H[V \setminus S]$, then C induces also a connected component in $G[V \setminus S]$.*
3. *$H = G_{\Delta(H)}$, which is the graph obtained from G by adding edges between all pairs of vertices contained in the same set S for every $S \in \Delta(H)$.*

2.3 Preliminaries on blocks

Blocks and realizations of blocks are useful concepts for designing treewidth and minimum fill-in algorithms that are based on minimal separators.

Definition 2.11 *A* 1-block *of G 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 making a clique of S is called the* realization *of B and is denoted by $R(S, C)$.*

The treewidth of a graph can be computed from the treewidth of the realizations of all 1-blocks of the graph (see also [25]).

Lemma 2.7 *Let $G = (V, E)$ be a graph which is not complete. Then*

$$tw(G) = \min_S \max_C \ tw(R(S, C))$$

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

A similar lemma can be obtained for the minimum fill-in of a graph (see [25]).

Lemma 2.8 *Let $G = (V, E)$ be a graph which is not complete. Then*

$$mfi(G) = \min_S \left(fill(S) + \sum_C mfi(R(S, C)) \right)$$

where the minimum is taken over all minimal separators S of G and the summation is over all connected components C of $G[V \setminus S]$, where $fill(S) = \binom{|S|}{2} - |E(G[S])|$ denotes the number of edges added to $G[S]$ for making S a clique.

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.

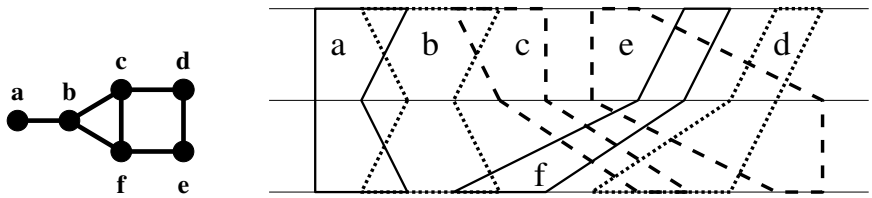

Figure 1: 3-trapezoid graph G and 3-trapezoid model $\mathcal{D}(G)$

3 The *d*-trapezoid graphs

In this section we define d-trapezoid graphs and d-trapezoid diagrams. Our definition of d-trapezoid graphs is a mixture of the definitions in [15, 31], chosen such that it generalizes the definition of cocomparability graphs of dimension d, given in [25], and such that d-trapezoid graphs are exactly the cocomparability graphs of partially ordered sets of interval dimension at most d.

Definition 3.1 *Let d be a fixed positive integer, $d \geq 1$. Then a d-*trapezoid diagram *of a graph $G = (V, E)$ assigns to each vertex v of G a collection of d intervals*

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

such that for each $i \in \{1, 2, \ldots d\}$ and any pair of different 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 $i \in \{2, 3, \ldots, d\}$ such that $l_v^{i-1} < r_v^{i-1} < l_w^{i-1} < r_w^{i-1}$ and $l_w^i < r_w^i < l_v^i < r_v^i$.

We use the following visualizing of a d-trapezoid diagram. 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. (See Fig. 1 for an example.)

Definition 3.2 *A graph G is a d-trapezoid graph if it has a d-trapezoid diagram.*

The following theorem is a consequence of Definition 3.1 (see [15, 31]).

Theorem 3.1 *The d-trapezoid graphs are exactly the cocomparability graphs of partially ordered sets of interval dimension at most d.*

Unfortunately, the problem `Given a partially ordered set P, decide whether the interval dimension of P is at most d' is NP-complete for any fixed $d \geq 3$ [37]. Hence for fixed $d \geq 3$, computing a d-trapezoid diagram of the given graph, if it is indeed a d-trapezoid graph, means solving an NP-complete problem. Moreover, at the present not even reasonable approximation algorithms for the interval dimension of a partially ordered set are known. Thus assuming that a d-trapezoid diagram is part of the input is a very strong assumption for $d \geq 3$.

The situation is much better for $d \in \{1, 2\}$. The 1-trapezoid graphs are exactly the interval graphs and the 2-trapezoid graphs are exactly the trapezoid graphs by Theorem 3.1. There is a linear time recognition algorithm for interval graphs computing an interval model if the input graph is an interval graph [11]. There is an $O(n^2)$ recognition algorithm for trapezoid graphs computing a trapezoid model if the input graph is a trapezoid graph [28].

Moreover, Theorem 3.1 shows that for any fixed d the d-trapezoid graphs form a subclass of the cocomparability graphs. Hence the class of d-trapezoid graphs is a subclass of the asteroidal triple-free graphs for any fixed d. Consequently, by Theorem 2.2, the treewidth and pathwidth of a d-trapezoid graph coincide, and the minimum fill-in and the minimum interval graph completion of a d-trapezoid graph coincide.

The concept of a scanline is crucial to various efficient treewidth algorithms developed in the last years [10, 19, 21, 31].

Definition 3.3 *A* scanline *in a d-trapezoid diagram is a polyline determined by one* endpoint *on a coordinate $s^i \in \{0.5, 1.5, \ldots, 2n + 0.5\}$, $i \in \{1, 2, \ldots, d\}$, on each of the d horizontal lines of the d-trapezoid diagram, obtained by drawing line segments between the endpoints on consecutive horizontal lines.*

Definition 3.4 *Let s be a scanline of a d-trapezoid diagram of a graph G. Then $S(s)$ is the set of those vertices v for which the scanline s has a nonempty intersection with Q_v.*

The following results can easily be obtained in the same fashion as the corresponding result for permutation graphs given in [10].

Lemma 3.1 *Let G be a d-trapezoid graph with d-trapezoid diagram $\mathcal{D}(G)$. For every minimal x, y-separator S of G there is a scanline s in $\mathcal{D}(G)$, which is between the d-trapezoids Q_x and Q_y, such that $S = S(s)$.*

Corollary 3.1 *The number of minimal separators of a d-trapezoid graph G on $n \geq 2$ vertices is at most $(2n - 3)^d$.*

4 Realizations

The next two sections give most of the technical results of the paper. In particular they contain all theorems and lemmas for verifying the correctness of our algorithms. Throughout this section we assume that $G = (V, E)$ is a d-trapezoid graph with a fixed d-trapezoid diagram $\mathcal{D}(G)$.

Definition 4.1 *Let s_1 and s_2 be two different scanlines and Q_v a d-trapezoid in a d-trapezoid diagram $\mathcal{D}(G)$. Then s_1 is* left of *s_2 if $s_1^i \leq s_2^i$ for all $i \in \{1, 2, \ldots d\}$. Furthermore, the d-trapezoid Q_v is* between *the scanlines s_1 and s_2 if $s_1^i \leq l_v^i < r_v^i \leq s_2^i$ for all $i \in \{1, 2, \ldots, d\}$.*

Definition 4.2 *Let s_1 and s_2 be two different scanlines such that s_1 is left of s_2. A* candidate component *$C = \mathcal{C}(s_1, s_2)$ is a subgraph of G induced by the set of those vertices v of G for which the corresponding d-trapezoid Q_v in the diagram $\mathcal{D}(G)$ has one of the following properties:*

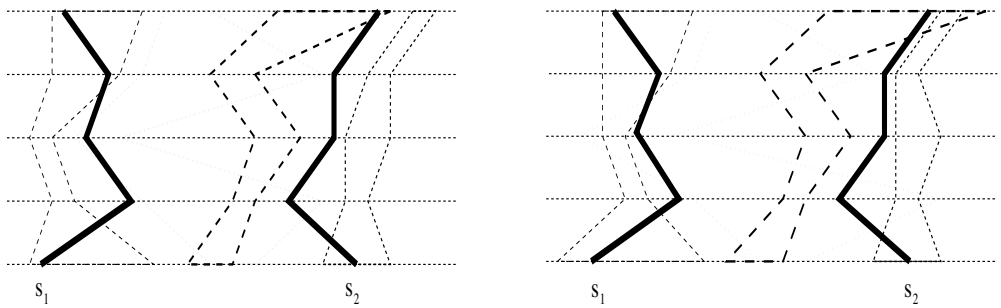

Figure 2: d-trapezoid diagram of a candidate component C and its realization $R(C)$.

