A partial k-arboretum of graphs with bounded treewidth

Hans L. Bodlaender*

Abstract

The notion of treewidth has seen to be a powerful vehicle for many graph algorithmic studies. This survey paper wants to give an overview of many classes of graphs that can be seen to have a uniform upper bound on the treewidth of graphs in the class. Also, some mutual relations between such classes are discussed.

1 Introduction

Many recent investigations in algorithmic graph theory have been on the topic of graphs with bounded treewidth. For instance, it appears that many problems that are intractable (e.g., NP-hard) for general graphs become polynomial or linear time solvable, when restricted to graphs of bounded treewidth. (See [19] for an overview.) Clearly, such results also hold for any class of graphs with the property that there is a uniform upper bound on the treewidth of the graphs in the class. Thus, it is interesting to know for a class of graphs, when such a uniform upper bound exists. This paper gives an overview of several such classes of graphs.

The notion of treewidth was introduced by Robertson and Seymour [75], and it plays an important role in their fundamental work on graph minors. The related notion of pathwidth was also introduced by Robertson and Seymour [70]. It appears that there are many other graph theoretic notions, that can be seen to be equivalent to either treewidth or pathwidth. In several cases, these notions have been studied independently, and only after some time the equivalence was realized.

The following notations and definitions will be used throughout this paper. Unless specified otherwise, a graph $G = (V, E)$ is an undirected graph without self loops or parallel edges, with vertex set V and edge set E. If not specified otherwise, n denotes the number of vertices of graph $G = (V, E)$, i.e., $n = |V|$. The subgraph of $G = (V, E)$, induced by W is denoted by $G[W] = (W, \{(v, w) \in E \mid v \in W \wedge w \in W\})$.

*Department of Computer Science, Utrecht University, P.O. Box 80.089, 3508 TB Utrecht, the Netherlands. Email: hansb@cs.ruu.nl.

2 Treewidth and pathwidth

In this section we give the definitions of the notions 'treewidth' and 'pathwidth' of a graph, and summarise some results, discussed in later sections of this paper, and give some useful lemmas. The notions of treewidth and pathwidth were introduced by Robertson and Seymour [70, 75].

Definition. A *tree decomposition* of a graph $G = (V, E)$ is a pair $(\{X_i \mid i \in I\}, T = (I, F))$ with $\{X_i \mid i \in I\}$ a family of subsets of V, one for each node of T, and T a tree such that

- $\bigcup_{i \in I} X_i = V$.
- for all edges $(v, w) \in E$, there exists an $i \in I$ with $v \in X_i$ and $w \in X_i$.
- for all $i, j, k \in I$: if j is on the path from i to k in T, then $X_i \cap X_k \subseteq X_j$.

The *width* of a tree decomposition $(\{X_i \mid i \in I\}, T = (I, F))$ is $\max_{i \in I} |X_i| \Leftrightarrow 1$. The *treewidth* of a graph G is the minimum width over all possible tree decompositions of G.

One obtains an equivalent definition, when the third condition in the definition of tree decomposition is replaced by:

> For all $v \in V$, the set of nodes $\{i \in I \mid v \in X_i\}$ forms a connected part (i.e., a subtree) of T.

The notion of pathwidth is obtained by restriction of the trees in the tree decompositions to paths.

Definition. A *path decomposition* of a graph $G = (V, E)$ is a sequence of subsets of vertices $(X_1, X_2, \ldots, X_r)$, such that

- $\bigcup_{1 \leq i \leq r} X_i = V$.
- for all edges $(v, w) \in E$, there exists an i, $1 \leq i \leq r$, with $v \in X_i$ and $w \in X_i$.
- for all $i, j, k \in I$: if $i \leq j \leq k$, then $X_i \cap X_k \subseteq X_j$.

The *width* of a path decomposition $(X_1, X_2, \ldots, X_r)$ is $\max_{1 \leq i \leq r} |X_i| \Leftrightarrow 1$. The pathwidth of a graph G is the minimum width over all possible path decompositions of G.

Note that each path decomposition can be written as a tree decomposition with T a path. Throughout this paper we will use the term 'vertices' to denote the vertices $v \in V$ of graphs $G = (V, E)$, and the term 'nodes' to denote the vertices '$i \in I$' of the decomposition tree $T = (I, F)$.

There are a large number of equivalent characterisations of the notions 'treewidth' and 'pathwidth'. We summarise these in the following theorems. For details, see later sections.

Theorem 1 *Let $G = (V, E)$ be a graph, and let $k \geq 0$. The following statements are equivalent:*

1. *The treewidth of G is at most k.*
2. *G is a partial k-tree.*
3. *G is a subgraph of a chordal graph with maximum clique size $k+1$.*
4. *The dimension of G is at most k.*
5. *G is k-decomposable.*
6. *The tangle number of G is at most k.*
7. *G has a convex tangle of order at most k.*
8. *$k+1$ cops can search G in the Seymour-Thomas search game.*
9. *$k+1$ cops can monotonely search G in the Seymour-Thomas search game.*
10. *The number of searchers, needed to search G in the fugitive search game with an inert fugitive is at most $k+1$.*
11. *The number of searchers, needed to monotonically search G in the fugitive search game with an inert fugitive is at most $k+1$.*

Theorem 2 *Let $G = (V, E)$ be a graph, and let $k \geq 0$. The following statements are equivalent:*

1. *The pathwidth of G is at most k.*
2. *The vertex separation number of G is at most k.*
3. *The interval thickness of G is at most $k+1$.*
4. *The node search number of G is at most $k+1$.*
5. *The minimum progressive black pebble demand over all directives of G is at most $k+1$.*
6. *The minimum progressive black and white pebble demand over all directives of G is at most $k+1$.*

The following lemma follows directly from the definitions.

Lemma 3 *For all graphs G, pathwidth$(G) \geq$ treewidth(G).*

A short proof of the following lemma can be found in [23]. Older, but longer proofs can be found in [15, 94].

Lemma 4 (See [23]) *Let $(\{X_i \mid i \in I\}, T = (I,F))$ be a tree decomposition of $G = (V,E)$, and let $W \subseteq V$ be a clique in G. Then there exists an $i \in I$ with $W \subseteq X_i$.*

Lemma 5 (Bodlaender, Möhring [23]) *Let $(\{X_i \mid i \in I\}, T = (I,F))$ be a tree decomposition of $G = (V,E)$. Let $W_1, W_2 \subseteq V$, and suppose $\{(v,w) \mid v \in W_1,\ w \in W_2\} \subseteq E$. Then there exists an $i \in I$ with $W_1 \subseteq X_i$ or $W_2 \subseteq X_i$.*

Proof: (The following short proof was noticed by Ton Kloks.) Suppose not. Consider $H = (V, E')$, with $E' = \{(v,w) \mid \exists i \in I : v,w \in X_i,\ v \neq w\}$. As in the proof of Theorem 27, H is chordal, and contains G as a subgraph. By Lemma 4, there are $v_1, w_1 \in W_1$, $v_1 \neq w_1$, with $(v_1, w_1) \notin E'$, and $v_2, w_2 \in W_2$, $v_2 \neq w_2$ with $(v_2, w_2) \notin E'$. This implies that v_1, w_1, v_2, w_2 form a chordless cycle of length four in H, contradicting the chordality of H. $\Box$

Lemma 6 (See [14, 94]) *For every graph $G = (V,E)$:*

1. *The treewidth of G equals the maximum treewidth of its connected components.*
2. *The pathwidth of G equals the maximum pathwidth of its connected components.*
3. *The treewidth of G equals the maximum treewidth of its biconnected components.*

Proof: (i) Given tree decompositions of all connected components of a graph G, a tree decomposition of G can be formed by taking the disjoint union of the tree decompositions, and then adding arbitrarily some edges between nodes in the disjoint trees to make a tree from this forest.

(ii) Similar.

(iii) Observe that a graph G is chordal, if and only if each biconnected component of G is chordal, and that the maximum clique size of G equals the maximum of the maximum clique size over all biconnected components. Then use Theorem 27. (Alternatively, one can give a direct construction.) $\Box$

As trees can have arbitrary large pathwidth, the pathwidth of a graph does not necessarily equal the maximum pathwidth of its biconnected components. The following lemma is useful for the design of algorithms on graphs with bounded treewidth. It can be noted, that a tree decomposition can be transformed into one of this form in linear time.

Lemma 7 (See [56]) *Suppose the treewidth of $G = (V,E)$ is at most k. G has a tree decomposition $\{(X_i \mid i \in I\}, T = (I,F))$, of width k, such that* *a root r of T can be chosen, such that every node $i \in I$ has at most two children in the rooted tree T with r as root, and*

- *If a node $i \in I$ has two children j_1, j_2, then $X_{j_1} = X_{j_2} = X_i$. (i is called a* **join** *node.)*
- *If a node $i \in I$ has one child j, then either $X_i \subset X_j$ and $|X_i| = |X_j| \Leftrightarrow 1$ (i is called a* **forget** *node), or $X_j \subset X_i$ and $|X_j| = |X_i| \Leftrightarrow 1$ (i is called an* **introduce** *node).*
- *If a node $i \in I$ is a leaf of T, then $|X_i| = 1$. (i is called a* **leaf** *node.)*
- *$|I| = O(k \cdot |V|)$.*

A tree decomposition of the form, described in the Lemma 7 above, is called *nice*.

Lemma 8 *Suppose the treewidth of $G = (V, E)$ is k. G has a tree decomposition $\{(X_i \mid i \in I\}, T = (I, F))$ of width k, such that*

- *For all $i \in I$: $|X_i| = k + 1$.*
- *For all $(i, j) \in F$: $|X_i \cap X_j| = k$.*

Proof: Take an arbitrary tree decomposition of G of width k, and repeatedly apply the following operations, until none is possible. The resulting tree decomposition will satisfy the conditions.

- If $(i, j) \in F$, and $X_i \subseteq X_j$ or $X_j \subseteq X_i$, then 'contract the edge (i, j)': replace X_i and X_j by one set $X_{i'} = X_i \cup X_j$, with i' adjacent to all nodes that were adjacent to i or j.
- If $(i, j) \in F$, $|X_i| < k + 1$, and $X_j \not\subseteq X_i$, then add a vertex $v \in X_j \Leftrightarrow X_i$ to X_i.
- If $(i, j) \in F$, $|X_i| = |X_j| = k + 1$, and $|X_i \cap X_j| < k$, then choose a vertex $v \in X_j \Leftrightarrow X_i$, $w \in X_i \Leftrightarrow X_j$, and let $X_{i'} = X_i \Leftrightarrow \{w\} \cup \{v\}$. Replace the edge (i, j) in T by edges (i, i') and (i', j). (So, i' is a new node in T.)

$\square$

Several other similar results can be derived.

3 Branchwidth

The notion of branchwidth was also introduced by Robertson and Seymour [82, 90].

Definition. A *branch decomposition* of a graph $G = (V, E)$ is a pair $(T = (I, F), \sigma)$, where T is a tree with every node in T of degree one or three, and σ is a bijection from E to the set of leaves in T.

The *order* of an edge $f \in F$ is the number of vertices $v \in V$, for which there exist adjacent edges $(v, w), (v, x) \in E$, such that the path in T from $\sigma(v, w)$ to $\sigma(v, x)$ uses f.

The *width* of branch decomposition $(T = (I, F), \sigma)$, is the maximum order over all edges $f \in F$. The *branchwidth* of G is the minimum width over all branch decompositions of G.

The following relationship between treewidth and branchwidth was presented in a somewhat more general form (for hypergraphs) in [82].

Theorem 9 (Robertson, Seymour [82]) *Let $G = (V, E)$ be a graph with treewidth k, and branchwidth l, $E \neq \emptyset$. Then* $\max(l, 2) \leq k + 1 \leq \max(\lfloor 3/2 \cdot l \rfloor, 2)$.

Graphs of small branchwidth are characterised by the following theorem.

Theorem 10 (Robertson, Seymour [82]) *(i) A graph G has branchwidth 0, if and only if every connected component of G contains at most one edge.*
(ii) A graph G has branchwidth at most 1, if and only if every connected component of G has at most one vertex of degree at least two.
(iii) A graph G has branchwidth at most 2, if and only if G does not contain K_4 as a minor.

See Section 4 for the notion of minor. As a direct corollary from Theorem 10(iii) and Theorem 17(ii), we have that a graph has branchwidth at most 2, if and only if it has treewidth at most 2. (This relationship is not know to hold for values other than 2, e.g., a tree with at least two non-leaves has treewidth 1 and branchwidth 2.)

4 Subgraphs and minors

In this section we give a simple lemma for the treewidth and pathwidth of subgraphs, and give several relations between treewidth, pathwidth, and graph minors.

Lemma 11 (See [94]) *Let $G = (V, E)$ be a subgraph of $H = (V', E')$. Then treewidth$(G) \leq$ treewidth(H), and pathwidth$(G) \leq$ pathwidth(H).*

Proof: If $\{(X_i \mid i \in I\}, T = (I, F))$ is a tree decomposition of G, then $\{X_i \cap V' \mid i \in I\}, T = (I, F))$ is a tree decomposition of H with the same or smaller width. The same argument holds for pathwidth. □

Definition. A graph $G = (V, E)$ is a minor of a graph $H = (W, F)$, if G can be obtained from H by a series of vertex deletions, edge deletions, and edge contractions, where an edge contraction is the operation that replaces two adjacent vertices v, w by one that is adjacent to all vertices that were adjacent to v or w.

In a long series of fundamental papers, Robertson and Seymour obtained several important results on graph minors (among others: [70, 75, 72, 79, 76, 77, 78, 81, 80, 82, 88, 89, 90, 83, 84, 85, 69, 86], see also [71, 73, 74, 87, 92].) Treewidth and pathwidth play an important role in these studies. We mention some of these results, that fit in the framework of this paper.

Theorem 12 (Robertson and Seymour [70]) *For every forest H there is an integer w_H such that every graph with no minor isomorphic to H has pathwidth at most w_H.*

Bienstock et al [13] show that in Theorem 12, one can take $w_H = |V_H| \Leftrightarrow 2$. A simpler and algorithmic proof, but with a much higher constant can be found in [28].

Theorem 13 (Robertson and Seymour [75]) *For every planar graph H there is an integer w_H such that every graph with no minor isomorphic to H has treewidth at most w_H.*

In [91], it is shown that one can take in Theorem 13 $c_H = 20^{4|V_H|+8|E_H|^5}$. A similar type of bound was proved by Gorbunov [44]. As there are forests with arbitrary large pathwidth, it is not possible to prove a variant of Theorem 12 for graphs H that are not a forest. Also, there are planar graphs with arbitrary large treewidth, and hence no variant of Theorem 13 can be proved for graphs that are not planar. Thus, Theorems 12 and 13 are sharp in the sense that they deal with the largest possible classes of graphs.

In some special cases, one can prove better bounds. For instance, for $H = C_k$, the cycle with k vertices, then one can take in Theorem 13, $c_H = k \Leftrightarrow 1$ [39]. For other special cases, similar bounds can be found in [18, 20, 24].

Perhaps the most important result proven by Robertson and Seymour in their series of papers on graph minors is the following.

Theorem 14 (Robertson and Seymour) *Let $G_1, G_2, \ldots$ be a countable sequence of graphs. Then there exist $j < i$ such that G_i is isomorphic to a minor of G_j.*

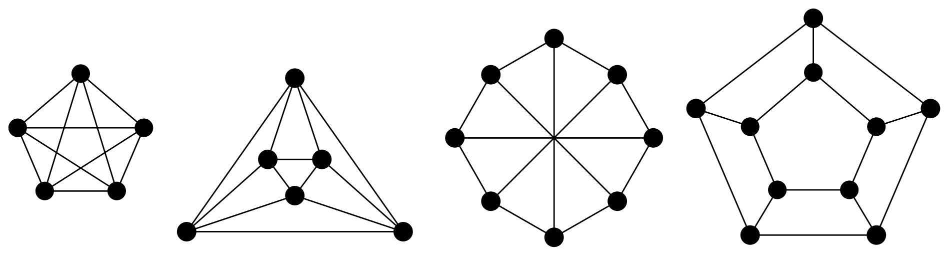
Figure 1: The forbidden minors for graphs with treewidth at most three

An equivalent way of stating this result is the following.

