The structure of separator graphs and efficient algorithms for minimal triangulations problems of permutation graphs

Daniel Meister

Report No. 361 May 2005

Preprint-Reihe
Fakultät für Mathematik und Informatik
Universität Würzburg

The structure of separator graphs and efficient algorithms for minimal triangulations problems of permutation graphs

Daniel Meister
Theoretische Informatik
Bayerische Julius-Maximilians-Universität Würzburg
97074 Würzburg, Germany
meister@informatik.uni-wuerzburg.de

Abstract. A chordal graph H is a triangulation of a graph G, if G is a spanning subgraph of H. H is a minimal triangulation of G, if no proper subgraph of H is a triangulation of G. Parra and Scheffler characterised minimal triangulations of a graph G by maximal independent sets of its separator graph. The separator graph of G has a vertex for every minimal separator of G, and two vertices of the separator graph are adjacent if and only if the corresponding minimal separators are non-crossing. We will show that non-crossing of minimal separators of permutation graphs can easily be identified and, as a corollary, that separator graphs of permutation graphs are permutation graphs. We will characterise the minimal separators of a permutation graph using its potential maximal cliques graph, that is introduced here. This graph can be computed in linear time. Using this graph, we can solve a number of problems for permutation graphs related to minimal triangulations in linear time such as treewidth and minimum fill-in.

1. Introduction. When Bodlaender, Kloks, Kratsch investigated the computation of treewidth and pathwidth of permutation graphs in 1995 [3], it was the last time that permutation graphs were in the focus of considerations about minimal triangulations. Actually, the results are some years older. Later, Bodlaender, Kloks, Kratsch, Müller extended the results and techniques of [3] and applied them to d-trapezoid graphs (a subclass of co-comparability graphs) [4]. So, a decade passed before minimal triangulations of permutation graphs became the central part of further investigations.

A minimal triangulation of a graph is the result of adding an inclusion-minimal set of edges to it to make it chordal. Minimal triangulations were first investigated by Rose, Tarjan, Lueker in connection with efficient Gaussian matrix elimination [22]. Their original problem was Minimum Fill-in: Given a graph, find the least number of edges that have to be added to obtain a chordal graph. Since the generated graph is a minimal triangulation of the given graph and the decision variant of Minimum Fill-in was expected to be NP-complete, minimal triangulations were studied instead. Five years later, Yannakakis actually showed the NP-completeness of the problem: Given a co-bipartite graph G and a natural number k, do k edges suffice to make G chordal [23]? So, Minimum Fill-in is hard even on co-comparability and therefore AT-free graphs.

Another problem that is closely related to minimal triangulations is the treewidth problem: Given a graph, determine the least clique number among its minimal triangulations. This problem setting is equivalent to asking for the smallest k such that the given graph is a partial k-tree. Motivation for this problem comes from algorithms, since for many problems there exist efficient, even linear-time algorithms on partial k-trees, where the running time depends also on k. However, Arnborg, Corneil, Proskurowski showed that the decision variant of Treewidth is NP-complete even on co-bipartite graphs [1].

During the 1980's, triangulation problems have been considered only seldomly. However, the last decade of the 20th century was very fertile in this area. Kloks, Bodlaender, Müller, Kratsch initiated the study of the connection between minimal triangulations and minimal separators [13], [14], and Parra, Scheffler and Kloks, Kratsch, Spinrad brought this general study to a first end [20], [17].

Motivated by a number of practical problems, several authors studied minimal triangulations of special graph classes. It was shown that minimal triangulations of cographs, i.e., P_4-free graphs, are cographs [5], by a sequel of papers that minimal triangulations of permutation, co-comparability and finally AT-free graphs are interval graphs [3], [12], [19]. By extending the notion of an asteroidal triple to arbitrary vertex sets, Kloks, Kratsch, Müller defined the asteroidal number of a graph. They showed that the asteroidal number of any minimal triangulation of a graph does not exceed its asteroidal number [16]. Now, consider the following graph classes, that are ordered by inclusion: P_3-free graph, P_4-free graphs, trapezoid graphs, co-comparability graphs, AT-free graphs, graphs with asteroidal number $k \geq 3$. By the results cited above every of the enumerated graph classes contains its minimal triangulations. The class of permutation graphs fits well between the P_4-free and trapezoid graphs, but it does not contain the set of its minimal triangulations. From a theoretical point of view, this property distiguishes the class of permutation graphs from the other mentioned graph classes and makes it a wunderful object of further investigations.

In this paper, we will study minimal separators of permutation graphs and its relations. Applying the idea of scanlines introduced by Bodlaender, Kloks, Kratsch we will show that the number of minimal separators of a permutation graph is linearly bounded above by the graph, and we will give a linear-time algorithm for enumerating all minimal separators. We will see that the separator graph of a permutation graph can also be generated in linear time. The separator graph was introduced by Parra and Scheffler to characterise the minimal triangulations of a graph [20]. However, the separator graph seems to be of great value only for theoretical considerations. Potential maximal cliques of graphs were introduced by Bouchitté and Todinca and are highly interesting in connection with algorithmic aspects [6]. A *potential maximal clique* of a graph is a set of vertices that can become a maximal clique in a minimal triangulation of the graph. We will identify the potential maximal cliques of permutation graphs, show that there are not too much of them and give a linear-time algorithm for generating them. We will define the *potential maximal cliques graph* of a permutation graph and show how this graph will help in solving problems about minimal triangulations of permutation graphs like Treewidth and Minimum Fill-in for permutation graphs.

This paper is organized as follows. In Section 3, we will consider minimal separators of permutation graphs. We will see that crossing of minimal separators directly translates into a geometric property verifiable in the permutation diagram. In Section 4, we will characterise potential maximal cliques of permutation graphs and define the potential maximal cliques graph of a permutation graph. In Section 5, we will consider minimal triangulations of permutation graphs and show a correspondence between the maximal paths of the potential maximal cliques graph and the minimal triangulations of a permutation graph. We will also give $\mathcal{O}(n)$-time algorithms for computing any minimal triangulation and some minimal triangulation of a given permutation graph represented by its permutation diagram. Finally, in Section 6, we will present linear-time algorithms for computing the treewidth and the minimum fill-in of permutation graphs and consider a number of further problems about minimal triangulations.

2. Preliminaries. We will consider only simple and finite graphs that may be directed or undirected. For a directed graph $G = (V, A)$, V denotes the set of vertices of G and A denotes the set of arcs. Arcs are denoted as uv, which means that u is start vertex and v is end vertex. For an undirected graph $G = (V, E)$, V and E denote the sets of vertices and edges, respectively. Since we mostly deal with undirected graphs, most of the following definitions are for undirected graphs. An edge is denoted as uv, which means that vertices u and v are *adjacent*. Two edges uv and $u'v'$ are adjacent if $u \in \{u', v'\}$ or $v \in \{u', v'\}$. A vertex is *isolated* if it is not adjacent to any other vertex. For $S \subseteq V$, the subgraph of G *induced* by S, denoted as $G[S]$, is the graph containing only the vertices of S, and two vertices are adjacent in $G[S]$ if and only if they are adjacent in G; we write $G \setminus S$ instead of $G[V \setminus S]$. Adding a set F of edges to G will be denoted by $G \cup F$, which is short for $(V, E \cup F)$; deleting an edge $e \in E$ will be denoted by $G-e$ instead of $(V, E \setminus \{e\})$; $G \setminus F$ for $F \subseteq E$ is short for $(V, E \setminus F)$. The *neighbourhood* of a vertex $u \in V$, denoted by $N_G(u)$, is the set of all vertices adjacent to u in G; $N_G[u] =_{\text{def}} N_G(u) \cup \{u\}$ denotes the *closed* neighbourhood of u. For $U \subseteq V$, the neighbourhood $N_G(U)$ of U is the union of the neighbourhoods of the vertices in U outside U, i.e., $N_G(U) =_{\text{def}} \bigcup_{v \in U} N_G(v) \setminus U$; $N_G[U] =_{\text{def}} N_G(U) \cup U$. For further definitions, we also refer to [11]. A *clique* in G is a set of pairwise adjacent vertices of G, whereas an *independent set* in G is a set of pairwise non-adjacent vertices. By $\omega(G)$ and $\alpha(G)$, we denote the numbers of vertices of a maximum clique and independent set of G, respectively.

A *path* of length k in an undirected graph $G = (V, E)$ is a sequence $(x_0, \ldots, x_k)$ of $k+1$ different vertices of G where x_{i-1} and x_i are adjacent for every $i \in \{1, \ldots, k\}$. Similarly, a path of length k in a directed graph $G' = (V, A)$ is a sequence $(x_0, \ldots, x_k)$ of $k+1$ different vertices of G' where $x_{i-1}x_i \in A$ for every $i \in \{1, \ldots, k\}$. G is *connected* if there is a path between any pair of vertices of G. A *connected component* of G is a maximal connected induced subgraph of G. A path $(x_1, \ldots, x_k)$ of G is a *cycle* of length k if $x_k x_1 \in E$; the definition for G' is similar. G' is *acyclic* if G' does not contain a cylce. A *topological ordering* for G is an ordering $\langle x_{\pi(i)}, \ldots, x_{\pi(n)} \rangle$ of the vertices of G' such that $uv \in A$ implies $i < j$ where $u = x_{\pi(i)}$ and $v = x_{\pi(j)}$. It holds that G' is acyclic if and only if there is a topological ordering for G'. A *chord* in a cycle C of G is an edge of G between two non-consecutive vertices in C; a chord is *unique* in C if it is the only edge in G that is a chord in C. C is *chordless* if there is no edge in G that is a chord in C. G is *chordal* if there is no chordless cycle of length at least 4 in G. An *interval graph* on n vertices is the intersection graph of a family of n closed intervals of the real line. Interval graphs are chordal. A triple u, v, w of pairwise independent vertices of G is an *asteroidal triple*, *AT* for short, if there is a path between any two of them not containing a neighbour of the third. G is *AT-free* if G does not contain three vertices that form an AT. Structural and algorithmic properties of AT-free graphs have been studied extensively in [7] and [8]. Interval graphs are exactly the chordal graphs that are AT-free [18].

Permutation graphs. For $n \geq 1$, let $[n]$ denote the set of the numbers $1, 2, \ldots, n$. Let $\pi : [n] \to [n]$ be a bijection. We also say that π is a *permutation sequence over* $[n]$. The graph $G(\pi)$ has vertex set $[n]$ and two vertices $u, v \in [n]$, $u \neq v$, are adjacent if and only if $(u-v)(\pi^{-1}(u)-\pi^{-1}(v)) < 0$. A *permutation graph over* $[n]$ is a graph $G(\pi)$ for some permutation sequence π over $[n]$. A graph G is a *permutation graph* if there is $n \geq 1$ such that G is isomorphic to a permutation graph over $[n]$. Permutation graphs can be represented by *permutation diagrams.* Let $G = G(\pi)$ be a permutation graph over $[n]$. The permutation diagram for G is defined as follows: for two parallel horizontal lines define n equidistant points on each line. On the upper line, label the points from 1 to n, on the lower line label the points from $\pi(1)$ to $\pi(n)$. Join two points by a line segment if and only if they have the same label. The intersection graph of this diagram is graph G, i.e., the vertices of G correspond with the line segments, and two vertices are adjacent if and only if the corresponding line segments intersect. Since we often use the permutation diagram representation, we identify line segments and the corresponding vertices of the graph. This will never cause confusion, but will make language easier.

Permutation graphs have a lot of characterisations. We relate the reader to Golumbic's book [11]. Without further definintion, we only want to mention that permutation graphs are exactly the graphs that are comparability and co-comparability graphs, in particular, complements of permutation graphs are permutation graphs. Many NP-complete problems become easy on permutation graphs, others remain NP-complete.

We will need special neighbours of vertices. Let $G = G(\pi)$ be a permutation graph over $[n]$. For every $x \in [n]$, the *closest left neighbour of* x is the vertex $\pi(z)$ where z is largest among all $i < \pi^{-1}(x)$ such that $\pi(i) > x$. If there is no such i, x is its own closest left neighbour. The *closest left non-neighbour of* x is the vertex $\pi(z)$ where z is largest among all $i < \pi^{-1}(x)$ such that $\pi(i) < x$. Equally, if there is no such i, x is its own closest left non-neighbour.

Lemma 1. *Let $G = G(\pi)$ be a permutation graph over $[n]$ given by π. The closest left neighbours and non-neighbours can be computed in time $\mathcal{O}(n)$.*

Proof: For computing the closest left neighbours, consider the following algorithm:

(1) initialize a stack with $n+1$

(2) for $i = 1, \ldots, n$: delete all vertices from the stack that are smaller than $\pi(i)$, assign $\pi(i)$ the top-of-stack element as closest left neighbour, push $\pi(i)$ on the stack

(3) for $x \in [n]$: if x has $n+1$ as closest left neighbour, assign x as closest left neighbour to x.

The described algorithm obviously runs in time $\mathcal{O}(n)$. For the correctness, first observe that deleted vertices in step 2 are non-neighbours of the current vertex and that all remaining vertices in the stack (except $n+1$) are left neighbours of the current vertex. It is clear that the top-of-stack element is rightmost in π among all vertices in the stack and is the closest left neighbour.

For computing the closest left non-neighbour, use the same algorithm on permutation graph $G(\pi')$ where $\pi' =_{\text{def}} n+1-\pi$. The position of the closest left neighbour of $\pi'(i)$ is the position of the closest left non-neighbour of $\pi(i)$. ■

Separators and triangulations. Let $G = (V, E)$ be a graph. A graph $H = (W, F)$ is a *triangulation* of G if H is chordal, $V = W$ and $E \subseteq F$. H is a *minimal* triangulation of G if there is no triangulation of G that is a proper subgraph of H. Rose, Tarjan and Lueker characterised minimal triangulations of a graph.

