ELSEVIER

Discrete Applied Mathematics 77 (1997) 259–270

DISCRETE APPLIED MATHEMATICS

Measuring the vulnerability for classes of intersection graphs

Dieter Kratsch[a,1], Ton Kloks[b], Haiko Müller[a,*]

[a] *Fakultät für Mathematik und Informatik, Friedrich-Schiller-Universität, 07740 Jena, Germany*
[b] *Department of Mathematics and Computing Science, Eindhoven University of Technology, P.O.Box 513, 5600 MB Eindhoven, The Netherlands*

Received 20 June 1995; revised 13 September 1996

Abstract

A general method for the computation of various parameters measuring the vulnerability of a graph is introduced. Four measures of vulnerability are considered, i.e., the toughness, scattering number, vertex integrity and the size of a minimum balanced separator. We show how to compute these parameters by polynomial-time algorithms for various classes of intersection graphs like permutation graphs, bounded dimensional cocomparability graphs, interval graphs, trapezoid graphs and circular versions of these graph classes.

Keywords: Graph algorithms; Trapezoid graphs

1. Introduction

Among the most studied graph parameters, is the connectivity of a graph. Connectivity (or edge connectivity) measures the 'vulnerability' of a graph (or network). In other words the connectivity determines in a certain sense the resistance of the graph to operations such as deletions of vertices or edges.

In the early 1970s it was found that the connectivity only partly reflects the ability of graphs to retain certain degrees of connectedness after operations such as the removal of vertices or edges (cf. [2]). Therefore, other measures were introduced and studied.

The names of these new graph parameters (binding number, toughness, etc.) suggest that if the value of the parameter is large enough, the vertices of the graph are well tied together. The fact that these new parameters are indeed a reasonably good measure for the vulnerability of graphs is perhaps best illustrated by the results on hamiltonicity.

* Corresponding author. E-mail: hm@minet.uni-jena.de.
[1] This research was done while the author was with IRISA Rennes, France.

PII S0166-218X(96)00133-3

Not only the connectedness of a graph can be guaranteed by a lower bound on some of these parameters, but even the existence of a hamiltonian circuit or path. For example, if the *binding number* of a graph is at least $\frac{3}{2}$, then the existence of a hamiltonian circuit is guaranteed [17].

One of the first of these newly introduced parameters was the *toughness* defined by Chvátal in 1973 [6]. The primal motivation for introducing the toughness was to obtain necessary conditions for the hamiltonicity of graphs. From the definition it is immediately clear that every hamiltonian graph has toughness at least one (or: is 1-tough). This condition however is not sufficient, which was illustrated in [6] by the construction of an infinite family of nonhamiltonian graphs with toughness $\frac{3}{2}$. This led Chvátal to the conjecture of the existence of a constant t such that every t-tough graph is hamiltonian. This conjecture is still open.

Apart from being hamiltonian, the importance of the toughness of graphs can by now be illustrated by a great number of papers. For example in [15], Plummer shows, among other things, that a t-tough graph is t-extendable. Other examples (and conjectures) of graph parameters (minimum degree sums of sets of independent vertices of fixed order) in relation to toughness are given in [16].

From an algorithmic point of view, it is somewhat unfortunate that the problem of recognizing t-tough graphs is coNP-complete for every fixed positive rational t [3]. On the other side, in this paper we show that for many important graph classes the toughness can be computed efficiently.

Another parameter we consider in this paper is the *scattering number* which was introduced in 1978 by Jung in [10]. In his paper Jung shows, among other things that, when the scattering number of graphs in a certain class (called D^*-graphs, which are comparability graphs of multitrees) is known, then it is also known whether the graph is hamiltonian or not.

In [10] the author calls the scattering number the 'additive dual' of the toughness. This additive appearance of the parameter, and the results on the integrity parameter (see below), perhaps suggest that the recognition of graphs with bounded scattering number would be easier than the recognition of graphs of bounded toughness. However, from the definition it easily follows that the toughness of a graph is more than one if and only if the scattering number is less than zero. Hence, it follows that the problem 'Given a graph G, decide whether the scattering number is larger than zero' is NP-complete. We show that our methods can be applied to compute the scattering number for many important graph classes like interval graphs, permutation graphs etc.

