Journal of Graph Algorithms and Applications

http://www.cs.brown.edu/publications/jgaa/

vol. 2, no. 5, pp. 1–23 (1998)

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.
http://www.cs.ruu.nl/~hansb/
hansb@cs.ruu.nl

Ton Kloks
Department of Applied mathematics and DIMATIA
Charles University, Malostranské nám. 25
11800 Praha 1, Czech Republic.
kloks@kam.ms.mff.cuni.cz

Dieter Kratsch
Friedrich-Schiller-Universität
Fakultät für Mathematik und Informatik
07740 Jena, Germany.
http://www.minet.uni-jena.de/~kratsch/
kratsch@minet.uni-jena.de

Haiko Müller
Friedrich-Schiller-Universität
Fakultät für Mathematik und Informatik
07740 Jena, Germany.
http://www.minet.uni-jena.de/~hm/
hm@minet.uni-jena.de

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 in time $O(n\,tw(G)^{d-1})$. In both cases, 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, each of the four graph parameters can be computed in time $O(n^2)$ for trapezoid graphs and thus for permutation graphs even if no intersection model is part of the input.

Communicated by S. Khuller: submitted August 1996; revised March 1998.

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 in graph theory and other areas. One reason for this interest is that many well known and important graph 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]. 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.

A related graph problem is the MINIMUM FILL-IN problem. In this problem, we want to add as few 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 [29]. Due to the lack of efficient algorithms for finding an optimal solution, in practice one usually has to work with certain heuristics.

By now there is a large number of results on the algorithmic complexity of the problems TREEWIDTH and PATHWIDTH when restricted to special graph classes. The TREEWIDTH (resp. PATHWIDTH) 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]. (For information on graph classes we refer to [11, 19]. For other definitions we refer to Section 2.) For various special classes of graphs, it has been shown that the treewidth can be computed in polynomial time, as e.g. cographs [10], circular-arc graphs [32], chordal bipartite graphs [22], permutation graphs [9], circle graphs [20] and cointerval graphs [17].

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 TREEWIDTH and PATHWIDTH. The MINIMUM FILL-IN 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 [36] and on bipartite graphs [33]. The only graph classes for which the MINIMUM FILL-IN problem were known to be polynomial time solvable were for almost ten years the relatively small classes of cographs [13] and bipartite permutation graphs [31]. Now polynomial time algorithms for chordal bipartite graphs [12], multitolerance graphs [27] as well as circle and circular-arc graphs [24] are available.

Our paper is organized as follows. Preliminaries on treewidth, pathwidth, minimum fill-in, minimal separators and triangulations are given in Section 2. In Section 3 we define d-trapezoid graphs and we summarize some of their structural and algorithmic properties. d-trapezoid graphs are a common generalization of interval, permutation and trapezoid graphs such that 1-trapezoid graphs coincide with interval graphs and 2-trapezoid graphs coincide with trapezoid graphs. d-trapezoid graphs are intersection graphs of d-trapezoids in a so-called d-trapezoid diagram which consists of d parallel lines in which a d-trapezoid is defined by a collection of d intervals, one interval on each of the parallel lines (For more details see Definition 12.) In Section 4 we present the major structural results of the paper on which our efficient algorithms are based. In Theorem 17 we establish a representation theorem of minimal triangulations of a d-trapezoid graph G in terms of scanlines of a d-trapezoid diagram $\mathcal{D}(G)$. Hence in contrast to previous work in this area as e.g. [9, 20, 28, 32], our algorithms are based on a general representation theorem that enables the design of an algorithm for TREEWITH, MINIMUM FILL-IN and possibly related problems (concerning the optimization of a graph parameter over all minimal triangulations of the graph). A similar representation theorem is given in [24] for circle and circular-arc graphs. In Section 5 we introduce so-called small scanlines and dense sequences of scanlines as tools to obtain significantly faster algorithms than by a straightforward application of the representation theorem.

In Section 6 we present our polynomial time algorithms to solve NP-complete graph problems when their input is restricted to d-trapezoid graphs. Both algorithms require that d is a constant and that a d-trapezoid diagram of the given graph is part of the input. The algorithm to compute the treewidth and the pathwidth has running time $O(n\, tw(G)^{d-1})$. The algorithm to compute the minimum fill-in and the minimum interval completion has running time $O(n^d)$. Up to now the best known algorithms for all four problems had running time $O(\max(n^{2.376\, d}, n^{2d+2}))$ [28].

Our algorithms are simple and efficient for trapezoid graphs ($d = 2$). In that case they do not even require a trapezoid diagram as part of the input. We obtain $O(n^2)$ algorithms to compute all the four graph parameters on trapezoid graphs (compared to running time $O(n^6)$ of the algorithm in [28]). Furthermore we obtain an $O(n^2)$ algorithm to compute the minimum fill-in of permutation graphs (compared to running time $O(n^5)$ of the algorithm for bipartite permutation graphs in [31]). A similar algorithm to compute the treewidth and the pathwidth for permutation graphs has been presented in [9].

For $d \geq 3$, our algorithms for d-trapezoid graphs require an intersection model as part of the input. This is clearly a disadvantage although it is a quite natural assumption. On one hand, all four problems that we consider are NP-complete on cocomparability graphs [2, 36], 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 is NP-complete for any fixed $d \geq 3$ [35]. Thus, for $d \geq 3$, to compute a d-trapezoid diagram, whenever the input graph is a d-trapezoid graph, means to solve an NP-complete problem.

Fortunately, a d-trapezoid diagram as part of the input is not a necessary assumption to establish algorithms with polynomial running time. $O(n^5 R + n^3 R^3)$ time algorithms to compute the treewidth and minimum fill-in of a given asteroidal triple-free graph on n vertices with R minimal separators imply that the treewidth, pathwidth, minimum fill-in and minimum interval completion of a d-trapezoid graph can be computed in time $O(n^{3d+3})$ without an intersection model as part of the input [23].