- *Q_v is between the scanlines s_1 and s_2.*
- *Q_v has nonempty intersection with at least one of the two scanlines.*

We identify the candidate component $C = \mathcal{C}(s_1, s_2)$ with the d-trapezoid diagram obtained from $\mathcal{D}(G)$ by removing all d-trapezoids for which the corresponding vertex does not belong to C and by adding the two scanlines s_1 and s_2.

Definition 4.3 *Let $C = \mathcal{C}(s_1, s_2)$ be a candidate component. We define the* realization *$R(C)$ as the graph obtained from C, by adding all edges between vertices of $S(s_1)$ and between vertices of $S(s_2)$ (i.e. the two sets $S(s_1)$ and $S(s_2)$ are cliques in $R(C)$).*

Lemma 4.1 *If $C = \mathcal{C}(s_1, s_2)$ is a candidate component of a d-trapezoid graph G with d-trapezoid diagram $\mathcal{D}(G)$ then the realization $R(C)$ is a d-trapezoid graph.*

Proof. Consider $\mathcal{D}(G)$. For every vertex $v \in S(s_1)$ there is a horizontal line D_i such that $l_v^i < s_1^i$, and for every pair of nonadjacent vertices $v, w \in S(s_1)$ there is a horizontal line D_i such that $l_v^i < s_1^i$ and $l_w^i < s_1^i$.

We construct a diagram of the realization $R(C)$ from the diagram of C, by reordering for any horizontal line D_i, $i \in \{1, 2, \ldots, d\}$, all points of d-trapezoids that are left of s_1^i and right of s_2^i, respectively. The diagram remains unchanged in the area between the two scanlines s_1 and s_2.

For each $i \in \{1, \ldots, d\}$ we replace the points $l_v^i, r_w^i < s_1^i$ by a collection of consecutive new points $l_v'^i$ positioned totally left of a collection of consecutive new points $r_w'^i$. Inside these collections we choose an arbitrary order. This procedure transforms disjoint intervals $[l_v^i, r_v^i]$ and $[l_w^i, r_w^i]$ with $l_v^i, l_w^i < s_1^i$ into intervals $[l_v'^i, r_v'^i]$ and $[l_w'^i, r_w'^i]$ with the point $\max\{l_u'^i : l_u^i < s_1^i\}$ in common. This implies that $S(s_1)$ is a clique in the transformed diagram.

Analogously we transform the points right of s_2. For every line D_i we reorder the points $l_v^i, r_w^i > s_2^i$ to obtain a collection of new left points $l_v'^i$ left of a collection of new right points $r_v'^i$. Similarly $S(s_2)$ is a clique in the transformed diagram. Hence this construction gives a d-trapezoid diagram for the realization $R(C)$, thus $R(C)$ is a d-trapezoid graph. (See Fig. 2 for an illustration of the described construction.) $\square$

Let $C = \mathcal{C}(s_1, s_2)$ be a candidate component of a d-trapezoid graph G. Consider the d-trapezoid diagram of $R(C)$, obtained from the diagram of C by the procedure,

described in the proof of Lemma 4.1. Suppose $R(C)$ has a minimal separator S. By Lemma 3.1 there is a scanline s in the d-trapezoid diagram of $R(C)$ such that $S = S(s)$.

Definition 4.4 *Let $C = \mathcal{C}(s_1, s_2)$ be a candidate component with realization $R(C)$. A scanline t in the d-trapezoid diagram is* nice *for $C = \mathcal{C}(s_1, s_2)$ if for all endpoints t^i, $i \in \{1, 2, \ldots, d\}$:*

$$s_1^i \leq t^i \leq s_2^i.$$

Lemma 4.2 *Let $C = \mathcal{C}(s_1, s_2)$ be a candidate component of a d-trapezoid graph G, $R(C)$ the realization of C and S a minimal a, b-separator of $R(C)$. Then there is a nice scanline s^* for $C = \mathcal{C}(s_1, s_2)$ such that $S = S(s^*)$.*

Proof. Consider the d-trapezoid diagram of $R(C)$ with the scanlines s_1, s_2 obtained by the construction of Lemma 4.1. Let S be a minimal a, b-separator of $R(C)$. By Lemma 3.1 there is a scanline s in the d-trapezoid diagram of $R(C)$ such that $S = S(s)$. Moreover, the scanline s is between the d-trapezoids Q_a and Q_b in the diagram of $R(C)$. Without loss of generality assume that Q_a is left of Q_b. Hence $l_a^i < r_a^i < s^i < l_b^i < r_b^i$ for each horizontal line D_i, $i \in \{1, 2, \ldots d\}$.

Suppose s is not nice for $C = \mathcal{C}(s_1, s_2)$. Hence either $s^i < s_1^i$ for some $i \in \{1, 2, \ldots, d\}$ or $s_2^j < s^j$ for some $j \in \{1, 2, \ldots, d\}$. Without loss of generality assume $s^i < s_1^i$ for some $i \in \{1, 2, \ldots, d\}$. Then $a \in S(s_1)$ implies $b \notin S(s_1)$ since a and b are not adjacent in $R(C)$. Let s^* be the scanline for which on each horizontal line its endpoint coincides with the one of s or s_1, more precisely $(s^*)^i := \max(s^i, s_1^i)$ for each $i \in \{1, 2, \ldots, d\}$ (see Fig. 3). Let $S^* = S(s^*)$. Suppose that $S \neq S^*$.

Case 1: $S \setminus S^* \neq \emptyset$.
Let $p \in S \setminus S^*$. By Lemma 2.4 every vertex of S has a neighbor in the component of $R(C) \setminus S$ that contains b. The d-trapezoid corresponding to such a neighbor q of p must be right of the scanline s. Hence Q_q intersects s_1, thus q and a are adjacent in $R(C)$ and q is in the component of $R(C) \setminus S$ containing a, contradicting the choice of q.

Case 2: $S^* \setminus S \neq \emptyset$.
Let $p \in S^* \setminus S$. Hence there is an $i \in \{1, 2, \ldots, d\}$ such that $s^i < l_p^i < s_1^i = (s^*)^i$ for the d-trapezoid Q_p. Hence $p \notin S$ implies that s is between Q_a and Q_p. However a and p are adjacent in $R(C)$ since Q_a and Q_p both intersect s_1, a contradiction.

Consequently s^* is a nice scanline with $S(s^*) = S$ unless s^* intersects s_2. In this case the analogous construction applied to s^* and s_2 gives the wanted nice scanline. □

Lemma 4.2 shows that all d-trapezoids intersecting the scanline s correspond to vertices of the candidate component $C = \mathcal{C}(s_1, s_2)$. Hence a nice scanline s generating a minimal separator S of a realization $R(C)$ can be chosen in the region of the d-trapezoid diagram of the candidate component C between s_1 and s_2 since in that region the d-trapezoid diagram of $R(C)$ is exactly the same as the original diagram $\mathcal{D}(G)$. (See the proof of Lemma 4.1.)

Definition 4.5 *$s_0, s_1, s_2, \ldots, s_{r-1}, s_r$ is a* nice sequence *of scanlines in a d-trapezoid diagram of a graph G if s_i is left of s_{i+1} for each $i \in \{0, 1, \ldots, r-1\}$. We denote by*

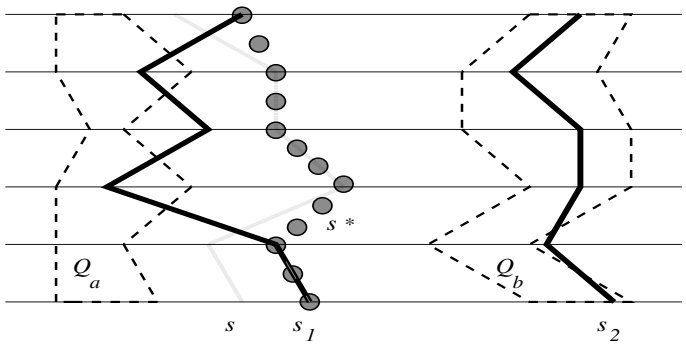

Figure 3: d-trapezoid diagram of $R(C)$ and the nice scanline s^*, that is indicated by the heavy dots.