The third parameter for which we illustrate our methods is the *vertex integrity* of a graph. This parameter was introduced in [2] and compared with other parameters such as connectivity, toughness and binding number. The authors compute for cliques, complete bipartite graphs, powers of cycles and some other graph classes the parameters, and conclude that the integrity is in some sense the better measure for the vulnerability of graphs.

'Given a graph G and an integer k, deciding whether the integrity of G is at most k' is an NP-complete problem, even when restricted to planar graphs [7]. However, in [7] it is shown that for every constant t the class of graphs with integrity at most t is minor closed, and it follows from the results of Robertson and Seymour that for every constant t this class is recognizable in polynomial time. From a practical point of view, these results are of limited use only, since, although the algorithm is known to exist, there is no constructive way known to obtain it. Even if it were known, it is believed that it would involve astronomical constants. Our algorithms computing the integrity of graphs in many important classes are efficient, constructive and do not hide any constants that are exponential in the integrity of the graph.

We also consider the *minimum balanced separator* problem. The best known algorithms computing balanced separators on general graphs are polynomial time approximation algorithms of worst case performance ratio $O(\log n)$ [11]. The importance of finding balanced separators for solving all kinds of problems using divide and conquer techniques, is perhaps best illustrated by the results on planar graphs [12]. These applications and results of Lipton and Tarjan [12] were extended in [1] to any class of graphs with an excluded minor. We present polynomial time algorithms exactly computing a balanced separator of minimum cardinality for graphs from many other special graph classes.

There are by now many results known for the parameters mentioned above. However, from an algorithmic point of view, our algorithms are the first to compute efficiently these parameters for many nontrivial graph classes. Indeed, our approach is widely applicable (cf. Tables 1 and 2).

The algorithms we present compute *component number vectors* and *maximum component order vectors* (cf. [14]) which could be of interest for solving other problems for these graph classes as well. In fact, these two vectors give a lot of information on the vulnerability of a graph; thus they should be of interest for characterizing vulnerability properties of graphs in general.

2. Preliminaries

First we give some notations for undirected graphs. For a graph G we denote the order (number of vertices) by n. $G[W]$ denotes the subgraph of the graph $G=(V,E)$ induced by the vertex set $W \subseteq V$. We denote the number of (connected) components of a graph G by $c(G)$ and the maximum order of a component of G by $n(G)$. A set $S \subseteq V$ is said to be a *separator* of a graph G if $c(G[V \setminus S]) > 1$. We denote the (vertex) connectivity of the graph G by $\kappa(G)$ and we denote the maximum size of an independent set of the graph G by $\alpha(G)$.

We refer to [5] for definitions and properties of graph classes not given here.

2.1. Problems

We now pose some definitions concerning the graph problems which we investigate here.

Definition 1. The *toughness* of a complete graph K_n is $t(K_n)=\infty$. If G is not complete, then

$$t(G)=\min\left\{\frac{|S|}{c(G[V\backslash S])}:S \text{ separator of } G\right\}.$$

Definition 2. The *scattering number* of a complete graph K_n is $\mathrm{sc}(K_n)=-\infty$. If G is not complete, then

$$\mathrm{sc}(G)=\max\{c(G[V\backslash S])-|S|:S \text{ separator of } G\}.$$

Definition 3. The *vertex integrity* $I(G)$ of a graph $G=(V,E)$ is defined as

$$I(G)=\min\{|S|+n(G[V\backslash S]):S\subseteq V\}.$$

The *minimum balanced separator* problem is considered under different formulations in the literature (cf. [1, 11]). We study a somewhat general version of the problem.

Definition 4. Let β be a real number between 0 and 1 and let W be a subset of vertices. A separator $S\subseteq V$ of the graph $G=(V,E)$ is an *β-balanced separator for W* if any component of $G[V\backslash S]$ contains at most $\beta\cdot|W|$ vertices of W.

2.2. Separation vectors

We explore vectors $(c_i(G))_{i=0}^n$ and $(n_i(G))_{i=0}^n$ which describe quite well the separation behaviours of a graph G. They provide a lot of information on the vulnerability and the reliability of a graph, when node failures are investigated. Some of the vulnerability measures proposed in the literature (cf. [2]) can easily be derived from our vectors.

The *component number vector* $(c_i(G))_{i=0}^n$ allows to compute the scattering number $\mathrm{sc}(G)$ and the toughness $t(G)$.