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 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 well-known class of graphs. For detailed information on chordal graphs and other special classes of graphs we refer to [11, 19]. (For more information on chordal graphs see also Subsection 2.3.)

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 referred to the survey paper [7]. In this paper we introduce the treewidth by means of triangulations.

Definition 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 3 *The* treewidth *of a graph G, denoted by $tw(G)$, is the minimum of $\omega(H) - 1$ where H ranges over all triangulations of G.*

The pathwidth can be defined in terms of triangulations of a special kind.

Definition 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 every interval graph is chordal [19].

Definition 5 *The* pathwidth *of a graph G, denoted by $pw(G)$, is the minimum of $\omega(H) - 1$ where H ranges over all triangulations of G which are interval graphs.*

Definition 6 *A* path-decomposition *of a graph $G = (V, E)$ is a sequence of subsets of V, $(X_1, \ldots, X_r)$, such that $\bigcup_{1 \le i \le r} X_i = V$, for all $\{v, w\} \in E$, there is an i, $1 \le i \le r$, $v, w \in X_i$, and for all $v \in V$, there are l_v, r_v, such that for all integers j, $1 \le l_v \le j \le r_v \le r \Leftrightarrow v \in X_j$. The* width *of path-decomposition $(X_1, \ldots, X_r)$ is $\max_{1 \le i \le r} |X_i| - 1$.*

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 [21, Lemma 2.2.8].

Lemma 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 $\omega(H) \le k+1$.*

The following characterization of interval graphs is due to Gilmore and Hoffman [18].

Theorem 2 *G is an interval graph if and only if the maximal cliques of G can be ordered $C_1, C_2, \ldots, C_t$ 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* 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 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, $1 \le i \le r$ a clique, (i.e., for all $v, w \in V, v \ne w$: $\{v, w\} \in F \Leftrightarrow \exists i : v, w \in X_i$), 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 or pathwidth at most k are recognizable in $O(n)$ time [6, 8]. The large constants depending on k involved in these algorithms make them usually not practical. It is therefore of importance to find polynomial algorithms to compute the treewidth and the pathwidth for special classes of graphs which are as large as possible, where the treewidth (resp. pathwidth) of input graphs is not supposed to be bounded by a constant k. The aim of this paper is to present fast algorithms to compute the treewidth and the pathwidth as well as the minimum fill-in and the minimum interval graph completion on a relatively large parameterized class of graphs.

Definition 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 minimum of $|E(H)| - |E(G)|$ where H ranges over all triangulations of G.*

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 minimum of $|E(H)| - |E(G)|$ where H ranges over all triangulations of G which are interval graphs.*

Hence solving the MINIMUM FILL-IN (resp. MINIMUM INTERVAL GRAPH COMPLETION) problem on a graph G is equivalent to finding a triangulation H of G (such that H is an interval graph) that has as few edges as possible.

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., [9, 21]).

Definition 8 *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 well-known lemma gives a useful characterization of minimal separators.

Lemma 4 *Let $G = (V, E)$ be a graph and $S \subseteq V$. Let C_a and C_b be the components of $G[V \setminus S]$, containing a and b respectively. Then S is a minimal a, b-separator of G if and only if every vertex of S has a neighbor in C_a and a neighbor in C_b.*

Using the characterization in Theorem 2, one can easily identify the minimal separators of an interval graph which has been shown in [23].

Lemma 5 *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 $A_i \cap A_{i+1}$, $i \in \{1, 2, \ldots, q-1\}$.*

Various types of triangulations are of importance in algorithms to compute the treewidth or the minimum fill-in for special graph classes.

Definition 9 *A triangulation H of a graph G is a* minimal triangulation *of G if no proper subgraph of H is a triangulation of G.*

In 1976 Rose, Tarjan and Lueker have given the following characterization of minimal triangulations [30].

Theorem 6 *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.*

Theorem 7 provides another characterization of minimal triangulations (see [23]).

Definition 10 *Let $G = (V, E)$ be a graph. For any collection $\mathcal{M}$ of subsets of V, we denote by $G_{\mathcal{M}}$ the graph obtained from G by adding edges between all pairs of nonadjacent vertices x and y of G for which an $S \in \mathcal{M}$ with $\{x, y\} \subseteq S$ exists. We denote by $\mathbf{Sep}(G)$ the set of all minimal separators of G.*

Notice that any set $S \in \mathcal{M}$ is a clique in $G_{\mathcal{M}}$.

Theorem 7 *Let H be a triangulation of the graph G. Then H is a minimal triangulation of G if and only if $H = G_{\mathbf{Sep}(H)}$.*

The following theorem is an immediate consequence of [21, Theorem 2.1.2],

Theorem 8 *Let H be a minimal triangulation of the graph $G = (V, E)$. Then the following two 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 $V(C)$ is the vertex set of a connected component C of $H[V \setminus S]$ then $G[V(C)]$ is a connected component of $G[V \setminus S]$.*

The following theorem of Möhring in [26] considers asteroidal-triple free graphs, which form a graph class containing interval, permutation, trapezoid and cocomparability graphs. (For information on asteroidal triple-free graphs see [14].)

Theorem 9 *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.*

Any d-trapezoid graph is a cocomparability graph (see Section 3) and thus asteroidal triple-free.

Corollary 10 *Any minimal triangulation of an d-trapezoid graph is an interval graph. Hence $pw(G) = tw(G)$ and $mfi(G) = mic(G)$ for each d-trapezoid graph G.*

Therefore we may restrict ourselves to algorithms to compute the treewidth and the minimum fill-in for d-trapezoid graphs.

2.3 Preliminaries on chordal graphs and simplicial vertices

In 1961 Dirac established some of the fundamental structural properties of chordal graphs [15] (see also [19]).