$H = G_{s_0,s_1,\ldots,s_r}$ the graph obtained from G by adding an edge between those pairs of nonadjacent vertices u and v of G for which there is a scanline s_i, $i \in \{0, 1, \ldots, r\}$, that intersects both d-trapezoids Q_u and Q_v.

Consider a d-trapezoid diagram of a graph G. We denote by s_{L} the scanline which lies totally to the left of all d-trapezoids and by s_{R} the scanline which lies totally to the right of all d-trapezoids. We are now ready to prove a characterization of minimal triangulations of d-trapezoid graphs via scanlines.

Theorem 4.1 *Let $G = (V, E)$ be a d-trapezoid graph with d-trapezoid diagram $\mathcal{D}(G)$. Let H be any minimal triangulation of G and let $\Delta(H)$ be the set of all minimal separators of H. Then there is a nice sequence of scanlines $s_{\mathrm{L}} = s_0, s_1, s_2, \ldots, s_{r-1}, s_r = s_{\mathrm{R}}$ in $\mathcal{D}(G)$ such that*

(i) $\Delta(H) = \{S(s_1), S(s_2), \ldots, S(s_{r-1})\}$,

(ii) $H = G_{s_0,s_1,\ldots,s_r}$,

(iii) $\mathcal{C}(s_0, s_1), \mathcal{C}(s_1, s_2), \ldots, \mathcal{C}(s_{r-1}, s_r)$ *is a consecutive clique arrangement of H.*

Proof. We prove the following claim by induction on the number of vertices.

Let H be any minimal triangulation of a d-trapezoid graph G with d-trapezoid diagram $\mathcal{D}(G)$. Let s', s'' be two scanlines in $\mathcal{D}(G)$ such that s' is left of s'' and G is the realization of $\mathcal{C}(s', s'')$. Then there is a nice sequence of scanlines $s' = s_0, s_1, s_2, \ldots, s_{r-1}, s_r = s''$ in $\mathcal{D}(G)$ such that

(i) $\Delta(H) = \{S(s_1), S(s_2), \ldots, S(s_{r-1})\}$,

(ii) $H = G_{s_0,s_1,\ldots,s_r}$,

(iii) $\mathcal{C}(s_0, s_1), \mathcal{C}(s_1, s_2), \ldots, \mathcal{C}(s_{r-1}, s_r)$ is a consecutive clique arrangement of H.

Notice that the claim immediately implies the theorem by taking $s' = s_{\mathrm{L}}$ and $s'' = s_{\mathrm{R}}$.

Let H be any minimal triangulation of the d-trapezoid graph G and $\Delta(H)$ be the set of all minimal separators of H. The graph H is an interval graph by Theorem 2.2. If H is a complete graph then $G = H$ by Theorem 2.3 and the assertion holds with the nice sequence s', s''. Hence we may assume that H is not complete.

Let S be any minimal separator of H. Suppose S is a minimal a, b-separator of H. Then S is a minimal a, b-separator of G. Since G is the realization of $\mathcal{C}(s', s'')$, by Lemma 4.2, there is a scanline s in $\mathcal{D}(G)$ such that s is nice for $\mathcal{C}(s', s'')$, $S = S(s)$ and s is between Q_a and Q_b. Consequently the sequence s', s, s'' is nice.

Consider the candidate components $C_1 = \mathcal{C}(s', s)$ and $C_2 = \mathcal{C}(s, s'')$ and their realizations $R_1 = R(C_1)$ and $R_2 = R(C_2)$. Since s is between Q_a and Q_b we get that C_1 and C_2 are both smaller than G. R_j, $j \in \{1, 2\}$, is obtained from C_j by making S into a clique. $H_j = H[C_j]$ is an interval graph. Hence H_j is a triangulation of R_j since $H = G_{\Delta(H)}$ by Theorem 2.3 and $S \in \Delta(H)$.

We claim that H_j is a minimal triangulation of R_j, $j \in \{1, 2\}$. Suppose not and assume w.l.o.g. that H_1' is a proper subgraph of H_1 and a minimal triangulation of R_1. Let H' be the graph that has the same vertex set as H and edge set $E(H_1') \cup E(H_2)$, thus H' is a proper spanning subgraph of H. Furthermore S is a clique of H' and $S = V(H_1') \cap V(H_2)$.

Suppose Z would be a chordless cycle in H' having length at least four. Since H_1' and H_2 are chordal there is a vertex $u \in V(H_1') \setminus S$ and a vertex $v \in V(H_2) \setminus S$ in Z. Since there is no edge between a vertex of $V(H_1') \setminus S$ and a vertex of $V(H_2) \setminus S$ in H', the cycle Z must contain at least two vertices of S, that are not consecutive vertices in Z. Thus the cycle Z has a chord connecting these two vertices of S, a contradiction. Consequently, H' is a chordal graph and therefore a triangulation of G that is a proper subgraph of the minimal triangulation H, a contradiction.

Let $s_0 := s'$, $s_1 := s$ and $s_2 := s''$. As both R_1 and R_2 have fewer vertices as G, by induction it follows that for each $j \in \{1, 2\}$, there is a nice sequence of scanlines $s_{j-1}, s_{j1}, \ldots, s_{jq_j}, s_j$ in $\mathcal{D}(R_j)$ such that

(a) $\Delta(H_j) = \{S(s_{j1}), \ldots, S(s_{jq_j})\}$,

(b) $H_j = R_{j_{s_{j-1}, s_{j1}, \ldots, s_{jq_j}, s_j}}$ and

(c) $\mathcal{C}(s_{j-1}, s_{j1}), \mathcal{C}(s_{j1}, s_{j2}), \ldots, \mathcal{C}(s_{jq_j}, s_j)$ is a CCA of H_j.

Hence $s', s_{11}, \ldots, s_{1q_1}, s, s_{21}, \ldots, s_{2q_2}, s''$ is a nice sequence of scanlines in $\mathcal{D}(G)$ since the sequence $s_{j-1}, s_{j1}, \ldots, s_{jq_j}, s_j$ in $\mathcal{D}(R_j)$ is nice for $j \in \{1, 2\}$ and the construction of $\mathcal{D}(R_j)$ from $\mathcal{D}(G)$ does not change scanlines, i.e., any scanline s of $\mathcal{D}(R_j)$ has exactly the same endpoint s^i on the horizontal line D_i, for all $i \in \{1, 2, \ldots, d\}$, in both diagrams $\mathcal{D}(R_j)$ and $\mathcal{D}(G)$.

Using induction, we can see that $\mathcal{C}(s', s_{11}), \mathcal{C}(s_{11}, s_{12}), \ldots, \mathcal{C}(s_{1q_1}, s)$ is a CCA of H_1 and $\mathcal{C}(s, s_{21}), \mathcal{C}(s_{21}, s_{22}), \ldots, \mathcal{C}(s_{2q_2}, s'')$ is a CCA of H_2. Moreover, S is a clique and a minimal separator of the interval graph H and $S = V(H_1) \cap V(H_2)$. Since $V(H_1)$ is the vertex set of $\mathcal{C}(s', s)$ and $V(H_2)$ is the vertex set of $\mathcal{C}(s, s'')$ we obtain that each component of $G[V \setminus S]$ is either completely contained in H_1 or it is completely contained in H_2. Since H is a minimal triangulation of G, Theorem 2.3 implies that S is a minimal separator of G and that the components of $G[V \setminus S]$ and

$H[V \setminus S]$ coincide. Thus there is no edge between $V(H_1) \setminus S$ and $V(H_2) \setminus S$ in H. Furthermore notice that S is not a maximal clique in H and that S is a maximal clique in neither H_1 nor H_2 since $S \subset \mathcal{C}(s_{1q_1}, s)$ and $S \subset \mathcal{C}(s, s_{21})$.

