Treewidth of Chordal Bipartite Graphs

T. Kloks, D. Kratsch

Treewidth of Chordal Bipartite Graphs

T. Kloks, D. Kratsch

Treewidth of Chordal Bipartite Graphs

T. Kloks *
Department of Computer Science
Utrecht University
P.O.Box 80.089
3508 TB Utrecht, The Netherlands

D. Kratsch †
Fakultät Mathematik
Friedrich-Schiller-Universität
Universitätshochhaus
O-6900 Jena, Germany

Abstract

Chordal bipartite graph are exactly those bipartite graph in which every cycle of length at least six has a chord. The treewidth of a graph G is the smallest maximum cliquesize among all chordal supergraphs of G decreased by one. We present a polynomial time algorithm for the exact computation of the treewidth of all chordal bipartite graphs.

keywords: Graph algorithms, Algorithms and Data Structures.

1   Introduction

In many recent investigations of computer science the notions of treewidth and pathwidth play an increasingly important role. One reason for this is that many NP-complete problems become solvable in polynomial and usually even linear time, when restricted to the class of partial k-trees for some *constant* k [Arn 91]. Of crucial importance for these algorithms is that an embedding in a k-tree is given in advance. The treewidth of a graph is the minimum k for which the graph is the subgraph of a k-tree. Much research has been done in finding embeddings in k-trees with a reasonable small k.

Determining whether the treewidth or pathwidth of a given graph is at most a given integer k is NP-complete [Arn 87]. In view of this, the results of Robertson and Seymour on minor closed classes of graph are of great interest. Robertson and Seymour proved that every minor closed class of graphs is recognizable in $O(n^3)$ time [Rob 85]. Since for every fixed k, the class of graphs with treewidth (pathwidth) at most k is minor closed, it follows that for every constant k there is a polynomial algorithm that recognizes graphs with treewidth (pathwidth) at most k. In fact, for these minor closed classes faster algorithms exist [Ree 92]. If the treewidth is not bounded by a constant, only very general approximations are known [Bod 91]. For an introductory overview of recent results dealing with treewidth and pathwidth, the reader is referred to [Bod 92/1].

Chordal bipartite graphs (or weakly chordal bipartite graphs) are bipartite graphs such that every induced cycle of length at least 6 has a chord. There is a strong connection between chordal bipartite graphs, strongly chordal graphs and totally balanced matrices. Indeed, a graph is chordal bipartite if and only if the adjacency matrix is totally balanced [Far 83]. If G is a bipartite graph, then let $split(G)$ be the graph obtained from G by making a clique of one of the color classes. Then G is chordal bipartite if and only if $split(G)$ is strongly chordal.

The chordal bipartite graphs form a large class of perfect graphs containing for example the convex and biconvex bipartite graphs, the bipartite permutation graphs and the bipartite distance hereditary graphs (or $(6,2)$-chordal bipartite graphs). Many NP-complete problems remain NP-complete when restricted to the class of chordal bipartite graphs. For example Hamiltonian Cycle (Hamiltonian Path), Dominating set (Connected Dominating Set and Independent Dominating Set) and Steiner Tree [Mül 87, Dam]. Recognizing chordal bipartite graphs can be done in $O(\min(m \log n, n^2))$ [Lub 87].

There are very few classes of graphs for which the treewidth can be computed in polynomial time. This can be done for example for chordal graphs (trivially), cographs, circular arc graphs and permutation graphs [Bod 90, Sun 91, Bod 92/2]. It is easy to see that treewidth is NP-complete for bipartite graphs in general. We present a polynomial time algorithm to compute the treewidth of chordal bipartite graphs. Since so many NP-complete problems remain NP-complete when restricted to chordal bipartite graphs, it is of great importance to be able to use the partial k-tree algorithms for these problems.

We do not claim that our algorithm is a very practical one, however we feel that it is one of the first non-trivial polynomial time algorithms for computing the treewidth of a relatively large class of graphs. Note that it narrows the gap between classes where treewidth is computable in polynomial time and classes where the corresponding decision problem is NP-complete significantly. Indeed, the gap between chordal bipartite graphs and the class of all bipartite graphs is relatively small.