Theorem 2. **[22]** *Let $G = (V, E)$ and $H = G \cup F$, $E \cap F = \emptyset$, be two graphs, and let H*

be chordal. The following statements are equivalent:

(i) H is a minimal triangulation of G,

(ii) For every $e \in F$, e is the unique chord in a cycle of length 4 in H,

(iii) For every $e \in F$, the graph $H-e$ is not chordal.

Let $G = (V, E)$ be a graph and $S \subseteq V$. We call S an *a, b-separator* of G for two non-adjacent vertices $a, b \in V$ if a and b are in different connected components of $G \setminus S$. S is a *minimal* a, b-separator of G if there is no a, b-separator S' such that $S' \subset S$. S is a *minimal separator* of G if S is a minimal a, b-separator of G for some non-adjacent vertices $a, b \in V$. A connected component C of $G \setminus S$ is *S-full* if every vertex of S has a neighbour in C; S-full components of G are S-full connected components of $G \setminus S$. Then, S is a minimal separator of G if and only if G has two S-full components [11]. S is a minimal a, b-separator of G if and only if a and b are in different S-full components of G. Two minimal separators S_1 and S_2 *cross* if there are two connected components in $G \setminus S_1$ that contain vertices from S_2. The crossing relation over minimal separators is symmetric [20].

Theorem 3. [9], [20] *Let G be a graph. The following statements are equivalent:*

(i) G is chordal,

(ii) Every minimal separator of G is a clique,

(iii) Every pair of minimal separators of G is non-crossing.

Let $G = (V, E)$ be an interval graph. Let $A_1, \ldots, A_k$ be the maximal cliques of G. A *consecutive clique arrangement* for G is a sequence $A_{\pi(1)}, \ldots, A_{\pi(k)}$ for π a permutation over $[k]$ such that, for every vertex u in G, the maximal cliques containing u appear consecutively in the sequence. A graph is an interval graph if and only if it has a consecutive clique arrangement [10].

Theorem 4. [17] *Let $G = (V, E)$ be an interval graph with consecutive clique arrangement $A_1, \ldots, A_k$. The sets $A_i \cap A_{i+1}$, $1 \leq i < k$, are exactly the minimal separators of G.*

There is a strong correspondence between maximal sets of pairwise non-crossing minimal separators of a graph and its minimal triangulations, given by the following theorem (see also [17]).

Theorem 5. [20] *Let $G = (V, E)$ be a graph.*

(i) Let $\mathcal{S}$ be a maximal set of pairwise non-crossing minimal separators of G. Graph H is obtained from G by making all separators in $\mathcal{S}$ into cliques. Then, H is a minimal triangulation of G.

(ii) Let H be a minimal triangulation of G and let $\mathcal{S}$ be the set of minimal separators of H. Then, $\mathcal{S}$ is a maximal set of pairwise non-crossing minimal separators of G and H originates from G by completing into cliques the separators in $\mathcal{S}$.

Theorem 6. [17] *Let $G = (V, E)$ be a graph and let H be a minimal triangulation of G.*

(i) For every pair a, b of non-adjacent vertices of H, every minimal a, b-separator in H is a minimal a, b-separator in G.

(ii) For every minimal separator S of H and every connected component C in $H \setminus S$, $V(C)$ induces a connected component in $G \setminus S$.

From a theoretical point of view, minimal triangulations can characterise graph classes. The first non-trivial such result was obtained by Bodlaender and Möhring, who showed that every minimal triangulation of a P_4-free graph, i.e., a cograph, is P_4-free [5]. An important result was the characterisation of AT-free graphs.

Theorem 7. **[19], [20]** *A graph G is AT-free if and only if every minimal triangulation of G is AT-free.*

Since every chordal AT-free graph is an interval graph, minimal triangulations of AT-free graphs are interval graphs. Characterisations of graph classes by minimal triangulations in the sense of Theorem 7 are highly interesting but rare.

3. Scanlines, separators and separator graphs. Listing the minimal separators of a graph is an interesting problem. Kloks and Kratsch showed that this can be done in polynomial time for a graph class, if the numbers of minimal separators of its graphs are polynomially bounded in the numbers of vertices [15]. We will show in this section that the minimal separators of permutation graphs can be listed in linear time, using an appropriate representation. In a second part, we will show that the separator graphs of permutation graphs can be generated in linear time.

For permutation graphs, minimal separators can be represented efficiently in the permutation diagram. For this purpose, Bodlaender, Kloks, Kratsch introduced *scanlines* [3], which play a central role in our subsequent studies.[1]

Definition 1. *Let $G = G(\pi)$ be a permutation graph over $[n]$. A* ***scanline*** *of G is a pair (a, e) where $a, e \in \{0.5, 1.5, \ldots, n+\frac{1}{2}\}$.*

Let $G = G(\pi)$ be a permutation graph over $[n]$, and let $\mathfrak{D} = \mathfrak{D}(\pi)$ be its permutation diagram. Let $s = (a, e)$ and $s' = (a', e')$ be scanlines of G. We say that $s \leq s'$ if and only if $a \leq a'$ and $e \leq e'$; $s < s'$ if and only if $s \leq s'$ and $s \neq s'$. By $\mathrm{int}(s)$, we mean the set of vertices $x \in [n]$ such that $(a-x)(e-\pi^{-1}(x)) < 0$, i.e., $\mathrm{int}(s)$ is the set of vertices scanline s intersects with. In terms of permutation diagrams, s can be thought of as a line segment and $\mathrm{int}(s)$ is the set of vertices intersecting with s. We say that a vertex x is *to the left of* s in $\mathfrak{D}$ if it does not intersect with s and is smaller than a. We similarly define what it means to be *to the right of* s. It is easy to see that $\mathrm{int}(s)$ is a separator of G if there are vertices to the left and right of s. For $C_1, C_2 \subseteq [n]$, s is *between* C_1 and C_2, if the vertices of C_1 are to the left of s and the vertices of C_2 are to the right of s, or if the vertices of C_2 are to the left of s and the vertices of C_1 are to the right of s. A scanline s is called *special* if $S =_{\mathrm{def}} \mathrm{int}(s)$ is a minimal separator of G and s is between two S-full components of G. The *enclosure* of s where $a, e \in \{1.5, \ldots, n-\frac{1}{2}\}$ is the set $\mathrm{en}(s) =_{\mathrm{def}} \{a-\frac{1}{2}, a+\frac{1}{2}, \pi(e-\frac{1}{2}), \pi(e+\frac{1}{2})\}$. So, the enclosure may contain between two and four vertices.

Lemma 8. *Let $G = G(\pi)$ be a permutation graph over $[n]$. A scanline s of G is special if and only if $\mathrm{en}(s) \cap \mathrm{int}(s) = \emptyset$.*

Proof: Let $s = (a+\frac{1}{2}, e+\frac{1}{2})$ be a scanline, $1 \leq a, e < n$, and $S =_{\mathrm{def}} \mathrm{int}(s)$. If $\mathrm{en}(s) \cap S = \emptyset$, then a and $\pi(e)$ belong to one connected component C_1 of $G \setminus S$ and $a+1$ and $\pi(e+1)$ belong to another connected component C_2 of $G \setminus S$. Since every vertex in S has neighbours in C_1 and C_2, C_1 and C_2 are S-full components of G and s is a special scanline. Now, let $u \in \mathrm{en}(s) \cap S$. Then, u cannot have a neighbour in the connected components of $G \setminus S$ to the left or the right of s, hence s is not a special scanline. Note that S may not be a minimal separator or that there is no S-full component to the left or right of s. ■

[1] Many graph classes based on geometric models provide a similar efficient characterisation of minimal separators; for a short overview see [15].

Corollary 9. *Let $G = G(\pi)$ be a permutation graph over $[n]$, and let m be the number of edges of G. Then, G has at most $\min\{n+m, \binom{n}{2} - m\} \leq \frac{1}{4}n \cdot (n+1)$ special scanlines, and they can be listed in linear time.*

Proof: Every special scanline $s = (a+\frac{1}{2}, e+\frac{1}{2})$ of G can be defined by the sets $\{a, \pi(e)\}$ and $\{a, \pi(e+1)\}$. In the former case, these sets can be of cardinality 1 or 2, and if it contains two vertices these vertices are adjacent. In the latter case, the sets always contain exactly two non-adjacent vertices. Hence, G has at most $n+m$ special scanlines and at most $\binom{n}{2} - m$ special scanlines. For the upper bound $\frac{1}{4}n \cdot (n+1)$, note that, if $n \geq 4$, $\min\{n+m, \binom{n}{2} - m\}$ is largest possible if $n+m$ equals $\binom{n}{2} - m$, and this is true for $m = \frac{1}{4}n \cdot (n-3)$. For listing the special scanlines, simply check for every vertex $x \in [n]$ and every edge $uv \in E$, $u < v$, whether $(x+\frac{1}{2}, \pi^{-1}(x)+\frac{1}{2})$ or $(v+\frac{1}{2}, \pi^{-1}(u)+\frac{1}{2})$ are special scanlines, which results in a linear-time listing algorithm. ∎

By definition, special scanlines represent minimal separators of permutation graphs. We will show that the converse is equally true. Furthermore, we will show that a result for minimal separators similar to Corollary 9 also holds. The following lemma is an extension of a result by Bodlaender, Kloks, Kratsch.

Lemma 10. **[3]** *Let $G = G(\pi)$ be a permutation graph over $[n]$. Let $C_1, C_2 \subseteq V$ induce connected subgraphs of G. If $N_G[C_1] \cap C_2 = \emptyset$, then there is a special scanline s between C_1 and C_2 such that $S =_{\text{def}} \text{int}(s)$ is a minimal u,v-separator in G for some vertices $u \in C_1$ and $v \in C_2$.*

Proof: We assume $\max C_1 < \min C_2$. Let $u \in C_1$ and $v \in C_2$ be vertices of G. Since $S' =_{\text{def}} V \setminus (C_1 \cup C_2)$ is a separator for any pair of vertices from C_1 and C_2, there is a minimal u,v-separator $S \subseteq S'$ of G. Let C_1' and C_2' be the connected components of $G \setminus S$ containing the vertices of C_1 and C_2, respectively. It holds that C_1' and C_2' are S-full components of G, and every vertex of S has neighbours in C_1' and C_2'. Let $a =_{\text{def}} \max C_1'$ and $e =_{\text{def}} \max \pi^{-1}(C_1')$, and let $s =_{\text{def}} (a+\frac{1}{2}, e+\frac{1}{2})$. Observe that $a < \min C_2'$ and $e < \min \pi^{-1}(C_2')$. Then, $S = \text{int}(s)$, and s is a scanline between the S-full components of G induced by C_1' and C_2'. Hence, s is a special scanline. ∎

Corollary 11. *Let $G = G(\pi)$ be a permutation graph over $[n]$. If $S \subseteq [n]$ is a minimal separator of G, then there is a special scanline s of G such that $S = \text{int}(s)$. Furthermore, if m denotes the number of edges of G, G has at most $\min\{n+m, \binom{n}{2} - m\} \leq \frac{1}{4}n \cdot (n+1)$ minimal separators.*

Proof: Let S be a minimal separator G, and let C_1 and C_3 induce two S-full components of G. Consider $G \setminus S$. Every vertex of G between C_1 and C_3 belongs to an S-full component of G. Let C_2 be the S-full component of G containing $\max C_1 + 1$. Applying Lemma 10 to C_1 and C_2 yields a special scanline of G representing S. For the number of minimal separators, we apply Corollary 9. ∎

We want to list all minimal separators. Instead of listing them as vertex sets, we choose scanlines as an efficient representation. By Corollary 11, only special scanlines have to be considered. However, two special scanlines can represent the same minimal separator, i.e., for achieving a linear-time listing algorithm we have to identify multiple representations of the same minimal separator in overall linear time. Clearly, it is not possible in linear time to compare the separators vertex by vertex. Instead, we will show that it suffices to compare only a few parameters to know whether two minimal separators are equal. For scanline $s = (a, e)$, the set of *enclosing extremal points* is defined as

$$\begin{aligned}\mathrm{exo}(s) =_{\mathrm{def}} \{&(\max\{x < a : x \in \mathrm{int}(s)\} \cup \{0\}, \mathrm{U}),\\ &(\min\{x > a : x \in \mathrm{int}(s)\} \cup \{n{+}1\}, \mathrm{U}),\\ &(\max\{x < e : \pi(x) \in \mathrm{int}(s)\} \cup \{0\}, \mathrm{L}),\\ &(\min\{x > e : \pi(x) \in \mathrm{int}(s)\} \cup \{n{+}1\}, \mathrm{L})\}\,.\end{aligned}$$

The second component of the pairs indicates the upper or lower line of the permutation diagram. If we interpret a scanline in the permutation diagram as a usual vertex line segment, the enclosing extremal points can be interpreted as closest neighbours. They define a trapezium, for which we can define interior and exterior. The *interior* of $\mathrm{exo}(s)$ is the set of points on the upper and lower lines that lie properly between two bounding points; the *exterior* is the set of points on the upper and lower lines that are not interior. If s is a special scanline, then the interior of $\mathrm{exo}(s)$ partitions the exterior into two non-empty "areas", i.e., (possibly degenerated) trapeziums, and $x \in \mathrm{int}(s)$ if and only if both endpoints of x are in both exterior areas.

Lemma 12. *Let $G = G(\pi)$ be a permutation graph over $[n]$. Let s_1 and s_2 be special scanlines of G where $S_1 = \mathrm{int}(s_1)$ and $S_2 = \mathrm{int}(s_2)$. Then, $\mathrm{exo}(s_1) = \mathrm{exo}(s_2)$ if and only if $S_1 = S_2$.*

Proof: If $S_1 = S_2$, we can assume without loss of generality that $s_1 < s_2$; otherwise $\mathrm{int}(s_2) \cap \mathrm{en}(s_1) \neq \emptyset$ would imply $S_1 \neq S_2$. Then, $\mathrm{exo}(s_1) = \mathrm{exo}(s_2)$. Conversely, if $\mathrm{exo}(s_1) = \mathrm{exo}(s_2)$, then $S_1 = S_2$ immediately holds. ■

Theorem 13. *Let $G = G(\pi)$ be a permutation graph over $[n]$. The minimal separators of G can be listed in linear time where every minimal separator is represented by a special scanline.*

Proof: The algorithm is obvious:

(1) list the special scanlines of G

(2) generate the enclosing extremal points of each scanline

(3) delete all but one scanlines with the same enclosing extremal points.