Consequently, any maximal clique of H is a subset of $V(H_1)$ or it is a subset of $V(H_2)$ and thus each maximal clique of H is a maximal clique either in H_1 or in H_2 but not in both. Furthermore each maximal clique in H_1 or H_2 is obviously a clique of H and it is a maximal one since there is no edge between $V(H_1) \setminus S$ and $V(H_2) \setminus S$ in H. Taking into account that $S = S(s)$ is a subset of the cliques $\mathcal{C}(s_{1q_1}, s)$ and $\mathcal{C}(s, s_{21})$ we get that $\mathcal{C}(s', s_{11}), \mathcal{C}(s_{11}, s_{12}), \ldots, \mathcal{C}(s_{1q_1}, s), \mathcal{C}(s, s_{21}), \mathcal{C}(s_{21}, s_{22}), \ldots, \mathcal{C}(s_{2q_2}, s'')$ is a CCA of H. Hence (iii) is true.

Let G' be any interval graph and let $A_1, A_2, \ldots, A_q$ be any CCA of G'. Then Lemma 2.6 implies that for each minimal separator S' of G' there is a $j \in \{1, 2, \ldots, q-1\}$ such that $S' = A_j \cap A_{j+1}$. By (c) we have that $\mathcal{C}(s', s_{11}), \mathcal{C}(s_{11}, s_{12}), \ldots, \mathcal{C}(s_{1q_1}, s)$ is a CCA of H_1 and that $\mathcal{C}(s, s_{21}), \mathcal{C}(s_{21}, s_{22}), \ldots, \mathcal{C}(s_{2q_2}, s'')$ is a CCA of H_2. Moreover $\mathcal{C}(s', s_{11}), \mathcal{C}(s_{11}, s_{12}), \ldots, \mathcal{C}(s_{1q_1}, s), \mathcal{C}(s, s_{21}), \mathcal{C}(s_{21}, s_{22}), \ldots, \mathcal{C}(s_{2q_2}, s'')$ is a CCA of H by (iii). Applying Lemma 2.6 to these three consecutive clique arrangements we obtain that any minimal separator of H is either equal to $S = \mathcal{C}(s_{1q_1}, s) \cap \mathcal{C}(s, s_{21})$ or it is a minimal separator of H_1 and H_2, respectively. Similarly, Lemma 2.6 applied to these three consecutive clique arrangements implies that each minimal separator of H_1 or H_2 is a minimal separator of H. Thus $\Delta(H) = \Delta(H_1) \cup \Delta(H_2) \cup \{S\}$. By (a), $\Delta(H_1) = \{S(s_{11}), \ldots, S(s_{1q_1})\}$ and $\Delta(H_2) = \{S(s_{21}), \ldots, S(s_{2q_2})\}$, thus $\Delta(H) = \{S(s_{11}), \ldots, S(s_{1q_1})\} \cup \{S(s_{21}), \ldots, S(s_{2q_2})\} \cup \{S(s)\}$ and (i) is true. Then, Theorem 2.3 immediately implies that (ii) is true, and this completes the proof. □

The theorem has interesting consequences for computing the treewidth and the minimum fill-in of d-trapezoid graphs.

Definition 4.6 *Let $s_{\mathrm{L}} = s_0, s_1, s_2, \ldots, s_{r-1}, s_r = s_{\mathrm{R}}$ be a nice sequence of scanlines in a d-trapezoid diagram $\mathcal{D}(G)$. Then for all $i \in \{1, 2, \ldots, r-1\}$, first$(s_i)$ is the set of those pairs $\{u, v\}$ of nonadjacent vertices of G for which s_i is the leftmost scanline of the nice sequence that intersects both Q_u and Q_v.*

Corollary 4.1 *Let $G = (V, E)$ be a d-trapezoid graph with a d-trapezoid diagram $\mathcal{D}(G)$.*

(i) *There is a nice sequence of scanlines $s_{\mathrm{L}} = s_0, s_1, s_2, \ldots, s_{r-1}, s_r = s_{\mathrm{R}}$ in $\mathcal{D}(G)$ such that $\mathcal{C}(s_0, s_1), \mathcal{C}(s_1, s_2), \ldots, \mathcal{C}(s_{r-1}, s_r)$ is a consecutive clique arrangement of a minimal triangulation of G into an interval graph H with*

$$tw(G) = pw(G) = \omega(H) - 1 = \max_{0 \le i \le r-1} \left(|\mathcal{C}(s_i, s_{i+1})| - 1\right).$$

(ii) *There is a nice sequence of scanlines $s_{\mathrm{L}} = s'_0, s'_1, s'_2, \ldots, s'_{l-1}, s'_l = s_{\mathrm{R}}$ in $\mathcal{D}(G)$ such that $H = G_{s'_0, s'_1, \ldots, s'_l}$ is a minimum triangulation and a minimum interval graph completion of G, $\bigcup_{i=1}^{l-1}$ fill$(S(s'_i))$ is a minimum fill-in of G, where fill$(S(s'_i))$, $i \in \{1, 2, \ldots, l-1\}$, is the set of edges added to G for making $S(s'_i)$*

a clique. Furthermore

$$mic(G) = mfi(G) = \sum_{i=1}^{l-1} |first(s'_i)|.$$

Proof. Let H be a minimal triangulation of G with $tw(G) = \omega(H) - 1$. By Theorem 4.1 there is a nice sequence of scanlines $s_{\mathrm{L}} = s_0, s_1, s_2, \ldots, s_{r-1}, s_r = s_{\mathrm{R}}$ in $\mathcal{D}(G)$ such that $H = G_{s_0,s_1,s_2,\ldots,s_r}$ and $\mathcal{C}(s_0, s_1), \mathcal{C}(s_1, s_2), \ldots, \mathcal{C}(s_{r-1}, s_r)$ is a CCA of H. Hence $tw(G) = pw(G) = \omega(H) - 1 = \max_{0\le i\le r-1} (|\mathcal{C}(s_i, s_{i+1})| - 1)$.

Let H be a minimum triangulation (and a minimum interval graph completion) of G. By Theorem 4.1 there is a nice sequence of scanlines $s_{\mathrm{L}} = s'_0, s'_1, s'_2, \ldots, s'_{l-1}, s'_l = s_{\mathrm{R}}$ in $\mathcal{D}(G)$ such that $H = G_{s'_0,s'_1,\ldots,s'_l}$ and $\Delta(H) = \{S(s'_1), S(s'_2), \ldots, S(s'_{l-1})\}$. Therefore the edges of the fill-in $E(H) \setminus E(G)$ are exactly those edges added to G for making $S(s'_i)$ a clique for each $i \in \{1, 2, \ldots, l-1\}$. Hence for any edge $\{u, v\} \in E(H) \setminus E(G)$, there is a leftmost scanline in the sequence that intersects both Q_u and Q_v. Consequently, we obtain $mic(G) = mfi(G) = \sum_{i=1}^{l-1} |first(s'_i)|$. □

5 Small scanlines and dense sequences

The notion of a small scanline has been introduced in [10]. It is useful in a treewidth algorithm since a minimal separator S with $|S| > k + 1$ can not be made into a clique for obtaining a minimal triangulation H with $\omega(H) \le k + 1$.

Definition 5.1 *Consider a d-trapezoid diagram. A scanline s is k-*small *if it intersects with at most $k + 1$ d-trapezoids.*

Lemma 5.1 *Any d-trapezoid diagram has $O(nk^{d-1})$ k-small scanlines. If s^i and s^j, $i, j \in \{1, 2, \ldots d\}$, are endpoints of a k-small scanline s then $|s^i - s^j| \le 2(k + 1)$.*

Proof. Consider two parallel horizontal lines D_i and D_j, $i, j \in \{1, 2, \ldots d\}$, of the diagram. Let s be a scanline with endpoints s^i and s^j on D_i and D_j, respectively.