Definition 11 *A vertex v of a graph $G = (V, E)$ is* simplicial *if $N[v]$ is a clique in G.*

Theorem 11 *Let $G = (V, E)$ be a graph. Then the following conditions are equivalent:*

(i) *G is chordal.*

(ii) *Every minimal separator of G is a clique.*

(iii) *Every induced subgraph of G has a simplicial vertex.*

Theorem 12 *Let $G = (V, E)$ be a chordal graph and let S be a minimal separator of G. Then every component C of $G[V \setminus S]$ contains a simplicial vertex of the graph G.*

The following lemma is of importance for the proof of our main structural theorem (Theorem 17).

Lemma 13 *Let $G = (V, E)$ be a graph. Then no simplicial vertex of G is contained in a minimal separator of G. Furthermore no minimal triangulation H of G contains an edge $e \in E(H) \setminus E(G)$ such that an endpoint of e is a simplicial vertex of G.*

Proof. Let S be a minimal a, b-separator of G and $s \in S$. Then s has a neighbor s_a in C_a and a neighbor s_b in C_b by Lemma 4. Clearly s_a and s_b are not adjacent. Hence s is not simplicial.

Now assume that H is a minimal triangulation of G. Then $H = G_{\mathbf{Sep}(H)}$ by Theorem 7. Thus no edge incident to a simplicial vertex is added to obtain H from G, since $\mathbf{Sep}(H) \subseteq \mathbf{Sep}(G)$ by Theorem 8 and since no simplicial vertex is contained in a minimal separator of G. □

3 *d*-Trapezoid graphs

Flotow introduced d-trapezoid graphs in [16].

Definition 12 *Let d be a positive integer. A d-*trapezoid diagram *of a graph $G = (V, E)$ assigns to each vertex v of G a collection of d intervals*

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

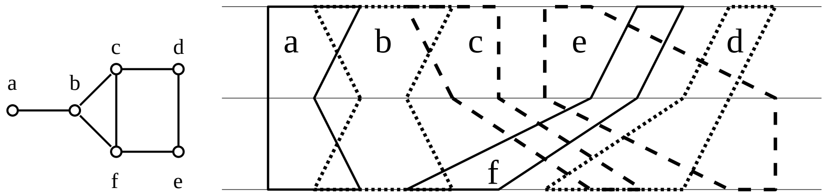

Figure 1: A 3-trapezoid graph G and a 3-trapezoid model $\mathcal{D}(G)$

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 labeled $D_1, D_2, \ldots D_d$ from bottom to the top. Identify points $1, 2, \ldots, 2n$ 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 13 *A graph G is a d-trapezoid graph if it has a d-trapezoid diagram.*

The following theorem is a consequence of Definition 12 (see [16]).

Theorem 14 *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 $d \geq 3$ [35]. Hence for fixed $d \geq 3$, to compute a d-trapezoid diagram of the given graph, if it is indeed a d-trapezoid graph, means to solve an NP-complete problem. Moreover, at present not even reasonable approximation algorithms for the interval dimension of a partially ordered set are known. Thus to assume for $d \geq 3$ that a d-trapezoid diagram is part of the input is a strong assumption. Theorem 14 also shows that for any fixed d the d-trapezoid graphs form a subclass of the cocomparability graphs. Hence every d-trapezoid graph is asteroidal triple-free. Consequently as already mentioned in Corollary 10, 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.

Details about the terminology used in Theorem 14 and the remarks above can be found in [34].

The definition of a d-trapezoid diagram implies that 1-trapezoid graphs and interval graphs coincide and that 2-trapezoid graphs and trapezoid graphs coincide. Furthermore permutation graphs are exactly those 2-trapezoid graphs having a 2-trapezoid diagram satisfying $r_v^i - l_v^i = 1$ for all vertices v and $i \in \{1, 2\}$. For $d = 2$ we can drop the requirement that a 2-trapezoid diagram must be part of the input since there is an $O(n^2)$ recognition algorithm for trapezoid graphs that also computes a trapezoid model if the input graph is a trapezoid graph [25]. (Notice that interval graphs are of no interest to us since the four problems of this paper are easy for interval graphs.)

Scanlines as a tool to represent all minimal separators of a graph have been used in various efficient algorithms to compute the treewidth or the minimum fill-in for special classes of intersection graphs, among them permutation, d-trapezoid, circle and circular-arc graphs [9, 20, 21, 24, 28].

Definition 14 *Let $G = (V, E)$ be a graph on n vertices with d-trapezoid diagram $\mathcal{D}(G)$. On each of the horizontal lines $D_1, D_2, \ldots, D_d$ of $\mathcal{D}(G)$, identify $2n + 1$ unit distance apart points $0.5, 1.5, \ldots, 2n + 0.5$ from left to right, such that the point j is between the points $j - 0.5$ and $j + 0.5$ for all $j \in \{1, 2, \ldots, 2n\}$.*

A scanline *s of the d-trapezoid diagram $\mathcal{D}(G)$ is a sequence $(s^1, s^2, \ldots, s^d)$ of d* scanpoints *$s^i \in \{0.5, 1.5, \ldots, 2n + 0.5\}$, $i \in \{1, 2, \ldots, d\}$.*

In the d-trapezoid diagram $\mathcal{D}(G)$ we represent the scanpoint s^i as point on the horizontal line D_i. The scanline s is represented by drawing a line segment between pairs of scanpoints on consecutive horizontal lines.

Definition 15 *Let $\mathcal{D}(G)$ be a d-trapezoid diagram. Let s_1 and s_2 be two different scanlines and let Q_u and Q_v be two different d-trapezoids of $\mathcal{D}(G)$. The scanline s_1 is* left of *s_2 if $s_1^i \le s_2^i$ for all $i \in \{1, 2, \ldots d\}$. Note that a scanline s_1, that is left of a scanline s_2, may share scanpoints with s_2.*

The d-trapezoid Q_u is left of *Q_v if $r_u^i < l_v^i$ for all $i \in \{1, 2, \ldots, d\}$. The d-trapezoid Q_v is* between *the scanlines s_1 and s_2 if $s_1^i < l_v^i < r_v^i < s_2^i$ for all $i \in \{1, 2, \ldots, d\}$. The scanline s_1 is* between *Q_u and Q_v if either $r_u^i < s_1^i < l_v^i$, for all $i \in \{1, 2, \ldots, d\}$, or $r_v^i < s_1^i < l_u^i$, for all $i \in \{1, 2, \ldots, d\}$.*

The following results extend corresponding ones for permutation graphs given in [9]. Some ideas of the proof of Theorem 15 will be reused in the proof of our main structural theorem in Section 4.