For step (1), we apply Corollary 9. The scanlines are stored in $n{-}1$ lists each containing the scanlines with the same lower endpoint, and each list is ordered decreasingly according to the upper endpoint. The extremal points are computed separately. We illustrate a procedure for the lower right point. Remember that enclosing extremal points can be seen as closest neighbours. We apply a procedure that is a modification of the one used for computing closest neighbours in Lemma 1:

(a) initialize a stack with top-of-stack element 0

(b) for $i = n, \ldots, 1$: delete all numbers greater than $\pi(i)$ and push $\pi(i)$ on the stack.

The $\mathcal{O}(n)$-time implementation is obvious. At the end of each execution of step (b), the set of vertices greater than 0 in the stack induces a clique in G. So, running the procedure and passing through the whole stack in every iteration step takes overall linear time. The lower right enclosing extremal point of scanline $s = (a{+}\frac{1}{2}, e{+}\frac{1}{2})$ is $\pi^{-1}(x)$ for the largest $x \leq a$ in the stack at the end of iteration $i = e{+}1$. We assume $\pi^{-1}(0) = n{+}1$. The correctness follows similar to Lemma 1: When the lower right enclosing extremal point of s is determined, the following iteration step of the closest-neighbour algorithm is simulated.

To compute the other three enclosing extremal points we apply similar procedures:

- lower left point: we initialize the stack with $n+1$, pass from 1 to n and delete smaller numbers, so that the top-of-stack element is the smallest number in the stack. The induced-clique property of the vertices in the stack still holds.
- for the two upper enclosing extremal points proceed similar to the lower points, but the special scanlines have to be reordered. This can be done by a single sweep and building new lists.

For every special scanline, its enclosing extremal points are known, and they can be sorted according to them, so that scanlines with the same extremal points occur consecutively. Now, execute the deletion step. This algorithm takes an overall linear time. ∎

Using the scanline representation of minimal separators, we can easily build the separator graph of a permutation graph. The separator graph was introduced by Parra and Scheffler as an auxiliary structure to characterise minimal triangulations [20]. Let $G = (V, E)$ be a graph. The *separator graph* $\mathfrak{S}_G$ of G has a vertex for every minimal separator of G and two vertices are adjacent if and only if the corresponding minimal separators are non-crossing. Since a maximal clique in the separator graph corresponds to a maximal set of pairwise non-crossing minimal separators, a maximal clique in the separator graph corresponds to a minimal triangulation of G due to Theorem 5. There is not much information known about the structure of separator graphs of arbitrary graphs or special graph classes. One of the rare results concerns chordal graphs: a graph is chordal if and only if its separator graph is complete (Theorem 3). We will see that separator graphs of permutation graphs are permutation graphs.

Definition 2. *Two scanlines s_1 and s_2 of a permutation graph $G = G(\pi)$ **intersect** if and only if neither $s_1 \leq s_2$ nor $s_2 < s_1$.*

Lemma 14. *Let $G = G(\pi)$ be a permutation graph. Let s_1 and s_2 be special scanlines of G, and let $S_1 =_{\text{def}} \text{int}(s_1)$ and $S_2 =_{\text{def}} \text{int}(s_2)$.*

(i) If s_1 and s_2 do not intersect then S_1 and S_2 are non-crossing.

(ii) If s_1 and s_2 intersect then S_1 and S_2 cross.

Proof: Remember that S_1 and S_2 are minimal separators of G by definition. Let s_1 and s_2 do not intersect. We assume $s_1 < s_2$. Let $C_1, \ldots, C_\ell$ be the connected components of $G \setminus S_1$, $\ell \geq 2$, where the vertices in C_i are smaller than the vertices in C_{i+1} for all $i \in [\ell-1]$. Suppose S_1 and S_2 cross. Then, there is $r \geq 1$ such that s_2 intersects with vertices from C_r and C_{r+1}. Since $s_1 < s_2$, the vertices in $\text{en}(s_2)$ that are to the right of s_2 are not contained in $\text{int}(s_1)$. Hence, there is a path between a vertex from C_r and a vertex from C_{r+1}, and C_r and C_{r+1} are not different connected components of $G \setminus S_1$, which contradicts the assumption that S_1 and S_2 cross. For the converse, let s_1 and s_2 intersect. Observe that the vertices of $\text{en}(s_1)$ belong to two connected components of $G \setminus S_1$, and one vertex from each of these components belongs to S_2. Hence, S_1 and S_2 cross. ∎

Lemma 14 implies two interesting results about the structure of separator graphs of permutation graphs.

Theorem 15. *Let $G = G(\pi)$ be a permutation graph. Then, $\mathfrak{S}_G$ is a permutation graph, and a permutation sequence representation of $\mathfrak{S}_G$ can be generated in linear time.*

Proof: Let $\mathfrak{D} = \mathfrak{D}(\pi)$ be the permutation diagram of G. In linear time, a minimal set of special scanlines of G representing the minimal separators of G can be generated according to Theorem 13. Add these special scanlines to $\mathfrak{D}$. Obviously, the special scanlines define

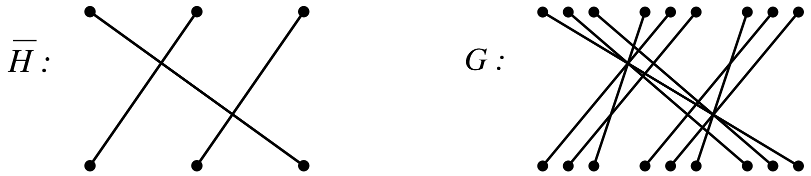

Figure 1. An example for the construction of Theorem 16. Graphs $\overline{H}$ and G are represented by appropriate permutation diagrams.

the permutation diagram of a permutation graph G', and two vertices in G' are adjacent if and only if the corresponding special scanlines intersect. Due to Lemma 14, intersection of special scanlines corresponds to crossing of the associated minimal separators. Hence, the complement of G' is the separator graph of G, which is a permutation graph. It remains to assign numbers to the scanlines to obtain a permutation sequence. Order the scanlines by their upper endpoints and within an equivalence class by their lower endpoints and assign numbers according to this ordering, which gives the numbers on the upper line in the permutation diagram. The sequence is represented on the lower line, where scanlines with the same lower endpoint are ordered according to the upper endpoint. Since the permutation diagram represents the complement of $\mathfrak{S}_G$ output the permutation sequence in reversed order. This is a linear-time algorithm. ■

Theorem 16. *For every permutation graph H, there is a permutation graph G such that H is induced subgraph of $\mathfrak{S}_G$.*

Proof: Let $\overline{H}$ be the complement of H, and let $\mathfrak{D}'$ be a permutation diagram for $\overline{H}$. Replace every line segment of $\mathfrak{D}'$ by the following line segments configuration: two parallel line segments and one crossing both from upper left to lower right. This special configuration represents the chordless path on three vertices. (For a sample replacement see Figure 1.) We obtain permutation diagram $\mathfrak{D}$, and let G be the corresponding graph. It is clear that G emerges from $\overline{H}$ by (iteratively) replacing every vertex by a P_3 but conserving the neighbourhood. By Lemma 8, the non-adjacent vertices of the inserted P_3 are the enclosures of special scanlines in $\mathfrak{D}$, and two of these scanlines intersect if and only if the corresponding vertices of H are non-adjacent. By Lemma 12, these special scanlines identify pairwise different minimal separators, so that H is induced subgraph of $\mathfrak{S}_G$. ■

It is an interesting question whether every permutation graph is the separator graph of a permutation graph. From Theorem 16, we conclude that it is not possible to find a characterisation by forbidden induced subgraphs that is stronger than Theorem 15. Our construction in the proof of Theorem 16 generates graphs with separator graphs with more vertices than the graph that is induced. Consider the example of Figure 1. The constructed graph G has four minimal separators, and its separator graph is isomorphic to the P_3 plus an isolated vertex.

4. Potential maximal cliques. The main focus of our work is the study of minimal triangulations of permutation graphs and associated problems. Theorem 5 establishes a strong correspondence between maximal cliques of separator graphs and minimal triangulations. However, from an algorithmic point of view, separator graphs are useful only in a restricted sence. It is often difficult to efficiently list its maximal cliques, and for most problems it is not necessary to know minimal triangulations explicitly. At the end of this section, we will present another structure which serves this purpose better. We will use *potential maximal cliques* that were introduced by Bouchitté and Todinca [6].

Definition 3. *Let $G = (V, E)$ be a graph. A set $C \subseteq V$ of vertices is a **potential maximal clique** of G if and only if there is a minimal triangulation H of G such that C is a maximal clique in H.*

We want to characterise the potential maximal cliques of permutation graphs. Remember the definition of special scanlines. In this section, we add two further scanlines. For a permutation graph G over $[n]$, $\mathrm{s}_0 =_{\text{def}} (\frac{1}{2}, \frac{1}{2})$ and $\mathrm{s}_e^n =_{\text{def}} (n+\frac{1}{2}, n+\frac{1}{2})$ are also special scanlines of G. Let s_1 and s_2 be two special scanlines of G such that $s_1 < s_2$. By $G[s_1, s_2]$ we denote the subgraph of G induced by the vertices that intersect with s_1 or s_2 or that lie between s_1 and s_2, i.e., to the right of s_1 and to the left of s_2. With $G[s_1, s_2]$ we associate the permutation diagram of G reduced to only the vertices of $G[s_1, s_2]$. The non-intersecting scanlines s_1 and s_2 are *neighbours* if there is no special scanline s' of G such that $s_1 < s' < s_2$. We say that s_1 is a *left neighbour* of s_2 and s_2 is a *right neighbour* of s_1. If s_1 and s_2 have a common endpoint they are *close* neighbours. By $N_\pi^<(s)$ we denote the set of all left neighbours of special scanline s of G.

Lemma 17. *Let $G = G(\pi)$ be a permutation graph over $[n]$, and let m be the number of edges of G. Then, G has at most $2m$ pairs of special scanlines that are close neighbours.*

Proof: Let $s = (a, e)$ be a special scanline of G. There is at most one special scanline $s' = (a, e')$ of G such that s and s' are neighbours and $e' < e$. Similarly, there is at most one special scanline $s'' = (a'', e)$ of G such that s and s'' are neighbours and $a'' < a$. Note that in both cases the enclosure of s must contain two vertices that are to the left of s. Since these vertices are adjacent, the lemma holds. ■

Lemma 18. *Let $G = G(\pi) = (V, E)$ be a permutation graph over $[n]$. Let s_1, s_1', s_2 and s_2' be special scanlines of G where $s_1 \in N_\pi^<(s_2)$ and $s_1' \in N_\pi^<(s_2')$. Let $S_1 =_{\text{def}} \operatorname{int}(s_1)$ and $S_2 =_{\text{def}} \operatorname{int}(s_2)$, and let $C =_{\text{def}} V(G[s_1, s_2])$. Then:*

(i) $u \in C \setminus (S_1 \cup S_2) \implies N_G[u] = C$,

(ii) $u \in S_1 \setminus S_2$ and $v \in S_2 \setminus S_1 \implies uv \in E$,

(iii) $C = V(G[s_1', s_2']) \implies s_1' = s_1$ and $s_2' = s_2$.

Proof: Let $u, v \in C$ such that $uv \notin E$. Let $u \in C \setminus (S_1 \cup S_2)$. If $v \in C \setminus (S_1 \cup S_2)$, there is a special scanline between u and v by Lemma 10 that does not intersect with neither s_1 nor s_2. If $v \in S_1$, there is a vertex in the enclosure of s_1 that is to the left of s_1 and adjacent to v such that there is a special scanline between $\{v, w\}$ and $\{u\}$ by Lemma 10, and this scanline does not intersect with neither s_1 nor s_2. The case $v \in S_2$ is similar to $v \in S_1$. Since s_1 and s_2 are neighbours there cannot be a special scanline between them, and statement (i) holds. If $u \in S_1 \setminus S_2$ and $v \in S_2 \setminus S_1$, let x and z be vertices of the enclosures of s_1 and s_2, respectively, that do not belong to C and that are adjacent to u and v, respectively, such that there is a special scanline between $\{u, x\}$ and $\{v, z\}$ by Lemma 10. We can apply Lemma 10, since $\{v, z\} \cap N_G[\{u, x\}] = \emptyset$. However, this is not possible since s_1 and s_2 are neighbours, and statement (ii) holds. Note that this case is not possible if $s_1 = \mathrm{s}_0$ or $s_2 = \mathrm{s}_e^n$, since $S_1 = \emptyset$ or $S_2 = \emptyset$.

For statement (iii) observe that s_1' does not intersect with neither s_1 nor s_2, since otherwise $C' =_{\text{def}} V(G[s_1', s_2'])$ would contain vertices from the enclosure of s_1 or s_2. Analogously, s_2' does not intersect with neither s_1 nor s_2. So, $\{s_1, s_1', s_2, s_2'\}$ is a set of pairwise non-intersecting special scanlines. Since C' must contain a vertex to the right of s_1 that belongs to the enclosure of s_1, $s_1 < s_2'$; similarly, $s_1' < s_2$, and this is only possible if $s_1' = s_1$ and $s_2' = s_2$. ■

Let s_1 and s_2 be special scanlines of G where $s_1 \in N^{<}_{\pi}(s_2)$. A vertex $x \in [n]$ is an *inner vertex* of $G[s_1, s_2]$ if $x \in V(G[s_1, s_2])$ and $x \notin \mathrm{int}(s_1) \cup \mathrm{int}(s_2)$. Inner vertices play a special role in our study, since $N_G[x] = V(G[s_1, s_2])$ by Lemma 18. For a potential maximal clique C of G, we say that $x \in C$ is an *inner vertex* of C if $N_G[x] = C$.