Theorem 15 (Robertson and Seymour) *Let $\mathcal{G}$ be a class of graphs, closed under taking of minors. Then there exists a finite set of graphs, called the* obstruction set*, or the set of* forbidden minors *of $\mathcal{G}$, such that for every graph H: H belongs to $\mathcal{G}$, if and only if H does not contain a graph in the obstruction set of $\mathcal{G}$ as a minor.*

The following well-known result gives a connection between Theorem 15 and the notions 'treewidth' and 'pathwidth'.

Lemma 16 *If G is a minor of H, then treewidth$(G) \leq$ treewidth(H), and pathwidth$(G) \leq$ pathwidth(H).*

Proof: Vertex and edge deletions can be handled as in Lemma 11. If x is obtained by contracting edge (v, w), then we can make a tree or path decomposition of the new graph by replacing all occurrences of v and w in sets X_i by occurrences of x. □

In other words, for every fixed k, the sets $\{G \mid G$ is a graph with treewidth at most $k\}$ and $\{G \mid G$ is a graph with pathwidth at most $k\}$ can be characterised by a finite set of forbidden minors. Some of these characterisations are known.

Theorem 17 *(i) A graph $G = (V, E)$ has treewidth at most 1, if and only if G does not contain K_3 as a minor.*
(ii) A graph $G = (V, E)$ has treewidth at most 2, if and only if G does not contain K_4 as a minor.
(iii) **(Arnborg, Proskurowski, and Corneil [6])** *A graph $G = (V, E)$ has treewidth at most 3, if and only if it does not contain any of the four graphs, shown in Figure 1 as a minor.*

The obstruction sets of graphs with pathwidth 1 and 2 are also known [52]. The size of the obstruction sets can grow very fast: for instance, the obstruction set of the graphs with pathwidth at most k contains at least $k!^2$ trees, each containing $\frac{5 \cdot 3^k - 1}{2}$ vertices [98]. Ramachandramurthi [68] investigated the graphs with $k+1$, $k+2$ and $k+3$ vertices that belong to the obstruction sets for graphs of treewidth or pathwidth k.

5 Separators

Several different, but closely related notions of 'balanced separators' exist. We restrict ourselves to vertex separators, and use the following two notions.

Definition. (i) A type-1 k-separator of a graph $G = (V, E)$ is a set $U \subseteq V$, such that $V \Leftrightarrow U$ can be partitioned into two disjoint sets A and V of at most k vertices each, such that no vertex in A is adjacent to a vertex in B.
(ii) A type-2 k-separator of a graph $G = (V, E)$ is a set $U \subseteq V$, such that each connected component of $G[V \Leftrightarrow U]$ contains at most k vertices.

Clearly, every type-1 k-separator is also a type-2 k-separator. In the other direction, we have the following relationship, which can be easily observed.

Lemma 18 *If S is a type-2 $\frac{1}{2}n$-separator of G, then S is a type-1 $\frac{2}{3}n$-separator of G.*

It was shown by Lipton and Tarjan [61] that every planar graph has a (type-1) $\frac{2}{3}n$-separator of size at most $\sqrt{8n}$. This result is known as the Planar Separator Theorem, see also Section 13.1. The constant factor in this result was later improved, e.g., by Djidjev, who showed that every planar graph has a (type-1) $\frac{2}{3}n$-separator of size at most $\sqrt{6n}$ and for every ϵ, $0 < \epsilon < 1$, a (type-1) ϵn-separator of size at most $4\sqrt{\frac{n}{\epsilon}}$. Related results exist for graphs of bounded genus, see e.g., [41].

Consider a tree decomposition $(\{X_i \mid i \in I\}, T = (I, F))$ of graph $G = (V, E)$. Note that 'most' sets X_i will be separators of G: if $v \in X_{i_1}$, $w \in X_{i_2}$ and i is an internal node on the path from i_1 to i_2 in T, then v and w will belong to different components of $G[V \Leftrightarrow X_i]$. Also, all vertices in one component of $G[V \Leftrightarrow X_i]$ will be in the same subtree of the forest, obtained by removing i from T.

Theorem 19 (See e.g., [75, 42, 62, 22]) *If the treewidth of $G = (V, E)$ is at most k, then G has a type-2 $\frac{1}{2}(n \Leftrightarrow k)$-separator of size at most $k+1$.*

Lingas studied the strongly related classes of $s(N)$-separable graphs [60].

Definition. Let $s : \mathbf{N} \rightarrow \mathbf{N}$ be a function. A graph $G = (V, E)$ is $s(N)$-separable, if it consists of one vertex, or has a type-1 $\frac{2}{3}n$-separator S of size at most $s(n)$, with A and B the two non-adjacent vertex sets that partition $V \Leftrightarrow S$, then $G[A]$ and $G[B]$ are again $s(N)$-separable.

As every subgraph of a planar graph is again planar, the results on separators of planar graphs mentioned above show that planar graphs are $s(N)$-separable, with s defined by $s(n) = \sqrt{6n}$. We will soon see that these results can also be stated in terms of treewidth and pathwidth. See also Theorem 86.

Theorem 20 *(i) A graph $G = (V, E)$ with treewidth k is k-separable.*
(ii) If $G = (V, E)$ is $s(N)$-separable, with s a non-decreasing function $\mathbf{N} \to \mathbf{N}$, then the pathwidth of G is $O(s(n) \cdot \log n)$.
(iii) If $G = (V, E)$ is $s(N)$-separable, with s a non-decreasing function $\mathbf{N} \to \mathbf{N}$, and there exist $c, \epsilon > 0$ with for all $N \in \mathbf{N}$, $s(N) \geq c \cdot n^{\epsilon}$, then the pathwidth of G is $O(s(n))$.

Proof: (i) This follows from Theorem 19.

(ii), (iii). Let S be the indicated separator, and A and B the corresponding vertex sets partitioning $V \Leftrightarrow S$. Recursively make path decompositions $(X_1, \ldots, X_r)$ and $(Y_1, \ldots, Y_q)$ of $G[A]$ and $G[B]$. Then take the path decomposition $(X_1 \cup S, X_2 \cup S, \ldots, X_r \cup S, Y_1 \cup S, \ldots, Y_q \cup S)$. If the maximum width of such a path-decomposition of a graph with n vertices is at most $k(n)$, then we have that $k(n) \leq k(\frac{2}{2}n) + s(n)$. Hence, $k(n) = O(s(n) \cdot \log n)($, and if $s(n) \geq c \cdot n^{\epsilon}$, then $k(n) = O(s(n))$. $\Box$

Similar to Theorems 20(ii) and 20(iii) we have:

Theorem 21 *Let $\mathcal{G}$ be a class of graphs that is closed under taking connected subgraphs. Let $f : \mathbf{N} \to \mathbf{N}$ be a non-decreasing function. Let $0 < c < 1$. Suppose every $G \in \mathcal{G}$ has a type-2 cn-separator of size $f(n)$. Then every $G \in \mathcal{G}$ has pathwidth at most $f(n) \cdot (\lceil {}^{1/c} \log n \rceil + 1)$.*

Theorem 22 *Let $\mathcal{G}$ be a class of graphs, that is closed under taking connected subgraphs. Let $f : \mathbf{N} \to \mathbf{N}$ be a non-decreasing function with*

$$\forall c, 0 < c < 1 : \exists c', 0 < c' < 1 : \forall n \in \mathbf{N} : f(\lceil cn \rceil) \leq c' f(n)$$

Then the following statements are equivalent.

1. *There exists a $c_1 \in \mathbf{N}$ such that for all $G \in \mathcal{G}$: treewidth$(G) \leq c_1 \cdot f(n)$.*

2. *There exists a $c_2 \in \mathbf{N}$ such that for all $G \in \mathcal{G}$: pathwidth$(G) \leq c_2 \cdot f(n)$.*

3. *There exist c_3, $c_4 > 0$, $c_3 < 1$: for all $G \in \mathcal{G}$: G has a type-2 $c_3 \cdot n$-separator of size at most $c_4 \cdot f_n$.*

Proof: (ii) $\Rightarrow$ (i): trivial.

(i) $\Rightarrow$ (iii): Cf. Theorem 19.

(iii) $\Rightarrow$ (ii): Use a construction, similar to the proof of Theorem 20. The resulting treewidth is bounded by $c_4 f(n) + c_4 f(\lfloor c_3 n \rfloor) + c_4 f(\lfloor {c_3}^2 n \rfloor + \cdots = O(f(n))$. $\Box$

Corollary 23 *For every planar graph $G = (V, E)$: pathwidth$(G) = O(\sqrt{n})$.*

Theorem 22 shows that Corollary 23 is in a certain sense nothing else as the Planar Separator Theorem [61] in disguise. See the discussion above and in Section 13.1.

Corollary 24 *For every graph $G = (V, E)$, pathwidth$(G) = O($treewidth$(G) \cdot \log n)$.*

The algorithm of Arnborg et al. [3] to determine in $O(n^{k+2})$ time whether a given graph $G = (V, E)$ has treewidth at most k is based on the characterisation of graphs with treewidth at most k as k-decomposable graphs.

Definition.[See [2]] A graph $G = (V, E)$ is k-decomposable, if one of the following two conditions holds:

1. G has at most $k + 1$ vertices.
2. G has a separator S, $|S| \leq k$, such that the components of $G[V \Leftrightarrow S]$ are $S_1, \ldots, S_m$, and all graphs $(S_i \cup S, \{(v, w) \mid v, w \in S_i \cup S,\ (v, w) \in E$ or $(v \in S$ and $w \in S)\})$ are k-decomposable $(1 \leq i \leq m)$. (I.e., take $G[S \cup S_i]$ and make every two vertices in S adjacent.)

Theorem 25 (Arnborg and Proskurowski [5]) *A graph $G = (V, E)$ is a partial k-tree, if and only if G is k-decomposable.*

(See also [2].) We will later see that a graph is a partial k-tree, if and only if it has treewidth at most k (Theorem 35).

Another related framework has been developed by Hohberg and Reischuk [47, 48].

Definition. A graph $G = (V, E)$ is (k, μ)-*decomposable*, if for any decomposition of G into k-connected components, the size of each component is bounded by μ.

In [47, 48], it is argued that the k-connected components form a tree. This tree can be used, to obtain tree decompositions of (k, μ)-decomposable graphs of treewidth $O(k + \mu)$. Also, graphs with treewidth k are $(O(k), O(k))$-decomposable. For some algorithmic purposes, the approach of Hohberg and Reischuk can have advantages, as it is here possible to consider graphs with small (constant sized) separators ($k = O(1)$), but somewhat larger sized components (e.g., taking $\mu = O(\log n)$).

6 Intersection graphs

In this section we consider classes of intersection graphs. Each vertex in an intersection graph has associated with it an object in some space, and there is an edge between two vertices if and only if the corresponding objects intersect.

We consider the chordal graphs, and several subclasses of these. Chordal graphs are perfect. See [43] for more information on chordal graphs and other perfect graphs.

Definition. A graph $G = (V, E)$ is a *chordal* (or: triangulated) graph, if and only if every cycle with length exceeding three has an edge between two non-consecutive vertices in the cycle (a 'chord').

Theorem 26 (Gavril [40]) *A graph $G = (V, E)$ is chordal, if and only if there exists a tree $T = (I, F)$ such that one can associate with each vertex $v \in V$ a subtree $T_v = (I_v, F_v)$ of T, such that $(v, w) \in E$, if and only if $I_v \cap I_w \neq \emptyset$.*

In other words, a graph is chordal, if and only if it is the intersection graph of subtrees of a tree.

Theorem 27 (Robertson and Seymour[75]) *For every $k \in \mathbf{N}^+$ and every graph $G = (V, E)$, the treewidth of G is at most $k \Leftrightarrow 1$, if and only if G is a subgraph of a chordal graph H that has maximum clique size at most k.*

Proof: Use the characterisation of Theorem 26.

$\Leftarrow$: Use the tree decomposition $(\{X_i \mid i \in I\}, T)$ with $X_i = \{v \in V \mid i \in I_v\}$.

$\Rightarrow$: Let a tree decomposition $(\{X_i \mid i \in I\}, T = (I, F))$ of G with treewidth at most k be given. Let T_v be the subtree of T, induced by the set of nodes $I_v = \{i \in I \mid v \in X_i\}$. The intersection graph H, corresponding to these subtrees, is a chordal graph that contains G as a subgraph and has maximum clique size at most k. $\Box$

Instead of the maximum clique size of a chordal graph H, one can also use its chromatic number, as these are equal for all perfect graphs. It follows that the treewidth of a chordal graph equals its maximum clique size minus one. It also follows that every chordal graph G has a tree decomposition $(\{X_i \mid i \in I\}, T = (I, F))$ such that the set $\{X_i \mid i \in I\}$ equals the set of maximal cliques in G.

Similar results can be obtained for subclasses of the chordal graphs.

Definition. (i) A graph $G = (V, E)$ is an *undirected path graph*, if and only if there exists a tree $T = (I, F)$ such that one can associate with each

vertex $v \in V$ a path $P_v = (I_v, F_v)$ in T, such that for all $v, w \in V$, $v \neq w$: $(v, w) \in E$, if and only if $I_v \cap I_w \neq \emptyset$.
(ii) A graph $G = (V, E)$ is a *directed path graph*, if and only if there exists a rooted tree $T = (I, F)$ such that one can associate with each vertex $v \in V$ a path $P_v = (I_v, F_v)$ in T which is a sub-path from a path from a leaf of T to the root of T, such that for all $v, w \in V$, $v \neq w$: $(v, w) \in E$, if and only if $I_v \cap I_w \neq \emptyset$.
(iii) A graph $G = (V, E)$ is an *interval graph*, if and only if one can associate with each vertex $v \in V$ an interval $I_v = [l_v, r_v] \subset \mathbf{R}$, such that for all $v, w \in V$, $v \neq w$: $(v, w) \in E$, if and only if $I_v \cap I_w \neq \emptyset$.
(iv) A graph $G = (V, E)$ is a *proper interval graph*, if and only if one can associate with each vertex $v \in V$ an interval $I_v = [l_v, r_v] \subset \mathbf{R}$, such that for all $v, w \in V$, $v \neq w$: $(v, w) \in E$, if and only if $I_v \cap I_w \neq \emptyset$, and no interval I_v is entirely contained in another interval I_w, $v, w \in V$.

It follows that each interval graph is a directed path graph, each directed path graph is an undirected path graph, and each undirected path graph is chordal.

Definition. Let $G = (V, E)$ be a graph. The *interval thickness* of G is the smallest maximum clique size of an interval graph G that contains G as a subgraph.

The following results can be obtained in a similar way as Theorem 27.

Theorem 28 *(i) $G = (V, E)$ is a subgraph of an undirected path graph H with maximum clique size at most k, if and only if G has a tree decomposition $(\{X_i \mid i \in I\}, T = (I, F))$ with treewidth at most $k \Leftrightarrow 1$, and for all $v \in V$, $I_v = \{i \in I \mid v \in X_i\}$ induces a path in T.*
(ii) $G = (V, E)$ is a subgraph of a directed path graph H with maximum clique size at most k, if and only if G has a tree decomposition $(\{X_i \mid i \in I\}, T = (I, F))$ with treewidth at most $k \Leftrightarrow 1$, and a root node $r \in I$ can be chosen such that for all $v \in V$, $I_v = \{i \in I \mid v \in X_i\}$ induces a path in T which is a sub-path of a path from a leaf in T to r.