Definition 16 *Let s be a scanline of a d-trapezoid diagram $\mathcal{D}(G)$. We denote by $S(s)$ the set of those vertices v of G for which s has nonempty intersection with the d-trapezoid Q_v in $\mathcal{D}(G)$.*

Theorem 15 *Let G be a d-trapezoid graph and let $\mathcal{D}(G)$ be a d-trapezoid diagram of G. For every minimal x, y-separator S of G there is a scanline s of $\mathcal{D}(G)$, which is between the d-trapezoids Q_x and Q_y, such that $S = S(s)$.*

Proof. Let S be a minimal x, y-separator of G. Consider the d-trapezoid diagram $\mathcal{D}(G[V \setminus S])$, obtained from $\mathcal{D}(G)$ by removing all d-trapezoids Q_v with $v \in S$.

For each connected component C of $G[V \setminus S]$, there is a *generalized d-trapezoid* Q_C in $\mathcal{D}(G[V \setminus S])$ defined by $l^i_C := \min_{z \in C} l^i_z$ and $r^i_C := \max_{z \in C} r^i_z$, for all $i \in \{1, 2, \ldots, d\}$. The generalized d-trapezoid contains all d-trapezoids Q_z for which z is a vertex of C and Q_C has empty intersection with Q_w for all $w \in V \setminus (S \cup C)$. Hence Q_C and $Q_{C'}$ have empty intersection for any pair C and C' of different components of $G[V \setminus S]$. By the construction of the scanpoints in $\mathcal{D}(G)$, this implies that for any two different components of $G[V \setminus S]$, there is a scanline s in $\mathcal{D}(G)$ between the generalized d-trapezoids Q_C and $Q_{C'}$.

Now let C_x and C_y be the components of $G[V \setminus S]$ containing x and y respectively. Without loss of generality we may assume that Q_{C_x} is left of Q_{C_y} in $\mathcal{D}(G[V \setminus S])$. We can choose a scanline s between Q_{C_x} and Q_{C_y} in $\mathcal{D}(G[V \setminus S])$ such that s does not intersect any generalized d-trapezoid Q_C of a component C of $G[V \setminus S]$. Hence for all d-trapezoids Q_v that intersect the scanline s in $\mathcal{D}(G)$, v must be a vertex of S. This implies that $S(s)$, i.e. the set of those vertices v of G for which s has nonempty intersection with Q_v in $\mathcal{D}(G)$, is a subset of S. On the other hand, every vertex of S is adjacent to some vertex in C_x and to some vertex in C_y by Lemma 4. Hence Q_v intersects the scanline s for every vertex $v \in S$, since Q_v intersects a d-trapezoid, which is left of s, and a d-trapezoid, which is right of s. Hence $S = S(s)$. □

Corollary 16 *The number of minimal separators of a d-trapezoid graph G on $n \geq 2$ vertices is at most $(2n-3)^d$.*

To avoid confusion let us emphasize that a scanline depends on a d-trapezoid diagram. When we study scanlines we always assume that a fixed d-trapezoid diagram of the graph under consideration is given.

Throughout the rest of the paper we assume that $G = (V, E)$ is a d-trapezoid graph and that $\mathcal{D}(G)$ is a fixed d-trapezoid diagram of G.

4 Minimal triangulations of d-trapezoid graphs

The main structural result of this paper is a representation theorem for minimal triangulations of a d-trapezoid graph G in terms of scanlines of its d-trapezoid diagram $\mathcal{D}(G)$. The approach taken to prove this theorem differs completely from the techniques used to prove related theorems in previous work as e.g. [9, 20, 24].

Definition 17 *Let $\mathcal{D}(G)$ be a d-trapezoid diagram of a graph G. A sequence of scanlines $s_0, s_1, s_2, \ldots, s_{r-1}, s_r$ of $\mathcal{D}(G)$ is* non-crossing *if s_i is left of s_{i+1} for each $i \in \{0, 1, \ldots, r-1\}$.*

We need some additional notations. Consider a d-trapezoid diagram $\mathcal{D}(G)$. We denote by s_{L} the leftmost scanline in $\mathcal{D}(G)$, i.e., $s_{\mathrm{L}} = (0.5, 0.5, \ldots, 0.5)$, and by

s_{R} the rightmost scanline in $\mathcal{D}(G)$, i.e., $s_{\mathrm{R}} = (2n+0.5,\ldots,2n+0.5)$. For a non-crossing sequence of scanlines $s_0, s_1, s_2, \ldots, s_{r-1}, s_r$ of $\mathcal{D}(G)$, we denote by $G_{s_0,s_1,\ldots,s_r}$ the graph obtained from G by adding an edge between any pair of nonadjacent vertices u and v of G for which there is an $i \in \{0,1,\ldots,r\}$ such that Q_u and Q_v intersect s_i in $\mathcal{D}(G)$.