Similar to the argumentation in the corresponding proof for permutation graphs in [10] the number of d-trapezoids having empty intersection with the scanline s is at most

$$\frac{\min(s^i, s^j) - 1/2}{2} + \frac{2n - \max(s^i, s^j) + 1/2}{2} = n - \frac{|s^i - s^j|}{2}.$$

Hence the number of d-trapezoids intersecting the scanline s is at least $1/2\,|s^i - s^j|$. Thus for any k-small scanline holds $1/2\,|s^i - s^j| \le k + 1$ for each $i, j \in \{1, 2, \ldots d\}$. Hence there are $O(nk^{d-1})$ k-small scanlines. □

Definition 5.2 *Scanline s is a* predecessor *of scanline t in a d-trapezoid diagram if s is left of t and both have common endpoints on all horizontal lines except one. On this horizontal line (say D_j) there is exactly one point of a d-trapezoid between the endpoints of s and t (i.e. $t^j = s^j + 1$).*

Definition 5.3 *A nice sequence of scanlines $s_{\mathrm{L}} = s_0, s_1, s_2, \ldots, s_{r-1}, s_r = s_{\mathrm{R}}$ in a d-trapezoid diagram is said to be a* dense sequence *of scanlines if s_i is a predecessor of s_{i+1} for each $i \in \{0, 1, \ldots, r-1\}$.*

Lemma 5.2 *Let $G = (V, E)$ be a d-trapezoid graph with a d-trapezoid diagram $\mathcal{D}(G)$ and let $s_{\mathrm{L}} = s_0, s_1, s_2, \ldots, s_{r-1}, s_r = s_{\mathrm{R}}$ be a nice sequence of scanlines. Then*

$$tw(G) = pw(G) \leq \max_{i=0,1,\ldots,r-1} |\mathcal{C}(s_i, s_{i+1})| - 1$$

and

$$mfi(G) = mic(G) \leq \sum_{i=1}^{r-1} |first(s_i)|.$$

Moreover, if $s_{\mathrm{L}} = s_0, s_1, s_2, \ldots, s_{r-1}, s_r = s_{\mathrm{R}}$ is a dense sequence of scanlines then $H = G_{s_0,s_1,\ldots,s_r}$ is a triangulation of G into an interval graph and $\mathcal{C}(s_i, s_{i+1})$ is a clique of H for each $i \in \{0, 1, \ldots, r-1\}$.

Proof. First note that Theorem 2.2 implies $tw(G) = pw(G)$ and $mfi(G) = mic(G)$ for all d-trapezoid graphs. Let $s_{\mathrm{L}} = s_0, s_1, s_2, \ldots, s_{r-1}, s_r = s_{\mathrm{R}}$ be a nice sequence of scanlines in a d-trapezoid diagram $\mathcal{D}(G)$ of a graph G.

The following convexity property is important. Let Q_v be a d-trapezoid in $\mathcal{D}(G)$ such that s_i and s_k both intersect Q_v. Then $i < j < k$ implies that s_j also intersects Q_v. Thus $v \in \mathcal{C}(s_i, s_{i+1})$ and $v \in \mathcal{C}(s_k, s_{k+1})$ implies $v \in \mathcal{C}(s_j, s_{j+1})$ for all $j \in \{i+1, i+2, \ldots, k-1\}$ by Definition 4.2 and the convexity property. Therefore each vertex $v \in V$ appears in the subsets $\mathcal{C}(s_i, s_{i+1}), \mathcal{C}(s_{i+1}, s_{i+2}), \ldots, \mathcal{C}(s_{k-1}, s_k)$ for some i, k with $0 \leq i \leq k \leq r$. Thus $\mathcal{C}(s_0, s_1), \mathcal{C}(s_1, s_2), \ldots, \mathcal{C}(s_{r-1}, s_r)$ is a path-decomposition of G. Consequently

$$tw(G) = pw(G) \leq \max_{i=0,1,\ldots,r-1} |\mathcal{C}(s_i, s_{i+1})| - 1.$$

Let H be the graph obtained from G by adding an edge between two nonadjacent vertices u and v of G, if there is a candidate component $\mathcal{C}(s_i, s_{i+1})$, for some $i \in \{0, 1, \ldots, r-1\}$, containing u and v. Therefore $V(G) = V(H)$, $E(G) \subseteq E(H)$ and $\mathcal{C}(s_0, s_1), \mathcal{C}(s_1, s_2), \ldots, \mathcal{C}(s_{r-1}, s_r)$ are cliques of H.

Then H is an interval graph by Lemma 2.3 and therefore H is a triangulation of G. Since $\sum_{i=1}^{r-1} |first(s_i)|$ is exactly the number of edges added to G to obtain H, we obtain

$$mfi(G) = mic(G) \leq \sum_{i=1}^{r-1} |first(s_i)|.$$

Let $s_{\mathrm{L}} = s_0, s_1, s_2, \ldots, s_{r-1}, s_r = s_{\mathrm{R}}$ be a nice sequence of scanlines that is also dense. Let $i \in \{0, 1, \ldots, r-1\}$. Then for each $u, v \in \mathcal{C}(s_i, s_{i+1})$ with u and v nonadjacent in G, the d-trapezoids Q_u and Q_v either both intersect s_i or both intersect s_{i+1}, thus u and v are adjacent in H. Consequently $\mathcal{C}(s_i, s_{i+1})$ is a clique of $H = G_{s_0,s_1,\ldots,s_r}$ for each $i \in \{0, 1, \ldots, r-1\}$.

Recall that $\mathcal{C}(s_0, s_1), \mathcal{C}(s_1, s_2), \ldots, \mathcal{C}(s_{r-1}, s_r)$ is a path-decomposition of G. Thus $H = G_{s_0,s_1,\ldots,s_r}$ is a triangulation of G into an interval graph by Lemma 2.3. □

Using Corollary 4.1 we obtain the following lemma.

Lemma 5.3 *Let $G = (V, E)$ be a d-trapezoid graph with a d-trapezoid diagram $\mathcal{D}(G)$.*

(i) *If $tw(G) \leq k$ then there is a dense sequence of k-small scanlines $s_{\mathrm{L}} = s_0, s_1, s_2, \ldots, s_{r-1}, s_r = s_{\mathrm{R}}$ in $\mathcal{D}(G)$ satisfying $|\mathcal{C}(s_i, s_{i+1})| \leq k + 1$ for all $i \in \{0, 1, \ldots, r-1\}$.*

(ii) *There is a dense sequence of scanlines $s_{\mathrm{L}} = s'_0, s'_1, s'_2, \ldots, s'_{l-1}, s'_l = s_{\mathrm{R}}$ in $\mathcal{D}(G)$ such that $mfi(G) = \sum_{i=1}^{l-1} |first(s'_i)|$.*

Proof. Consider (i). By Corollary 4.1 there is a nice sequence of scanlines $s_{\mathrm{L}} = s_0, s_1, s_2, \ldots, s_{r-1}, s_r = s_{\mathrm{R}}$ in $\mathcal{D}(G)$ such that $\mathcal{C}(s_0, s_1), \mathcal{C}(s_1, s_2), \ldots, \mathcal{C}(s_{r-1}, s_r)$ is a consecutive clique arrangement of a minimal triangulation of G into an interval graph H with $tw(G) = pw(G) = \omega(H) - 1 = \max_{0 \leq i \leq r-1} (|\mathcal{C}(s_i, s_{i+1})| - 1)$. Thus each scanline s_i is k-small, for $i \in \{1, 2, \ldots, r-1\}$, since $S(s_i) \subseteq \mathcal{C}(s_i, s_{i+1})$ implies $|S(s_i)| \leq |\mathcal{C}(s_i, s_{i+1})| \leq tw(G) + 1 \leq k + 1$. (Trivially s_{L} and s_{R} are k-small for each positive integer k.) Therefore $s_{\mathrm{L}} = s_0, s_1, s_2, \ldots, s_{r-1}, s_r = s_{\mathrm{R}}$ is a nice sequence of k-small scanlines.

All the sets $\mathcal{C}(s_i, s_{i+1})$, $i \in \{0, 1, \ldots, r-1\}$, are cliques in $H = G_{s_0, s_1, s_2, \ldots, s_r}$ by Theorem 4.1. Hence for any scanline s^* between s_i and s_{i+1} and any pair of d-trapezoids Q_u and Q_v that both intersect s^*, the vertices u and v belong to $\mathcal{C}(s_i, s_{i+1})$, thus they are adjacent in H.