2 Preliminaries

2.1 Preliminaries on chordal bipartite graphs

In this section we start with some definitions and easy lemmas. For more information the reader is referred to [Gol 80] or [Bran 91].

Definition 2.1 *A graph is called* chordal bipartite *(or weakly chordal bipartite) if it is bipartite and each cycle of length at least six has a chord.*

A chord (x, y) in a cycle C of even length is *odd* if the distance between x and y in the cycle is odd.

Definition 2.2 *A graph is called* strongly chordal *if it is chordal and each cycle of even length at least six has an odd chord.*

Definition 2.3 *For a bipartite graph* $G = (X, Y, E)$ *let* $split(G) = (X, Y, \hat{E})$ *with* $\hat{E} = E \cup \{(x, x') \mid x, x' \in X \land x \neq x'\}$.

The following characterization of chordal bipartite graphs appeared in [Dah 89].

Lemma 2.1 $G = (X, Y, E)$ *is chordal bipartite if and only if* $split(G)$ *is strongly chordal.*

If x is a vertex of a graph $G = (V, E)$, we denote by $N[x]$ the *closed neighborhood* of x, i.e. $N[x] = \{y \mid (x, y) \in E\} \cup \{x\}$.

Definition 2.4 *A vertex* v *is* simple *if for all* $x, y \in N[v]$, $N[x] \subseteq N[y]$ *or* $N[y] \subseteq N[x]$.

Notice that a simple vertex is simplicial (i.e. the neighborhood is complete). We shall use the following property of strongly chordal graphs [Far 83].

Lemma 2.2 *A graph* G *is strongly chordal if and only if every induced subgraph has a simple vertex.*

Definition 2.5 *Let* $G = (X, Y, E)$ *be a bipartite graph. Then* $(u, v) \in E$ *is called a* bisimplicial edge *if* $N(x) \cup N(y)$ *induces a complete bipartite subgraph of* G.

Definition 2.6 *Let* $G = (X, Y, E)$ *be a bipartite graph. Let* $(e_1, \ldots, e_k)$ *be an ordering of the edges of* G. *For* $i = 0, \ldots, k$ *define the subgraph* $G_i = (X_i, Y_i, E_i)$ *as follows.* $G_0 = G$ *and for* $i \geq 1$ G_i *is the subgraph of* G_{i-1} *with* $X_i = X_{i-1}$, $Y_i = Y_{i-1}$ *and* $E_i = E_{i-1} \setminus \{e_i\}$ *(i.e. the edge* e_i *is removed but not the endvertices). The ordering* $(e_1, \ldots, e_k)$ *is a* perfect edge without vertex elimination ordering *for* G *if each edge* e_i *is bisimplicial in* G_i, *and* G_k *has no edge.*

The following lemma appears for example in [Bran 91].

Lemma 2.3 *G is chordal bipartite if and only if there is a perfect edge without vertex elimination ordering.*

The following lemma shows that we can start a perfect edge without vertex elimination ordering with *any* bisimplicial edge.

Lemma 2.4 *Let G be chordal bipartite. Let e be a bisimplicial edge in G. Let G' be the graph obtained from G by deleting the edge e but not the endvertices of e. Then G' is chordal bipartite.*

Proof. Assume G' has a chordless cycle C of length ≥ 6. Let $e = (x, y)$. Then, clearly, x and y must be elements of C. The neighbors of x and y in the cycle form a square. This shows that C cannot be chordless in G'. □

In [Goh 82] it is shown that a bisimplicial edge in a chordal bipartite graph with n vertices can be found in $O(n^2)$ time.

Corollary 2.1 *We can find a perfect edge without vertex elimination scheme in time $O(n^2 m)$, where n is the number of vertices and m is the number of edges.*

2.2 Preliminaries on treewidth

In this paper we show how to compute the *treewidth* of chordal bipartite graphs. For more general information on treewidth the reader is referred to the survey paper [Bod 92/1].

Definition 2.7 *A k-tree is defined recursively as follows: A clique with $k + 1$ vertices is a k-tree. Given a k-tree T_n with n vertices, a k-tree with $n + 1$ vertices is constructed by making a new vertex x_{n+1} adjacent to a k-clique of T_n and nonadjacent to the $n - k$ other vertices of T_n. A partial k-tree is a subgraph of a k-tree. The treewidth of a graph G is the minimum k for which G is a partial k-tree.*

There are equivalent definitions of the notion treewidth [Bod 92/1].