Definition 18 *Let s and s' be scanlines of $\mathcal{D}(G)$ such that s is left of s'. Then $\mathcal{V}(s,s')$ is the set of all vertices v for which Q_v is either between the scanlines s and s' or has a nonempty intersection with s or with s' in $\mathcal{D}(G)$.*

Theorem 17 *Let $G = (V,E)$ be a d-trapezoid graph and let $\mathcal{D}(G)$ be a d-trapezoid diagram of G. Let H be any minimal triangulation of G and let $\mathbf{Sep}(H)$ be the set of all minimal separators of H. Then there is a non-crossing sequence of scanlines $s_{\mathrm{L}} = s_0, s_1, s_2, \ldots, s_{r-1}, s_r = s_{\mathrm{R}}$ of $\mathcal{D}(G)$ for which*

(i) *$\mathcal{V}(s_0,s_1), \mathcal{V}(s_1,s_2), \ldots, \mathcal{V}(s_{r-1},s_r)$ is a consecutive clique arrangement of the interval graph H,*

(ii) *$\mathbf{Sep}(H) = \{S(s_1), S(s_2), \ldots, S(s_{r-1})\}$, and*

(iii) *$H = G_{s_0,s_1,\ldots,s_r}$.*

Proof. The main property is (i) which easily implies the other two. To see this note that (i) implies (ii) by Lemma 5. Furthermore Theorem 7 and property (ii) imply $H = G_{\{S(s_1),S(s_2),\ldots,S(s_{r-1})\}}$. Thus the definition of $G_{s_0,s_1,\ldots,s_r}$ immediately implies (iii). Hence it suffices to prove property (i).

Let H be any minimal triangulation of the d-trapezoid graph G with d-trapezoid diagram $\mathcal{D}(G)$. By Corollary 10, H is an interval graph, and thus chordal. Let $\tilde{H}$ be the subgraph of H induced by the set of all simplicial vertices of H. Then all components of $\tilde{H}$ are complete and we call them simplicial cliques of H. Each simplicial clique A of H corresponds to a generalized d-trapezoid Q_A of $\mathcal{D}(G)$ with $l_A^i := \min_{z\in A} l_z^i$ and $r_A^i := \max_{z\in A} r_z^i$, for all $i \in \{1,2,\ldots,d\}$. Furthermore, all these generalized d-trapezoid Q_A have pairwise empty intersection (see the proof of Theorem 15). Let Q_{A_1} be the leftmost of all generalized d-trapezoids Q_A of the simplicial cliques A of H in the d-trapezoid diagram obtained from $\mathcal{D}(G)$ by removing all d-trapezoids Q_v for which v is not simplicial in H.

Claim 1: *There is no d-trapezoid Q_u and no $a_1 \in A_1$ such that Q_u is left of Q_{a_1} in $\mathcal{D}(G)$.*
Suppose not. Let $a_1 \in A_1$ and let Q_u be a d-trapezoid that is left of Q_{a_1} in $\mathcal{D}(G)$. Then u has no neighbor in A_1 since A_1 is a simplicial clique of H. Furthermore $S^* = N_H[A_1] \setminus A_1$ is a minimal separator of H and A_1 is a component of $H[V \setminus S^*]$, since A_1 is a simplicial clique of H. (Here we denote by $N_G[V']$ the set of all vertices of the graph $G=(V,E)$ that either belong to $V' \subseteq V$ or have a neighbor in V'.)

Let x be a vertex of H such that S^* is a minimal x, a_1-separator of H. By Theorem 8, S^* is also a minimal x, a_1-separator of G and A_1 is a component of $G[V \setminus S^*]$. Consider the d-trapezoid diagram $\mathcal{D}(G[V \setminus S^*])$. Recall that A_1 is a component of $G[V \setminus S^*]$. Notice that $u \notin S^*$ since $u \notin N_H[A_1]$. Let C_u be the component of $G[V \setminus S^*]$ containing u. Then the generalized d-trapezoid Q_{C_u} is left of Q_{A_1} in $\mathcal{D}(G[V \setminus S^*])$ and thus in $\mathcal{D}(G)$, since Q_u is left of Q_{a_1} in $\mathcal{D}(G)$. By Theorem 8, C_u is also a component of $H[V \setminus S^*]$. Hence, Theorem 12 implies that C_u contains a simplicial vertex of H and this contradicts the choice of A_1. This proves the claim.

Now take as s_1 the scanline of $\mathcal{D}(G)$ with $s_1^i = r_{A_1}^i + 0.5$ for all $i \in \{1, 2, \ldots, d\}$.

Claim 2: *$\mathcal{V}(s_0, s_1)$ is a maximal clique in H.*
$N_H[A_1]$ is a maximal clique in H. Furthermore $N_G[A_1] = A_1 \cup S(s_1)$ by the choice of s_1. Hence $\mathcal{V}(s_0, s_1) = N_G[A_1]$. Finally Lemma 13 implies $N_G[A_1] = N_H[A_1]$ and therefore $\mathcal{V}(s_0, s_1) = N_H[A_1]$ is a maximal clique in H. This proves the claim.

Now assume that we have constructed a non-crossing sequence of scanlines $s_{\mathrm{L}} = s_0, s_1, s_2, \ldots, s_{j-1}$, $j \geq 2$, of $\mathcal{D}(G)$ such that $\mathcal{V}(s_0, s_1), \mathcal{V}(s_1, s_2), \ldots, \mathcal{V}(s_{j-2}, s_{j-1})$ are maximal cliques of H. We show how to find s_j.

Let H_j be the subgraph of H induced by the vertex set V_j that consists of all those vertices v for which the d-trapezoid Q_v is not left of s_{j-1} in $\mathcal{D}(G)$. Clearly H_j is an interval graph.

Claim 3: Every minimal separator of H_j is also a minimal separator of H.
Let S be a minimal separator of the interval graph H_j. By assumption, $\mathcal{V}(s_{j-2}, s_{j-1})$ is a maximal clique of H. Hence $S_{j-1} = S(s_{j-1})$ is a clique. The construction of H_j implies $N_H[v] \cap V_j \subseteq S_{j-1}$ for all $v \in V \setminus V_j$. Hence it is impossible that there are two components of $H_j[V_j \setminus S]$ that both contain vertices of S_{j-1}. Imagine we take the components of $H_j[V_j \setminus S]$ and we add all vertices of $V \setminus V_j$. Then we obtain the components of $H[V \setminus S]$. Now in $H[V \setminus S]$ all vertices of $V \setminus V_j$ either form a collection of new components (if $S_{j-1} \subseteq S$) or they will be added to the unique component of $H_j[V_j \setminus S]$ containing vertices of S_{j-1}. In any case at most one component of $H_j[V_j \setminus S]$ will be changed and that one will only be enlarged.