Theorem 29 *For all graphs $G = (V, E)$, the pathwidth of G is at most $k \Leftrightarrow 1$, if and only if the interval thickness is at most k.*

Proof: '$\leq$': Suppose $(X_1, \ldots, X_r)$ is a path decomposition of G of width $k \Leftrightarrow 1$. Associate to each $v \in V$ the interval $[\min\{i \mid v \in X_i\}, \max\{i \mid v \in X_i\}]$; the corresponding interval graph contains G and has maximum clique size at most k.

'$\geq$': Without loss of generality, one may assume that all endpoints $l_v, r_v \in \{1, 2, \ldots, 2n \Leftrightarrow 1, 2n\}$. Now take $X_i = \{v \in V \mid l_v \leq i \leq r_v\}$. $(X_1, \ldots, X_{2n})$ is a path decomposition of G of width at most $k \Leftrightarrow 1$. $\Box$

Definition. A graph $G = (V, E)$ is an *intersection graph of a graph* $H = (W, F)$, if one can associate to each vertex $v \in V$ a connected subgraph of G, such that for all $u, v \in V$, $u \neq v$: $(u, v) \in E$, if and only if the subgraphs, associated to u and v intersect.

Theorem 30 (Scheffler [94]) *Let $G = (V, E)$ be an intersection graph of a graph $H = (V', E')$, and let c be the maximum clique size of G. Then treewidth$(G) \leq c\cdot$ treewidth$(H) \Leftrightarrow 1$, and pathwidth$(G) \leq c\cdot$ pathwidth$(H) \Leftrightarrow 1$.*

The *circular arc graphs* are the intersection graphs of circles (or: graphs that are a cycle). The *proper circular arc graphs* are those circular arc graphs, that have an intersection model (in a circle), such that no arc is contained entirely in another arc.

Corollary 31 *Every circular arc graph G with maximum clique size k has pathwidth at most $2k \Leftrightarrow 1$.*

In Corollary 55, we see that the pathwidth of a proper circular arc graph with maximum clique size k is at most $2k \Leftrightarrow 2$. We close this section with a useful result on edge (or line) graphs.

Definition. The *edge graph* of a graph $G = (V, E)$ is the graph $(E, \{(e, e') \mid e$ and e' have one endpoint in common$\})$.

Lemma 32 (Bodlaender [17]) *Let the treewidth of graph $G = (V, E)$ be at most k, and the maximum degree of a vertex in G at most d. Then the treewidth of the edge graph of G is at most $(k + 1)d \Leftrightarrow 1$.*

Proof: If $(\{X_i \mid i \in I\}, T = (I, F))$ is a tree decomposition of G of width k, then $(\{Y_i \mid i \in I\}, T = (I, F))$ with $Y_i = \{(v, w) \in E \mid v \in X_i \vee w \in X_i\}$ is a tree decomposition of the edge graph of G of width at most $(k+1)d \Leftrightarrow 1$. $\Box$

The same result holds if we use pathwidth instead of treewidth. This bound may not be sharp.

7 Graph rewriting and elimination orderings

In this section, we consider several aspects of the rewriting of graphs to other, in general smaller, graphs. The related subject of graph grammars is discussed in the next section.

Arnborg and Proskurowski [5] have derived sets of rules to rewrite graphs into smaller graphs with the same treewidth for graphs with treewidth at most 1, 2, or 3. These rules are shown in Figure 2. See also [2].

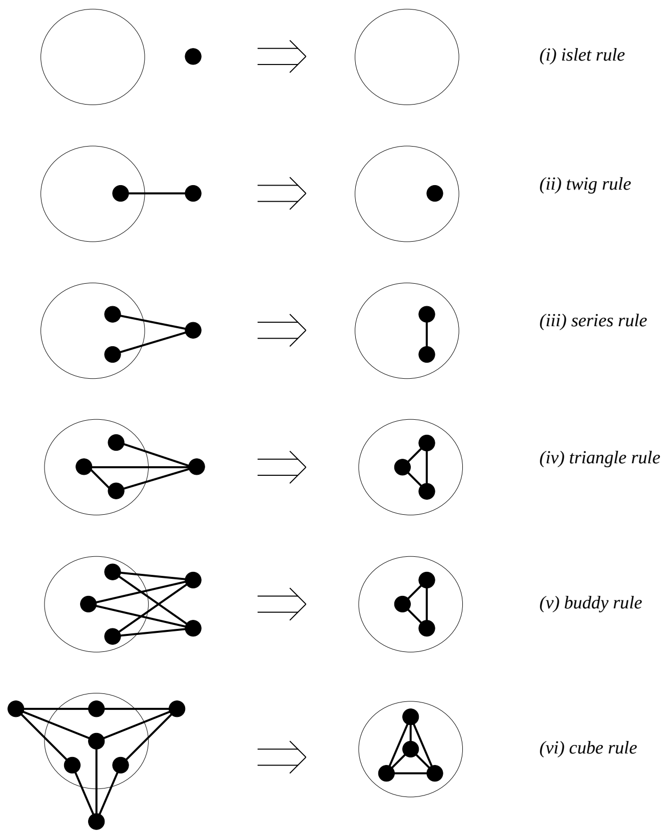

Figure 2: Safe and complete rules for rewriting graphs of treewidth at most 3

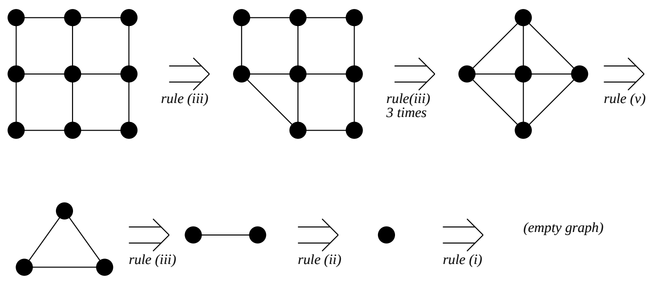

Figure 3: Rewriting the 3×3 grid to the empty graph

Theorem 33 (Arnborg and Proskurowski [5])
(i) A graph has treewidth 0, if and only if it can be rewritten to the empty graph, using rule (i) from figure 2.
(ii) A graph has treewidth 1, if and only if it can be rewritten to the empty graph, using rules (i) and (ii) from figure 2.
(iii) A graph has treewidth 2, if and only if it can be rewritten to the empty graph, using rules (i) – (iii) from figure 2.
(iv) A graph has treewidth 3, if and only if it can be rewritten to the empty graph, using rules (i) – (iv) from figure 2.

An example of a graph with treewidth 3 that is rewritten to the empty graph is given in Figure 3.

In [4], Arnborg et al show a much more general result. For many graph properties (including all that can be formulated in monadic second order logic) and all constants k, it holds that there is a finite set of 'local' graph reduction rules, that rewrite a graph to the empty graph, if and only if it fulfils the property and has treewidth at most k. Lagergren [57] shows that such rules, that are 'local' and 'based on star substitution' do not exist for the graphs of treewidth four.

Next we consider perfect elimination orderings of graphs.

Definition. A *perfect elimination ordering* of a graph $G = (V, E)$ is a bijection $f : V \to \{1, 2, \ldots, n\}$, such that for all $v \in V$, the set $\{w \in V \mid (v, w) \in W \wedge f(w) > f(v)\}$ forms a clique in G. A graph is a *perfect elimination graph*, if it has a perfect elimination ordering,

Note that if f is a perfect elimination ordering, then when we remove the vertices $v \in V$ one by one, in the order $f^{-1}(1), f^{-1}(2), \ldots, f^{-1}(n)$, then when we remove v, the neighbours of v form a clique in the remaining graph. A graph is a perfect elimination graph, if and only if it is a chordal graph (see [43], Chapter 4). A special type of perfect elimination graphs are the k-trees.

Definition. A graph $G = (V, E)$ is a k-tree, if and only if one of the following two conditions holds:

1. G is isomorphic to K_k, the complete graph with k vertices.
2. There exists a k-tree $H = (W, F)$, a vertex $v \in V \Leftrightarrow W$, and vertices $w_1, w_2, \ldots, w_k$ that form a clique in H, and G is the graph, obtained by adding v to H and an edge from v to each vertex in $\{w_1, \ldots, w_k\}$, i.e., $G = (W \cup \{v\}, F \cup \{(v, w_i) \mid 1 \leq i \leq k\})$.

Rose [93] obtained several equivalent characterisations of the class of k-trees. An x-y-separator of a graph $G = (V, E)$ is a set $W \subseteq V$ such that x and y are in different connected components of $G[V \Leftrightarrow W]$.

Theorem 34 (Rose [93]) *Let $G = (V, E)$ be a graph. The following four conditions are equivalent.*

1. *G is a k-tree.*
2. *G is connected, G has a k-clique, but not a $k + 2$-clique, and every minimal x-y-separator of G is a k-clique.*
3. *G is connected, $|E| = k \cdot |V| \Leftrightarrow \frac{1}{2}k(k + 1)$, and every minimal x-y-separator of G is a k-clique.*
4. *G has a k-clique, but not a $k + 2$-clique, and every minimal x-y-separator of G is a clique, and for all distinct non-adjacent pairs of vertices x, $y \in V$, there exist exactly k vertex disjoint paths from x to y.*

One can also characterise k-trees as perfect elimination graphs with a perfect elimination ordering f, such that $\forall v \in V : |\{w \in V \mid (v, w) \in E \wedge f(w) > f(v)\}| = k$ or $(f(v) > |V| \Leftrightarrow k \wedge |V| \geq k)$.

Definition. A graph $G = (V, E)$ is a *partial k-tree*, if and only if G is the subgraph of a k-tree.

Theorem 35 (See Scheffler [94] or van Leeuwen [101]) *A graph $G = (V, E)$ is a partial k-tree, if and only if the treewidth of G is at most k.*

Proof: $\Rightarrow$: It is sufficient to show that every k-tree $G = (V, E)$ has treewidth at most k. If $|V| \leq k + 1$, then we are done. Suppose $|V| > k + 1$. There is a vertex $v \in V$, such that $G[V \Leftrightarrow \{v\}]$ is a k-tree, and the neighbours of v in G form a clique of size k in G. Using induction, there is a tree decomposition $(\{X_i \mid i \in I\}, T = (I, F))$ of $G[V \Leftrightarrow \{v\}]$ of width at most k. By Lemma 4, there is an $i_0 \in I$ with X_{i_o} containing all neighbours of v in G. Now $(\{X_i \mid i \in I \cup \{j\}\}, T = (I \cup \{j\}, F \cup \{(i_0, j)\}))$ with $X_j = \{v\} \cup \{w \in V \mid (v, w) \in E\}$ is a tree decomposition of G of width k.

$\Leftarrow$: W.l.o.g, suppose $|V| \geq k+1$. Let $(\{X_i \mid i \in I\}, T = (I,F))$ be a tree decomposition of G with treewidth at most k. We claim, with induction to $|V|$, that there is a k-tree $H = (V,E')$, such that for every $i \in I$, X_i is a subset of a clique with $k+1$ vertices in G. If $|V| = k+1$, we are done. Otherwise, take a leaf node $i \in I$, with neighbour $j \in I$ in T. If $X_i \subseteq X_j$, we can remove node i from the tree decomposition, and continue with the resulting tree decomposition of G. Otherwise, take a vertex $v \in X_i \Leftrightarrow X_j$. Suppose induction on $G \Leftrightarrow \{v\}$, and $(\{X_i \Leftrightarrow \{v\} \mid i \in I\}, T)$ gives k-tree H'. As $v \notin X_j$, all neighbours of v must belong to X_i. Hence, the neighbours of v form a subset of a clique C with k vertices in G, so we can add v with edges to all vertices in C to H' and obtain the desired graph H. $\Box$

A bijection $f : V \to \{1, \ldots, |V|\}$ (called here *elimination ordering*) has a *fill-in* F_f, which is a set of unordered pairs of vertices in V, and is computed as follows. (Intuitively, it is the set of edges that must be added to G to make f a perfect elimination ordering.)

```
$F := \emptyset$;
**for** $i := 1$ **to** $|V|$
**do begin** $S_i := \{w \mid (f^{-1}(i), w) \in E \cup F \wedge f(w) > i\}$;
             **for all** $v,\ w \in S_i$:
             **do if** $(v,w) \notin E \cup F$ **then** add $(v,w)$ to $F$
    **end**;
$F_f := F$.
```

For a perfect elimination ordering f, $F_f = \emptyset$. A graph G has *dimension k with respect to f*, if $\max_{1 \leq i \leq |V|} |S_i| = k$. The *dimension* of a graph G is the minimum dimension over all elimination orderings of G.

Theorem 36 (See Arnborg [2]) *For every graph $G = (V,E)$, the dimension of G equals the treewidth of G.*

8 Graph grammars and recursive families of graphs

In this section, we consider some types of graph grammars, the 'recursive families of graphs' and their relationships to treewidth. We will limit the presentation here to a few notions and results, and direct the readers for further reading to other sources, e.g. [30, 36, 46, 45].

8.1 Hyperedge replacement grammars

First, we consider the notion of *hyperedge replacement grammar*, introduced by Habel and Kreowski. We only give an informal description here: for a good introduction to this topic, see e.g. [45, 46]. The framework of *context free graph grammars* of Bauderon and Courcelle is essentially similar [8]. See also [66].

Hyperedge replacement grammars work with hypergraphs, where each hyperedge is represented as a *sequence* of vertices. A hyperedge also has a label, which is either a *terminal* label, or a *non-terminal* label. More or less similar to context free string grammars, a hyperedge replacement grammar consists of a set of non-terminal labels, a set of terminal labels, a starting hypergraph (the 'axiom'), and a set of rewrite rules. We consider labelled directed hypergraphs $H = (V, S)$. V is a finite set of vertices. Graphs are seen as special cases of hypergraphs where each hyperedge has cardinality 2.

Each rewrite rule has as its left hand side an edge label and a number α. The right hand side is a directed labelled hypergraph with a sequence of α distinguished vertices in this hypergraph. Applying a rule consists of: taking a hyperedge with the corresponding label and α vertices, and replacing it by the right hand side graph. All vertices, except those which are distinguished are newly added to the graph. The vertices of the replaced hyperedge are identified with the corresponding vertices in the sequence of distinguished vertices.

A hyperedge replacement grammar (HRG) consists of sets of terminal and non-terminal labels, rewrite rules, and a start graph. (This concept clearly generalises the concept of context free string grammars.) We say a (labelled directed) hypergraph is generated by a hyperedge replacement grammar, if its edges are only labelled with terminal labels, and it can be produced from the start graph by a sequence of applications of rewrite rules.

Define the *width* of a hyperedge replacement grammar as the maximum number of vertices of a graph at the right hand side of a rule or the start graph, minus 1. (The result below can also be proved for hypergraphs, where for a tree-decomposition of a hypergraph it also must hold that for every hyperedge Z, there must be an $i \in I$ with $Z \subseteq X_i$, and the treewidth of a hypergraph defined accordingly.)

Theorem 37 (Lautemann [59]) *(i) Every graph, generated by a hyperedge replacement grammar of width k, has treewidth at most k.*
(ii) For every k, there exists a hyperedge replacement grammar of width k that generates exactly the directed graphs with treewidth at most k.

Proof: The result can be shown by establishing a direct correspondence between 'derivation trees' for hyperedge replacement grammars and tree decompositions. Such derivation trees are of a more or less similar form as derivation trees for context free string rewriting grammars. To each non-root node of the tree, we associate a hyperedge, and to each non-root and non-leaf node of the tree, we associate an isomorphic copy of a right hand side: the hypergraph to which the corresponding hyperedge is rewritten. To the root node, we associate the start graph. The children of a node are the hyperedges of its associated hypergraph.

Using this tree, and associating to each node i the set X_i, consisting of all vertices in its associated hypergraph ($\emptyset$ for leaf nodes), we obtain a tree decomposition of the generated graph of treewidth, at most the width of the hyperedge replacement grammar. This proves (i).