Lemma 19. *Let $G = G(\pi)$ be a permutation graph over $[n]$. Let s_1, s_2, s be special scanlines of G where $s_1 \in N^{<}_{\pi}(s_2)$ and $G[s_1, s_2]$ contains an inner vertex x. If s intersects with s_1 or s_2, then s also intersects with x.*

Proof: Observe that x is to the right of s_1 and to the left of s_2. If s intersects with s_1 and s_2 then s intersects with every vertex between s_1 and s_2. Let s intersect with only s_1. Two non-adjacent vertices of the enclosure of s belong to $\mathrm{int}(s_1)$, and by Lemma 18, they are neighbours of x. Then, s is between these two vertices and intersects with x. Similarly, x belongs to $\mathrm{int}(s)$, if s intersects with only s_2. ∎

Corollary 20. *Let $G = G(\pi)$ be a permutation graph over $[n]$. Then, G has at most n pairs of special scanlines that are neighbours but not close neighbours.*

Proof: Let s_1 and s_2 be special scanlines of G that are neighbours but not close neighbours. Then, $G[s_1, s_2]$ must contain an inner vertex x; otherwise, i.e., every vertex of $G[s_1, s_2]$ intersects with s_1 or s_2, G would have a special scanline that is close to s_1 and s_2, hence between s_1 and s_2 (this also follows from Lemma 18, statement (ii)). Suppose there is another pair (s'_1, s'_2) of special scanlines of G where $s'_1 < s'_2$ and s'_1 and s'_2 are neighbours but not close neighbours such that x is inner vertex in $G[s'_1, s'_2]$. By definition, x is to the right of s_1 and s'_1 and to the left of s_2 and s'_2. If $s_1 \neq s'_1$, then s_1 and s'_1 intersect, but then $x \in \mathrm{int}(s_1) \cup \mathrm{int}(s'_1)$ by Lemma 19. Hence, $s_1 = s'_1$. Analogously, we conclude $s_2 = s'_2$, so that there is a unique pair (s, s') of neighbouring special scanlines of G that are not close such that x is inner vertex of $G[s, s']$. We conclude that there is a 1-1 correspondence between the pairs of special scanlines of G that are neighbours but not close neighbours and a subset of $[n]$, which proves the corollary. ∎

The main result of this section is a characterisation of potential maximal cliques of permutation graphs in terms of special scanlines. We will partition the proof into several lemmata.

Lemma 21. *Let $G = (V, E)$ be a graph, and let H be a minimal triangulation of G. Let $u \in V$ such that there is only one maximal clique C in H containing u. Then, u is an inner vertex of C.*

Proof: Since $\{u, v\}$ for every $v \in N_G(u)$ is a clique in H, $N_G[u] \subseteq C$. Suppose there is $v \in C$ such that $uv \notin E$. Since $uv \in E(H)$, uv is unique chord in a cylce (u, x, v, z) in H by Theorem 2. But since $x, z \in C$, x and z are adjacent in H, hence (u, x, v, z) contains two chords in H, which contradicts the assumption. ∎

For chordal graphs, it holds that maximal cliques are not minimal separators. So, we can alternatively formulate the prerequisites of Lemma 21 by the following: u does not belong to a minimal separator of H.

Lemma 22. *Let $G = G(\pi)$ be a permutation graph over $[n]$. Let $C \subseteq [n]$ be a potential maximal clique of G that has an inner vertex. Then, there are special scanlines s_1, s_2 such that $s_1 \in N^{<}_{\pi}(s_2)$ and $C = V(G[s_1, s_2])$, where s_1 and s_2 are not close neighbours.*

Proof: Let H be a minimal triangulation of G such that C is a maximal clique of H. By Theorem 7, H is an interval graph. Let $A_1, \ldots, A_k$, $k \geq 1$, be a consecutive clique arrangement for H. If $k = 1$, then G is complete. Hence, s_0 and s^n_e are neighbouring special scanlines of G, and $C = V(G[\mathrm{s}_0, \mathrm{s}_e^n])$. Obviously, s_0 and s_e^n are not close neighbours. Let $k \geq 2$. Let $C = A_i$ for $1 \leq i \leq k$, and let $S_1 =_{\text{def}} A_{i-1} \cap A_i$ and $S_2 =_{\text{def}} A_i \cap A_{i+1}$—we assume $A_0 =_{\text{def}} A_{k+1} =_{\text{def}} \emptyset$. By Theorems 4 and 6, S_1 and S_2 are minimal separators of G (except for the particular cases $i = 1$ and $i = k$). Observe that $S_1 \cup S_2 \subset C$, since C contains an inner vertex. Let $S =_{\text{def}} S_1 \cup S_2$ and $C' =_{\text{def}} C \setminus S$. Note that every vertex in S has a neighbour outside C in G. Remember that $N_G[u] = C$ for every $u \in C'$ according to Lemma 21. In particular, C' is a clique of G, and every vertex of G is either a neighbour of every vertex in C' or of none, i.e., C' is a clique module of G, even a maximal clique module of G (see also the results in Section 5). Let

$$\begin{aligned}
a_1 &=_{\text{def}} \max\{x < \min(C') : x \in [n] \setminus C\} \cup \{0\} \\
a_2 &=_{\text{def}} \min\{x > \max(C') : x \in [n] \setminus C\} \cup \{n+1\} \\
e_1 &=_{\text{def}} \max\{i < \min \pi^{-1}(C') : i \in [n] \setminus \pi^{-1}(C)\} \cup \{0\} \\
e_2 &=_{\text{def}} \min\{i > \max \pi^{-1}(C') : i \in [n] \setminus \pi^{-1}(C)\} \cup \{n+1\}\,,
\end{aligned}$$

and let $s_1 =_{\text{def}} (a_1+\frac{1}{2}, e_1+\frac{1}{2})$ and $s_2 =_{\text{def}} (a_2-\frac{1}{2}, e_2-\frac{1}{2})$. Every vertex in C' is to the right of s_1 and to the left of s_2 by definition of a_1, a_2, e_1, e_2, hence between s_1 and s_2. In particular, s_1 and s_2 do not have a common endpoint. Observe that $a_1+1, \ldots, a_2-1$ and $\pi(e_1+1), \ldots, \pi(e_2-1)$ are vertices of C. Then, s_1 and s_2 are special scanlines of G by Lemma 8. Furthermore, every vertex in S has a neighbour to the left of s_1 or to the right of s_2 in G, since between s_1 and s_2 there are only vertices of C. Hence, $S \subseteq \mathrm{int}(s_1) \cup \mathrm{int}(s_2)$. Suppose there is $x \in (\mathrm{int}(s_1) \cup \mathrm{int}(s_2)) \setminus S$. By the aforesaid, $x \in \mathrm{int}(s_1) \cap \mathrm{int}(s_2)$ and must be a neighbour of $u \in C'$ in G, hence $x \in C$ and $x \in S$, which is a contradiction. So, $\mathrm{int}(s_1) \cup \mathrm{int}(s_2) = S$, and $V(G[s_1, s_2]) = C$. Finally, suppose there is a special scanline s of G such that $s_1 < s < s_2$. Since C has inner vertices, s is a close neighbour of s_1 and s_2, and C contains two vertices u, v that are non-adjacent in G such that u intersects with s_1 and not with s_2 and v intersects with s_2 and not with s_1 (u and v belong to the enclosure of s). Since u and v are adjacent in H, there is a minimal separator $S' = A_j \cap A_{j+1}$ of G and H that contains u and v due to Theorem 5. Let s' be a special scanline of G such that $S' = \mathrm{int}(s')$, that exists by Lemma 10. It holds that s' cannot intersect with s_1 or s_2. If s' intersects with both scanlines, a vertex to the left of s_1 would be adjacent to the vertices in C' in H, which is not possible. If s' intersects with either s_1 or s_2, it must have a common endpoint with the other scanline, and again, the vertices from C' would be adjacent in H with a vertex to the left of s_1 or to the right of s_2. Hence, s' is a close neighbour of s_1 and s_2 and intersects with s. But then, $C' \subseteq S'$, which contradicts the assumption that C' are inner vertices of C. So, $s_1 \in N_\pi^<(s_2)$. ■

Lemma 23. *Let $G = G(\pi)$ be a permutation graph over $[n]$. Let $C \subseteq [n]$ be a potential maximal clique of G without inner vertices. Then, there are minimal separators S_1, S_2 of G such that $C = S_1 \cup S_2$. Furthermore, there are special scanlines s_1, s_2 of G such that $s_1 \in N_G^<(s_2)$, $\{S_1, S_2\} = \{\mathrm{int}(s_1), \mathrm{int}(s_2)\}$ and $C = V(G[s_1, s_2])$. In particular, s_1 and s_2 are close neighbours.*

Proof: Let H be a minimal triangulation of G such that C is a maximal clique of H. By Theorem 7, H is an interval graph. Let $A_1, \ldots, A_k$, $k \geq 1$, be a consecutive clique arrangement for H. Let $i \in [k]$ such that $A_i = C$. Since A_1 and A_k have inner vertices, $k \geq 3$ and $1 < i < k$. Let $S_1 =_{\text{def}} A_{i-1} \cap A_i$ and $S_2 =_{\text{def}} A_i \cap A_{i+1}$. By Theorems 4 and 6, S_1 and S_2 are minimal separators of G. Due to Lemma 21, $C = S_1 \cup S_2$, and $S_1 \not\subseteq S_2$ and $S_2 \not\subseteq S_1$. Let $u \in C \setminus S_1$, $v \in A_{i-1} \setminus S_1$ and $w \in C \setminus S_2$, $x \in A_{i+1} \setminus S_2$. We assume $v < u$;

if $u < v$ we use $A_k, \ldots, A_1$ as consecutive clique arrangement for H and the same vertices with their new meanings. It holds that v, w, u, x induce a chordless path in H, and u and w are adjacent in G. We want to show that $w < x$. Suppose $x < w$. Remember that S_1 and S_2 are minimal u,v- and w,x-separators in H, respectively, and in G by Theorem 6. If $x < v$, i.e., $x < v < u$ and u and x are non-adjacent in G, then the u,v-separator S_1 also separates u and x, which contradicts Theorem 6. If $v < x$, i.e., $v < x < w$ and w and v are non-adjacent in G, similarly, S_2 also separates w and v contradicting Theorem 6. Hence, $w < x$. Due to Theorem 6, $C \setminus S_1$ is contained in a connected component of $G \setminus S_1$ induced by D_1. Let $a_1 =_{\text{def}} \min(D_1)$ and $e_1 =_{\text{def}} \min \pi^{-1}(D_1)$ and $s_1 =_{\text{def}} (a_1 - \frac{1}{2}, e_1 - \frac{1}{2})$. Observe that $a_1 - 1$ and $\pi(e_1 - 1)$ exist and $v \leq a_1 - 1$ and $\pi^{-1}(v) \leq e_1 - 1$ and $a_1 - 1$ and $\pi(e_1 - 1)$ are to the left of s_1. Since $a_1, \pi(e_1) \in D_1$, a_1 and $\pi(e_1)$ are to the right of s_1, hence s_1 is a special scanline of G due to Lemma 8. Let D_1' induce the connected component of $G \setminus S_1$ containing v. Since S_1 is a minimal u,v-separator in H and G, every vertex in S_1 has a neighbour in D_1 and D_1', and $S_1 = \text{int}(s_1)$. By a similar construction using S_2, define D_2 and D_2' and $s_2 =_{\text{def}} (a_2 + \frac{1}{2}, e_2 + \frac{1}{2})$, where $a_2 =_{\text{def}} \max(D_2)$ and $e_2 =_{\text{def}} \max \pi^{-1}(D_2)$. It holds that $S_2 = \text{int}(s_2)$. Since $u < w$ or $\pi^{-1}(u) < \pi^{-1}(w)$, $a_1 \leq u < w \leq a_2$ or $e_1 \leq \pi^{-1}(u) < \pi^{-1}(w) \leq e_2$, and since S_1 and S_2 are non-crossing minimal separators, $s_1 < s_2$ by Lemma 14. Hence, $C \subseteq V(G[s_1, s_2])$. Since H is a minimal triangulation of G, for every pair a, b of vertices, $a \in S_1 \setminus S_2$, $b \in S_2 \setminus S_1$, a and b are adjacent in G. Suppose there is $z \in V(G[s_1, s_2]) \setminus C$. Then, z must be adjacent to all vertices in S_1 or all vertices in S_2 in G. Since $z \notin C$, $z \in A_1 \cup \cdots \cup A_{i-1}$ or $z \in A_{i+1} \cup \cdots \cup A_k$. In the former case, S_1 is a z,u-separator, in the latter case, S_2 is a w,z-separator. Hence, z has no neighbours in $S_2 \setminus S_1$, which means that z is not contained in D_1 and to the left of s_1, or in $S_1 \setminus S_2$, which means that z is not contained in D_2 and to the right of s_2. So, $C = V(G[s_1, s_2])$, and s_1 and s_2 have a common endpoint. Finally, suppose there is a special scanline s of G such that $s_1 < s < s_2$. If $a_1 = a_2 + 1$, then there is $j \in \{e_1, \ldots, e_2 - 1\}$ such that $\pi(e_1), \ldots, \pi(j) \in S_2$ and $\pi(j+1), \ldots, \pi(e_2) \in S_1$. If $e_1 = e_2 + 1$, then there is $j \in \{a_1, \ldots, a_2 - 1\}$ such that $a_1, \ldots, j \in S_2$ and $j+1, \ldots, a_2 \in S_1$. But then, there cannot be a special scanline s such that $s_1 < s < s_2$, and $s_1 \in N_\pi^{<}(s_2)$. ■

Since scanlines s_1 and s_2 of Lemma 23 are unique, S_1 and S_2 of Lemma 23 are also unique up to indices according to Lemma 23.