Suppose S is a minimal a, b-separator of H_j. By Lemma 4, every vertex of S has a neighbor in the components C_a and C_b of $H_j[V_j \setminus S]$. When adding the vertices of $V \setminus V_j$ to $H_j[V_j \setminus S]$ this does not change, i.e. every vertex of S has a neighbor in the components C_a and C_b of $H[V \setminus S]$. Hence S is a minimal a, b-separator of H by Lemma 4. This proves the claim.

Suppose H_j is not complete. Let $\tilde{H}_j$ be the subgraph of H_j induced by the set of all simplicial vertices of H_j. The components of $\tilde{H}_j$ are the simplicial cliques of H_j and each simplicial clique A of H_j corresponds to a generalized d-trapezoid

Q_A of $\mathcal{D}(G)$. Let Q_{A_j} be the leftmost of all the generalized d-trapezoids Q_A of a simplicial clique A of H_j in the d-trapezoid diagram obtained from $\mathcal{D}(G)$ by removing all d-trapezoids Q_v for which v is not simplicial in H_j.

Claim 4: *There is no $u \in V_j$ and no $a_j \in A_j$ such that the d-trapezoid Q_u is left of Q_{a_j} in $\mathcal{D}(G)$.*
This can be obtained by applying Claim 1 to H_j and A_j.

Now take as s_j the scanline of $\mathcal{D}(G)$ with $s_j^i = r_{A_j}^i + 0.5$ for all $i \in \{1, 2, \ldots, d\}$. Notice that our construction ensures that s_{j-1} is left of s_j and recall that S_{j-1} is a clique of H.

Claim 5: *$\mathcal{V}(s_{j-1}, s_j)$ is a maximal clique in H.*
Since A_j is a simplicial clique of H_j, we obtain $N_{H_j}[a_j] = N_{H_j}[A_j]$ for all $a_j \in A_j$. By construction of s_j, every vertex of $S_j = S(s_j)$ has a neighbor in A_j. Hence $S_j \subseteq N_{H_j}[A_j]$. Furthermore by Claim 4, every vertex of $S_{j-1} = S(s_{j-1})$ has a neighbor in A_j. Consequently $N_{H_j}[A_j] = S_{j-1} \cup A_j \cup S_j = \mathcal{V}(s_{j-1}, s_j)$ is a clique in H_j. Furthermore $\mathcal{V}(s_{j-1}, s_j)$ is a maximal clique in H since $N_{H_j}[A_j] = N_H[A_j]$. This proves the claim.

Finally consider the case that H_j is complete. Then we finish the construction by taking as s_j the scanline s_{R} of $\mathcal{D}(G)$. Clearly $\mathcal{V}(s_{j-1}, s_j)$ is a maximal clique of H.

By now we have shown how to construct a non-crossing sequence of scanlines $s_{\mathrm{L}} = s_0, s_1, s_2, \ldots, s_{r-1}, s_r = s_{\mathrm{R}}$ such that $\mathcal{V}(s_0, s_1), \mathcal{V}(s_1, s_2), \ldots, \mathcal{V}(s_{r-1}, s_r)$ contains all maximal cliques of H. Clearly our construction guarantees that no maximal clique appears twice. Furthermore the definition of a d-trapezoid diagram and the definition of the sets $\mathcal{V}(s, s')$ imply that for every vertex of H the maximal cliques containing it occur consecutively. Consequently $\mathcal{V}(s_0, s_1), \mathcal{V}(s_1, s_2), \ldots, \mathcal{V}(s_{r-1}, s_r)$ is a consecutive clique arrangement of the interval graph H. □

Our algorithms are based on the above representation theorem for minimal triangulations of d-trapezoid graphs. In fact algorithms obtained from Theorem 17 in a straightforward manner (by dynamic programming) would already run in polynomial time (roughly $O(n^{3d})$). However the tools that we develop in the next section allow us to exploit the information of the d-trapezoid diagram more cleverly. As a consequence we obtain significantly faster algorithms.

5 Small scanlines and dense sequences

The notion of a small scanline has been introduced in [9]. 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 $\omega(H) \le k$.

Definition 19 *Let $\mathcal{D}(G)$ be a d-trapezoid diagram. A scanline s of $\mathcal{D}(G)$ is* k-small *if it intersects with at most $k+1$ d-trapezoids.*

Lemma 18 *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 $|s^i - s^j| \leq 2(k+1)$.*

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.

d-Trapezoids do not have common endpoints in the diagram. Thus 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 $1/2\,|s^i-s^j| \leq k+1$ for each $i,j \in \{1,2,\ldots d\}$. Hence there are $O(nk^{d-1})$ k-small scanlines. □

Definition 20 *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. $t^j = s^j + 1$).*

A non-crossing 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 $i \in \{0,1,\ldots,r-1\}$.*

The following definition is needed to describe the algorithm to compute the minimum fill-in of d-trapezoid graphs.

Definition 21 *Let $\mathcal{D}(G)$ be a d-trapezoid diagram of a graph $G=(V,E)$. Let $s_{\mathrm{L}} = s_0, s_1, s_2, \ldots, s_{r-1}, s_r = s_{\mathrm{R}}$ be a non-crossing sequence of scanlines of $\mathcal{D}(G)$. Then for all $i \in \{1,2,\ldots,r-1\}$, we denote by first(s_i) the set of those pairs $\{u,v\}$ of nonadjacent vertices of G for which s_i is the leftmost scanline of the sequence that intersects both Q_u and Q_v.*

The following theorem justifies the correctness of our algorithms.