Consequently our particular nice sequence of scanlines can be transformed into a dense sequence of scanlines by adding a suitable sequence of scanlines $s^*_{i1}, s^*_{i2}, \ldots, s^*_{i,q_i}$ between s_i and s_{i+1} for all $i \in \{0, 1, \ldots, r-1\}$. Then $|S(s^*)| \leq |\mathcal{C}(s_i, s_{i+1})| \leq k + 1$. Hence each scanline s^* added between s_i and s_{i+1} is k-small. Thus we obtain a new sequence $s_{\mathrm{L}} = \hat{s}_0, \hat{s}_1, \hat{s}_2, \ldots, \hat{s}_{q-1}, \hat{s}_q = s_{\mathrm{R}}$ which is a dense sequence of k-small scanlines. Furthermore for every $j \in \{0, 1, \ldots, q-1\}$, $\mathcal{C}(\hat{s}_j, \hat{s}_{j+1}) \subseteq \mathcal{C}(s_i, s_{i+1})$ for some $i \in \{0, 1, \ldots, r-1\}$ by the construction of the new sequence, implying $|\mathcal{C}(\hat{s}_j, \hat{s}_{j+1})| \leq k + 1$.

Consider (ii). By Corollary 4.1 there is a nice sequence of scanlines $s_{\mathrm{L}} = s'_0, s'_1, s'_2, \ldots, s'_{l-1}, s'_l = s_{\mathrm{R}}$ in $\mathcal{D}(G)$ such that $H = G_{s'_0, s'_1, \ldots, s'_l}$ is a minimum triangulation and $mic(G) = mfi(G) = \sum_{i=1}^{l-1} |first(s'_i)|$. Analogously to (i) this sequence can be transformed into a dense sequence $s_{\mathrm{L}} = \hat{s}'_0, \hat{s}'_1, \hat{s}'_2, \ldots, \hat{s}'_{p-1}, \hat{s}'_p = s_{\mathrm{R}}$ satisfying $mic(G) = mfi(G) = \sum_{i=1}^{p-1} |first(\hat{s}'_i)|$. □

Lemmas 5.2 and 5.3 justify the correctness of our algorithms that will be described in the next section.

6 Algorithms

We present in this section our two polynomial time algorithms computing the treewidth and the pathwidth as well as the minimum fill-in and the minimum interval graph completion of a d-trapezoid graph that is given with a d-trapezoid diagram, d a fixed positive integer. Notice that for any input to one of the algorithms the constant d is equal to the number of horizontal lines in the given diagram.

6.1 Treewidth and pathwidth

We start with the algorithm computing the treewidth and pathwidth. Let k be a positive integer. First we present a procedure that checks whether the treewidth of the given d-trapezoid graph does not exceed k.

Construct a directed acyclic graph $W_k(G)$ as follows. The vertices of the graph are the k-small scanlines of $\mathcal{D}(G)$. There is an arc from scanline s to t in $W_k(G)$ if and only if

- the scanline s is a predecessor of the scanline t in $\mathcal{D}(G)$ and
- the candidate component $\mathcal{C}(s,t)$ has at most $k+1$ vertices.

The next lemma follows immediately from Lemma 5.2 and 5.3(i).

Lemma 6.1 *G has treewidth at most k if and only if there is a directed path from s_{L} to s_{R} in $W_k(G)$.*

Lemma 6.2 *The graph $W_k(G)$ has $O(nk^{d-1})$ vertices and $O(nk^{d-1})$ edges.*

Proof. The bound on the number of vertices is shown in Lemma 5.1. For each scanline s there are at most d scanlines t for which s is a predecessor. Hence the outdegree of a vertex is at most d. □

Now we describe the procedure which determines if the treewidth of a d-trapezoid graph G given with a d-trapezoid diagram $\mathcal{D}(G)$ is at most k.

Step 1 Construct the acyclic digraph $W_k(G)$ as follows. Compute all k-small scanlines in $\mathcal{D}(G)$ and compute all ordered pairs of k-small scanlines s and t for which s is a predecessor of t and $|\mathcal{C}(s,t)| \leq k+1$.

Step 2 If there exists a path in $W_k(G)$ from s_{L} to s_{R}, then report that the treewidth of G is at most k. If such a path does not exist, then report that the treewidth of G is larger than k.

Lemma 6.3 *The running time of the procedure is $O(nk^{d-1})$.*

Proof. We describe any scanline s by the vector $(s^1, s^2, \ldots, s^d)$ of its endpoints on the horizontal lines of the diagram. The procedure processes in step 1 only those scanlines $s = (s^1, s^2, \ldots, s^d)$ satisfying $s^1 - 2k - 2 \leq s^i \leq s^1 + 2k + 2$ for all $i \in \{2, 3, \ldots, d\}$ (with obvious boundary conditions), since all k-small scanlines fulfil this condition by Lemma 5.1.

We denote by $A(s^1, s^2, \ldots, s^d)$ the number of d-trapezoids that intersect the scanline $s = (s^1, s^2, \ldots, s^d)$ and we denote by $B^i(s^1, s^2, \ldots, s^d)$, $i \in \{1, 2, \ldots, d\}$, the number of d-trapezoids intersecting the scanline $s = (s^1, s^2, \ldots, s^d)$ or the scanline $t = (s^1, s^2, \ldots, s^{i-1}, s^i + 1, s^{i+1}, \ldots, s^d)$. Thus $s = (s^1, s^2, \ldots, s^d)$ is k-small if and only if $A(s^1, s^2, \ldots, s^d) \leq k+1$ and $|\mathcal{C}(s,t)| \leq k+1$ if and only if $B^i(s^1, s^2, \ldots, s^d) \leq k+1$. Notice that $A(0.5, 0.5, \ldots, 0.5) = 0$. All other values of $A(s^1, s^2, \ldots, s^d)$ and $B^i(s^1, s^2, \ldots, s^d)$ are computed using the following rules. For all $i \in \{1, 2, \ldots, d\}$ holds

(i) $B^i(s^1, s^2, \ldots, s^d) = A(s^1, s^2, \ldots, s^d)$ if the unique d-trapezoid with a point between the scanlines $s = (s^1, s^2, \ldots, s^d)$ and $t = (s^1, s^2, \ldots, s^{i-1}, s^i + 1, s^{i+1}, \ldots, s^d)$ on the horizontal line D_i intersects the scanline s, otherwise $B^i(s^1, s^2, \ldots, s^d) = A(s^1, s^2, \ldots, s^d) + 1$.

(ii) $A(s^1, s^2, \ldots, s^{i-1}, s^i + 1, s^{i+1}, \ldots s^d) = B^i(s^1, s^2, \ldots, s^d)$ if the unique d-trapezoid with a point between the scanlines $s = (s^1, s^2, \ldots, s^d)$ and $t = (s^1, s^2, \ldots, s^{i-1}, s^i + 1, s^{i+1}, \ldots, s^d)$ on the horizontal line D_i intersects t, otherwise $A(s^1, s^2, \ldots, s^d) = B^i(s^1, s^2, \ldots, s^d) - 1$.

During step 1 the procedure computes $O(nk^{d-1})$ values of $A(s^1, s^2, \ldots, s^d)$ and $O(nk^{d-1})$ values of $B^i(s^1, s^2, \ldots, s^d)$. Clearly it can be checked in constant time whether the unique d-trapezoid with a point between the scanlines $s = (s^1, s^2, \ldots, s^d)$ and $t = (s^1, s^2, \ldots, s^{i-1}, s^i + 1, s^{i+1}, \ldots, s^d)$ on the horizontal line D_i, $i \in \{1, 2, \ldots, d\}$, intersects s and t, respectively. Hence step 1 takes time $O(nk^{d-1})$.

Computing whether there is a directed path from s_{L} to s_{R} in $W_k(G)$ takes $O(nk^{d-1})$ time by a standard single source shortest-path algorithm in a directed acyclic graph. Hence the total procedure can be implemented to run in $O(nk^{d-1})$ time. □

Finally we show that the procedure can be used for obtaining an algorithm that computes the treewidth.