For the proof of (ii), we can for example take one terminal and one non-terminal label, a start graph with one node and a 1-vertex hyperedge with non-terminal label, and all possible rules, rewriting a hyperedge with at most $k + 1$ vertices to a hypergraph with at most $k + 1$ vertices. Now, each tree-decomposition with a root node r with $|X_r| = 1$ has an associated derivation tree, which gives this tree-decomposition, when the process described above is applied to it. $\Box$

The result also appears in [32], with a less direct proof. Vogler has shown that hyperedge replacement grammars generate the same languages of simple graphs as BNLC grammars of bounded nonterminal degree [103].

8.2 Recursive families of graphs

In this section, we discuss the recursive families of graphs, as introduced in the work of Borie [26] (see also [25, 27]). Similar frameworks have been introduced by Courcelle (see e.g. [30, 31, 32, 33]) and by Wimer [104]. While the differences between Bories and Wimers formalisms are not very large, I prefer the former as I find it somewhat simpler and more elegant. Courcelles framework is much more general and more precise, but is difficult to master for readers with little algebraic backgrounds. See also [9].

A *terminal graph* is a triple $G = (V, E, T)$, where (V, E) is an undirected graph, and $T \subseteq V$ is on ordered set of distinguished vertices, known as the *terminals* of G. A terminal graph $G = (V, E, T)$ is a *k-terminal* graph, when $|T| \leq k$. The number of terminals of $G = (V, E, T)$ is denoted by $t(G) = |T|$.

A *c-ary k-terminal recursive operation* is a function f, acting on c k-terminal graphs, and yielding another k-terminal graph, of the following form. f can be represented by pair $(M(f), t)$, where $M(f)$ is a matrix, which has r rows and c columns, such that each value $M_{i,j}(f)$ is an integer in the range $0 \ldots t(G_j)$. t is an integer, $t \leq k$. Given k-terminal graphs G_1, G_2, $\ldots$, G_c, $f(G_1, G_2, \cdots, G_c)$ is obtained in the following way: First, take the disjoint union of G_1, G_2, $\ldots$, G_c. Then, for each row i, $1 \leq i \leq r$, a number of terminal vertices are identified, namely, we take for each j, $1 \leq j \leq c$, if $M_{i,j}(f) \neq 0$, the $M_{i,j}(f)$th terminal of G_j, and identify (or merge) these vertices. If $M_{i,j} = 0$ for all j, $1 \leq j \leq c$, then a new vertex is formed. Finally, the vertices corresponding to the first t rows become the terminals of $f(G_1, \cdots, G_c)$, vertices corresponding to other rows become non-terminals.

A *base graph* is a k-terminal graph with no nonterminal vertices. A *k-terminal recursive family of graphs* is a set $\mathcal{G}$ of terminal graphs, for which there exists a set B of k-terminal base graphs and a finite set R of k-terminal recursive operations, such that $\mathcal{G}$ is the closure of B under the operations in R.

Theorem 38 (Wimer [104]) *Any k-terminal recursive family of graphs with only binary composition operators is a subclass of the partial $(2k \Leftrightarrow 1)$-trees.*

This can be generalised in the following way:

Theorem 39 *For any k-terminal recursive family of graphs $\mathcal{G}$, there is a constant $c_{\mathcal{G}}$, such that every graph $G \in \mathcal{G}$ has treewidth at most $c_{\mathcal{G}}$.*

Proof: Suppose $\mathcal{G}$ is generated by set of operations R and base set B. Let r_0 be the maximum number of rows of any matrix, associated with an operation in R. Now, with induction, we claim that any $G = (V, E, T) \in \mathcal{G}$ has a tree decomposition $(\{X_i \mid i \in I\}, T = (I, F))$ of treewidth at most $\max(r \Leftrightarrow 1, k \Leftrightarrow 1)$, such that there is an $i \in I$ with $T \subseteq X_i$, call i the distinguished node of the tree decomposition. Clearly this holds for base graphs (just take a tree decomposition with one node.) Suppose we have such tree decompositions for $G_1, \ldots, G_c$, and $f \in R$ is a c-ary operator. Now take one new set $X_{i_{\text{new}}}$, containing all vertices that were formed from a row of $M(f)$, and make i_{new} adjacent to all distinguished nodes in the tree decompositions of $G_1, \ldots, G_c$. $\Box$

Many classes of graphs, known to have a uniform treewidth upper bound, can be seen to be a k-terminal recursive family of graphs (see [26, 104]). Specifically, we mention the following:

Theorem 40 *(See [104].) For every k: the class of graphs of treewidth at most k is a $(k+1)$-terminal recursive family of graphs.*

Proof: A possible method to proof this is to use nice tree decompositions, see Lemma 7. We can give a set of operations, generating for graphs G with a nice tree decomposition $(\{X_i \mid i \in I\}, T = (I, F))$ of width k, the terminal graphs $G_i = (V_i, E_i, X_i)$, where $i \in I$, $V_i = \bigcup \{X_i \mid i = j \text{ or } j \text{ is a descendant of } i\}$, and $E_i \subseteq E$ the set of edges between vertices in V_i.

If i is a leaf node, then G_i can be assumed to be a base graph.

If i is a join node, with children j_1 and j_2, then G_i can be obtained from G_{j_1} and G_{j_2} by identifying corresponding terminals, which can be expressed by the pair $(M, |X_i|)$, M being the matrix

$$\begin{pmatrix} 1 & 1 \\ 2 & 2 \\ \vdots & \vdots \\ k+1 & k+1 \end{pmatrix}$$

Introduce nodes can be handled by first adding an isolated terminal vertex (a unary operation, with matrix $(1, 2, \cdots, k+1, 0)^T$), and then joining the graph with a base graph, giving the necessary edges to the new vertex. Forget nodes can be easily handled with another unary operation. □

See also [32] for similar results.

8.3 Series parallel graphs

A class of graphs with a recursive definition that we want to mention specifically is the class of series parallel graphs. In the literature, different definitions are given for this notion, in some cases expressing different classes of graphs. We use here one of the most common definitions.

Definition. A *series parallel graph* is a multi graph $G = (V, E)$ with two distinguished vertices, called *source* s and *sink* t, which can be formed with the following rules:

1. A graph with two vertices: source s and sink t and one edge (s, t) is a series parallel graph.
2. If $G_1 = (V_1, E_1)$ with source $s_1 \in V_1$ and sink $t_1 \in V_1$ and $G_2 = (V_2, E_2)$ with source $s_2 \in V_2$ and sink $t_2 \in V_2$ are series parallel graphs, then
   (a) the graph, obtained by taking the disjoint union of G_1 and G_2 and then identifying s_1 and s_2, and identifying t_1 and t_2 is a series parallel graph, with source the node, representing s_1 and s_2, and with sink the node, representing t_1 and t_2. This operation is called a *parallel composition.* See Figure 4 (i).
   (b) the graph, obtained by taking the disjoint union of G_1 and G_2 and then identifying t_1 and s_2, is a series parallel graph, with source s_1, and with sink t_2. This operation is called a *series composition.* See Figure 4 (ii).

A graph G without distinguished source and sink is called series parallel, if one can select a source and sink, such that G with this source and sink is a series parallel graph.

Theorem 41 *The treewidth of a series parallel graph G is at most 2.*

Proof: We prove with induction to the construction of G that the treewidth of the graph, obtained by adding edge (s, t) to G is at most 2. This clearly holds for a graph with two vertices.

Both in the case of a parallel and of a series composition, recursively make tree decompositions of treewidth 2 of $G_1 + (s_1, t_1)$ and $G_2 + (s_2, t_2)$. Add a new node, containing (the vertices, resulting from identifications of)

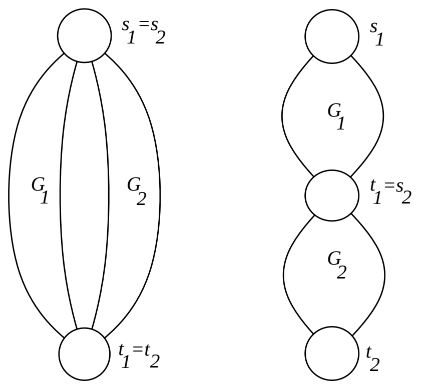

Figure 4: Parallel and series composition

s_1, s_2, t_1, t_2. (These are at most three vertices.) Make this new node adjacent to a node, containing s_1 and t_1, and to a node containing s_2 and t_2. □

Note that for instance $K_{1,3}$ is not a series parallel graph. Hence, the series parallel graphs, as defined above, are a *proper* subset of the graphs of treewidth 2. However, the following relationship exists between graphs of treewidth 2 and series parallel graphs.

Theorem 42 *A graph $G = (V, E)$ has treewidth at most 2, if and only if every biconnected component of G is a series parallel graph.*

Proof: If every biconnected component of G is series parallel, then every biconnected component of G has treewidth at most 2 (Theorem 41, hence G has treewidth at most 2 (Lemma 6).

To prove the other direction, it is sufficient to prove that every biconnected graph of treewidth 2 is series parallel. Consider a biconnected series parallel graph $G = (V, E)$, with a tree-decomposition $(\{X_i \mid i \in I\}, T = (I, F))$ of G of width 2, of the form of Lemma 8. We use induction to $|V|$. If $|V| \leq 3$, then G clearly is series parallel. Suppose $|V| > 3$. Now, T has at least two nodes. Take a leaf i from T with neighbour j in T. Let $\{v\} = X_i \Leftrightarrow X_j$. Suppose $X_i = \{v, w, x\}$. The set of neighbours of v must be $\{w, x\}$ (by definition of tree-decomposition and biconnectivity). Let G' be the multi graph, obtained by removing v from G and adding an edge between w and x if not already present. By induction G' is series parallel. Look to the construction of G' as series parallel graph. Where the edge $\{w, x\}$ is taken, we instead take the series parallel graph $(\{v, w, x\}, F)$ with terminals w and x and edges (w, v), (v, x), and (w, x) in case $(w, x) \in E$. This gives a construction of G as series parallel graph. □

9 Linear orderings

In this section, we consider linear orderings of the vertices of graphs, and several different measures of 'width' of these linear orderings.

Definition. A *linear ordering* of a graph $G = (V, E)$ is a bijection $f : V \to \{1, 2, \ldots, |V|\}$.

Definition. Let $G = (V, E)$ be a graph, and let $f : V \to \{1, 2, \ldots, n\}$ be a linear ordering of G.

1. The *bandwidth* of f is $\max\{|f(v) \Leftrightarrow f(w)| \mid (v, w) \in E\}$.
2. The *cutwidth* of f is $\max_{1 \leq i \leq n} |\{(u, v) \in E \mid f(u) \leq i < f(v)\}|$.
3. The *modified cutwidth* of f is $\max_{1 \leq i \leq n} |\{(u, v) \in E \mid f(u) < i < f(v)\}|$.
4. The *vertex separation number* of f is $\max_{1 \leq i \leq n} |\{u \in V \mid \exists v \in V : (u, v) \in E \ \wedge f(v) \geq i \ \wedge f(u) < i\}|$.
5. The *cyclic bandwidth* of f is $\max\{\min(|f(v) \Leftrightarrow f(w)|, n \Leftrightarrow |f(v) \Leftrightarrow f(w)|) \mid (v, w) \in E\}$.

The bandwidth, cutwidth, modified cutwidth, vertex separation number, cyclic bandwidth of a graph G is the minimum bandwidth, cutwidth, etc., over all possible linear orderings of G.

The *topological bandwidth* of a graph G is the minimum bandwidth over all graphs G' which are obtained by addition of an arbitrary number of vertices along edges of G.

Lemma 43 *For every graph $G = (V, E)$, the cyclic bandwidth of G is at most the bandwidth of G, and the bandwidth of G is at most twice the cyclic bandwidth of G.*

Proof: From the definitions, it directly follows that the cyclic bandwidth of G is at most the bandwidth of G. Suppose f is a linear ordering of G with cyclic bandwidth k. We suppose n is even; a similar construction can be used for odd n. The linear ordering $g : V \to \{1, \ldots, n\}$, defined by

$$g(v) = \begin{cases} 2 \cdot f(v) & \text{if } f(v) \leq n/2 \\ 2n + 1 \Leftrightarrow 2 \cdot f(v) & \text{if } f(v) > n/2 \end{cases}$$

has bandwidth at most $2k$. $\Box$

Theorem 44 *For every graph G, the pathwidth of G is at most the bandwidth of G, and at most twice the cyclic bandwidth of G.*

Proof: Let $f : V \to \{1, \ldots, n\}$ be a linear ordering of G with bandwidth k. Then $(X_1, \ldots, X_{n-k})$ with $X_i = \{f^{-1}(i),\ f^{-1}(i+1), \ldots, f^{-1}(i+k)\}$ is a path decomposition of G with pathwidth k. The second bound follows directly with Lemma 43. □

These bounds are sharp. Each graph $G_{n,k} = (\{1, 2, \ldots, n\}, \{(i,j) \mid i,j \in \{1, \ldots, n\}, 0 < |i \Leftrightarrow j| \leq k\})$ with $n \geq k+1$ has bandwidth k, and pathwidth (and treewidth) k, as it contains a clique with $k+1$ vertices as a subgraph, and each graph $G'_{n,k} = (\{1, 2, \ldots, n\}, \{(i,j) \mid i,j \in \{1, \ldots, n\}, 0 < |i \Leftrightarrow j| \leq k \vee |i \Leftrightarrow j| \geq n \Leftrightarrow k\})$ with $n \geq 2k+1$, and k divides $n \Leftrightarrow 2k+1$ has cyclic bandwidth k, and pathwidth and treewidth $2k$, as it contains K_{2k+1} as a minor (for all i, $1 \leq i \leq k$, contract the vertices $k+1+i$, $2k+1+i$, $\ldots$, $n \Leftrightarrow k+i$.)

Theorem 45 *For every graph G, the pathwidth of G is at most the topological bandwidth of G.*

Proof: If G' is obtained by adding vertices along edges of G, then G is a minor of G'. So, there exists a graph G' with G a minor of G', and topological bandwidth(G) = bandwidth$(G') \geq$ pathwidth$(G') \geq$ pathwidth(G). □

Makedon et al. claim a result, that states that the node search number of a graph is at most its topological bandwidth [63]. However, they use searcher moves that are not allowed in the node search game; a counter example is a clique with k vertices: it has topological bandwidth $k \Leftrightarrow 1$, but node search number k (any move with $\leq k \Leftrightarrow 1$ searchers would result in recontamination.) (See Section 10.)

By considering the graphs $G_{n,k}$ with $n \geq k+1$, it follows that the bound of Theorem 45 is sharp.

A sharp relation between the topological bandwidth and the cutwidth of trees was obtained by Chung [29].

Theorem 46 (Chung [29]) *Let T be a tree with topological bandwidth k and cutwidth l. Then $k \leq l \leq k + \log_2 k + 2$.*

Chung [29] also gives for every k, a tree with topological bandwidth k and cutwidth $k + \log_2 k \Leftrightarrow 1$.

Theorem 47 *For every graph G, the pathwidth of G is at most the cutwidth of G.*

Proof: Let $f : V \to \{1, \ldots, n\}$ be a linear ordering of G with cutwidth k. Let $X_i = \{w \in V \mid f(w) > i \wedge\ \exists v \in V : (v,w) \in E \wedge\ f(v) \leq i\} \cup \{f^{-1}(i)\}$. We claim that $(X_1, X_2, \ldots, X_n)$ is a path decomposition of G of width at most k.

Consider $(v,w) \in E$. If $f(v) < f(w)$, then $v,w \in X_{f^{-1}(v)}$, otherwise $v,w \in X_{f^{-1}(w)}$.