Definition 5. Let $G=(V,E)$ be a graph. For $i\in\{0,1,\ldots,n\}$ we define $c_i(G)$ to be the maximum number of components of the graph $G[V\backslash S]$ taken over all subsets $S\subseteq V$ with $|S|=i$, i.e., $c_n(G)=0$ and for $i<n$:

$$c_i(G)=\max\{c(G[V\backslash S]):S\subseteq V,\ |S|=i\}.$$

Remark 1. For any graph G, $c_{i-1}(G)\leqslant c_i(G)$ for $1\leqslant i\leqslant n-\alpha(G)$ and $c_i(G)=n-i$ for $n-\alpha(G)\leqslant i\leqslant n$ hold. For any non-complete graph G, the following hold:

$$t(G)=\min\left\{\frac{i}{c_i(G)}:0\leqslant i\leqslant n \text{ and } c_i(G)>1\right\},$$

$$\mathrm{sc}(G)=\max\{c_i(G)-i:0\leqslant i\leqslant n \text{ and } c_i(G)>1\},$$

$$\alpha(G)=\max\{c_i(G):0\leqslant i\leqslant n\},$$

$$\kappa(G)=\min\{i:0\leqslant i\leqslant n \text{ and } c_i(G)>1\}.$$

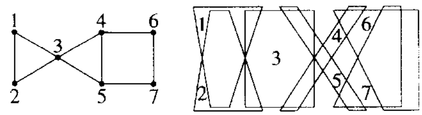

Fig. 1. Trapezoid graph and trapezoid diagram.

The *maximum component order vector* $(n_i(G))_{i=0}^n$ allows to compute the vertex integrity and the size of a minimum β-balanced separator for V.

Definition 6. Let $G=(V,E)$ be a graph. For $i \in \{0,1,\ldots,n\}$ we define $n_i(G)$ to be the maximum cardinality of a component of the graph $G[V\backslash S]$ taken over all subsets $S \subseteq V$ with $|S|=i$, i.e., $n_n(G)=0$ and for $i<n$:

$$n_i(G)=\min\{n(G[V\backslash S]) : S \subseteq V,\ |S|=i\}.$$

Remark 2. For any graph G, $n_{i-1}(G) \geqslant n_i(G)$ hold for $1 \leqslant i \leqslant n$ and

$$I(G)=\min\{i+n_i(G) : 0 \leqslant i \leqslant n\},$$

$$\alpha(G)=n-\min\{i : 0 \leqslant i \leqslant n \text{ and } n_i(G)=1\}.$$

2.3. Trapezoid graphs

Trapezoid graphs are the intersection graphs of finite collections of trapezoids between two parallel lines [8]. Both the interval graphs and the permutation graphs form subclasses of the trapezoid graphs [8].

Definition 7. A *trapezoid diagram* consists of two parallel horizontal lines and a collection of trapezoids having two corners on each of the horizontal lines. A graph $G=(V,E)$ is a *trapezoid graph* if there is a trapezoid diagram and a bijection assigning to each vertex v of V a trapezoid $\mathrm{td}(v)$ such that $u,v \in V$ are joined by an edge if and only if $\mathrm{td}(u)$ and $\mathrm{td}(v)$ have a nonempty intersection.

There is a $O(n^2)$ time recognition algorithm for trapezoid graphs [13]. Moreover, this algorithm also computes a trapezoid diagram if the given graph G is a trapezoid graph. An example of a trapezoid graph and one of its trapezoid diagrams is given in Fig. 1.

We assume for the remainder of the paper that the trapezoid graph G is given by a trapezoid diagram and we identify the vertex v and its trapezoid $\mathrm{td}(v)$. We may assume that each point on a horizontal line is a corner of at most one trapezoid in the diagram. Thus the graph is uniquely determined by the sequence of corners on both of the horizontal lines.

Definition 8. A *scanline* in the trapezoid diagram is any straight line segment with one end point on each horizontal line such that these end points do not coincide with any corner of a trapezoid $\mathrm{td}(v)$.

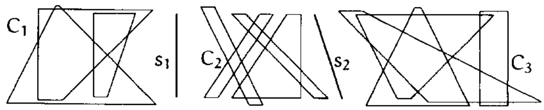

Fig. 2. Components and scanlines.

Any scanline s *generates* a set $S(s)$ of vertices of the graph, namely those vertices v for which $\mathrm{td}(v)$ and s have nonempty intersection.