Theorem 19 *Let $G=(V,E)$ be a d-trapezoid graph with a d-trapezoid diagram $\mathcal{D}(G)$. Then the following statements hold:*

(i) $tw(G) = pw(G) = \min\Big\{ \max_{i\in\{0,1,\ldots,r-1\}} |\mathcal{V}(s_i,s_{i+1})| - 1 : s_{\mathrm{L}} = s_0, s_1, s_2, \ldots,$ $s_{r-1}, s_r = s_{\mathrm{R}}$ *non-crossing sequence of scanlines of* $\mathcal{D}(G)\Big\}$.

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

(iii) $mfi(G) = mic(G) = \min\Big\{ \sum_{i=1}^{r-1} |first(s_i)| : s_{\mathrm{L}} = s_0, s_1, s_2, \ldots, s_{r-1}, s_r = s_{\mathrm{R}}$ *dense sequence of scanlines of* $\mathcal{D}(G) \Big\}$.

Proof. First note that $tw(G) = pw(G)$ and $mfi(G) = mic(G)$ for every d-trapezoid graph G by Corollary 10.

Consider (i). Let $s_{\mathrm{L}} = s_0, s_1, s_2, \ldots, s_{r-1}, s_r = s_{\mathrm{R}}$ be any non-crossing sequence of scanlines of $\mathcal{D}(G)$. 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. By the definition of $\mathcal{V}(s, s')$ and the convexity property, $v \in \mathcal{V}(s_i, s_{i+1})$ and $v \in \mathcal{V}(s_k, s_{k+1})$ implies $v \in \mathcal{V}(s_j, s_{j+1})$ for all $j \in \{i+1, i+2, \ldots, k-1\}$. Therefore each vertex $v \in V$ appears in the subsets $\mathcal{V}(s_i, s_{i+1}), \mathcal{V}(s_{i+1}, s_{i+2}), \ldots, \mathcal{V}(s_{k-1}, s_k)$ for some i, k with $0 \leq i \leq k \leq r$. Thus $\mathcal{V}(s_0, s_1), \mathcal{V}(s_1, s_2), \ldots, \mathcal{V}(s_{r-1}, s_r)$ is a path-decomposition of G. Consequently

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

Now let H be a minimal triangulation of G with $\omega(H) - 1 = tw(G)$. By Theorem 17, there is a non-crossing sequence of scanlines $s_{\mathrm{L}} = s_0, s_1, s_2, \ldots, s_{r-1}, s_r = s_{\mathrm{R}}$ of $\mathcal{D}(G)$ such that $H = G_{s_0, s_1, \ldots, s_r}$ and $\mathcal{V}(s_0, s_1), \mathcal{V}(s_1, s_2), \ldots, \mathcal{V}(s_{r-1}, s_r)$ is a consecutive clique arrangement of the interval graph H. Thus $pw(G) = tw(G) = \omega(H) - 1 = \max_{0 \leq i \leq r-1} (|\mathcal{V}(s_i, s_{i+1})| - 1)$. This completes the proof of (i).

Consider (ii). Let H be a minimal triangulation of G with $\omega(H) - 1 = tw(G)$ and let $s_{\mathrm{L}} = s_0, s_1, s_2, \ldots, s_{r-1}, s_r = s_{\mathrm{R}}$ be a non-crossing sequence of scanlines of $\mathcal{D}(G)$ for which $\mathcal{V}(s_0, s_1), \mathcal{V}(s_1, s_2), \ldots, \mathcal{V}(s_{r-1}, s_r)$ is a consecutive clique arrangement of H. Then each scanline s_i, $i \in \{1, 2, \ldots, r-1\}$, is k-small, since $S(s_i) \subseteq \mathcal{V}(s_i, s_{i+1})$ implies $|S(s_i)| \leq |\mathcal{V}(s_i, s_{i+1})| \leq tw(G) + 1 \leq k + 1$. (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 non-crossing sequence of k-small scanlines.

Recall that each set $\mathcal{V}(s_i, s_{i+1})$, $i \in \{0, 1, \ldots, r-1\}$, is a clique in H. 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 $\mathcal{V}(s_i, s_{i+1})$, thus they are adjacent in H. Consequently our particular non-crossing 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 $|S(s^*_{il})| \leq |\mathcal{V}(s_i, s_{i+1})| \leq k + 1$. Hence each scanline s^*_{il} 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 $j \in \{0, 1, \ldots, q-1\}$, $\mathcal{V}(\hat{s}_j, \hat{s}_{j+1}) \subseteq \mathcal{V}(s_i, s_{i+1})$ for some $i \in \{0, 1, \ldots, r-1\}$ by the construction of the new sequence, implying $|\mathcal{V}(\hat{s}_j, \hat{s}_{j+1})| \leq k+1$. This completes the proof of (ii).

Consider (iii). Let $s_{\mathrm{L}} = s_0, s_1, s_2, \ldots, s_{r-1}, s_r = s_{\mathrm{R}}$ be any dense sequence of scanlines of $\mathcal{D}(G)$. Hence s_i is a predecessor of s_{i+1} for all $i \in \{0, 1, \ldots, r-1\}$. Therefore for all $u, v \in \mathcal{V}(s_j, s_{j+1})$ with u and v nonadjacent in G, the d-trapezoids Q_u and Q_v either both intersect s_j or both intersect s_{j+1}. Consequently there is a unique leftmost scanline s_i in $s_{\mathrm{L}} = s_0, s_1, s_2, \ldots, s_{r-1}, s_r = s_{\mathrm{R}}$ that intersects both Q_u and Q_v, i.e. $\{u, v\} \in \mathit{first}(s_i)$ for exactly one $i \in \{0, 1, \ldots, r-1\}$.