Theorem 24. *Let $G = G(\pi)$ be a permutation graph over $[n]$. $C \subseteq V$ is a potential maximal clique of G if and only if there are special scanlines s_1 and s_2 of G such that $s_1 \in N_\pi^{<}(s_2)$ and $C = V(G[s_1, s_2])$.*

Proof: The "only if" part of the theorem holds by Lemmata 22 and 23. For the "if" part of the theorem, let s_1 and s_2 be special scanlines such that $s_1 \in N_\pi^{<}(s_2)$, and let $C =_{\text{def}} V(G[s_1, s_2])$. Observe that there are $u, v \in C$ such that $u \notin \text{int}(s_1)$ and $v \notin \text{int}(s_2)$; u and v may be identical. Let G' emerge from G by completing $\text{int}(s_1)$ and $\text{int}(s_2)$ into cliques. Due to Lemma 18, C induces a clique in G'. Since no vertex to the left of s_1 is adjacent to u and no vertex to the right of s_2 is adjacent to v, C is a maximal clique in G'. Obtain H' from G' by making every connected component of $G' \setminus C$ complete. C induces a maximal clique in H'. H' is a triangulation of G', and there is a subgraph H of H' that is a minimal triangulation of G'. C induces a maximal clique in H, and since every minimal triangulation of G' is a minimal triangulation of G, C is a potential maximal clique of G. ■

Corollary 25. *A permutation graph on $n \geq 1$ vertices and with m edges has at most $n + 2m$ potential maximal cliques.*

Proof: Let $G = G(\pi)$ be a permutation graph over $[n]$. Let $G[s_1, s_2]$ be a potential maximal clique of G, $s_1 < s_2$ and $s_1 \in N_\pi^<(s_2)$. Then, either s_1 and s_2 are close neighbours or $G[s_1, s_2]$ has an inner vertex. Applying Lemma 17 and Corollary 20, we obtain the desired result. ■

The *potential maximal cliques graph* of a permutation graph $G(\pi)$ over $[n]$ is a directed graph that is denoted by $\mathcal{PC}(\pi)$ and defined as follows: $\mathcal{PC}(\pi)$ has a vertex for every special scanline of $G(\pi)$ (including s_0 and s_e^n), and there is an arc from vertex u to vertex v if and only if $s_u \in N_\pi^<(s_v)$, where s_u and s_v denote the special scanlines the vertices u and v are labelled with, respectively. Hence, there is a 1-1 correspondence between the potential maximal cliques of $G(\pi)$ and the arcs in $\mathcal{PC}(\pi)$ due to Theorem 24.

Lemma 26. *Let $G = G(\pi)$ be a permutation graph over $[n]$.*

(i) The potential maximal cliques graph $\mathcal{PC}(\pi)$ of G can be generated in linear time.

(ii) $\mathcal{PC}(\pi)$ is acyclic, and a topological ordering can be computed in linear time.

(iii) $(x_0, \ldots, x_k)$, $k \geq 1$, is a maximal path in $\mathcal{PC}(\pi)$ if and only if $\mathrm{s}_0, s_1, \ldots, s_{k-1}, \mathrm{s}_e^n$ are the labels of $x_0, \ldots, x_k$, respectively, and $\{\mathrm{s}_0, s_1, \ldots, s_{k-1}, \mathrm{s}_e^n\}$ is a maximal set of pairwise non-intersecting special scanlines of G.

Proof: Due to Corollary 9, the set of special scanlines of G can be generated in linear time. It is clear by Theorem 24 that the sets $N_\pi^<(s)$ for s a special scanline of G, $\mathrm{s}_0 < s$, represent the arc set of $\mathcal{PC}(\pi)$. So, it suffices to show that these sets can be generated in linear time. Similar to the proof of Theorem 15, we construct a permutation diagram of a graph that is related to the separator graph of G in that sense that a minimal separator of G may be assigned to several vertices. Order the special scanlines of G increasingly by their upper endpoints and within an equivalence class by their lower endpoints. In that order, assign numbers $1, 2, \ldots$ to the scanlines. Reorder the scanlines such that they are ordered increasingly by their lower endpoints and within an equivalence class by their upper endpoints. This second order defines a permutation sequence τ. Note that the described algorithm is a linear-time algorithm and that τ preserves $<$ and intersection of the special scanlines of G. Now, assign to every special scanline $s' > \mathrm{s}_0$ the special scanline to its left with respect to τ that is rightmost and does not intersect with s'; let g denote this mapping. For convenience, let $g(\mathrm{s}_0) =_{\mathrm{def}} \mathrm{s}_0$. Let h denote the mapping that assigns to every special scanline $s' \geq \mathrm{s}_0$ the special scanline to its left with respect to τ that is rightmost and intersects with s'; if there is no such scanline, let $h(s') =_{\mathrm{def}} s'$. For permutation graphs, g and h compute the closest left non-neighbours and neighbours, respectively. So, g and h can be computed in linear time by Lemma 1 and Corollary 9. We claim that

$$N_\pi^<(s) = \{g^i(h(s)) : i \geq 0\} \cap \{s' < s\}$$

where $g^0(h(s)) =_{\mathrm{def}} h(s)$ and $g^{i+1}(h(s)) =_{\mathrm{def}} g(g^i(h(s)))$. The inclusion from left to right is clear, since the rightmost element of $N_\pi^<(s)$ with respect to τ is $g(s)$ and the other elements are closest left neighbours of other scanlines in $N_\pi^<(s)$. The inclusion from right to left is obvious. Since every pair s, s' of special scanlines of G where $s' \in N_\pi^<(s)$ defines a unique potential maximal clique (Theorem 24 and Lemma 18) and since G has at most $n + 2m$ potential maximal cliques (Corollary 25), the sets $N_\pi^<$ can be computed in overall linear time. Note that, if $g^i(h(s)) \notin \{s' : s' < s\}$, then $g^{i+1}(h(s)) \notin \{s' : s' < s\}$. This shows claim (i).

For claim (ii) it suffices to observe that ordering τ actually is a topological ordering of the vertices of $\mathcal{PC}(\pi)$. Now, let $(x_0, \ldots, x_k)$ be a maximal path in $\mathcal{PC}(\pi)$. Obviously, the

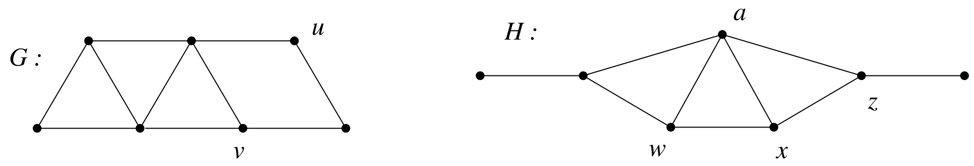

Figure 2. About minimal triangulations of permutation graphs.

labels of x_0 and x_k are s_0 and s_e^n, respectively, since $\mathrm{s}_0 < s < \mathrm{s}_e^n$ for every special scanline s of G. Let $s_1, \ldots, s_{k-1}$ be the labels of $x_1, \ldots, x_{k-1}$, respectively. By definition of $\mathcal{PC}(\pi)$, $\{\mathrm{s}_0, s_1, \ldots, s_{k-1}, \mathrm{s}_e^n\}$ is a maximal set of pairwise non-intersecting special scanlines of G. Conversely, let $S =_{\mathrm{def}} \{s_0, \ldots, s_k\}$ be a maximal set of pairwise non-intersecting special scanlines of G where $s_0 < \cdots < s_k$. Then, $s_{i-1} \in N_\pi^<(s_i)$ for $i \in [k]$, and $s_0 = \mathrm{s}_0$ and $s_k = \mathrm{s}_e^n$. This, however, uniquely defines a maximal path in $\mathcal{PC}(\pi)$. ∎

5. Minimal triangulations. A triangulation H of a graph G is a chordal graph on the same vertex set as G such that G is a subgraph of H. If there is no proper subgraph of H that is a triangulation of G then H is a minimal triangulation of G. By Theorem 7, every minimal triangulation of an AT-free graph is an interval graph. In particular, minimal triangulations of permutation graphs are interval graphs. It is easy to see that minimal triangulations of permutation graphs may not be permutation graphs. Consider graph G of Figure 2. This graph is a permutation graph, and graph $G+uv$ is a minimal triangulation of G that is not a permutation graph. A second question arises, however: Is every interval graph a minimal triangulation of a permutation graph? Consider graph H of Figure 2: H is an interval graph and not a permutation graph. Furthermore, H is a minimal triangulation of only the following four graphs: H, $H-aw$, $H-ax$, $(H-aw)-ax$, but none of these graphs is a permutation graph. It is interesting to note that H is induced subgraph of a minimal triangulation of G' where G' emerges from H by deleting az and adding a new vertex that is adjacent to a and z (i.e., subdividing edge az). G' is a permutation graph. A graph class $\mathcal{C}$ is called *hereditary* if, for every graph $G \in \mathcal{C}$, every induced subgraph of G belongs to $\mathcal{C}$.

Fact 27. *The set of minimal triangulations of permutation graphs is properly contained in the set of the interval graphs and not hereditary.*

Proof: By Theorem 7, every minimal triangulation of a permutation graph is an interval graph. Graph H of Figure 2 is not a minimal triangulation of a permutation graph but is induced subgraph of a minimal triangulation of a permutation graph. ∎

In the following, we will give an efficient characterisation of minimal triangulations of a permutation graph, and we will show that special minimal triangulations of a given permutation graph can be computed fast. An interval graph over $[n]$ is an interval graph on n vertices where the names of the vertices are $1, \ldots, n$. Let $G = (V, E) = G(\pi)$ be a permutation graph over $[n]$. Let $P = (x_0, \ldots, x_k)$ be a maximal path in $\mathcal{PC}(\pi)$. We will see that P defines a minimal triangulation of G. Due to Lemma 26, the special scanlines $s_0, \ldots, s_k$ of G that are the labels of $x_0, \ldots, x_k$ in $\mathcal{PC}(\pi)$, respectively, form a maximal set of non-intersecting special scanlines of G and $s_{i-1} \in N_\pi^<(s_i)$ for $i \in [k]$; furthermore, $s_0 = \mathrm{s}_0$ and $s_k = \mathrm{s}_e^n$. Every vertex x of G is assigned an interval $I_x = [\ell_x, r_x]$ where

$$\ell_x =_{\mathrm{def}} \max\{i : s_i = (a_i, e_i) \in S \text{ and } a_i < x \text{ and } e_i < \pi^{-1}(x)\} + 1$$

$$r_x =_{\mathrm{def}} \min\{i : s_i = (a_i, e_i) \in S \text{ and } x < a_i \text{ and } \pi^{-1}(x) < e_i\}\,.$$

It is easy to verify that $\ell_x \le r_x$ always holds. Let $H(P)$ be the interval graph defined by $\{I_x\}_{x\in[n]}$, and let $A_i =_{\text{def}} \{x \in [n] : \ell_x \le i \le r_x\}$ for $i \in [k]$. Note that $A_1, \ldots, A_k$ is a consecutive clique arrangement of $H(P)$.

Lemma 28. *(i) $A_i = V(G[s_{i-1}, s_i])$ for $i \in [k]$.*

(ii) $H(P)$ is a minimal triangulation of G.

(iii) Given π and $\{s_0, \ldots, s_k\}$, $\{I_x\}_{x\in[n]}$ can be computed in time $\mathcal{O}(n)$.

Proof: For statement (i), let $i \in [k]$, and let $s_{i-1} = (a, e)$ and $s_i = (a', e')$. It holds that $x \in V(G[s_{i-1}, s_i])$ if and only if x is between s_{i-1} and s_i or $x \in \mathrm{int}(s_{i-1}) \cup \mathrm{int}(s_i)$ if and only if $(i-1)+1 = \ell_x = r_x = i$ or $\ell_x < (i-1)+1 = i \le r_x$ or $\ell_x \le (i-1)+1 = i < r_x$.

For statement (ii), let $uv \in E$, $u < v$. Observe that $u \in A_{\ell_v}$. Hence, u and v are adjacent in $H(P)$, and $H(P)$ is a triangulation of G. Let $uv \in E(H(P)) \setminus E$. Then, there is $j \in [k-1]$ such that $u, v \in A_j \cap A_{j+1}$, since otherwise u and v are adjacent in G by statement (i) and Lemma 18. Note that there is a vertex in A_j that is not contained in A_{j+1} (A_j contains a vertex of the enclosure of s_j that is to the left of s_j, hence not contained in A_{j+1}); analogously, there is a vertex in A_{j+1} that is not contained in A_j. Then, uv is the unique chord in a cycle of length 4 in $H(P)$. Hence, $H(P)$ is a minimal triangulation of G due to Theorem 2.