We say that two scanlines s and s' are *equivalent* if there is no corner of a trapezoid between the endpoints of s and s' on both of the two horizontal lines. (Notice that for equivalent scanlines s and s': $S(s)=S(s')$).

Observation 1. A maximal set of pairwise nonequivalent scanlines of any trapezoid diagram of a trapezoid graph of order n consists of $(2n+1)^2$ scanlines.

The usefulness of scanlines becomes clear as follows. Consider a scanline s such that there is at least one trapezoid to the left of it and at least one trapezoid to the right of it. Take all those trapezoids out of the diagram that intersect the scanline s. Consider the graph corresponding with the trapezoids that are left in the diagram. Since there is no path in the new diagram connecting the trapezoids to the left and the trapezoids to the right of s, the corresponding graph $G[V \backslash S(s)]$ is disconnected. Hence, in this manner, the scanline corresponds with a separator in the graph $G=(V,E)$.

Definition 9. The scanline s_1 is *left* of the scanline s_2 if the end point of s_1 is left of the end point of s_2 on both horizontal lines. Let s_1 and s_2 be nonequivalent scanlines such that s_1 is left of s_2. Then the *piece* $\mathscr{P}(s_1,s_2)$ consists of all vertices v for which its trapezoid $\mathrm{td}(v)$ is *between* s_1 and s_2, i.e., the corners of $\mathrm{td}(v)$ are between the end points of s_1 and s_2 on both horizontal lines.

We say that two scanlines s and s' are *noncrossing* if s is left of s' or s' is left of s.

Lemma 2.1. *Let $G=(V,E)$ be a trapezoid graph. For any $t \in \{c(G)+1, c(G)+2,\ldots,\alpha(G)\}$ there exists a set $S \subseteq V$ and a collection of pairwise noncrossing scanlines $(s_j)_{j=1}^{r-1}$ such that $S=\bigcup_{j=1}^{r-1} S(s_j)$, $c(G[V\backslash S])=r \geqslant t$ and $|S|=\min\{i : c_i(G) \geqslant t\}$.*

Proof. Clearly, for any $t \in \{c(G)+1,\ldots,\alpha(G)\}$ there is a set S with $c(G[V\backslash S]) \geqslant t$, such that $c(G[V\backslash S']) < t$ for any set S' with $|S'| < |S|$. Consider the diagram of $G[V\backslash S]$, i.e., remove all trapezoids $\mathrm{td}(v)$ for $v \in S$ from the diagram of G. Let $C_1, C_2, \ldots, C_r$, $r \geqslant t$, be the components in the diagram of $G[V\backslash S]$ taken from left to right. We choose scanlines s_j for $j \in \{1,2,\ldots,r-1\}$ such that s_j is between C_j and C_{j+1}; see Fig. 2. This creates $r-1$ scanlines such that $S \supseteq \bigcup_{j=1}^{r-1} S(s_j)$. Finally, by the construction of the scanlines $(s_j)_{j=1}^{r-1}$, we have $c(G[V\backslash \bigcup_{j=1}^{r-1} S(s_j)]) \geqslant r \geqslant t$; thus $S\backslash \bigcup_{j=1}^{r-1} S(s_j) \neq \emptyset$ would contradict the choice of S. $\square$

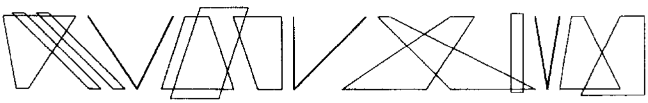

Fig. 3. Scanlines s_j and s'_j.

Lemma 2.2. *Let $G=(V,E)$ be a trapezoid graph. For any $t \in \{0,1,\dots,n\}$ there exists a set $S \subseteq V$ and a collection of pairwise noncrossing scanlines $(s_j)_{j\in J}$ such that $S=\bigcup_{j\in J} S(s_j)$, $n(G[V\backslash S]) \leqslant t$ and $|S| = \min\{i : n_i(G) \leqslant t\}$.*