Next, suppose $i_1 < i_2 < i_3$, and $w \in X_{i_1} \cap X_{i_3}$. From $w \in X_{i_3}$, it follows that $f(w) \geq i_3 > i_2 > i_1$. As $w \in X_{i_1}$, there must be a $v \in V$ with $(v, w) \in E$ and $f(v) \leq i_1$. Now we have that $w \in X_{i_2}$. So, $(X_1, X_2, \ldots, X_n)$ is a path decomposition of G. The width of this path decomposition is at most k, as for all i, $|X_i| \leq 1 + |\{(v, w) \in E \mid f(v) \leq i < f(w)\}| \leq 1 + k$. □

Probably, this bound is not sharp. As the cutwidth of an n by n grid is at most $2n + 1$, and the treewidth and pathwidth of an n by n grid are n (see Section 13.2), it follows that there are graphs G with pathwidth$(G) \geq \frac{1}{2}$(cutwidth$(G) \Leftrightarrow 1)$.

Theorem 48 *For every graph G, the pathwidth of G is at most one larger than the modified cutwidth of G.*

For the relation between the topological bandwidth and cutwidth of a graph, see [29]. We also have the following relation between cutwidth and pathwidth.

Theorem 49 *For every class of graphs $\mathcal{G}$, the following statements are equivalent.*

1. $\exists c : \forall G \in \mathcal{G}$: *cutwidth*$(G) \leq c$.
2. $\exists c : \forall G \in \mathcal{G}$: *modified cutwidth*$(G) \leq c$.
3. $\exists c, d : \forall G \in \mathcal{G}$: *pathwidth*$(G) \leq c$ *and degree*$(G) \leq d$.

Proof: (i) $\rightarrow$ (iii): Use Theorem 47. Note that the cutwidth of G is at least degree$(G)/2$.

(iii) $\rightarrow$ (i): Suppose $(X_1, \ldots, X_r)$ is a path decomposition of G of width at most c. For all $v \in V$, let $g(v) = \min\{i \mid v \in X_i\}$. Now take a linear ordering f of G such that $g(v) < g(w) \Rightarrow f(v) < f(w)$ for all $v, w \in V$. We claim that the cutwidth of f is at most $c\cdot$ degree(G).

Consider j, $1 \leq j \leq |V|$, with $f^{-1}(j) = v$. We claim that if $f(w) \leq j < f(x)$ and $(w, x) \in E$, then $w \in X_{g(v)}$ or $x \in X_{g(v)}$. Suppose not. Then $g(w) < g(v) < g(x)$ and $w, x \notin X_{g(v)}$. This contradicts the definition of path decomposition. (Use that $w \in X_{g(w)}$, $x \in X_{g(x)}$, and $(w, x) \in E$.) So $|\{(u, v) \in E \mid f(u) \leq j < f(v)\}| \leq |X_{g(v)}|\cdot$degree$(G)$.

(i) $\rightarrow$ (ii): Trivial, as the modified cutwidth of a graph is at most its cutwidth.

(ii) $\rightarrow$ (iii): Use Theorem 48. The modified cutwidth of a graph G is at least degree$(G)/2 \Leftrightarrow 2$; if v had degree d, and f is a linear ordering of G, then either $\{(w, v) \mid f(w) < f(v) \Leftrightarrow 1\}$ or $\{(w, v) \mid f(w) > f(v) + 1\}$ contains at least $d/2 \Leftrightarrow 2$ edges. □

Several authors have noted equivalence between vertex separation number and pathwidth or an equivalent notion.

Theorem 50 (Kirousis and Papadimitriou [54]) *For every graph G, the node search number of G equals the vertex separation number of G plus one.*

Theorem 51 (Kinnersley [51]) *For every graph G, the vertex separation number of G equals the pathwidth of G.*

To close this section, we give two results on intersection graphs (cf. Section 6).

Theorem 52 *Let $G = (V, E)$ be a proper interval graph with maximum clique size k. Then the bandwidth of G is at most $k \Leftrightarrow 1$.*

Proof: We may assume that we can associate to all $v \in V$ an interval $I_v = [l_v, r_v] \subseteq \{1, \ldots, m\}$, such that for all $v, w \in V$, $v \neq w$, $(v, w) \in E \Leftrightarrow I_v \cap I_w \neq \emptyset$; $l_v, r_v \in \mathbf{N}$, and $I_v \subseteq I_w \Rightarrow v = w$. So, different vertices have different values for l_v and r_v. There exists a linear ordering $f : V \rightarrow \{1, \ldots, n\}$ of G, such that $f(v) < f(w) \Leftrightarrow l_v < l_w$. We claim that the bandwidth of f is at most $k \Leftrightarrow 1$. Consider an edge $(v, w) \in E$ with $f(v) < f(w)$. We have $l_v < l_w < r_v$. Let $X = \{x \in V \mid f(v) \leq f(x) \leq f(w)\}$. For all $x \in X$, $l_w \in I_x$, so X is a clique with $|f(v) \Leftrightarrow f(w)| + 1$ vertices. Hence $|f(v) \Leftrightarrow f(w)| \leq k \Leftrightarrow 1$. □

A somewhat stronger result has been proven by Kaplan and Shamir [50].

Definition. A path decomposition $(X_1, \ldots, X_r)$ of $G = (V, E)$ is a *proper path decomposition*, if there are no $v, w \in V$, $v \neq w$, with

$$\min\{i \mid v \in X_i\} \leq \min\{i \mid w \in X_i\} \leq \max\{i \mid w \in X_i\} \leq \max\{i \mid v \in X_i\}$$

The *proper pathwidth* of a graph G is the minimum width of a proper path decomposition of G.

Theorem 53 (Kaplan, Shamir [50]) *The bandwidth of a graph G equals its proper pathwidth, and is one smaller than the smallest clique size of any proper interval supergraph of G.*

It should be remarked that the notion of proper pathwidth of Kaplan and Shamir is different from the notion of proper pathwidth of Takahashi, Ueno, and Kajitani [99]. To distinguish, we use the term 3-proper for discussing the notion of proper pathwidth of Takahashi et al., in Section 10. With a proof, similar to the proof of Theorem 52 one can show:

Theorem 54 *Let $G = (V, E)$ be a proper circular arc graph with maximum clique size k. Then the cyclic bandwidth of G is at most $k \Leftrightarrow 1$.*

Corollary 55 *For every proper circular arc graph G with maximum clique size k, the pathwidth of G is at most $2k \Leftrightarrow 2$.*

10 Search and pebble games, and tangles

In this section we consider characterisations of the pathwidth and treewidth of graphs by the number of searchers, needed to capture a fugitive in certain "capturing games", played on the graph, and by pebble games. A more extensive overview on graph searching, treewidth, pathwidth, and related notions has been made by Bienstock [10]. We also discuss the notion of tangles in this section.

Definition. A *search strategy* of a graph $G = (V, E)$ is a sequence of the following types of moves:

1. place a searcher on a vertex.
2. delete a searcher from a vertex.
3. move a searcher over an edge.

Initially, all edges are *contaminated.* An edge $(v, w) \in E$ can become *cleared* by moving a searcher from v to w, while there is a second searcher on v or while all other edges adjacent to v are already cleared. An edge e can become *recontaminated*, when a move results in a path without searchers from a contaminated edge to edge e. The *search number* of G is the minimum number of searchers, needed to clear all edges.

It has been shown by LaPaugh [58] that for every graph G, there exists a search sequence, using the minimum number of searchers and clearing all edges, that does not allow recontamination. Such a search sequence is called *progressive.*

A variant of this notion, called *node search number*, was introduced by Kirousis and Papadimitriou [53]. In this variant, edges are cleared by having a searcher on both endpoints of the edge.

Lemma 56 (Kirousis and Papadimitriou [53]) *Let $G = (V, E)$ be a graph with search number k and node search number l. Then $l-1 \leq k \leq l+1$.*

Proof: A 'node search' can be transformed to an 'edge search' by moving a searcher over each edge when it is cleared. An 'edge search' can be transformed to a node search, by instead of moving a searcher over an edge, putting a searcher on the second endpoint and then removing the searcher from the first endpoint of the edge. □

Theorem 57 (Kirousis and Papadimitriou [53]) *For every graph $G = (V, E)$, the node search number of G equals the interval thickness of G.*

Recall from Theorem 29 that for every graph, its interval thickness is exactly one larger than its pathwidth.

Corollary 58 *(See also [37].) For every graph $G = (V, E)$, the node search number of G equals the pathwidth of G plus one. Furthermore, pathwidth$(G) \leq$ search number$(G) \leq$ pathwidth$(G) + 2$.*

Seymour and Thomas [96] give a characterisation of treewidth by a search game where a number of cops try to capture a robber, that is seen by the cops but has infinite speed. Informally, this game is as follows. The robber stands at a vertex of the graph and can at any time run at great speed to any other vertex along a path of the graph, but may not run through or to a vertex containing a cop. Each of the k cops is either on a vertex or in a helicopter. The objective of the player, controlling the movement of the cops is to land a cop via helicopter on the vertex occupied by the robber — the objective of the robber is to prevent this capture. The robber can see the helicopter approaching its landing spot, and may run to a new vertex before the helicopter actually lands.

We proceed with a more formal description of the game. We use the following notations: $G \Leftrightarrow X$ denotes the graph $G[V \Leftrightarrow X]$. The vertex set of a connected component of $G \Leftrightarrow X$ is called an X-*flap*. $[V]^{\leq k} = \{W \subseteq V \mid |W| \leq k\}$.

Positions in the game are pairs (X, R), $X \in [V]^{\leq k}$, R an X-flap. Player 2 (the robber) starts the game by choosing a connected component W and the game starts in position $(\emptyset, W)$. Suppose at the start of the ith step in the game we are in position (X_{i-1}, R_i). Player 1 (the cops) chooses a new set $X_i \in [V]^{\leq k}$, such that either $X_i \subset X_{i-1}$ or $X_{i-1} \subset X_i$. Player 2 then chooses, if possible, an X_i-flap R_i with $R_{i-1} \subseteq R_i$ or $R_i \subseteq R_{i-1}$. If this is not possible, the cops win. Otherwise, proceed with position (X_i, R_i). The robber wins if the game lasts for ever. If there is a winning strategy for the cops player, we say that 'k cops can search G in the Seymour-Thomas search game'. If there is a winning strategy for the cops player such that for all $h < i < j$: $X_h \cap X_j \subseteq X_i$, we say that '$k$ cops can monotonely search G in the Seymour-Thomas search game'. (The sets X_i denote the set of vertices containing a cop. R_i denotes the component of $G[V \Leftrightarrow X_i]$ where the robber is — its exact location is not important due to its speed.)

Seymour and Thomas [96] also define the tangle number of a graph $G = (V, E)$: a *tangle of order* k in $G = (V, E)$ is a function β, mapping each $X \in [V]^{\leq k}$ to an X-flap, such that for all $X \subseteq Y \in [V]^{\leq k}$: $\beta(Y) \subseteq \beta(X)$. The *tangle number* of G is the maximum order of all tangles in G. A tangle β is *convex*, if and only if for all X, $Y \in [V]^{\leq k}$:

$$(\beta(X) \cap \beta(Y)) \cup (X \cap \beta(Y)) \cup (\beta(X) \cap Y) \neq \emptyset$$

Theorem 59 (Seymour, Thomas [96]) *Let $G = (V, E)$ be a graph, $k \geq 0$. The following statements are equivalent.*

1. *The treewidth of G is at most k.*

2. *The tangle number of G is at most k.*
3. *G has a convex tangle of order $\leq k$.*
4. *$k+1$ cops can search G*
5. *$k+1$ cops can monotonely search G.*

Dendris, Kirousis and Thilikos [34] studied search games where the fugitive is supposed to be *inert*, i.e., it can it can only move just before a searcher visits the vertex that it occupies. The *(monotone) search number for an inert fugitive with unbounded speed* is defined accordingly.

Theorem 60 (Dendris, Kirousis, Thilikos [34]) *Let $G = (V, E)$ be a graph, $k \geq 0$. The following statements are equivalent.*

1. *The treewidth of G is at most k.*
2. *The number of searchers, needed to search G in the fugitive search game with an inert fugitive is at most $k+1$.*
3. *The number of searchers, needed to monotonically search G in the fugitive search game with an inert fugitive is at most $k+1$.*

Dendris et al. also consider variants where the fugitive has bounded speed, and obtain, amongst others, the following result.

Theorem 61 (Dendris, Kirousis, Thilikos [34]) *Suppose $G = (V, E)$ has no chordless cycle of length more than $s+2$. The treewidth of G is one less than the monotone search number of G for an inert fugitive with speed s.*

Takahashi, Ueno, and Kajitani [99] made a connection between a *mixed search game* and their notion of proper pathwidth. As Kaplan and Shamir [50] use the same term to describe a different notion, we use the term 3-proper pathwidth here.

Definition. A path decomposition $(X_1, \ldots, X_r)$ of $G = (V, E)$ of width at most k is called *3-proper*, if $|X_{j_1} \cap X_{j_2} \cap X_{j_3}| < k$ for any X_{j_1}, X_{j_2}, X_{j_3} none of which is a subset of the others. The *3-proper pathwidth* of G is the minimum width of a 3-proper path decomposition of G.

In the mixed search game, edges can be cleared in two ways: either by having a searcher on both endpoints of the edge, or by having a searcher move over the edge. In all other aspects, the game is similar to the standard search game, discussed first in this section. Define the mixed search number of G accordingly.

Theorem 62 (Takahashi, Ueno, and Kajitani [99]) *Let G be a graph, $k \in \mathbf{N}$. The following three statements are equivalent.*
(i) G has proper pathwidth at most k.
(ii) The mixed search number of G is at most k.
(iii) There is a search strategy that clears all edges of G in the mixed search game, without allowing recontamination, and that uses at most k searchers.

Another type of games are the pebble games. These model sequential computation (see e.g. [49]), and have been studied extensively. An interesting connection between some of these pebble games and node search number (and hence pathwidth) was found by Kirousis and Papadimitriou [54].

A pebble game is a 'one person game', played on a directed acyclic graph. We consider two variants: the black pebble game, and the black and white pebble game.

The black pebble game has the following types of moves:

1. Placing a pebble on a vertex with all predecessors of that vertex containing a pebble. (Hence, vertices with in-degree 0 can always be pebbled.)
2. Removing a pebble.

In the black and white pebble game, the following moves can be made:

1. Placing a black pebble on a vertex with all predecessors of that vertex containing a (black or white) pebble.
2. Placing a white pebble on a vertex.
3. Removing a black pebble.
4. Turning a white pebble black when all predecessors of the vertex containing the pebble, contain a (black or white) pebble.

The games start with no pebbles on any vertex of the directed acyclic graph $G = (V, E)$, and ends when each vertex has been pebbled at least once. The *black pebble demand* of G, and the *black and white pebble demand*, respectively, is the minimum over all possible pebble strategies of the maximum number of pebbles simultaneously on G when carrying out the respective game. One can also study 'progressive versions' of the games: in these versions, each vertex can be pebbled only once.

For an undirected graph G, the *set of directives* of G, is the set of all directed acyclic graphs whose underlying undirected graph equals G.

Theorem 63 (Kirousis and Papadimitriou [54]) *For every undirected graph $G = (V, E)$, the following three numbers are equal:*

1. *the minimum progressive black pebble demand over all directives of G.*

2. *the minimum progressive black and white pebble demand over all directives of G.*

3. *the node search number of G.*

Recall that the node search number of G is one larger as the pathwidth of G, see Theorems 29 and 57.

Theorem 64 (Kirousis and Papadimitriou [54]) *For every directed acyclic graph $H = (V, F)$ with underlying undirected graph $G = (V, E)$, with maximum in-degree of a vertex in H k, the progressive black and white pebble demand, and hence the black and white pebble demand of H is at most $k + 1$ times the node search number of G.*

11 Trees and forests

In this section we consider the pathwidth and treewidth of trees, forests, and give a characterisation of treewidth with help of depth-first-search spanning forests. We also introduce, with help of spanning forests, the notions of vertex and edge remember number, for use in a later section.