For statement (iii), consider a vertex $x \in [n]$. It holds that $\ell_x = \min\{\ell_x^1, \ell_x^2\}$ where

$$\begin{aligned}
\ell_x^1 &=_{\text{def}} \max\{i : s_i = (a_i, e_i) \in S \text{ and } a_i < x\} + 1 \\
&= \min\{i : s_i = (a_i, e_i) \in S \text{ and } a_i > x\} \\
\ell_x^2 &=_{\text{def}} \max\{i : s_i = (a_i, e_i) \in S \text{ and } e_i < \pi^{-1}(x)\} + 1 \\
&= \min\{i : s_i = (a_i, e_i) \in S \text{ and } e_i > \pi^{-1}(x)\}\,.
\end{aligned}$$

A similar equivalence holds for r_x. The $\mathcal{O}(n)$-time algorithm is straight forward, since computing ℓ^1, r^1 and ℓ^2, r^2 is grouping the vertices of G into classes and done by simple sweeps through $1, \ldots, n$ and $\pi(1), \ldots, \pi(n)$, respectively. ∎

Theorem 29. *Let $G = G(\pi)$ be a permutation graph over $[n]$. An interval graph H over $[n]$ is a minimal triangulation of G if and only if there is a maximal path $(x_0, \ldots, x_k)$ in $\mathcal{PC}(\pi)$ such that $A_1, \ldots, A_k$ is a consecutive clique arrangement of H where $A_i =_{\text{def}} V(G[s_{i-1}, s_i])$ and the special scanlines s_{i-1} and s_i are the labels of x_{i-1} and x_i in $\mathcal{PC}(\pi)$ for $i \in [k]$.*

Proof: By Lemma 28, the discussions and constructions before, every path of $\mathcal{PC}(\pi)$ defines a minimal triangulation of G in the required sense. So, let H be a minimal triangulation of G, and let $A_1, \ldots, A_k$ be a consecutive clique arrangement of H. Let $S_0 =_{\text{def}} S_k =_{\text{def}} \emptyset$ and $S_i =_{\text{def}} A_i \cap A_{i+1}$, $i \in [k-1]$. By Theorem 4, $S_1, \ldots, S_{k-1}$ are the minimal separators of H, and by Theorem 5, $\{S_1, \ldots, S_{k-1}\}$ is a maximal set of pairwise non-crossing minimal separators of G. Since $A_1, \ldots, A_k$ are potential maximal cliques of G, there are special scanlines $s_1, s_1', s_2, \ldots, s_k'$ of G such that $A_i = V(G[s_i, s_i'])$ and $s_i \in N_\pi^{\le}(s_i')$ for $i \in [k]$ due to Theorem 24. Suppose there are $i, j \in [k]$, $i < j$, such that s_j or s_j' intersects with s_i or s_i'. Since an inner vertex of the one set does not belong to the other set, A_i and A_j do not contain inner vertices by Lemma 19. Hence, $1 < i < j < k$ and $A_i = S_{i-1} \cup S_i$ and $A_j = S_{j-1} \cup S_j$. By Lemma 23, $\{\mathrm{int}(s_i), \mathrm{int}(s_i')\} = \{S_{i-1}, S_i\}$ and $\{\mathrm{int}(s_j), \mathrm{int}(s_j')\} = \{S_{j-1}, S_j\}$. By Theorem 5 and Lemma 14, s_i, s_i', s_j, s_j' are pairwise non-intersecting. So, $S =_{\text{def}} \{s_1, \ldots, s_k'\}$ is a set of pairwise non-intersecting special scanlines of G. Suppose S is properly contained in a maximal set S' of pairwise non-intersecting special scanlines of G. S' uniquely defines a path in $\mathcal{PC}(\pi)$ by Lemma 26, and let $\mathcal{C}_{S'}$ be the set of potential maximal cliques defined by S' (i.e., on the path of $\mathcal{PC}(\pi)$ defined by S'). It holds that $\mathcal{C}_S =_{\text{def}} \{A_1, \ldots, A_k\} \subseteq \mathcal{C}_{S'}$. By Lemma 28, $\mathcal{C}_{S'}$ defines a minimal triangulation H' of G, and since $\mathcal{C}_S \subseteq \mathcal{C}_{S'}$, H is a subgraph of H'. Then, H and H' are equal, and $\mathcal{C}_S$ and $\mathcal{C}_{S'}$ are equal, and the theorem holds. ■

We have seen that we can compute a minimal triangulation of a permutation graph in linear time using its potential maximal cliques graph. However, given a permutation sequence, faster algorithms are desirable. Lemma 28 shows that the interval model of a minimal triangulation of a permutation graph can be generated in time $\mathcal{O}(n)$, if the defining set of special scanlines is known. We will show that we can compute in time $\mathcal{O}(n)$ such a set of special scanlines.

Lemma 30. *For a permutation graph $G = G(\pi)$ over $[n]$, given by π, a maximal set of pairwise non-intersecting special scanlines of G can be computed in time $\mathcal{O}(n)$.*

Proof: Note that there cannot be more than $n{+}1$ pairwise non-intersecting special scanlines of G. The algorithm to be presented works in two steps: first, a set of pairwise non-intersecting special scanlines is computed, and second, the set is completed to be maximal in a very special way. To accomplish the first step, we make the following definitions. For convenience, let $\pi(0) =_{\text{def}} 0$ and $\pi(n{+}1) =_{\text{def}} n{+}1$. For every $x \in [n{+}1]$,

$$S(x) =_{\text{def}} \{z : \exists v(z < v \text{ and } \pi^{-1}(z) < \pi^{-1}(v) \leq x\} \cup \{0\}$$
$$A(x) =_{\text{def}} \max S(x)$$
$$C =_{\text{def}} \left\{ \left(A(x){+}\frac{1}{2}, x{-}\frac{1}{2}\right) : x \in [1, n{+}1] \text{ and } \pi(x-1) < \pi(x) \text{ and } A(x) < \pi(x) \right\}.$$

Observe that $\mathrm{s}_0, \mathrm{s}_e^n \in C$. Furthermore, $S(x) \subseteq S(x+1)$ for all $x \in [1, n]$.

(*a*) The scanlines in C are pairwise non-intersecting: This is an immediate consequence of the monotonicity of S.

(*b*) C contains only special scanlines: Let $x \in [2, n]$ such that $(A(x){+}\frac{1}{2}, x{-}\frac{1}{2}) \in C$. In particular, $\pi(x{-}1) < \pi(x)$, i.e., $\pi(x{-}1) \in S(x)$. By definition of $S(x)$, $\pi(x-1) \leq A(x)$ and $\pi^{-1}(A(x)) < x$. Since $A(x) < \pi(x)$, if $\pi^{-1}(A(x)+1) < x$ then $A(x){+}1 < \pi(x)$ and $A(x){+}1 \in S(x)$, contradicting $A(x) = \max S(x)$. Hence, $(A(x){+}\frac{1}{2}, x{-}\frac{1}{2})$ is a special scanline of G.

(*c*) If there is a special scanline $s = (a, e) \notin C$ of G that does not intersect with any scanline in C, then there is a scanline $s' = (a', e') \in C$ such that $a < a'$ and $e = e'$: Let s be as defined. Suppose there is no $s' = (a', e') \in C$ such that $e = e'$. Then, it holds that $\pi(e{-}\frac{1}{2}) < \pi(e{+}\frac{1}{2})$ and $A(e{+}\frac{1}{2}) > \pi(e{+}\frac{1}{2})$. Note that $a \leq \pi(e{+}\frac{1}{2})$. Let $x \in [1, n]$ be smallest such that $A(x) = A(e{+}\frac{1}{2})$. Then, $1 < x < e$. It holds that $\pi(x{-}1) < \pi(x)$ and $A(x) < \pi(x)$, so that $(A(x){+}\frac{1}{2}, x{-}\frac{1}{2}) \in C$. But then, s intersects with a scanline in C. Now, suppose there is $s' = (a', e') \in C$ such that $e = e'$ and $a' < a$. Then, $a{-}\frac{1}{2} \in S(e{+}\frac{1}{2})$, i.e., $A(e{+}\frac{1}{2}) > a'$, and $s' \notin C$.

Observe that scanline s' of the last item is unique.

During the second step of our algorithm, for every $a \in [1, n{-}1]$, check whether scanline $(a{+}\frac{1}{2}, e)$ is special where e is smallest such that there is $(a', e) \in C$ and $a < a'$. Adding these special scanlines to the computed set, we obtain C' as a maximal set of pairwise non-intersecting special scanlines of G according to property (c).

It remains to show the running time. Obviously, the second step can be performed in time $\mathcal{O}(n)$, since the scanlines in C uniquely partition the numbers in $[1, n{-}1]$, and this

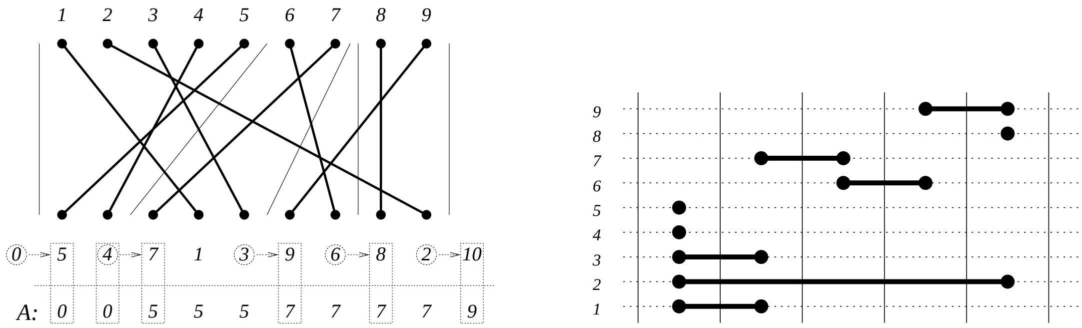

Figure 3. Computing a minimal triangulation of a permutation graph.

partition can be computed in time $\mathcal{O}(n)$. To compute A we consider another definition of A. For $x \in [1, n]$,

$$\begin{aligned} A(1) &= 0 \\ A(x{+}1) &= \max\{z : \pi^{-1}(z) < x{+}1 \text{ and } z < \pi(x{+}1)\} \cup \{A(x)\} \\ &= \max\{z : \pi^{-1}(z) < x{+}1 \text{ and } A(x) < z < \pi(x{+}1)\} \cup \{A(x)\}\,. \end{aligned}$$

With this definition, A can be computed in time $\mathcal{O}(n)$ using a stack that contains the candidates for $A(x{+}1)$ in increasing order. ■

Corollary 31. *An interval model of a minimal triangulation of a permutation graph over* $[n]$ *that is given by a permutation sequence can be computed in time* $\mathcal{O}(n)$.

Proof: This is an immediate consequence of Lemmata 30 and 28. ■

The following observation links permutation models and interval models for chordal permutation graphs in an interesting way.

Corollary 32. *An interval model of a chordal permutation graph over* $[n]$ *that is given by a permutation sequence can be computed in time* $\mathcal{O}(n)$.

Let us discuss the last results on a small example. In Figure 3, we find a permutation diagram on the left side, and we want to compute a minimal triangulation of the related graph by applying the algorithm of the proof of Lemma 30. On the lower end of the diagram, the values of function A are noted. The two conditions of the definition of set C are marked by small rectangles and circles (and arrows). The scanlines of C are inserted into the diagram as thin line segments. To find a maximal set of pairwise non-intersecting special scanlines, we execute the second step and verify whether $(1{+}\frac{1}{2}, 2{+}\frac{1}{2})$, $(2{+}\frac{1}{2}, 2{+}\frac{1}{2})$, $(3{+}\frac{1}{2}, 2{+}\frac{1}{2})$, $(4{+}\frac{1}{2}, 2{+}\frac{1}{2})$, $(5{+}\frac{1}{2}, 5{+}\frac{1}{2})$, $(6{+}\frac{1}{2}, 5{+}\frac{1}{2})$ are special scanlines. The only special scanline that we find is $(5{+}\frac{1}{2}, 5{+}\frac{1}{2})$. The right picture of Figure 3 shows the constructed interval model. The vertical lines represent the computed scanlines, and the inner four lines also mark the minimal separators of the interval graph in the sense of Theorem 4.

Let $G = G(\pi)$ be a permutation graph over $[n]$. The problem considered in Lemma 30 can be stated differently: Given the scanlines s_0 and s_e^n, compute a maximal set of pairwise non-intersecting special scanlines containing s_0 and s_e^n. A generalization of this problem is obvious. Let $\mathcal{S}$ be a set of special scanlines of G. Compute a maximal set $\mathcal{S}'$ of special scanlines of G such that $\mathcal{S} \subseteq \mathcal{S}'$. We will see that this problem can also be solved in time $\mathcal{O}(n)$. The central part is an $\mathcal{O}(n)$-time algorithm that lists all potential maximal cliques of G with inner vertices. This, however, relies on an algorithm that finds all maximal clique-modules of G.

Definition 4. *Let $G = (V, E)$ be a graph. A set $M \subseteq V$ is a* ***module*** *of G if and only if every vertex of G not contained in M is a neighbour of every vertex in M or of none. A set $C \subseteq V$ is a* ***clique-module*** *of G if and only if C is a module of G and a clique. C is* ***maximal*** *clique-module of G if and only if C is not properly contained in another clique-module of G.*

Clique-modules were first considered by Roberts in connection with indifference graphs [21]. It holds that the set of maximal clique-modules of a graph G partitions the vertex set of G, i.e., every vertex of G is contained in a unique maximal clique-module of G.

Fact 33. *Let $G = (V, E)$ be a graph, and let $C \subseteq V$ be a maximal clique-module of G. Then, $N_G[C]$ is a potential maximal clique of G if and only if for every pair of non-adjacent vertices (u, v) of $N_G[C]$ there is a connected component of $G \setminus N_G[C]$ containing neighbours of u and v.*

Since a maximal clique-module contains only vertices that are pairwise *true twins*, the maximal clique-modules of an arbitrary graph can be listed in linear time. For permutation graphs given by permutation sequences, this time can be improved.

Lemma 34. *Let $G = G(\pi)$ be a permutation graph over $[n]$ that is given by π. The maximal clique-modules of G can be listed in time $\mathcal{O}(n)$.*

Proof: Let $C \subseteq [n]$ be a clique-module. Since every vertex of G that is not contained in C is adjacent to every vertex of G or to none, there are $r, s \in [n]$ and $k \geq 0$ such that $C = \{r, \ldots, r{+}k\}$ and $r{+}i = \pi(s{-}i)$, $i \in \{0, \ldots, k\}$. In other words, the vertices of C appear consecutively in π in reversed order. Then, the algorithm is obvious:

(i) 1 belongs to a maximal clique-module of G,

(ii) for every $i \in \{2, \ldots, n\}$: if $\pi^{-1}(i) + 1 = \pi^{-1}(i{-}1)$, then i belongs to the same maximal clique-module of G as i; otherwise i belongs to a new maximal clique-module of G.