Proof. Again, for any $t \in \{0,1,\dots,n(G)-1\}$ there is a set T with $n(G[V\backslash T]) \leqslant t$, such that $n(G[V\backslash T'])>t$ for any set T' with $|T'|<|T|$. Let $C_1, C_2, \dots, C_r$ be the components in the diagram of $G[V\backslash T]$, taken from left to right. We choose scanlines s_0 and s_r being totally to the left and totally to the right, respectively, of all trapezoids and for $j=1,\dots,r-1$ we choose scanlines s_j between C_j and C_{j+1}. Let $S'=\bigcup_{j=0}^{r} S(s_j)$. Then $c(G[V\backslash S'])=r$. Let $C'_1, C'_2, \dots, C'_r$ be the components in the diagram of $G[V\backslash S']$, taken from left to right. Clearly, we have $C_j \subseteq C'_j$ for $j=1,\dots,r$. If $T=S'$ we are done with $S=S'$. Otherwise there is an index j such that $C_j \subset C'_j$. In this case we create a new scanline s'_j sharing the endpoint on the bottom line with s_j while on the top line the endpoint of s'_j is left of that of s_j; more precisely, the interval between these endpoints contains $|C'_j \backslash C_j|$ right upper corners of trapezoids $\mathrm{td}(v)$ for vertices v of C'_j. Hence $|\mathscr{P}(s_{j-1}, s'_j)| = |C_j|$. We define $S = S' \cup \bigcup_{j:C_j \subset C'_j} S(s'_j)$. This implies $|S|=|T|$, $n(G[V\backslash S]) = n(G[V\backslash T])$, hence $|S| = \min\{i : n_i(G) \leqslant t\}$. $\square$

3. The algorithms for trapezoid graphs

Given a trapezoid diagram of a trapezoid graph $G=(V,E)$, the algorithms computing $(c_i(G))_{i=0}^n$ and $(n_i(G))_{i=0}^n$ solve suitable shortest-path problems on auxiliary directed acyclic graphs whose vertex set is a maximal set of pairwise nonequivalent scanlines in the diagram. Among these scanlines we denote by s_{L} and s_{R} the scanline totally to the left and totally to the right, respectively, of all trapezoids $\mathrm{td}(v)$ of the trapezoid diagram.

3.1. The auxiliary graph for the component number vector

Construct the following auxiliary graph $D^c(G)$. The vertex set of $D^c(G)$ is a maximal set of pairwise nonequivalent scanlines in the diagram. There is an edge directed from s_1 to s_2 in D^c if $\mathscr{P}(s_1, s_2)$ is nonempty and induces a connected subgraph of G. The weight of an edge (s_1, s_2) of D^c is $w(s_1,s_2) = |S(s_1)\backslash S(s_2)|$.

Recall that for any graph G holds $c_i(G)=1$ if $i<\kappa(G)$ and $c_i(G)=n-i$ if $i>n-\alpha(G)$.

Lemma 3.1. *Let $w^c(t)$, $c(G)+1 \leqslant t \leqslant \alpha(G)$, be the minimum weight $\sum_{j=1}^{r} w(s_{j-1}, s_j)$ of a path $P_t = (s_0, s_1, \ldots, s_r)$, $r \geqslant t$, $s_0 = s_L$ and $s_r = s_R$, among all paths in $D^c(G)$ from s_L to s_R on at least t edges. Then $c_i(G) = \max\{t : w^c(t) \leqslant i\}$ for any $i \in \{\kappa(G), \ldots, n - \alpha(G)\}$.*

Proof. The paths from s_L to s_R in the acyclic directed graph $D^c(G)$ have at most $\max\{c_i(G) : 0 \leqslant i \leqslant n\} = \alpha(G)$ edges. By Lemma 2.1, there is a collection of pairwise noncrossing scanlines $(s_j)_{j=1}^{r-1}$, $r \geqslant t$, such that each of $\mathscr{P}(s_L, s_1)$, $\mathscr{P}(s_j, s_{j+1})$ for all $j \in \{1, 2, \ldots, r-2\}$ and $\mathscr{P}(s_{r-1}, s_R)$ is nonempty, induces a connected subgraph of G and $S = \bigcup_{j=1}^{r-1} S(s_j)$ is a set of minimum cardinality with $c(G[V \setminus S]) \geqslant t$.