Theorem 6.1 *For each positive integer d, there is an $O(n\, tw(G)^{d-1})$ algorithm computing the treewidth and the pathwidth of a d-trapezoid graph G for which a d-trapezoid diagram $\mathcal{D}(G)$ is part of the input.*

Proof. The algorithm first computes a number L such that $L/2 \leq tw(G) \leq L$. This can be done, using the procedure described above $O(\log tw(G))$ times, in overall time $O(n\, tw(G)^{d-1})$, by calling the procedure for $k = 1, 2, 4, \ldots$ until it reports `$tw(G) \leq k$' for the first time. Take this value of k as L and construct the directed graph $W_L(G)$. Then modify $W_L(G)$ as follows. Put weights on the arcs, saying how many vertices are in the corresponding candidate component. Then search for a path from s_{L} to s_{R}, such that the maximum over the weights of arcs in the path is minimized. This maximum weight minus one gives the exact treewidth $tw(G)$. A corresponding shortest-path algorithm for directed acyclic graphs has running time $O(n\, tw(G)^{d-1})$. □

6.2 Minimum fill-in and interval graph completion

Now we show how to compute the minimum fill-in and the minimum interval graph completion of a d-trapezoid graph G with d-trapezoid diagram $\mathcal{D}(G)$. The algorithm we present directly computes $mfi(G)$ by solving a single source shortest-path problem on a suitable directed acyclic graph.

Construct a directed acyclic graph $\widetilde{W}(G)$ as follows. The vertices of the graph are the scanlines of $\mathcal{D}(G)$. There is an arc from scanline s to t in $\widetilde{W}(G)$ if and only if the scanline s is a predecessor of t. The length of an arc from s to t is the number of pairs of d-trapezoids Q_u and Q_v in the diagram that have empty intersection, do not both intersect s but both intersect t.

Lemma 6.4 *mfi*(G) *is equal to the length of a shortest directed path from* s_{L} *to* s_{R} *in* $\widetilde{W}(G)$.

Proof. By Lemma 5.2, *mfi*$(G) \leq \sum_{i=1}^{r-1} |$*first*$(s_i)|$ for each nice sequence of scanlines $s_{\mathrm{L}} = s_0, s_1, s_2, \ldots, s_{r-1}, s_r = s_{\mathrm{R}}$ in $\mathcal{D}(G)$. Furthermore by Lemma 5.3(ii), there is a dense sequence of scanlines $s_{\mathrm{L}} = s'_0, s'_1, s'_2, \ldots, s'_{l-1}, s'_l = s_{\mathrm{R}}$ in $\mathcal{D}(G)$ such that *mfi*$(G) = \sum_{i=1}^{l-1} |$*first*$(s'_i)|$. The shortest paths from s_{L} to s_{R} are in one-to-one correspondence to the dense sequences of scanlines in $\mathcal{D}(G)$. Finally the length of an arc from s to t is defined such that it is equal to $|$*first*$(t)|$ if the dense sequence is $s_{\mathrm{L}} = s'_0, \ldots, s, t, \ldots, s'_l = s_{\mathrm{R}}$. This completes the proof. □

Similar to Lemma 6.2 one obtains the following.

Lemma 6.5 *The graph* $\widetilde{W}(G)$ *has* $O(n^d)$ *vertices and* $O(n^d)$ *edges.*

Hence the algorithm that computes the minimum fill-in of a d-trapezoid graph G given with a d-trapezoid diagram $\mathcal{D}(G)$ is as follows.

Step 1 Construct the acyclic digraph $\widetilde{W}(G)$. Compute the length of all arcs in $\widetilde{W}(G)$.

Step 2 Compute the length of a shortest path from s_{L} to s_{R} in $\widetilde{W}(G)$.

Theorem 6.2 *For each positive integer d, there is an* $O(n^d)$ *algorithm computing the minimum fill-in and the minimum interval graph completion of a given* d*-trapezoid graph* G *where a* d*-trapezoid diagram* $\mathcal{D}(G)$ *is part of the input.*

Proof. Again we describe any scanline s by the vector $(s^1, s^2, \ldots, s^d)$ of its endpoints on the horizontal lines of the diagram. For any scanline $s = (s^1, s^2, \ldots, s^d)$ we denote by $L(s^1, s^2, \ldots, s^d)$ the number of d-trapezoids that are left of the scanline s. In a preprocessing the algorithm computes $L(s^1, s^2, \ldots, s^d)$ for all scanlines $s = (s^1, s^2, \ldots, s^d)$ in the diagram.

This can be done quite similar to step 1 of the procedure in the previous subsection. Clearly $L(0.5, 0.5, \ldots, 0.5) = 0$. All other values of $L(s^1, s^2, \ldots, s^d)$ can be computed by the following rule. For all $i \in \{1, 2, \ldots, d\}$, $L(s^1, s^2, \ldots, s^{i-1}, s^i + 1, s^{i+1}, \ldots, s^d) = L(s^1, s^2, \ldots, s^d) + 1$ if the unique d-trapezoid with a point between the scanlines $s = (s^1, s^2, \ldots, s^d)$ and $t = (s^1, s^2, \ldots, s^{i-1}, s^i + 1, s^{i+1}, \ldots, s^d)$ on the horizontal line D_i is left of t, otherwise $L(s^1, s^2, \ldots, s^{i-1}, s^i + 1, s^{i+1}, \ldots, s^d) = L(s^1, s^2, \ldots, s^d)$. This preprocessing can be done in time $O(n^d)$ since there are $O(n^d)$ scanlines by Corollary 3.1.

Then the algorithm computes the length of all arcs of $\widetilde{W}(G)$. Consider an arc from s to t. First the unique d-trapezoid Q_v with a point between the scanlines $s = (s^1, s^2, \ldots, s^d)$ and $t = (s^1, s^2, \ldots, s^{i-1}, s^i + 1, s^{i+1}, \ldots, s^d)$ is determined. If Q_v intersects s then the length of the arc is 0. Otherwise the length is equal to the number of d-trapezoids that intersect s but not Q_v. Hence the length of the arc is exactly $L(l_v^1 - 0.5, l_v^2 - 0.5, \ldots, l_v^d - 0.5) - L(s^1, s^2, \ldots, s^d)$. Consequently the directed acyclic graph $\widetilde{W}(G)$ can be constructed in time $O(n^d)$.

Computing a shortest path from s_{L} to s_{R} in $\widetilde{W}(G)$ can be done in time $O(n^d)$ by a standard single source shortest-path algorithm in a directed acyclic graph. Hence the overall running time of the algorithm is $O(n^d)$. □

7 Conclusions

In this paper we described two algorithms designed by a `scanline approach' that has been introduced in [10]. Both polynomial time algorithms solve NP-complete problems when these problems are restricted to d-trapezoid graphs, if a d-trapezoid diagram of the given graph is part of the input (d a fixed positive integer). An $O(n\, tw(G)^{d-1})$ algorithm computes the treewidth and the pathwidth. An $O(n^d)$ algorithm computes the minimum fill-in and the minimum interval completion of the input graph. Both algorithms have running times that are significantly better than the best known up to know, i.e., $O(\max(n^{2.376\,d}, n^{2d+2}))$ [31] (giving running time $O(n^6)$ for trapezoid graphs). Indeed we show that there are algorithms for both problems for which the order of magnitude of the running time is equal to the order of magnitude of the number of (k-small) scanlines in a d-trapezoid diagram of the input graph. Hence improving our algorithms seems to require completely new ideas.

Our algorithms are simple and efficient for small d, in particular for $d = 2$. The 2-trapezoid graphs are exactly the trapezoid graphs and there is an $O(n^2)$ algorithm computing a trapezoid diagram for a given trapezoid graph [28]. Hence we obtain $O(n^2)$ algorithms, even if no trapezoid diagram is part of the input. Furthermore we obtain an $O(n^2)$ algorithm for computing the minimum fill-in of permutation graphs, as announced in [22], since permutation graphs are a subclass of the trapezoid graphs. (Notice that the known minimum fill-in algorithm for bipartite permutation graphs of [34] has running time $O(n^5)$.) Clearly our algorithms also apply to cocomparability graphs of dimension at most d if they are given with a suitable intersection model (see [25]) since they are a subclass of the d-trapezoid graphs.

Our algorithms for d-trapezoid graphs require an intersection model as part of the input. This is not surprising. On one hand, all four problems that we considered are NP-complete on cocomparability graphs [2, 38], hence there is no polynomial time algorithm for d-trapezoid graphs if the parameter d is unbounded, unless P=NP. On the other hand, the recognition problem for d-trapezoid graphs, $d \geq 3$, is NP-complete [37]. Nevertheless it has been shown that using a different approach an $O(n^{3d+3})$ time algorithm computing the treewidth and pathwidth for d-trapezoid graphs, $d \geq 3$, can be designed that does not require an intersection model as part of the input [21, 25].