Theorem 65 *A graph $G = (V, E)$ is a forest, if and only if the treewidth of G is at most 1.*

Proof: By Lemma 6, it is sufficient to proof the result for trees, and connected graphs. Take an arbitrary vertex $r \in V$ as root of tree G. Now, $(\{X_v \mid v \in V\}, G)$ with $X_r = \{r\}$, and for $v \neq r$, X_v consists of v and the parent of v, is a tree decomposition of G with treewidth at most 1. Note that if G is not a tree, then it contains K_3 as a minor, and hence has treewidth at least 2 (use Lemma 17). □

(Alternately, we can note that every biconnected component of a forest has treewidth 1, and use Lemma 6.)

From Corollary 24, we know that the pathwidth of a graph with treewidth k is $O(k \log n)$, so for trees, the pathwidth is $O(\log n)$. More precise bounds were obtained by Scheffler [94].

Theorem 66 (Scheffler [94]) *For every tree $T = (V, E)$, the pathwidth of T is at most ${}^3\log(2n + 1)$.*

Theorem 67 (Scheffler [94]) *The pathwidth of a complete binary tree of depth k (i.e., with $2^{k+1} \Leftrightarrow 1$ vertices) equals $\lceil k/2 \rceil$.*

Theorem 68 (Kirousis and Papadimitriou [54]) *The pathwidth of a complete ternary tree T equals its height.*

(To be precise, Kirousis and Papadimitriou formulate and prove this result in the equivalent notion of node search, cf. Section 10.)

Ellis et al. [37] obtained among others a precise characterization of what trees have a specific pathwidth (in terms of separation number).

Another characterisation of treewidth can be obtained by looking at depth-first-search spanning trees or forests.

Definition. (i) A *DFS-tree* (or: palm tree) of a connected graph $G = (V, E)$ is a rooted spanning subtree $T = (V, F)$ with root $r \in V$, such that for every $(v, w) \in E$: v is an ancestor of w in T or w is an ancestor of v in T.
(ii) Given a DFS-tree T with root r of G, then the *value* of T is the maximum over all vertices $v \in V$, of the number of ancestors of v that is adjacent to v or a descendant of v.

Proposition 69 (Kloks [55]) *Let $G = (V, E)$ be a connected graph, and let $r \in V$. The treewidth of a graph equals the minimum value of a DFS-tree with root r of a supergraph $G = (V, E')$ of G ($E \subseteq E'$).*

We now introduce the notions of vertex remember number and edge remember number of maximal spanning forests of a graph, which are used for some proofs in Section 13.1, and were introduced in [16].

Consider a maximal spanning forest $T = (V, F)$ of a graph $G = (V, E)$. (I.e., T contains a spanning tree of every connected component of G.) To every edge $e = (v, w) \in E \Leftrightarrow F$, we can associate its *fundamental cycle*, that is: the unique cycle, consisting of e and the simple path from v to w in T. We define the *vertex remember number* of G, relative to T, denoted as: $vr(G, T)$, to be the maximum over all $v \in V$ of the number of fundamental cycles that use v. Similarly, the *edge remember number* of G, relative to T, is denoted by $er(G, T)$, and defined as the maximum over all edges $e \in V$ of the number of fundamental cycles that use e.

Theorem 70 *Let $G = (V, E)$ be a graph with maximal spanning forest $T = (V, F)$, and with maximum vertex degree d. Then*

$$er(G, T) \leq vr(G, T) \leq \frac{d}{2} \cdot er(G, T)$$

Proof: Observe that when k cycles use a vertex v with degree d', at least one edge adjacent to v is used by $2k/d'$ of these cycles, as each of these cycles must use two of the edges adjacent to v. The result now follows from this observation and the definitions. $\Box$

Theorem 71 *Let $T = (V, F)$ be a maximal spanning forest of graph $G = (V, E)$. The treewidth of G is at most* $\max(vr(G, T), er(G, T) + 1)$.

Proof: Let T' be the tree $(V \cup F, F')$, with $F' = \{(v,e) | v \in V, e \in F, \exists w \in V : e = (v,w)\}$, i.e. T' is obtained by adding an extra vertex in the middle of each edge in T. We show how to construct sets X_i, $i \in V \cup F$, such that $(\{X_i | i \in V \cup F\}, T' = (V \cup F, F'))$ is a tree decomposition of G.

First, for all $v \in V$, take $v \in X_v$ and for all $(v,w) \in F$, take $v, w \in X_{(v,w)}$.

Secondly, for each edge $(v,w) \in E \Leftrightarrow F$, choose arbitrarily one of v or w, say v. Now add v to each X_x, for all vertices $x \in V, x \neq w$ that are on the fundamental cycle of (v,w). Only do not add v to X_w. Also, add v to X_e, for all edges $e \in F$ on the fundamental cycle of (v,w).

One can verify that this indeed yields a tree decomposition of G. For $v \in V$: $|X_v| \leq 1 + vr(G,T)$, and for $e \in F$: $|X_e| \leq 2 + er(G,T)$. So the width of this tree decomposition is at most $\max(vr(G,T), er(G,T) + 1)$. $\Box$

12 Graphs, that 'almost' have treewidth k

In this section, we give two easy to obtain lemmas, that help in many cases to establish an upper bound on the treewidth (or pathwidth) of graphs.

Lemma 72 *Let $G = (V,E)$ be a graph, let $W \subseteq V$ be a set of vertices, and suppose that $G[V \Leftrightarrow W]$ has treewidth (pathwidth) at most k. Then the treewidth (pathwidth) of G is at most $k + |W|$.*

Proof: Let $(\{X_i \mid i \in I\}, T = (I,F))$ be a tree decomposition (path decomposition) of $G[V \Leftrightarrow W]$ of treewidth at most k. Write for all $i \in I$: $Y_i = X_i \cup W$. Then $(\{Y_i \mid i \in I\}, T)$ is a tree decomposition (path decomposition) of G of treewidth at most $k + |W|$. $\Box$

Lemma 73 *Let $G = (V,E)$ be a graph, let $E' \subseteq E$ be a set of edges, and suppose that the graph $(V, E \Leftrightarrow E')$ has treewidth (pathwidth) at most k. Then the treewidth (pathwidth) of G is at most $k + |E'|$.*

Proof: Let $W \subseteq V$ be a set, obtained by choosing for every edge $e \in E'$ an arbitrary endpoint and putting it in W. $G[V \Leftrightarrow W]$ is a subgraph of $(V, E \Leftrightarrow E')$, hence $G[V \Leftrightarrow W]$ has treewidth at most k, hence, by lemma 72, G has treewidth at most $k + |W| \leq k + |E'|$. $\Box$

Next we consider almost trees with parameter k, or, in short, 'almost k-trees'. Almost 1-trees are also known as cacti or cactus graphs.

Definition. A graph $G = (V,E)$ is an *almost tree with parameter* $\leq k$, if and only if there exists a maximal spanning forest $T = (V,F)$ of G, such that in each biconnected component of G there are at most k edges of G that do not belong to T.

With other words, $G = (V,E)$ is an almost tree with parameter $\leq k$, if and only if for each biconnected component $G_i = (V_i, E_i)$ of G one has $|E_i| \Leftrightarrow |V_i| + 1 \leq k$.

Theorem 74 (Bodlaender [14]) *The treewidth of an almost tree with parameter k is at most $k+1$.*

Proof: For a biconnected graph, we directly can apply Lemma 72, using the fact that forests have treewidth 1. By Lemma 6, the result now follows. $\Box$

A set of vertices $V' \subseteq V$ is called a *feedback vertex set* of a graph $G = (V,E)$, if $G[V \Leftrightarrow V']$ does not contain a cycle, i.e., is a forest. A set of edges $E' \subseteq E$ is called a *feedback edge set* of a graph $G = (V,E)$, if $(V, E \Leftrightarrow E')$ does not contain a cycle.

Corollary 75 *If G has a feedback vertex set of size k, then the treewidth of G is at most $k+1$.*

In [67], the reduction complexity of *st*-dags (directed acyclic graphs, with a unique vertex s of in-degree 0 and a unique vertex t of out-degree 0) is considered. Consider the following reductions: parallel reduction: remove all but one of two or more parallel edges; series reduction: replace edges (v,w) and (w,x), by an edge (v,x) in case w has in- and out-degree 1, and node reduction: take a vertex v of in-degree 1 and out-degree 1, draw an edge from every predecessor of v to every successor of v, and remove v and its adjacent edges. The *reduction complexity* of *st*-dag G is the minimum number of node reductions in a series of parallel, series, and node reductions that reduces G to a single edge.

Theorem 76 *If G is an st-dag with reduction complexity k, then the treewidth of (the undirected graph underlying) G is at most $k+2$.*

Proof: First note that we can modify each series of parallel, series, and k node reductions that reduces G to a single edge to a series of parallel and series reductions and k vertex removals (take a vertex, and remove it and its adjacent edges), that also reduces G to a number of components, each having at most one single edge. (Apply vertex removals on the same vertices as node reductions in the original sequence.)

Hence, if G has reduction complexity k, there exists a subgraph G' of G obtained by removing at most k vertices from G, that can be reduced to single edge components by series and parallel reductions. Now, G' can be seen to be a series parallel graph, hence the treewidth of G' is at most 2 (Theorem 41), and by Lemma 72, the treewidth of G is at most $k+2$. $\Box$

13 Planar graphs

13.1 Planar graphs with small radius

In this section, we consider planar graphs with small radius, and some related classes of graphs, including the Halin graphs.

There are several related notions, that deal with the maximum distance from vertices or faces of a planar or plane graph to the exterior face. Bienstock and Monma [12] list the following four notions:

- Call two faces adjacent when they share a vertex. The maximum distance of a face to the exterior face is called the *radius* [72].
- The maximum distance (using the usual notion of distance in a graph) in G of a vertex to a vertex on the outer face is called the *width* [35] or *gauge* [1].
- Call two vertices adjacent, if they share a face, and call the outer face adjacent to all vertices on the outer facial cycle. The maximum distance of a vertex to the outer face is called the *outerplanarity* [7].
- Call two faces adjacent when they share an edge. The maximum distance of a face to the outer face is called the *depth* [11].

The radius, width, outerplanarity, and depth of a planar graph G is the minimum radius, width, etc., over all possible plane embeddings of G.

Another, equivalent definition of the notion 'outerplanarity' is the following (see [7]).

Definition. An embedding of a graph $G = (V, E)$ is 1-outerplanar, if it is planar, and all vertices lie on the exterior face. For $k \geq 2$, an embedding of a graph $G = (V, E)$ is k-outerplanar, if it is planar, and when all vertices on the outer face are deleted, then a $(k \Leftrightarrow 1)$-outerplanar embedding of the resulting graph is obtained. A graph is k-outerplanar, if it has a k-outerplanar embedding.

The 1-outerplanar graphs are usually called *outerplanar graphs.* Bienstock and Monma [12] notice the following easy relations.

Lemma 77 (Bienstock and Monma [12]) *Let $G = (V, E)$ be a planar graph with radius r, width w, outerplanarity p, and depth d. Then $r \leq p \leq r + 1$, $r \leq w + 1$, $r \leq d$, $p \leq w + 1$, $p \leq d + 1$.*

Lemma 78 *Every outerplanar graph $G = (V, E)$ has treewidth at most 2.*

Proof: Observe that an outerplanar graph can be rewritten to the empty graph by the rules (i) – (iii) from Theorem 33. □

The class of outerplanar graphs is closed under minor taking: its obstruction set consists of the graphs $K_{2,3}$ and K_4 (see [97] for some related results.)

We will show a bound of $3k \Leftrightarrow 1$ on the treewidth of k-outerplanar graphs. We need a series of lemma's.

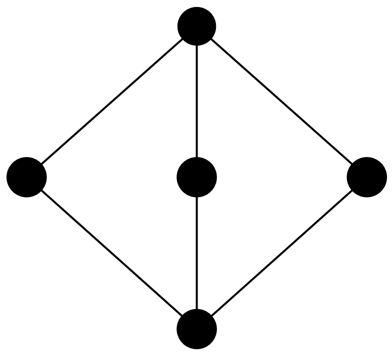
Figure 5: $K_{2,3}$

Lemma 79 *Let $G = (V, E)$ be a planar graph with some given planar embedding. Let $H = (V, E')$ be the graph, that is obtained from G by removing all edges on the exterior face. Let $T' = (V, F')$ be a maximal spanning forest of H. Then there exists a maximal spanning forest $T = (V, F)$ of G, such that $er(G, T) \leq er(H, T') + 2$, and $vr(G, T) \leq vr(H, T') +$ degree(G).*

Proof: Consider the graph $K = (V, (E \Leftrightarrow E') \cup F')$, i.e. the graph with edges in T', or in G but not in H. Let $T = (V, F)$ be a maximal spanning forest of K, such that $T' \subseteq T$, i.e. T is obtained by adding edges from $E \Leftrightarrow E'$ to T'. Note that every fundamental cycle in K, relative to T, must form the boundary of an interior face in K. As each edge is adjacent to at most 2 interior faces, and each vertex is adjacent to at most degree(G) interior faces, it follows that $er(K, T) \leq 2$, and $vr(K, T) \leq$ degree(G).

As T is also a maximal spanning forest of G, and each fundamental cycle in G either is a fundamental cycle in H, or a fundamental cycle in $K, er(G, T) \leq er(K, T) + er(H, T') \leq er(H, T') + 2$, and $vr(G, T) \leq vr(K, T) + vr(H, T') \leq vr(H, T') +$ degree(G). $\square$

Lemma 80 *Let $G = (V, E)$ be an outerplanar graph with degree$(G) \leq 3$. Then there exists a maximal spanning forest $T = (V, F)$, with $er(G, T) \leq 2$ and $vr(G, T) \leq 2$.*

Proof: If one removes all edges on the exterior face of G, a tree or forest $T' = (V, F')$ results. Clearly $er(T', T') = vr(T', T') = 0$. The result follows now as in Lemma 79 by observing that each vertex is adjacent to at most 2 interior faces. $\square$

Lemma 81 *Let $G = (V, E)$ be a k-outerplanar graph with degree$(G) \leq 3$. Then there exists a maximal spanning forest $T = (V, F)$ with $er(G, T) \leq 2k$, and $vr(G, T) \leq 3k \Leftrightarrow 1$.*

Proof: Use induction to k. The case $k = 1$ was shown in Lemma 80. Let $k \geq 2$. If we remove all edges on the exterior face of G, then each vertex on the exterior face has degree 0 or 1, so a $(k \Leftrightarrow 1)$-outerplanar graph is obtained. The result now follows with induction and Lemma 79. □

Lemma 82 *For every k-outerplanar graph $G = (V, E)$, there exists a k-outerplanar graph $H = (V', E')$, such that G is a minor of H, and degree$(H) \leq 3$.*

Proof: We can replace every vertex with degree $d \geq 4$ by a path of $d \Leftrightarrow 2$ vertices of degree 3, in such a way that the graph stays k-outerplanar. □

Now we are ready to prove the main results.

Theorem 83 *The treewidth of a k-outerplanar graph $G = (V, E)$ is at most $3k \Leftrightarrow 1$.*

Proof: For $k = 1$, use Lemma 78. Suppose $k \geq 2$. By Lemma 82, there exists a k-outerplanar graph H, such that G is a minor of H, and degree$(H) \leq 3$. By Lemma 81, there exists a maximal spanning forest T of H, such that $er(H, T) \leq 2k$ and $vr(H, T) \leq 3k \Leftrightarrow 1$. By Theorem 71, treewidth $(H) \leq \max\{3k \Leftrightarrow 1, 2k + 1\} = 3k \Leftrightarrow 1$. By Lemma 16, treewidth $(G) \leq 3k \Leftrightarrow 1$. □

Robertson and Seymour proved a very similar result, but based on the notion of radius.

Theorem 84 (Robertson and Seymour [72]) *The treewidth of a planar graph with radius d is at most $3d + 1$.*

We next consider the Halin graphs.

Definition. A graph $G = (V, E)$ is a *Halin graph*, if it can be obtained by embedding a tree without vertices with degree 2 and with at least 4 vertices in the plane, and connecting its leaves by a cycle that crosses none of its edges.