Hence, a shortest (minimum weight) path $P_t = (s_0, s_1, \ldots, s_r)$, $r \geqslant t$, from s_L to s_R in $D^c(G)$ determines such a set as $S = \bigcup_{j=1}^{r-1} S(s_j)$. Moreover, $w^c(t) = \min\{i : c_i(G) \geqslant t\}$ for any $t \in \{c(G)+1, \ldots, \alpha(G)\}$. Inverting this formulae gives $c_i(G) = \max\{t : w^c(t) \leqslant i\}$ for any $i \in \{\kappa(G), \ldots, n - \alpha(G)\}$. $\square$

3.2. The auxiliary graph for the maximum component order vector

Construct the following auxiliary graphs $D_t^n(G)$, $t \in \{0, 1, \ldots, n\}$. The vertex set of $D_t^n(G)$ is a maximal set of pairwise nonequivalent scanlines of the diagram. There is an edge directed from s_1 to s_2 in D_t^n if $|\mathscr{P}(s_1, s_2)| \leqslant t$. The weight of an edge (s_1, s_2) of $D_t^n(G)$ is $w(s_1, s_2) = |S(s_1) \setminus S(s_2)|$.

Lemma 3.2. *We define w_t^n to be the minimum weight $\sum_{j=1}^{r} w(s_{j-1}, s_j)$ of a path $P = (s_0, s_1, \ldots, s_r)$, $s_0 = s_L$ and $s_r = s_R$, among all paths in $D_t^n(G)$ from s_L to s_R. Then $n_i(G) = \min\{t : w_t^n \leqslant i\}$ for any $i \in \{0, 1, \ldots, n\}$.*

Proof. By Lemma 2.2, for any $t \in \{0, 1, \ldots, n\}$ there is a collection of pairwise noncrossing scanlines $(s_j)_{j=1}^{r-1}$, s_{j-1} left of s_j, such that $|\bigcup_{j=1}^{r-1} S(s_j)| = \min\{i : n_i(G) \leqslant t\}$. This sequence corresponds to the path $P = (s_0, s_1, \ldots, s_r)$ in $D_t^n(G)$. Thus, $w_t^n = \min\{i : n_i(G) \leqslant t\}$. Inverting this formulae gives $n_i(G) = \min\{t : w_t^n \leqslant i\}$ for any $i \in \{0, 1, \ldots, n\}$. $\square$

Observation 2. The auxiliary directed graphs $D^c(G)$ and $D_t^n(G)$ have both $O(n^2)$ vertices and $O(n^4)$ edges for any trapezoid graph G of order n.

Theorem 3.3. *There are $O(n^5)$ algorithms computing for given trapezoid graph G the scattering number $\mathrm{sc}(G)$, the toughness $t(G)$, the vertex integrity $I(G)$ and the vectors $(c_i(G))_{i=0}^{n}$ and $(n_i(G))_{i=0}^{n}$.*

Proof. It is not very hard to show that for given trapezoid diagram each of the two auxiliary graphs and the corresponding edge weights can be computed in time $O(n^4)$. A suitable method is described in detail in [4] for permutation graphs.

Given $D^c(G)$, the minimum weight of a path from s_L to s_R on at least t edges for all $t \in \{c(G)+1, \ldots, \alpha(G)\}$ can be determined in time $O(|V(D^c)| + n \cdot |E(D^c)|)$ by a dynamic programming computing for any vertex v the values $w_k(v)$, $1 \leqslant k \leqslant \alpha(G)$, i.e., the minimum weight of a path from s_L to v on at least k edges. Given $D_t^n(G)$ for some fixed $t \in \{0, 1, \ldots, n\}$, the minimum weight of a path from s_L to s_R can clearly be determined in time $O(|V(D_t^n)| + |E(D_t^n)|)$ by standard methods.

This together with Lemmas 3.1 and 3.2 verifies that $(c_i(G))_{i=0}^n$ and $(n_i(G))_{i=0}^n$ can be computed in time $O(n^5)$. Remarks 1 and 2 show that $\mathrm{sc}(G)$, $t(G)$ and $I(G)$ can easily be computed within the same time bound. □

3.3. *The auxiliary graph for minimum β-balanced separators*

Given a trapezoid graph $G = (V, E)$, a set $W \subseteq V$ and a real β between 0 and 1 we construct the auxiliary graph D_t^* for $t = \lfloor \beta \cdot |W| \rfloor$ as follows. The vertex set of $D_t^*(G)$ is a maximal set of pairwise nonequivalent scanlines of the diagram. There is an edge directed from s_1 to s_2 in D_t^* if the piece $\mathscr{P}(s_1, s_2)$ has at most t vertices in W. The weight of such an edge of $D_t^*(G)$ is $w(s_1, s_2) = |S(s_1) \backslash S(s_2)|$.