Consider the graph $H = G_{s_0, s_1, \ldots, s_r}$. We have seen that $\mathcal{V}(s_i, s_{i+1})$ is a clique of $H = G_{s_0, s_1, \ldots, s_r}$ for each $i \in \{0, 1, \ldots, r-1\}$. Recall that $\mathcal{V}(s_0, s_1), \mathcal{V}(s_1, s_2), \ldots, \mathcal{V}(s_{r-1}, s_r)$ is a path-decomposition of G. Hence $H = G_{s_0, s_1, \ldots, s_r}$ is a triangulation of G into an interval graph by Lemma 3 and $|E(H)| - |E(G)| = \sum_{i=1}^{r-1} |\mathit{first}(s_i)|$. Consequently $\mathit{mic}(G) \leq \sum_{i=1}^{r-1} |\mathit{first}(s_i)|$.

Now let H be a minimal triangulation of the graph G such that $\mathit{mic}(G) = \mathit{mfi}(G) = |E(H)| - |E(G)|$. By Theorem 17, there is a non-crossing sequence of scanlines $s_{\mathrm{L}} = s_0, s_1, s_2, \ldots, s_{r-1}, s_r = s_{\mathrm{R}}$ of $\mathcal{D}(G)$ such that $H = G_{s_0, s_1, \ldots, s_r}$ and $\mathcal{V}(s_0, s_1), \mathcal{V}(s_1, s_2), \ldots, \mathcal{V}(s_{r-1}, s_r)$ is a consecutive clique arrangement of the interval graph H. Hence $\mathit{mic}(G) = \mathit{mfi}(G) = \sum_{i=1}^{r-1} |\mathit{first}(s_i)|$.

Analogously to (ii), the non-crossing sequence $s_{\mathrm{L}} = s_0, s_1, s_2, \ldots, s_{r-1}, s_r = s_{\mathrm{R}}$ can be transformed into a dense sequence of scanlines $s_{\mathrm{L}} = \hat{s}_0, \hat{s}_1, \hat{s}_2, \ldots, \hat{s}_{p-1}, \hat{s}_p = s_{\mathrm{R}}$ of $\mathcal{D}(G)$ satisfying $\mathit{mic}(G) = \mathit{mfi}(G) = \sum_{i=1}^{p-1} |\mathit{first}(\hat{s}_i)|$. □

6 Algorithms

In this section we present our two polynomial time algorithms to compute 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 to compute 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
- $|\mathcal{V}(s,t)| \leq k+1$.

The next lemma follows immediately from Theorem 19 (i) and (ii).

Lemma 20 *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 21 *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 18. 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 $|\mathcal{V}(s,t)| \leq k+1$.

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 22 *The procedure can be implemented to run in time $O(nk^{d-1})$.*

Proof. We describe any scanline s by the vector $(s^1, s^2, \ldots, s^d)$ of its scanpoints. 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 fulfill this condition by Lemma 18.

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 $|\mathcal{V}(s,t)| \leq k+1$ 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 23 *For each positive integer d, there is an $O(n\, tw(G)^{d-1})$ algorithm to compute 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 $k = 1, 2, 4, 8, \ldots$ 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 (s, t), saying how many vertices are in the corresponding vertex set $\mathcal{V}(s, t)$. 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 $\mathcal{D}(G)$. Our algorithm 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 24 *mfi(G) is equal to the length of a shortest directed path from* s_{L} *to* s_{R} *in* $\widetilde{W}(G)$.

Proof. By Theorem 19 (iii), *mfi*(G) is equal to the minimum of $\sum_{i=1}^{r-1} |\mathit{first}(s_i)|$, where $s_{\mathrm{L}} = s_0, s_1, s_2, \ldots, s_{r-1}, s_r = s_{\mathrm{R}}$ ranges over all dense sequences of scanlines of $\mathcal{D}(G)$. The shortest paths from s_{L} to s_{R} in $\widetilde{W}(G)$ are in one-to-one correspondence to the dense sequences of scanlines of $\mathcal{D}(G)$. Finally the length of an arc from s to t is defined such that it is equal to $|\mathit{first}(t)|$ 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 21 one obtains the following.

Lemma 25 *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 lengths 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 26 *For each positive integer* d*, there is an* $O(n^d)$ *algorithm to compute 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 scanpoints. 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 16.

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 $O(n^d)$. □

It is worth mentioning that the order of magnitude of the running time of our algorithm is equal to the order of magnitude of the number of scanlines in a d-trapezoid diagram. Hence improving our algorithm seems to require completely new ideas.

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 tree-width, *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, *SIAM Journal on Computing* **25**, 1305 – 1317, 1996.

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

[8] H.L. Bodlaender and T. Kloks, Efficient and constructive algorithms for the path-width and treewidth of graphs, *Journal of Algorithms* **21**, 358 – 402, 1996.

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

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

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

[12] M.S. Chang, Algorithms for maximum matching and minimum fill-in on chordal bipartite graphs, *Proceedings of the 7th International Symposium Algorithms and Computation*, 146 – 155, Springer-Verlag, Lecture Notes in Computer Science 1178, 1996.

[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, *SIAM Journal on Discrete Mathematics* **10**, 399 – 430, 1997.

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

[16] Flotow, C., On powers of m-trapezoid graphs, *Discrete Applied Mathematics* **63**, 187 – 195, 1995.

[17] 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.

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

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

[20] T. Kloks, Treewidth of circle graphs, *International Journal of Foundations of Computer Science* **7**, 111 – 120, 1996.

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

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

[23] T. Kloks, D. Kratsch and J. Spinrad, On treewidth and minimum fill-in of asteroidal triple-free graphs, *Theoretical Computer Science* **175**, 309 – 335, 1997.

[24] 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.

[25] T.-H. Ma and J.P. Spinrad, On the 2-chain subgraph cover and related problems, *Journal of Algorithms* **17**, 251 – 268, 1994.

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

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

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

[29] 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.

[30] 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.

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

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

[33] R.E. Tarjan, Decomposition by clique separators, *Discrete Mathematics* **55**, 221–232, 1985.

[34] W.T. Trotter, *Combinatorics and Partially Ordered Sets: Dimension Theory*, The John Hopkins University Press, Baltimore, Maryland, 1992.

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

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