References

[1] S. Arnborg, Efficient algorithms for combinatorial problems on graphs with bounded decomposability — A survey. *BIT* **25**, $2 - 23$, 1985.

[2] S. Arnborg, D.G. Corneil and A. Proskurowski, Complexity of finding embeddings in a k-tree, *SIAM Journal on Algebraic and Discrete Methods* **8**, $277 - 284$, 1987.

[3] S. Arnborg, J. Lagergren and D. Seese, Easy problems for tree-decomposable graphs, *Journal of Algorithms* **12**, $308 - 340$, 1991.

[4] S. Arnborg and A. Proskurowski, Linear time algorithms for NP-hard problems restricted to partial k-trees. *Discrete Applied Mathematics* **23**, $11 - 24$, 1989.

[5] H.L. Bodlaender, Dynamic programming algorithms on graphs with bounded treewidth, *Proceedings of the* $15th$ *International Colloquium on Automata, Languages and Programming*, $105 - 119$, Springer-Verlag, Lecture Notes in Computer Science 317, 1988.

[6] H.L. Bodlaender, A linear time algorithm for finding tree-decompositions of small treewidth, *Proceedings of the* $25th$ *Annual ACM Symposium on Theory of Computing*, $226 - 234$, 1993. To appear in *SIAM Journal on Computing*.

[7] H.L. Bodlaender, A tourist guide through treewidth, *Acta Cybernetica* **11**, $1-23$, 1993.

[8] H.L. Bodlaender. A partial k-arboretum of graphs with bounded treewidth. Technical Report UU-CS-1996-02, Department of Computer Science, Utrecht University, Utrecht, the Netherlands, 1996.
URL: ftp://ftp.cs.ruu.nl/pub/RUU/CS/techreps/CS-1996/1996-02.ps.gz

[9] H. Bodlaender and R.H. Möhring, The pathwidth and treewidth of cographs, *SIAM Journal on Discrete Mathematics* **6**, $181 - 188$, 1993.

[10] H. Bodlaender, T. Kloks and D. Kratsch, Treewidth and pathwidth of permutation graphs, *SIAM Journal on Discrete Mathematics* **8**, $606 - 616$, 1995.

[11] K.S. Booth and G.S. Lueker, Testing for the consecutive ones property, interval graphs, and graph planarity using PQ-tree algorithms, *Journal of Computer and System Sciences* **13**, $335 - 379$, 1976.

[12] A. Brandstädt, Special graph classes — a survey, Schriftenreihe des Fachbereichs Mathematik, SM-DU-199, Universität Duisburg Gesamthochschule, 1991.

[13] D.G. Corneil, Y. Perl and L.K. Stewart, Cographs: recognition, applications and algorithms, *Congressus Numerantium* **43**, $249 - 258$, 1984.

[14] D.G. Corneil, S. Olariu and L. Stewart, Asteroidal triple-free graphs, *Proceedings of the* $19th$ *International Workshop on Graph-Theoretic Concepts in Computer Science*, $211 - 224$, Springer-Verlag, Lecture Notes in Computer Science 790, 1994.

[15] S. Felsner, R. Müller and L. Wernisch, Trapezoid graphs and generalizations, geometry and algorithms, *Proceedings of the* $4th$ *Scandinavian Workshop on Algorithm Theory*, $143-154$, Springer-Verlag, Lecture Notes in Computer Science 842, 1994.

[16] R. Garbe, Tree-width and path-width of comparability graphs of interval orders, *Proceedings of the 20th International Workshop on Graph-Theoretic Concepts in Computer Science*, $26-37$, Springer-Verlag, Lecture Notes in Computer Science 903, 1995.

[17] P.C. Gilmore, A.J. Hoffman, A characterization of comparability graphs and interval graphs, *Canadian Journal of Mathematics* **16**, $539 - 548$, 1964.

[18] M.C. Golumbic, *Algorithmic Graph Theory and Perfect Graphs*, Academic Press, New York, 1980.

[19] T. Kloks, Treewidth of circle graphs, *Proceedings of the 4th International Symposium on Algorithms and Computation*, $108 - 117$, Springer-Verlag, Lecture Notes in Computer Science 762, 1993.

[20] T. Kloks, Minimum fill-in for chordal bipartite graphs, Technical Report RUU-CS-93-11, Department of Computer Science, Utrecht University, The Netherlands, 1993.

[21] T. Kloks, *Treewidth — Computations and Approximations*, Springer-Verlag, Lecture Notes in Computer Science 842, 1994.

[22] T. Kloks, H. Bodlaender, H. Müller and D. Kratsch, Computing treewidth and minimum fill-in: all you need are the minimal separators, *Proceedings of the First Annual European Symposium on Algorithms*, $260 - 271$, Springer-Verlag, Lecture Notes in Computer Science 726, 1993.
Erratum: *Proceedings of the Second Annual European Symposium on Algorithms*, 508, Springer-Verlag, Lecture Notes in Computer Science 855, 1994.

[23] T. Kloks and D. Kratsch, Treewidth of chordal bipartite graphs, *Journal of Algorithms* **19**, $266 - 281$, 1995.

[24] T. Kloks, D. Kratsch and H. Müller, Approximating the bandwidth of asteroidal-triple free graphs, *Proceedings of the Third Annual European Symposium on Algorithms*, $434 - 447$, Springer-Verlag, Lecture Notes in Computer Science 979, 1995.

[25] T. Kloks, D. Kratsch and J. Spinrad, On treewidth and minimum fill-in of asteroidal triple-free graphs, to appear in *Theoretical Computer Science*.

[26] T. Kloks, D. Kratsch and C.K.Wong, Minimum fill-in on circle and circular-arc graphs, *Proceedings of the* 23 *International Colloquium on Automata, Languages and Programming*, $256 - 267$, Springer-Verlag, Lecture Notes in Computer Science 1099, 1996.

[27] J. van Leeuwen, Graph algorithms, *Handbook of Theoretical Computer Science, A: Algorithms and Complexity Theory*, $527 - 631$, North Holland Publishing Company, Amsterdam, 1990.

[28] T.-H. Ma and J.P. Spinrad, An $O(n^2)$ Algorithm for the 2-Chain Cover Problem and Related Problems, *Proceedings of the* 2*nd Symposium on Discrete Algorithms*, $363 - 372$, 1991.

[29] R.H. Möhring, Triangulating graphs without asteroidal triples, *Discrete Applied Mathematics* **64**, $281 - 287$, 1996.

[30] A. Parra, Triangulating multitolerance graphs, Technical Report $392/1994$, Technische Universität Berlin, Germany, 1994.

[31] A. Parra and P. Scheffler, How to use the minimal separators of a graph for its chordal triangulation, *Proceedings of the* 22*nd International Colloquium on Automata, Languages and Programming*, $123-134$, Springer-Verlag, Lecture Notes in Computer Science 944, 1995.

[32] D.J. Rose, A graph-theoretic study of the numerical solution of sparse positive definite systems of linear equations, in: R.C. Reed, (ed.); Graph Theory and Computing, $183-217$, Academic Press, New York, 1972.

[33] D.J. Rose, R.E. Tarjan and G.S. Lueker, Algorithmic aspects of vertex elimination on graphs, *SIAM Journal on Computing* **5**, $266-283$, 1976.

[34] J. Spinrad, A. Brandstädt and L. Stewart, Bipartite permutation graphs, *Discrete Applied Mathematics* **18**, $279-292$, 1987.

[35] R. Sundaram, K. Sher Singh and C. Pandu Rangan, Treewidth of circular arc graphs, *SIAM Journal on Discrete Mathematics* **7**, $647-655$, 1994.

[36] R.E. Tarjan, Decomposition by clique separators, *Discrete Mathematics* **55**, $221-232$, 1985.

[37] M. Yannakakis, The complexity of the partial order dimension problem, *SIAM Journal on Algebraic and Discrete Methods* **3**, $351-358$, 1982.

[38] M. Yannakakis, Computing the minimum fill-in is NP-complete, *SIAM Journal on Algebraic and Discrete Methods* **2**, $77-79$, 1981.