Lemma 3.4. *Let $G = (V, E)$ be a trapezoid graph and $W \subseteq V$. For any $t \in \{0, 1, \ldots, n\}$ there exists a set $S \subseteq V$ and a collection of pairwise noncrossing scanlines $(s_j)_{j \in J}$ such that each component of $G[V \backslash S]$ contains at most t vertices of W and among all such sets S has minimum cardinality.*

Proof. The proof follows the lines of the proof of Lemma 2.2, except that the interval between the endpoints of s_j and s_j' on the top line contains $|C_j' \backslash C_j|$ right upper corners of trapezoids $\mathrm{td}(v)$ for vertices v of $C_j' \cap W$. □

Lemma 3.5. *We define w_t^* to be the minimum weight $\sum_{j=1}^r w(s_{j-1}, s_j)$ of a path $P = (s_0, s_1, \ldots, s_r)$, $s_0 = s_L$ and $s_r = s_R$, among all paths in $D_t^*(G)$ from s_L to s_R. Then $S = \bigcup_{j=1}^{r-1} S(s_j)$ is a β-balanced separator of minimum cardinality w_t^*.*

Proof. By Lemma 3.4 and the one-to-one correspondence between $s_L - s_R$-paths in D_t^* and sets of pairwise noncrossing scanlines in D_t^*. □

By Lemma 3.5 and since the auxiliary directed graph D_t^* with $t = \lfloor \beta \cdot |W| \rfloor$ has $O(n^2)$ vertices and $O(n^4)$ edges for any trapezoid graph G of order n one obtains a $O(n^4)$ algorithm for the minimum balanced separator problem on trapezoid graphs.

This can be improved using the k-small scanline approach described in detail in [4] for permutation graphs.

Theorem 3.6. *There is a $O(k^2 n^2)$ algorithm computing a minimum β-balanced separator for W, given a trapezoid graph $G = (V, E)$, a real β between 0 and 1 and*

$W \subseteq V$. Here k denotes the size of a minimum β-balanced separator for W of the input graph G.

We mention that k-small scanlines as well as modified edge weights of the auxiliary directed graphs can be used to speed up some of the algorithms, e.g., there is a $O(n^4)$ algorithm computing the scattering number of trapezoid graphs.

4. Other classes of intersection graphs

In this section we report that the approach demonstrated for trapezoid graphs can be used for many other classes of intersection graphs.

There are classes of intersection graphs with an intersection model similar to the trapezoid diagram, listed in Table 1, which we propose to call *graph classes with linear model*. (d-trapezoid graphs were recently introduced in [9].) Any of the problems considered in Section 3 for trapezoid graphs can be solved on these classes with the same approach (cf. Table 1).

The graph classes of Table 1 have natural generalizations (in the way circular permutation graphs generalize permutation graphs) by somehow transforming the linear

Table 1
Graph classes with linear model

Graph class	Number of vertices of the auxiliary graphs	Number of edges of the auxiliary graphs	Running time of the algorithms computing $(c_i)_{i=0}^n$ and $(n_i)_{i=0}^n$
Interval graphs	$O(n)$	$O(n^2)$	$O(n^3)$
Permutation graphs	$O(n^2)$	$O(n^4)$	$O(n^5)$
Trapezoid graphs	$O(n^2)$	$O(n^4)$	$O(n^5)$
Cocomparability graphs of dimension at most d	$O(n^d)$	$O(n^{2d})$	$O(n^{2d+1})$
d-trapezoid graphs	$O(n^d)$	$O(n^{2d})$	$O(n^{2d+1})$

Table 2
Graph classes with circular model

Graph class with circular model	Related graph class with linear model	Running time of the algorithms computing $(c_i)_{i=0}^n$ and $(n_i)_{i=0}^n$
Circular-arc graphs	Interval graphs	$O(n^4)$
Circular permutation graphs	Permutation graphs	$O(n^7)$
Circular trapezoid graphs	Trapezoid graphs	$O(n^7)$
Circular d-cocomparability graphs	Cocomparability graphs of dimension at most d	$O(n^{3d+1})$
Circular d-trapezoid graphs	d-trapezoid graphs	$O(n^{3d+1})$

intersection model into a 'circular' one. The corresponding classes are given in Table 2 and we propose to call them *graph classes with circular model.*

The last three of these are new graph classes. Trying to give the reader an idea how to define them we describe the intersection model of circular trapezoid graphs. Instead of trapezoids between two parallel lines as in the model of trapezoid graphs we consider generalized trapezoids between two concentric circles. Thereby two parallel lines of the generalized trapezoid are arcs of each of the two circles and the two other lines of the generalized trapezoid are spiral segments.