The $\mathcal{O}(n)$-time implementation is straight-forward. ■

For the following results we assume that a permutation sequence π over $[n]$ is also defined for 0 and $n{+}1$ in the following way: $\pi(0) =_{\text{def}} 0$ and $\pi(n{+}1) =_{\text{def}} n{+}1$. If we made this a global assumption, it would immediately be clear why s_0 and s_e^n are special scanlines.

Lemma 35. *Let $G = G(\pi)$ be a permutation graph over $[n]$. Let $C = \{r, \ldots, r{+}k\}$, $r \in [n]$, $k \geq 0$, be a maximal clique-module of G, and let $\pi(s) = r$ for $s \in [n]$. Then, $N_G[C]$ is a potential maximal clique of G if and only if none of the following cases holds:*

(a) $r{-}1 \in N_G[C]$ and $\pi(s{+}1) \in N_G[C]$

(b) $r{+}k{+}1 \in N_G[C]$ and $\pi(s{-}k{-}1) \in N_G[C]$.

Proof: Since C is a maximal clique-module, $r{-}1 \neq \pi(s{+}1)$ and $r{+}k{+}1 \neq \pi(s{-}k{-}1)$. If $r{-}1 \notin N_G[C]$ and $\pi(s{-}k{-}1) \notin N_G[C]$, then $s_1 =_{\text{def}} (r{-}\frac{1}{2}, s{-}k{-}\frac{1}{2})$ is a special scanline of G. Let a_2 and e_2 be smallest such that $a_2 > r$ and $e_2 > s$ and $a_2 \notin N_G[C]$ and $\pi(e_2) \notin N_G[C]$ (since we extended π to $\{0, \ldots, n{+}1\}$, these numbers exist). Let $s_2 =_{\text{def}} (a_2{-}\frac{1}{2}, e_2{-}\frac{1}{2})$. Since $\pi^{-1}(a_2{-}1) \leq s$ and $\pi(e_2{-}1) \leq r$ and $\pi^{-1}(a_2) \geq e_2$ and $\pi(e_2) \geq a_2$, s_2 is a special scanline of G due to Lemma 8. It is clear by the construction that s_1 and s_2 are neighbouring scanlines of G and that $N_G[C] = G[s_1, s_2]$. A similar construction holds if $r{+}k{+}1 \notin N_G[C]$ and $\pi(s{+}1) \notin N_G[C]$.

If $r{-}1 \notin N_G[C]$ and $r{+}k{+}1 \notin N_G[C]$, let e_1 be the largest number such that $e_1 < s$ and $\pi(e_1) \notin N_G[C]$ and let e_2 be the smallest number such that $e_2 > s$ and $\pi(e_2) \notin N_G[C]$.

By a similar argumentation as above, $s_1 =_{\mathrm{def}} (r-\frac{1}{2}, e_1+\frac{1}{2})$ and $s_2 =_{\mathrm{def}} (r+k+\frac{1}{2}, e_2-\frac{1}{2})$ are neighbouring special scanlines of G and $N_G[C] = G[s_1, s_2]$. The case $\pi(s-k-1) \notin N_G[C]$ and $\pi(s+1) \notin N_G[C]$ is similar. Hence, if none of the two cases (a) and (b) holds, $N_G[C]$ is a potential maximal clique of G by Theorem 24.

Now, suppose that none of the four above cases holds, i.e., $r-1$ and $\pi(s+1)$ are contained in $N_G[C]$ or $r+k+1$ and $\pi(s-k-1)$ are contained in $N_G[C]$. Assume the former case; the latter case can be proved analogously. By maximality of C, $r-1 \neq \pi(s+1)$, and since they are neighbours, they are non-adjacent in G. Consider $G' =_{\mathrm{def}} G \setminus N_G[C]$. All vertices from $N_G(r-1)$ that are in G' are to the right of C whereas all vertices from $N_G(\pi(s+1))$ that are in G' are to the left of C. Hence, there is no connected component of G' containing vertices from $N_G(r-1)$ and $N_G(\pi(s+1))$, so that $N_G[C]$ cannot be a potential maximal clique due to Fact 33. ■

Corollary 36. *Let $G = G(\pi)$ be a permutation graph over $[n]$. Then, the potential maximal cliques of G with inner vertices can be listed in time $\mathcal{O}(n)$, where every potential maximal clique is represented in the sense of Theorem 24.*

Proof: Due to Lemmata 34 and 35, in time $\mathcal{O}(n)$ the maximal clique-modules of G that define potential maximal cliques can be listed. It remains to find the special scanlines. The proof of Lemma 35 gives an appropriate algorithm: it suffices to find closest non-neighbours, which can be done in time $\mathcal{O}(n)$ by Lemma 1. ■

Theorem 37. *Let $G = G(\pi)$ be a permutation graph over $[n]$ given by π. Let $\mathcal{S}$ be a set of pairwise non-intersecting special scanlines of G. In time $\mathcal{O}(n)$, a maximal set $\mathcal{S}'$ of special scanlines of G such that $\mathcal{S} \subseteq \mathcal{S}'$ can be computed.*

Proof: Without loss of generality, we can assume that $\mathrm{s}_0, \mathrm{s}_e^n \in \mathcal{S}$. For solving the problem, we give the following first procedure:

(0) verify that $\mathcal{S}$ is a set of pairwise non-intersecting special scanlines of G

(1) compute the potential maximal cliques of G with inner vertices and the corresponding pairs $(s_1^{(i)}, s_2^{(i)})$ of special scanlines of G; let their number be k

(2) compute the maximal $I \subseteq [k]$ such that, for every $i \in I$, $\mathcal{S} \cup \{s_1^{(i)}, s_2^{(i)}\}$ is a set of pairwise non-intersecting scanlines of G

(3) compute a maximal set $J \subseteq I$ such that $\mathcal{S}' =_{\mathrm{def}} \mathcal{S} \cup \{s_1^{(i)}, s_2^{(i)} : i \in J\}$ is a set of pairwise non-intersecting scanlines of G.

In time $\mathcal{O}(n)$, the scanlines in $\mathcal{S}$ can be ordered with respect to $<$, and it can be verified whether they are pairwise non-intersecting. Applying Lemma 8, it can be verified whether every scanline in $\mathcal{S}$ is indeed special. Step (1) can equally be performed in time $\mathcal{O}(n)$ by Corollary 36. For step (2), assume that the scanlines in $\mathcal{S}$ are already ordered according to $<$, say $s'_1 < \cdots < s'_\ell$ for $\ell =_{\mathrm{def}} |\mathcal{S}|$. It holds that $\mathcal{S} \cup \{s_1^{(i)}, s_2^{(i)}\}$, $i \in [k]$, is a set of pairwise non-intersecting scanlines if and only if there is $j \in [\ell-1]$ such that $s'_j \leq s_1^{(i)}$ and $s_2^{(i)} \leq s'_{j+1}$. This can only be the case, if s'_j and s'_{j+1} have no common endpoint. We will show that we can find the correct j, if it exists, in constant time for each pair of scanlines. To achieve this we need an $\mathcal{O}(n)$-time preprocessing. Consider the permutation diagram $\mathfrak{D}$ of G. For every $j \in [\ell-1]$, s'_j and s'_{j+1} define a pair of intervals, one on the upper and one on the lower line of $\mathfrak{D}$: $([a_j, a_{j+1}], [e_j, e_{j+1}])$ where $s'_j = (a_j, e_j)$ and $s'_{j+1} = (a_{j+1}, e_{j+1})$. If one of these intervals is empty, i.e., the corresponding scanlines have a common endpoint, we mark the pair of intervals. We order the unmarked pairs canonically and assign indices to them. In two arrays, corresponding to the upper and lower line of $\mathfrak{D}$, we store the interval index of each number in $[n]$, if it has one. For example, if $([a_2, a_3], [e_2, e_3])$ is unmarked and has index 2 (note that both intervals have the same index), then the numbers $a_2+\frac{1}{2}, a_2+1\frac{1}{2}, \ldots, a_3-\frac{1}{2}$ in the one array, the numbers $e_2+\frac{1}{2}, \ldots, e_3-\frac{1}{2}$ in the other array all have index 2. Now, for $i \in [k]$, $\mathcal{S} \cup \{s_1^{(i)}, s_2^{(i)}\}$ is a set of non-intersecting scanlines of G if and only if $a_1^{(i)}+\frac{1}{2}$, $a_2^{(i)}-\frac{1}{2}$, $e_1^{(i)}+\frac{1}{2}$, $e_2^{(i)}-\frac{1}{2}$, $s_1^{(i)} = (a_1^{(i)}, e_1^{(i)})$ and $s_2^{(i)} = (a_2^{(i)}, e_2^{(i)})$, all have the same index where the first two indices are looked up in the array corresponding to the upper line of $\mathfrak{D}$, the second two indices are looked up in the other array. Hence, step (2) can be performed in time $\mathcal{O}(n)$.

For the final step, observe that it suffices to compute a maximal set $J \subseteq I$ such that $\{s_1^{(i)}, s_2^{(i)} : i \in J\}$ is a set of pairwise non-intersecting special scanlines of G. Note that $\{s_1^{(i)}, s_2^{(i)}, s_1^{(j)}, s_2^{(j)}\}$, $i, j \in I$, $i \neq j$, is not a set of pairwise non-intersecting scanlines of G if and only if the inner vertices of $G[s_1^{(i)}, s_2^{(i)}]$ and $G[s_1^{(j)}, s_2^{(j)}]$ are adjacent, which is an immediate corollary of Lemmata 18 and 19. So, for each potential maximal clique $G[s_1^{(i)}, s_2^{(i)}]$, $i \in I$, select an inner vertex; let that set be $\mathcal{I}$. Then, a maximal independent set of $G(\mathcal{I})$ exactly corresponds to a desired maximal set J. Since step (3) can be carried out in time $\mathcal{O}(n)$, $\mathcal{S}'$ can be computed in time $\mathcal{O}(n)$. Note that it is straight-forward to eliminate multiple copies of one scanline in $\mathcal{S}'$.

It is clear that $\mathcal{S}'$ may not be a maximal set of pairwise non-intersecting special scanlines of G. However, it is easy to find further special scanlines to make $\mathcal{S}'$ maximal. Note that, if $s, s' \in \mathcal{S}'$ are not neighbouring special scanlines of G, there is a sequence of special scanlines $s = r_0, r_1, \ldots, r_q = s'$ of G such that r_{i-1} and r_i are close neighbours and $r_{i-1} < r_i$, $i \in [q]$. We can find such a sequence in the following way. Let $s = (a, e)$. Look for the first scanline in the sequence $(a+1, e), (a, e+1), (a+2, e), (a, e+2), \ldots$ that is a special scanline of G (scanlines that intersect with s' are not considered). This must exist (but may be identical to s') and is a close neighbour of s. Restart the process with the found scanline in place of s as long as possible. It is not hard to see that this process takes time $\mathcal{O}(n)$, since at most $2 \cdot (a' - a + e' - e)$ scanlines are considered between s and $s' = (a', e')$. Hence, in overall time $\mathcal{O}(n)$, we can compute a maximal set of pairwise non-intersecting special scanlines of G containing $\mathcal{S}$. ∎

Corollary 38. *Let $G = G(\pi)$ be a permutation graph over $[n]$ given by π. Let $\mathcal{S}$ be a set of scanlines of G. It can be verified in time $\mathcal{O}(n)$ whether $\mathcal{S}$ is a maximal set of pairwise non-intersecting special scanlines of G.*

6. Minimal triangulations problems. In this section, we will use the results obtained in the previous sections to solve problems fundamental in the minimal triangulations context efficiently, i.e., in linear time, and we want to consider new problems that have not yet been part of considerations.

The probably most fundamental problems are the treewidth and minimum-fill-in problems. We consider both problems as computational problems, i.e., the problems of computing the treewidth and the minimum fill-in of a given graph. Treewidth and minimum fill-in of a graph G can be defined as follows:

$$\mathrm{tw(G)} =_{\mathrm{def}} \min\{\omega(H) : H \text{ is a minimal triangulation of } G\} - 1$$
$$\mathrm{mfi(G)} =_{\mathrm{def}} \min\{|E(H)| - |E(G)| : H \text{ is a minimal triangulation of } G\}.$$

In other words, the treewidth problem looks for a minimal triangulation of the given graph with smallest clique number, and the minimum fill-in problem (also known as *chordal completion*) looks for a minimal triangulation with smallest number of edges. We will show that both graph parameters can be computed in linear time for permutation graphs. It will turn out that our algorithms are *constructive*, which means that they even produce such minimal triangulations as a proof of their result (but they will not prove the correctness of the result). We only remark that in the same manner the *pathwidth* and *interval completion* problems can be defined by replacing *minimal triangulation* by *triangulation that is an interval graph*. Since minimal triangulations of AT-free graphs are always interval graphs, treewidth and pathwidth as well as chordal completion and interval completion are the same notions on AT-free graphs. Bodlaender, Kloks, Kratsch, Müller presented algorithms for computing treewidth and minimum fill-in of permutation graphs with running times $\mathcal{O}(|V(G)| \cdot \mathrm{tw(G)})$ and $\mathcal{O}(n^2)$, respectively, [4]. Our linear-time algorithms for both problems improve their results, also due to Lemma 39.

Lemma 39. **[2]** *Let* $G = (V, E)$ *be a graph, and let* $\mathrm{tw(G)} = k$. *Then,* $|E| \leq k \cdot |V| - \frac{1}{2}k(k+1)$.