Theorem 85 (See [104]) *The treewidth of a Halin graph equals 3.*

Proof: A similar proof as above can be used. Let G be a Halin graph. G is a minor of a Halin graph H with maximum vertex degree 3. The latter clearly has a spanning tree T with $vr(H, T) = 3$, and $er(H, T) = 2$. The construction of Theorem 71 gives a tree decomposition of H of width 3. Hence the treewidth of G is at most 3.

As G contains K_4, a clique with 4 vertices, as a minor (contract all interior vertices to one vertex, a "wheel" results, and then contract further to K_4), treewidth $(G) \geq$ treewidth $(K_4) = 3$. □

We now give an alternate proof of the famous 'planar separator theorem' of Lipton and Tarjan [61] — although the constant factor yielded by the proof below is higher. See the discussion in Section 5.

Theorem 86 *Every planar graph $G = (V, E)$ has a (type-2) $\frac{1}{2}n$-separator of size $2\sqrt{6n} \approx 4.90\sqrt{n}$.*

Proof: Let $c = \frac{1}{2}\sqrt{6}$ in this proof. Let $G = (V, E)$ be a planar graph, and consider a fixed plane embedding of G. Let $V_0 = \emptyset$, V_1 be the set of vertices on the exterior face, and let V_i be the set of vertices that are adjacent to at least one vertex in V_{i-1}, and do not belong to $V_0 \cup V_1 \cup \cdots V_{i-2}$ ($i \geq 2$). Take i_0 such that $\sum_{j=0}^{i_0-1} |V_j| \leq \frac{1}{2}n$, and $\sum_{j=i_0+1}^{\infty} |V_j| \leq \frac{1}{2}n$. (Such i_0 always exists.) Note that each set V_i separates $V_0 \cup V_1 \cup \cdot \cup V_{i-1}$ from $V_{i+1} \cup V_{i+2} \cup \cdots$. If $V_{i_0} < c\sqrt{n}$, then we are done. Otherwise, let $i_1 = \max\{j < i_0 \mid |V_j| \leq c\sqrt{n}\}$, and $i_2 = \min\{j > i_0 \mid |V_j| \leq c\sqrt{n}\}$. Note that $G[V_{i_1+1} \cup V_{i_1+2} \cup \cdots \cup V_{i_2-2} \cup V_{i_2-1}]$ is an $(i_2 \Leftrightarrow i_1 \Leftrightarrow 1)$-outerplanar graph, hence has treewidth at most $3(i_2 \Leftrightarrow i_1 \Leftrightarrow 1) \Leftrightarrow 1 \leq \frac{3}{c}\sqrt{n} \Leftrightarrow 1$ (use that all V_j with j in the range $i_1 + 1 \cdots i_2 \Leftrightarrow 1$ have size more than $c\sqrt{n}$), so this graph has a (type-2) separator S of size $\frac{3}{c}\sqrt{n}$ (Theorem 19). Now $S \cup V_{i_1} \cup V_{i_2}$ is a (type-2) separator of G of size at most $2c\sqrt{n} + \frac{3}{c}\sqrt{n} = 2\sqrt{6n}$. □

13.2 Grid graphs

We consider in this section the n by r grid graph.

Definition. The $n \times r$ grid graph is the graph $GR_{n\times r} = (V_{n\times r}, E_{n\times r})$, defined by $V_{n\times r} = \{(i, j) \mid i \in \{1, 2, \ldots, n\},\ j \in \{1, 2, \ldots, r\}\}$, and $E_{n\times r} = \{((i_1, j_1), (i_2, j_2)) \mid (i_1, j_1), (i_2, j_2) \in V_{n\times r}$ and $(i_1 = i_2 \wedge |j_1 \Leftrightarrow j_2| = 1)$ or $(j_1 = j_2 \wedge |i_1 \Leftrightarrow i_2| = 1)\}$.

Lemma 87 *The pathwidth of an $n \times r$ grid graph $GR_{n\times r}$ is at most* $\min(n, r)$.

Proof: W.l.o.g., suppose $n = \min(n, r)$. Take $X_{j\times n+i-n-1} = \{(i, j),\ (i + 1, j),\ , (i + 2, j), \ldots, (n, j),\ (1, j + 1),\ (2, j + 1), \ldots, (i \Leftrightarrow 1, j + 1),\ (i, j + 1)\}$, for all i, j, $1 \leq i \leq n$, $1 \leq j \leq r \Leftrightarrow 1$. Then $(X_1, X_2, \ldots, X_{n(r-1)-1})$ is a path decomposition of $GR_{n\times r}$ of width n. □

Lemma 88 *The treewidth of an $n \times n$ grid graph $GR_{n\times n}$ is at least n.*

Proof: In [82], it is shown that a $GR_{n\times n}$ has a tangle of order n. Hence, the tangle number of $GR_{n\times n}$ is at least n, and by Theorem 59, the treewidth of $GR_{n\times n}$ is at least n. □

Corollary 89 *The treewidth and pathwidth of the $n \times r$ grid graph $GR_{n\times r}$ equal* $\min\{n, r\}$.

14 Graphs of bounded degree

In this section we mention some different results on the treewidth of graphs of bounded degree.

Lemma 90 (Scheffler [94]) *Every graph of treewidth at most k contains a vertex of degree at most k.*

Proof: If every vertex of G has degree more than k, then G cannot be a subgraph of a k-tree, because every k-tree has at least one vertex of degree k, hence G is no partial k-tree, hence G has treewidth more than k. (See Theorem 35). $\Box$

From Lemma 90, the following result follows easily by induction.

Lemma 91 (Rose [93].) *If $G = (V, E)$ has treewidth at most k, then $|E| \leq k \cdot |V| \Leftrightarrow \frac{1}{2}k(k+1)$.*

Ramachandramurthi introduced the graph parameter $\gamma(G) = \min(n \Leftrightarrow 1, \min_{v,w \in V, v \neq w, (v,w) \notin E} \max(\text{degree}(v), \text{degree}(w)))$. I.e., $\gamma(G) = n \Leftrightarrow 1$, if G is a clique.

Lemma 92 (Ramachandramurthi [68]) *The treewidth of a graph G is at least $\gamma(G)$.*

Ramachandramurthi [68] also gives characterisations of pathwidth and treewidth in terms of subgraphs that fulfil certain degree restrictions.

Bodlaender and Engelfriet [21] introduced the notion of domino treewidth.

Definition. A tree decomposition $(\{X_i \mid i \in I\}, T = (I, F))$ of a graph $G = (V, E)$ is a *domino tree-decomposition*, if for every $v \in V$, there are at most two nodes $i \in I$ with $v \in X_i$. The *domino treewidth* of a graph G is the minimum width over all domino tree decompositions of G.

Theorem 93 (Bodlaender and Engelfriet [21]) *For every k, $d \in \mathbf{N}$, there exists $k' \in \mathbf{N}$, such that every graph with treewidth at most k and maximum degree at most d has domino treewidth at most k'.*

There is also a connection of this notion with the notion of *strong treewidth*, as introduced by Seese [95].

Definition. A *strong tree decomposition* of a graph $G = (V, E)$ is a pair $(\{X_i \mid i \in I\}, T = (I, F))$ with $\{X_i \mid i \in I\}$ a collection of *disjoint* subsets of V, and $T = (I, F)$ a tree, such that

- $\bigcup_{i \in I} X_i = V$

- for all edges $\{v, w\} \in E$, either there is an $i \in I$ with $v, w \in X_i$, or there are i, $i' \in I$, that are adjacent in T ($(i, i') \in F$), and $v \in X_i$, $w \in X_{i'}$.

The *width* of a strong tree decomposition $(\{X_i \mid i \in I\}, T = (I, F))$ is $\max_{i \in I} |X_i|$. The *strong treewidth* of a graph $G = (V, E)$ is the minimum width over all strong tree decompositions of G.

Note that in general, a strong tree-decomposition of a graph G, is *not* a tree-decomposition of G. Trees have strong treewidth 1: take singleton sets X_i, one for each vertex in the tree.

Lemma 94 (Seese [95]) *If the strong treewidth of G is at most k, then the treewidth of G is at most $2k \Leftrightarrow 1$.*

Theorem 95 (Bodlaender and Engelfriet [21]) *For every class of graphs $\mathcal{G}$, the following statements are equivalent:*

1. *There exists a constant $c \in \mathbf{N}$, such that every graph in $\mathcal{G}$ has domino treewidth at most c.*
2. *There exist constants k, $d \in \mathbf{N}$, such that every graph in $\mathcal{G}$ has treewidth at most k and maximum degree at most d.*
3. *There exist constants k', $d \in \mathbf{N}$, such that every graph in $\mathcal{G}$ has strong treewidth at most k' and maximum degree at most d.*

15 Applications

In this section, we mention some of the applications of treewidth to (mostly) non graph-theoretic applications.

15.1 Gate Matrix Layout

The gate matrix layout problem arises from problems in VLSI design. It can be formulated as follows. An instance of the problem consists of an $n \times m$ boolean matrix M, and an integer k. We are asked whether we can permute the columns of the matrix M, such that if in each row, we change every 0 which lies between the rows leftmost and rightmost 1 into a 1, then no column contains more than k 1's.

Fellows and Langston [38] showed that there is an elegant translation of instances of gate matrix layout to instances of pathwidth, as follows. (We give a slightly more compact transformation, avoiding a first step replacing every column by a number of columns with exactly two 1's.)

Given matrix M, let G_M be the graph, obtained by taking a vertex v_i for each row i, and taking an edge between vertices v_i and v_j, if there exists a column c, with $M_{ic} = M_{jc} = 1$. Effectively, this means that each column corresponds to a clique in G, formed by the rows that have a 1 on the entry in that column.

Lemma 96 (Fellows, Langston [39]) *G_M has pathwidth at most k, if and only if there exists a permutation of the columns of M, such that if in each row, we change every 0 which lies between the rows leftmost and rightmost 1 into a 1, then no column contains more than $k+1$ 1's.*

Proof: First, suppose $(X_1, \ldots, X_r)$ is a path decomposition of G_M of width at most k. For each column c, note that $\{v_i \mid M_{ic} = 1\}$ forms a clique in G_M, hence, by Lemma 4, there exists an $\alpha_c \in \{1, \ldots, r\}$, such that $\{v_i \mid M_{ic} = 1\} \subseteq X_{\alpha_c}$. Permute the columns c by a permutation π, with $\alpha_c < \alpha_{c'} \Rightarrow \pi(c) < \pi(c')$ for all columns c, c'. One can verify that this column permutation π actually is of the form requested.

Alternatively, suppose we have permuted the columns of M, and changed in every row every 0 which lies between the rows leftmost and rightmost 1 into a 1, such that no column contains more than $k+1$ 1's. Let M' be the resulted matrix. Now $(X_1, \ldots, X_m)$, with $X_c = \{v_i \mid M_{ic} = 1\}$ can be seen to be a path decomposition of G_M of width at most k. $\Box$

See also [68].

A good overview of many of the issues involved here has been made by Möhring [64].

15.2 Interval Routing Schemes

Consider a distributed processor network, in which processors want to send messages to each other. Research has been done on so called compact routing methods (see [102] for an overview), methods in which processors decide over what link to forward messages that take relatively little space for storing such routing information. One type of these methods is interval routing. In the case of k-interval routing, each processor is numbered with a unique integer, and each outgoing link is labelled with at most k cyclic intervals of processor names (integers). (I.e., each edge is labelled with two labels, one at each endpoint.) A message (when not arriving at its final destination) is forwarded over the link whose label has an interval that contains the name of the destination processor. It is required that messages arrive at their destination, using this method, by the shortest route.

In the dynamic link cost setting, one assumes that weights of links can vary. An (undirected) graph is said to be in k-IRS, if there exists a numbering of the vertices (processors), such that for all weight assignments to edges (links), there exists a label assignment to links, fulfilling the requirements described above.

Theorem 97 (Bodlaender et al. [24]) *If $G \in k$-IRS, then the treewidth of G is at most $4k$.*

For details and more related results, the reader is referred to [24].

15.3 Structured programs

In a very recent paper, Thorup [100] makes a connection between the control-flow graphs of structured programs and treewidth. Under a rather general definition of 'structured program' (including goto-freeness), he shows the following result.

Theorem 98 (Thorup [100]) *All control-flow graphs of structured programs have treewidth at most 6.*

Thorup also mentions that control graphs of programs written in Modula-2 have treewidth at most five, and control graphs of goto-free programs written in Pascal have treewidth at most three. The application of this result lies in algorithms, solving the register allocation problem.

16 Miscellaneous results

16.1 AT-free graph

An asteroidal triple in a graph $G = (V, E)$ is a set of three distinct vertices $v, w, x \in Y$, such that between any two of them, there is a path that does not contain a neighbour of the third. A graph is AT-free, if it does not contain an asteroidal triple. Möhring proved the following interesting result.

Theorem 99 (Möhring [65]) *If G is an AT-free graph, then the treewidth of G equals its pathwidth.*

Postscript

An arboretum is a garden, containing many different kinds of trees, in many cases made and maintained for study of biologists. This partial k-arboretum was meant to be a collection of many different kinds of partial k-trees. However, the 'partial' from the title is also meant to reflect the incompleteness of the overview. I express my apologies to those, whose work I misrepresented or have missed to mention. I welcome any comments and suggestions.

This paper benefited especially from comments from, help of, discussions with and collaborations with Babette de Fluiter, Joost Engelfriet, Michael Fellows, Jens Gustedt, Ton Kloks, Andrzej Proskurowski, Petra Scheffler, Detlef Seese, Jan van Leeuwen, and several others, to which I apologise here for forgetting to mention their names.

References

[1] A. Aggarwal, M. Klawe, D. Lichtenstein, N. Linial, and A. Wigderson, *A lower bound on the area of permutation layents*, Algorithmica, 6 (1991), pp. 241–255.

[2] S. Arnborg, *Efficient algorithms for combinatorial problems on graphs with bounded decomposability – A survey*, BIT, 25 (1985), pp. 2–23.

[3] S. Arnborg, D. G. Corneil, and A. Proskurowski, *Complexity of finding embeddings in a k-tree*, SIAM J. Alg. Disc. Meth., 8 (1987), pp. 277–284.

[4] S. Arnborg, B. Courcelle, A. Proskurowski, and D. Seese, *An algebraic theory of graph reduction*, J. ACM, 40 (1993), pp. 1134–1164.

[5] S. Arnborg and A. Proskurowski, *Characterization and recognition of partial 3-trees*, SIAM J. Alg. Disc. Meth., 7 (1986), pp. 305–314.

[6] S. Arnborg, A. Proskurowski, and D. G. Corneil, *Forbidden minors characterization of partial 3-trees*, Disc. Math., 80 (1990), pp. 1–19.

[7] B. S. Baker, *Approximation algorithms for NP-complete problems on planar graphs*, J. ACM, 41 (1994), pp. 153–180.

[8] M. Bauderon and B. Courcelle, *Graph expressions and graph rewritings*, Mathematical Systems Theory, 20 (1987), pp. 83–127.

[9] M. W. Bern, E. L. Lawler, and A. L. Wong, *Linear time computation of optimal subgraphs of decomposable graphs*, J. Algorithms, 8 (1987), pp. 216–235.

[10] D. Bienstock, *Graph searching, path-width, tree-width and related problems (a survey)*, DIMACS Ser. in Discrete Mathematics and Theoretical Computer Science, 5 (1991), pp. 33–49.

[11] D. Bienstock and C. L. Monma, *On the complexity of covering vertices by faces in a planar graph*, SIAM J. Comput., 17 (1988), pp. 53–76.

[12] ——, *On the complexity of embedding planar graphs to minimize certain distance measures*, Algorithmica, 5 (1990), pp. 93–109.

[13] D. Bienstock, N. Robertson, P. D. Seymour, and R. Thomas, *Quickly excluding a forest*, J. Comb. Theory Series B, 52 (1991), pp. 274–283.