Any of the problems considered in this paper can be solved in polynomial time for the graph classes of Table 2 by reducing the problem on a given graph to the same problem on a 'reasonable small' collection of induced subgraphs belonging to the related graph class with linear model (cf. Table 2).

Note that the stated running times in Tables 1 and 2 assume that the input graph is given together with a corresponding intersection model. For the classes of interval graphs, permutation graphs, trapezoid graphs, circular arc graphs and circular permutation graphs there are algorithms which compute such a model within the stated running time. For the classes of cocomparability graphs of dimension at most d ($d \geqslant 3$), d-trapezoid graphs ($d \geqslant 3$), circular trapezoid graphs, circular d-cocomparability graphs ($d \geqslant 3$) and circular d-trapezoid graphs ($d \geqslant 2$) polynomial time algorithms computing such an intersection model are not known.

Acknowledgements

The first author is grateful to Hans Bodlaender (Utrecht) for valuable discussions on the most efficient variants of our algorithms. The authors are grateful to Annelies Jacobs for doing a lot of the bibliographical research for this topic.

References

[1] N. Alon, P. Seymour and R. Thomas, A separator theorem for graphs with an excluded minor and its applications, Proceedings of the 22nd Symposium on Theory of Computing (1980) 293–299.

[2] C.A. Barefoot, R. Entringer and H. Swart, Vulnerability in graphs – a comparative survey, J. Combin. Math. Combin. Comput. 1 (1987) 13–22.

[3] D. Bauer, S.L. Hakimi and E. Schmeichel, Recognizing tough graphs is NP-hard, Discrete Appl. Math. 28 (1990) 191–195.

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

[5] A. Brandstädt, Special graph classes – a survey, Schriftenreihe des Fachbereichs Mathematik, SM-DU-199, Universität Duisburg Gesamthochschule (1991).

[6] V. Chvátal, Tough graphs and hamiltonian circuits, Discrete Math. 5 (1973) 215–228.

[7] L.H. Clark, R.C. Entringer and M. Fellows, Computational complexity of integrity, J. Combin. Math. Combin. Comput. 2 (1987) 179–191.

[8] I. Dagan, M.C. Golumbic and R.Y. Pinter, Trapezoid graphs and their coloring, Discrete Appl. Math. 21 (1988) 35–46.

[9] S. Felsner, R. Müller and L. Wernisch, Trapezoid graphs and generalizations, geometry and algorithms, Proceedings of the 4th Scandinavian Workshop on Algorithm Theory (1994).

[10] H.A. Jung, On a class of posets and the corresponding comparability graphs, J. Combin. Theory Ser. B 24 (1978) 125–133.
[11] T. Leighton and S. Rao, An approximate max-flow min-cut theorem for uniform multicommodity flow problems with applications to approximation algorithms, Proceedings of the 29th Symposium on Foundations of Computer Science (1988) 422–431.
[12] R.J. Lipton and R.E. Tarjan, Applications of a planar separator theorem, SIAM J. Comput. 9 (1980) 615–627.
[13] Tze-Heng Ma and J.P. Spinrad, An $O(n^2)$ Algorithm for the 2-chain cover problem and related problems, Proceedings of the 2nd Symposium on Discrete Algorithms (1991) 363–372.
[14] R.E. Pippert and M.J. Lipman, Towards a measure of vulnerability – I. The edge connectivity vector, in: Graph Theory with Applications to Algorithms and Computer Science (Wiley, New York, 1985) 507–517.
[15] M.D. Plummer, Toughness and matching extension in graphs, Discrete Math. 72 (1988) 311–320.
[16] Z. Skupień, Research problems: problems 106, 107 and 108, Discrete Math. 76 (1989) 157–159.
[17] D.R. Woodall, The binding number of a graph and its Anderson number, J. Combin. Theory Ser. B 15 (1973) 225–255.