Our algorithms work on the potential maximal cliques graph whose vertices and edges are assigned weights. These weights are numbers of vertices of sets that are represented by the vertices and edges. By the definition of the potential maximal cliques graph and Theorem 24, edges define potential maximal cliques, and the edge weights will be the sizes of these sets. The vertices represent minimal separators, and their weights will be the sizes of the corresponding separators. We begin by showing that the weights can be assigned in linear time. Let $G(\pi)$ be a permutation graph over $[n]$. We need two auxiliary sets. Let $s = (a, e)$ be a scanline of $G(\pi)$. By $\overline{\mathrm{int}}^{<}(s) =_{\mathrm{def}} \{x < a : \pi^{-1}(x) < e\}$, we denote the set of all vertices of G that are smaller than a and that are not contained in $\mathrm{int}(s)$. By $\overline{\mathrm{int}}^{>}(s) =_{\mathrm{def}} \{x > a : \pi^{-1}(x) > e\}$, we denote the set of vertices greater than a outside $\mathrm{int}(s)$.

Lemma 40. *The cardinalities of* int, $\overline{\mathrm{int}}^{<}$ *and* $\overline{\mathrm{int}}^{>}$ *for the special scanlines of a permutation graph can be computed in linear time.*

Proof: Let $G = G(\pi)$ be a permutation graph over $[n]$. By Corollary 9, the set of special scanlines of G can be listed in linear time. We assume that the scanlines are ordered decreasingly by the lower endpoint. We first describe an algorithm for the cardinalities of $\overline{\mathrm{int}}^{<}$. Consider the following algorithm. We will iteratively manipulate a list of sets which is inspired by permutation diagrams. Let $\mathcal{L}(n)$ be an ordered list of singleton sets containing the numbers from 0 to n. Assign to every set the contained number as label. Obtain $\mathcal{L}(i)$ from $\mathcal{L}(i{+}1)$, $i \in [n{-}1]$, by the following procedure:

(a) find set L in $\mathcal{L}(i{+}1)$ containing $\pi(i{+}1)$ (it will be the smallest element in that set); let it have label a

(b) find set L' in $\mathcal{L}(i{+}1)$ with the largest label smaller than a

(c) add the elements of L to L'

(d) decrease the labels of all sets in $\mathcal{L}(i{+}1)$ with label greater than a by 1

(e) delete L.

For every $\mathcal{L}(i)$, $i \in [n]$, the labelling numbers of the sets are ordered increasingly and successively (remember that we have lists whose elements are ordered by their labels). It is the case that the cardinality of $\overline{\mathrm{int}}^{<}(s)$ for a scanline $s = (a{+}\frac{1}{2}, e{+}\frac{1}{2})$ is the label of the set in $\mathcal{L}(e)$ that contains a. Our algorithm corresponds to the following procedure on permutation diagrams: for $i \in [n]$ in decreasing order, in step i delete vertex $\pi(i)$. For a scanline $s = (a, e)$, all vertices x where $\pi^{-1}(x) > e$ do not contribute to $\overline{\mathrm{int}}^{<}(s)$.

We first show that $\mathcal{L}(n), \ldots, \mathcal{L}(1)$ can be computed in linear time. We do not keep every $\mathcal{L}(i)$ but manipulate on the same object. In an array, pointers to the list elements are kept to achieve constant time access to the sets by given label. Every set is kept as a chained list. Then, it is clear that steps (b) and (c) take only constant time, since L' has label $a{-}1$ and computing the union in step (c) is mere pointer manipulation. However, instead of adding L to L', we add L' to L. For changing the labels in step (d), pointers in the array and the labels themselves have to be adjusted. Observe that every set in $\mathcal{L}(i{+}1)$ with label greater than a contains a neighbour of $\pi(i)$, which gives linear time. To provide constant time access to the label of the set containing a given number, we keep another array containing a pointer for each number from $[n]$ to the set containing it. Steps (c) and (e) require updating these pointers, but since all but the smallest elements in L' are neighbours of $\pi(i)$ (they are smaller than $\pi(i)$ but have been deleted before), we still have linear time.

To compute $\overline{\mathrm{int}}^{>}(s)$ for the special scanlines s of G, we use the same algorithm on permutation sequence $\pi' =_{\mathrm{def}} \langle n{+}1{-}\pi(n), \ldots, n{+}1{-}\pi(1)\rangle$. Note that $G(\pi)$ and $G(\pi')$ are isomorphic and that $S = (a, b)$ is a special scanline in $G(\pi)$ if and only if $s' = (n{+}1{-}a, n{+}1{-}e)$ is a special scanline of $G(\pi')$. Finllay, it holds that $|\mathrm{int}(s)| = n - |\overline{\mathrm{int}}^{<}(s)| - |\overline{\mathrm{int}}^{>}(s)|$. ■

A final remark on the provided implementation in the proof of Lemma 40: It is not absolutely necessary to keep the numbers in chained lists, since the second array may provide all information via a pointer to the list element containing the label and the smallest element in the assigned set.

Lemma 41. *Let $G = G(\pi)$ be a permutation graph over $[n]$. Let s_1 and s_2 be special scanlines such that $s_1 \in N_\pi^{<}(s_2)$. For $C =_{\mathrm{def}} V(G[s_1, s_2])$, $S_1 =_{\mathrm{def}} \mathrm{int}(s_1)$ and $S_2 =_{\mathrm{def}} \mathrm{int}(s_2)$, it holds: $|C| = \frac{1}{2} \cdot (|S_1| + |S_2| + (a_2 - a_1) + (e_2 - e_1))$.*

Proof: Every vertex in $S_1 \cap S_2$ contributes exactly 2 to $|S_1| + |S_2|$. Every vertex in $C \setminus (S_1 \cup S_2)$ contributes exactly 2 to $(a_2 - a_1) + (e_2 - e_1)$. Every vertex in $(S_1 \cup S_2) \setminus (S_1 \cap S_2)$ contributes exactly 1 to $|S_1| + |S_2|$ and $(a_2 - a_1) + (e_2 - e_1)$. Vertices not contained in C do not contribute to the sum. ■

The *weighted* potential maximal cliques graph of a permutation graph $G(\pi)$ is obtained from $\mathcal{PC}(\pi)$ by assigning to its vertices the sizes of the corresponding minimal separators and to its arcs the sizes of the corresponding potential maximal cliques.

Corollary 42. *The weighted potential maximal cliques graph of a permutation graph can be computed in linear time.*

Proof: This follows from Lemmata 26, 40 and 41 and Corollaries 9 and 25. ■

Theorem 43. *Treewidth and minimum fill-in for permutation graphs can be computed in linear time.*

Proof: Let $G = G(\pi)$ be a permutation graph over $[n]$. By Theorem 29, the treewidth of G is realised by a minimal triangulation of G that is represented by a maximal path of the weighted potential maximal cliques graph of G with the least maximal arc weight. Constructing the weighted potential maximal cliques graph of G and finding such a path can be done in linear time by traversing the weighted potential maximal cliques graph following a topological ordering of $\mathcal{PC}(\pi)$.

For computing the minimum fill-in, consider the following observation. Let $A_1, \ldots, A_k$ be a consecutive clique arrangement of an interval graph H. For every $i \in [k]$:

$$|E(H[A_0 \cup \cdots \cup A_i])| + |E(H[A_{i-1} \cap A_i])| = |E(H[A_0 \cup \cdots \cup A_{i-1}])| + |E(H[A_i])|\,,$$

where $A_0 =_{\text{def}} \emptyset$. Since $H[A_{i-1} \cap A_i]$ and $H[A_i]$ are complete graphs and if the numbers of vertices of these graphs are known, it is easy to determine the numbers of edges of these graphs. Hence, a path of $\mathcal{PC}(\pi)$ that represents a minimal triangulation of G with the smallest number of edges among all minimal triangulations of G can be found similarly to the treewidth algorithm above. ■

The algorithmic approach of Theorem 43 can be used to solve a number of further, related problems. Let $G = G(\pi)$ be a permutation graph over $[n]$. We define the following problems:

(P1) Minimal triangulation of G with the smallest number of maximal cliques;

(P2) Minimal triangulation of G with the largest number of maximal cliques;

(P3) Minimal triangulation of G with the largest clique number among all minimal triangulations of G;

(P4) Minimal triangulation of G with the largest number of edges among all minimal triangulations of G;

(P5) Number of minimal triangulations of G.

Theorem 44. *The problems (P1)–(P5) can be solved in linear time for permutation graphs.*

Proof: Let $G = G(\pi)$ be a permutation graph over $[n]$, and let $\mathcal{PC}(\pi)$ be the potential maximal cliques graph of G. Obviously, problems (P1) and (P2) are solved by finding a shortest and longest path in $\mathcal{PC}(\pi)$, respectively. Problem (P3) is easily solved by determining an arc with the largest weight, and for problem (P4), we apply an algorithm similar to that for minimum fill-in but instead of choosing the minimum among the possible numbers of edges we choose the maximum. For problem (P5), note that the number of minimal triangulations of G corresponds to the number of maximal paths in $\mathcal{PC}(\pi)$. This number can be computed in the following way: the source vertex of $\mathcal{PC}(\pi)$ is assigned 1, and every other vertex, considered following a topological ordering, is assigned the sum of the number assigned to its predecessor vertices. The number assigned to the sink is the number of maximal paths. ■

Note that problems (P1) and (P2) can be formulated also about minimal separators; however, these problems seem more complicated since the same minimal separator may "appear" more than once in the interval model.

There are also minimal triangulations problems that seem to need more complicated algorithms. Let $\mathcal{MT}(G)$ denote the set of minimal triangulations of G.

(P6) Smallest number k such that there are $H_1, \ldots, H_k \in \mathcal{MT}(G)$ such that

$$G = \bigcap_{i=1}^{k} H_i\,,$$

i.e., the interval dimension of G;

(P7) Minimal independent set number among all minimal triangulations of G;

(P8) Smallest distance of (different) edge numbers among the minimal triangulations of G, i.e.,

$$\min\{|E(H_1)| - |E(H_2)| : H_1, H_2 \in \mathcal{MT}(g) \text{ and } H_1 \neq H_2\}\,;$$

(P9) $\min\{|E(H_1) \cap E(H_2)| > 0 : H_1, H_2 \in \mathcal{MT}(G)\}$;

(P10) $\max\{|E(H_1) \cap E(H_2)| : H_1, H_2 \in \mathcal{MT}(G)\}$;

(P11) For $H \in \mathcal{MT}(G)$:

(a) $\min\{||E(H)| - |E(H')|| : H' \in \mathcal{MT}(G) \setminus \{H\}\}$,

(b) $\min\{||E(H)| - |E(H')|| > 0 : H' \in \mathcal{MT}(G) \setminus \{H\}\}$,

(c) $\max\{||E(H)| - |E(H')|| : H' \in \mathcal{MT}(G)\}$.

All stated problems can be considered also as constructive, i.e., including the problem of finding appropriate minimal triangulations of G.

Bibliography.

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

[2] H.L. Bodlaender, *A linear-time algorithm for finding tree-decompositions of small treewidth*, SIAM Journal on Computing 25, No. 6, pp. 1305–1317, 1996.

[3] H.L. Bodlaender, T. Kloks, D. Kratsch, *Treewidth and Pathwidth of Permutation Graphs*, SIAM Journal on Discrete Mathematics 8, No. 4, pp. 606–616, 1995.

[4] H.L. Bodlaender, T. Kloks, D. Kratsch, H. Müller, *Treewidth and minimum fill-in on d-trapezoid graphs*, Journal of Graph Algorithms and Applications 2, No. 3, pp. 1–23, 1998.

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

[6] V. Bouchitté, I. Todinca, *Treewidth and Minimum Fill-in: Grouping the Minimal Separators*, SIAM Journal on Computing 31, pp. 212–232, 2001.

[7] D.G. Corneil, St. Olariu, L. Stewart, *Asteroidal triple-free graphs*, SIAM Journal on Discrete Mathematics 10, No. 3, pp. 399–430, 1997.

[8] D.G. Corneil, St. Olariu, L. Stewart, *Linear time algorithms for dominating pairs in asteroidal triple-free graphs*, SIAM Journal on Computing 28, No. 4, pp. 1284–1297, 1999.

[9] G.A. Dirac, *On rigid circuit graphs.*, Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg 25, pp. 71–76, 1962.

[10] D.R. Fulkerson, O.A. Gross, *Incidence matrices and interval graphs*, Pacific Journal of Mathematics 15, pp. 835–855, 1965.

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

[12] M. Habib, R.H. Möhring, *Treewidth of cocomparability graphs and a new order-theoretic parameter*, Technical report 336/1992, Fachbereich 3 Mathematik, Technische Universität Berlin, 1992.

[13] T. Kloks, H.L. Bodlaender, H. Müller, D. Kratsch, *Computing treewidth and minimum fill-in: All you need are the minimal separators*, Proceedings of the 1st Annual European Symposium on Algorithms, ESA'93, Lecture Notes in Computer Science 726, pp. 260–271, 1993.

[14] T. Kloks, H.L. Bodlaender, H. Müller, D. Kratsch, *Erratum to the ESA'93 proceedings*, Proceedings of the 2nd Annual European Symposium on Algorithms, ESA'94, Lecture Notes in Computer Science 855, p. 508, 1994.

[15] T. Kloks, D. Kratsch, *Listing all minimal separators of a graph*, SIAM Journal on Computing 27, No. 3, pp. 605–613, 1998.

[16] T. Kloks, D. Kratsch, H. Müller, *On the structure of graphs with bounded asteroidal number*, Graphs and Combinatorics 17, pp. 295–306, 2001.

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

[18] C.G. Lekkerkerker, J.Ch. Boland, *Representation of finite graphs by a set of intervals on the real line*, Fundamenta Mathematicae 51, pp. 45–64, 1962.

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

[20] A. Parra, P. Scheffler, *Characterizations and algorithmic applications of chordal graph embeddings*, Discrete Applied Mathematics 79, pp. 171–188, 1997.

[21] F.S. Roberts, *Indifference graphs*, in: F. Harary (Ed.), Proof techniques in graph theory, pp. 139–146, Academic Press, New York, 1969.

[22] D.J. Rose, R.E. Tarjan, G.S. Lueker, *Algorithmic aspects of vertex elimination on graphs*, SIAM Jounal on Computing 5, pp. 266–283, 1976.

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