[14] H. L. Bodlaender, *Classes of graphs with bounded treewidth*, Tech. Rep. RUU-CS-86-22, Dept. of Computer Science, Utrecht University, Utrecht, the Netherlands, 1986.

[15] ——, *Dynamic programming algorithms on graphs with bounded treewidth*, in Proceedings of the 15th International Colloquium on Automata, Languages and Programming, Springer Verlag, Lecture Notes in Computer Science, vol. 317, 1988, pp. 105–119.

[16] ——, *Planar graphs with bounded treewidth*, Technical Report RUU-CS-88-14, Dept. of Computer Science, Utrecht University, Utrecht, the Netherlands, 1988.

[17] ——, *Complexity of path-forming games*, Theor. Comp. Sc., 110 (1993), pp. 215–245.

[18] ——, *On linear time minor tests with depth first search*, J. Algorithms, 14 (1993), pp. 1–23.

[19] ——, *A tourist guide through treewidth*, Acta Cybernetica, 11 (1993), pp. 1–23.

[20] ——, *On disjoint cycles*, Int. J. Found. Computer Science, 5 (1994), pp. 59–68.

[21] H. L. Bodlaender and J. Engelfriet, *Domino treewidth*, in Proceedings 20th International Workshop on Graph Theoretic Concepts in Computer Science WG'94, E. W. Mayr, G. Schmidt, and G. Tinhofer, eds., Springer Verlag, Lecture Notes in Computer Science, vol. 903, 1995, pp. 1–13.

[22] H. L. Bodlaender, J. R. Gilbert, H. Hafsteinsson, and T. Kloks, *Approximating treewidth, pathwidth, and minimum elimination tree height*, J. Algorithms, 18 (1995), pp. 238–255.

[23] H. L. Bodlaender and R. H. Möhring, *The pathwidth and treewidth of cographs*, SIAM J. Disc. Meth., 6 (1993), pp. 181–188.

[24] H. L. Bodlaender, R. B. Tan, D. M. Thilikos, and J. van Leeuwen, *On interval routing schemes and treewidth*, in Proceedings 21th International Workshop on Graph Theoretic Concepts in Computer Science WG'95, M. Nagl, ed., Springer Verlag, Lecture Notes in Computer Science, vol. 1017, 1995, pp. 181–186.

[25] R. Borie and A. Gupta, *Balanced decompositions for partial k-trees*, Congressus Numerantium, 98 (1993), pp. 33–38.

[26] R. B. Borie, *Recursively Constructed Graph Families*, PhD thesis, School of Information and Computer Science, Georgia Institute of Technology, 1988.

[27] R. B. Borie, R. G. Parker, and C. A. Tovey, *Automatic generation of linear-time algorithms from predicate calculus descriptions of problems on recursively constructed graph families*, Algorithmica, 7 (1992), pp. 555–581.

[28] K. Cattell, M. J. Dinneen, and M. R. Fellows, *A simple linear-time algorithm for finding path-decompositions of small width.* Manuscript. To appear in Inform. Proc. Letters, 1994.

[29] F. R. K. Chung, *On the cutwidth and topological bandwidth of a tree*, SIAM J. Alg. Disc. Meth., 6 (1985), pp. 268–277.

[30] B. Courcelle, *Graph rewriting: an algebraic and logical approach*, in Handbook of Theoretical Computer Science, volume B, J. van Leeuwen, ed., Amsterdam, 1990, North Holland Publ. Comp., pp. 192–242.

[31] ——, *The monadic second-order logic of graphs I: Recognizable sets of finite graphs*, Information and Computation, 85 (1990), pp. 12–75.

[32] ——, *The monadic second-order logic of graphs III: Treewidth, forbidden minors and complexity issues*, Informatique Théorique, 26 (1992), pp. 257–286.

[33] B. Courcelle and M. Mosbah, *Monadic second-order evaluations on tree-decomposable graphs*, Theor. Comp. Sc., 109 (1993), pp. 49–82.

[34] N. D. Dendris, L. M. Kirousis, and D. M. Thilikos, *Fugitive-search games on graphs and related parameters*, in Proceedings 20th International Workshop on Graph Theoretic Concepts in Computer Science WG'94, E. W. Mayr, G. Schmidt, and G. Tinhofer, eds., Springer Verlag, Lecture Notes in Computer Science, vol. 903, 1995, pp. 26–37.

[35] D. Dolev, T. Leighton, and H. Trickey, *Planar embeddings of planar graphs*, Adv. in Comput. Res., 2 (1984), pp. 53–76.

[36] F. Drewes and H. Kreowski, *A note on hyperedge replacement*, in Proc. 4th International Workshop on Graph Grammars and Their Application to Computer Science, Springer Verlag, Lecture Notes in Computer Science, vol. 532, 1994, pp. 1–11.

[37] J. A. Ellis, I. H. Sudborough, and J. Turner, *The vertex separation and search number of a graph*, Information and Computation, 113 (1994), pp. 50–79.

[38] M. R. Fellows and M. A. Langston, *Nonconstructive advances in polynomial-time complexity*, Inform. Proc. Letters, 26 (1987), pp. 157–162.

[39] ——, *On search, decision and the efficiency of polynomial-time algorithms*, in Proceedings of the 21rd Annual Symposium on Theory of Computing, 1989, pp. 501–512.

[40] F. Gavril, *The intersection graphs of subtrees in trees are exactly the chordal graphs*, J. Comb. Theory Series B, 16 (1974), pp. 47–56.

[41] J. R. Gilbert, J. P. Hutchinson, and R. E. Tarjan, *A separator theorem for graphs of bounded genus*, J. Algorithms, 5 (1984), pp. 391–407.

[42] J. R. Gilbert, D. J. Rose, and A. Edenbrandt, *A separator theorem for chordal graphs*, SIAM J. Alg. Disc. Meth., 5 (1984), pp. 306–313.

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

[44] K. Y. Gorbunov, *An estimate of the tree-width of a graph which has not a given planar grid as a minor.* Manuscript, 1993.

[45] A. Habel, *Hyperedge Replacement: Grammars and Languages*, Lecture Notes in Computer Science, Vol. 643, Springer-Verlag, Berlin, 1992.

[46] A. Habel and H. J. Kreowski, *May we introduce to you: hyperedge replacement*, in Proc. Graph-Grammars and their Applications to Computer Science '86, H. Ehrig, M. Nagl, and A. Rosenberg, eds., Springer Verlag, Lecture Notes in Computer Science, vol. 291, 1987, pp. 15–26.

[47] W. Hohberg and R. Reischuk, *Decomposition of graphs — a uniform approach for the design of fast sequential and parallel algorithms on graphs.* Draft paper, 1989.

[48] ——, *A framework to design algorithms for optimization problems on graphs.* Preprint, April 1990.

[49] J. E. Hopcroft, W. Paul, and L. Valiant, *On time versus space*, J. ACM, 24 (1977), pp. 332–337.

[50] H. Kaplan and R. Shamir, *Pathwidth, bandwidth and completion problems to proper interval graphs with small cliques*, Technical Report 285/93, Inst. for Computer Science, Tel Aviv University, Tel Aviv, Israel, 1993. To appear in SIAM J. Comput.

[51] N. G. Kinnersley, *The vertex separation number of a graph equals its path width*, Inform. Proc. Letters, 42 (1992), pp. 345–350.

[52] N. G. Kinnersley and M. A. Langston, *Obstruction set isolation for the gate matrix layout problem*, Tech. Rep. CS-91-126, Computer Science Department, University of Tennessee, Knoxville, USA, 1991.

[53] L. M. Kirousis and C. H. Papadimitriou, *Interval graphs and searching*, Disc. Math., 55 (1985), pp. 181–184.

[54] ——, *Searching and pebbling*, Theor. Comp. Sc., 47 (1986), pp. 205–218.

[55] T. Kloks. Personal communication.

[56] ——, *Treewidth. Computations and Approximations*, Lecture Notes in Computer Science, Vol. 842, Springer-Verlag, Berlin, 1994.

[57] J. Lagergren, *The nonexistence of reduction rules giving an embedding into a k-tree*, Disc. Appl. Math., 54 (1994), pp. 219–223.

[58] A. S. LaPaugh, *Recontamination does not help to search a graph*, J. ACM, 40 (1993), pp. 224–245.

[59] C. Lautemann, *Decomposition trees: structured graph representation and efficient algorithms*, in Proceedings CSSP'88, Springer Verlag, Lecture Notes in Computer Science, vol. 299, 1988, pp. 28–39.

[60] A. Lingas, *Subgraph isomorphism for easily separable graphs with bounded valence*, in Proc. 11th Workshop on Graph-Theoretic concepts in Computer Science, 1985, pp. 217–229.

[61] R. J. Lipton and R. E. Tarjan, *A separator theorem for planar graphs*, SIAM J. Appl. Math., 36 (1979), pp. 177–189.

[62] J. W. H. Liu, *The role of elimination trees in sparse factorization*, SIAM J. Matrix Analysis and Applications, 11 (1990), pp. 134–172.

[63] F. S. Makedon, C. H. Papadimitriou, and I. H. Sudborough, *Topological bandwidth*, SIAM J. Alg. Disc. Meth., 6 (1985), pp. 418–444.

[64] R. H. Möhring, *Graph problems related to gate matrix layout and PLA folding*, in Computational Graph Theory, Comuting Suppl. 7, E. Mayr, H. Noltemeier, and M. Sysło, eds., Springer Verlag, 1990, pp. 17–51.

[65] ——, *Triangulating graphs without asteroidal triples*, Technical Report 365/1993, Technical University Berlin, 1993.

[66] M. H. Mosbah, *Constructions d'Algorithmes Pour les Graphes Structurés par des Méthodes Algébriques et Logiques*, PhD thesis, Université Bordeaux-I, 1992.

[67] V. Naumann, *Measuring the distance to series-parallel graphs by path expressions*, Tech. Rep. 376/1994, Technische Universität Berlin, Berlin, Germany, 1994.

[68] S. Ramachandramurthi, *Algorithms for VLSI Layout Based on Graph Width Metrics*, PhD thesis, Computer Science Department, University of Tennessee, Knoxville, Tennessee, USA, 1994.

[69] N. Robertson and P. D. Seymour, *Graph minors. XX. Wagner's conjecture.* In prepartion.

[70] ——, *Graph minors. I. Excluding a forest*, J. Comb. Theory Series B, 35 (1983), pp. 39–61.

[71] ——, *Generalizing Kuratowskis theorem*, Congressus Numerantium, 45 (1984), pp. 129–138.

[72] ——, *Graph minors. III. Planar tree-width*, J. Comb. Theory Series B, 36 (1984), pp. 49–64.

[73] ——, *Graph width and well-quasi ordering: a survey*, in Progress in Graph Theory, J. A. Bondy and U. S. R. Murty, eds., Toronto, 1984, Academic Press, pp. 399–406.

[74] ——, *Graph minors — a survey*, in Surveys in Combinatorics, I. Anderson, ed., Cambridge Univ. Press, 1985, pp. 153–171.

[75] ——, *Graph minors. II. Algorithmic aspects of tree-width*, J. Algorithms, 7 (1986), pp. 309–322.

[76] ——, *Graph minors. V. Excluding a planar graph*, J. Comb. Theory Series B, 41 (1986), pp. 92–114.

[77] ——, *Graph minors. VI. Disjoint paths across a disc*, J. Comb. Theory Series B, 41 (1986), pp. 115–138.

[78] ——, *Graph minors. VII. Disjoint paths on a surface*, J. Comb. Theory Series B, 45 (1988), pp. 212–254.

[79] ——, *Graph minors. IV. Tree-width and well-quasi-ordering*, J. Comb. Theory Series B, 48 (1990), pp. 227–254.

[80] ——, *Graph minors. IX. Disjoint crossed paths*, J. Comb. Theory Series B, 49 (1990), pp. 40–77.

[81] ——, *Graph minors. VIII. A Kuratowski theorem for general surfaces*, J. Comb. Theory Series B, 48 (1990), pp. 255–288.

[82] ——, *Graph minors. X. Obstructions to tree-decomposition*, J. Comb. Theory Series B, 52 (1991), pp. 153–190.

[83] ——, *Graph minors. XV. Etending an embedding.* Manuscript, 1991.

[84] ——, *Graph minors. XVI. Excluding a non-planar graph.* Manuscript, 1991.

[85] ——, *Graph minors. XVII. Taming a vortex.* Manuscript, 1991.

[86] ——, *Graph minors. XXII. Irrelevant vertices in linkage problems.* Manuscript, 1992.

[87] ——, *Excluding a graph with one crossing*, in Graph Structure Theory, Proceedings of the AMS-IMS-SIAM Joint Summer Research Conference, Seattle WA, June 1991, N. Robertson and P. Seymour, eds., Providence, RI, 1993, American Math. Soc., pp. 669–675. Contemp. Math. 147.

[88] ——, *Graph minors. XI. Distance on a surface*, J. Comb. Theory Series B, 60 (1994), pp. 72–106.

[89] ——, *Graph minors. XII. Excluding a non-planar graph.* To appear in J. Comb. Theory, Ser. B, 1995.

[90] ——, *Graph minors. XIII. The disjoint paths problem*, J. Comb. Theory Series B, 63 (1995), pp. 65–110.

[91] N. Robertson, P. D. Seymour, and R. Thomas, *Quickly excluding a planar graph*, Tech. Rep. TR89-16, DIMACS, 1989.

[92] ——, *A survey of linkless embeddings*, in Graph Structure Theory, Proceedings of the AMS-IMS-SIAM Joint Summer Research Conference, Seattle WA, June 1991, N. Robertson and P. Seymour, eds., Providence, RI, 1993, American Math. Soc., pp. 125–136. Contemp. Math. 147.

[93] D. J. Rose, *On simple characterization of k-trees*, Disc. Math., 7 (1974), pp. 317–322.

[94] P. Scheffler, *Die Baumweite von Graphen als ein Maß für die Kompliziertheit algorithmischer Probleme*, PhD thesis, Akademie der Wissenschaften der DDR, Berlin, 1989.

[95] D. Seese, *Tree-partite graphs and the complexity of algorithms*, in Proc. 1985 Int. Conf. on Fundamentals of Computation Theory, Lecture Notes in Computer Science 199, L. Budach, ed., Berlin, 1985, Springer Verlag, pp. 412–421.

[96] P. D. Seymour and R. Thomas, *Graph searching and a minimax theorem for tree-width*, Technical Report 89-1, DIMACS, 1989.

[97] M. M. Sysło, *Characterisations of outerplanar graphs*, Disc. Math., 26 (1979), pp. 47–53.

[98] A. Takahashi, S. Ueno, and Y. Kajitani, *Minimal acyclic forbidden minors for the family of graphs with bounded path-width*, in SIGAL 91-19-3, IPSJ, 1991. To appear in: *Annals of discrete mathematics* (Proceedings of 2nd Japan conference on graph theory and combinatorics, 1990).

[99] ——, *Mixed-searching and proper-path-width*, tech. rep., Tokyo Institute of Technology, 1991.

[100] M. Thorup, *Structured programs have small tree-width and good register allocation*, Technical Report DIKU-TR-95/18, Department of Computer Science, University of Copenhagen, Denmark, 1995.

[101] J. van Leeuwen, *Graph algorithms*, in Handbook of Theoretical Computer Science, A: Algorithms and Complexity Theory, Amsterdam, 1990, North Holland Publ. Comp., pp. 527–631.

[102] J. van Leeuwen and R. B. Tan, *Compact routing methods: A survey*, Technical Report UU-CS-1995-05, Department of Computer Science, Utrecht University, Utrecht, 1995.

[103] W. Vogler, *On hyperedge replacement and BNLC graph grammars*, Disc. Appl. Math., 46 (1993), pp. 253–273.

[104] T. V. Wimer, *Linear Algorithms on k-Terminal Graphs*, PhD thesis, Dept. of Computer Science, Clemson University, 1987.

Contents

Index