Definition 2.8 *A chordal graph (or triangulated graph) is a graph with no chordless cycle of length ≥ 4.*

Notice that k-trees are chordal, and have maximum clique size $k + 1$.

Definition 2.9 *A triangulation of a graph G is a chordal graph H with the same vertex set as G such that G is a subgraph of H.*

We make extensive use of the following (well known) result.

Lemma 2.5 *A graph G is a partial k-tree if and only if there is a triangulation of G with maximum cliquesize at most $k + 1$.*

4

3 Triangulations

In this section, let $G = (X, Y, E)$ be chordal bipartite. We denote complete bipartite graphs as $M = (A, B)$, i.e. the vertex set of this graph M is $A \cup B$ and the edge set $E = \{(a, b) \mid a \in A \wedge b \in B\}$. In this paper, by definition, a complete bipartite graph (A, B) is such that $|A| \geq 2$ and $|B| \geq 2$. If $G = (X, Y, E)$ is a bipartite graph, then we call the sets X and Y the color classes of G.

Lemma 3.1 *If $G = (X, Y, E)$ is chordal bipartite, then it contains at most $|E|$ maximal complete bipartite subgraphs.*

Proof. G is chordal bipartite, hence there is a perfect edge without vertex elimination ordering $(e_1, \ldots, e_k)$. Consider a maximal complete bipartite subgraph, (A, B). Let e_i be the first edge in the ordering which is an edge of (A, B). Let $e_i = (x, y)$ with $x \in A$ and $y \in B$. Since e_i is bisimplicial and (A, B) is maximal we have $A = N(y)$ and $B = N(x)$. This proves the lemma. $\square$

Remark. It is not difficult to see that there exist chordal bipartite graphs for which the number of maximal complete bipartite subgraphs is $\Omega(n^2)$.

If (A, B) is a complete bipartite graph, and H is a triangulation, then either $H[A]$ or $H[B]$ is a complete subgraph of H (otherwise there would be a chordless square). Now let G be chordal bipartite, and let $\mathcal{M}$ be the set of maximal complete bipartite subgraphs (A, B) of G (with $|A| \geq 2$ and $|B| \geq 2$). If H is a triangulation of G, then for each $(A, B) \in \mathcal{M}$, either $H[A]$ or $H[B]$ is a complete subgraph of H. Consider the following process. For *each* $(A, B) \in \mathcal{M}$, choose one color class $C \in \{A, B\}$, and add all edges between vertices of C. We say the color class C is completed. The following example shows, that the resulting graph need not be chordal. Take the $K(4, 4)$ with color classes $A = \{a, b, c, d\}$ and $B = \{p, q, r, s\}$ and delete the edges (p, d) and (a, s). Call this graph G. It is easy to see that G is chordal bipartite (it is even bipartite permutation). For the maximal complete bipartite subgraph $(\{a, b, c, d\}, \{q, r\})$ we choose the color class $\{a, b, c, d\}$ and change this into a clique. For the maximal complete bipartite subgraph $(\{b, c\}, \{p, q, r, s\})$ we choose the color class $\{p, q, r, s\}$ and make this complete. The resulting graph is not triangulated, because there is a chordless square induced by $\{a, d, p, s\}$.

Definition 3.1 *Let $M_1 = (A_1, B_1)$ and $M_2 = (A_2, B_2)$ be two maximal complete bipartite subgraphs. We say that M_1 and M_2 cross if either $A_2 \subset A_1$ and $B_1 \subset B_2$, or $A_1 \subset A_2$ and $B_2 \subset B_1$.*

In the example above, the maximal complete bipartite subgraphs $(\{b, c\}, \{p, q, r, s\})$ and $(\{a, b, c, d\}, \{q, r\})$ cross.

Definition 3.2 *For each $M \in \mathcal{M}$ choose one color class $C(M)$. The set $\mathcal{C} = \{C(M) \mid M \in \mathcal{M}\}$ is called* feasible, *if for each pair $(A_1, B_1), (A_2, B_2) \in \mathcal{M}$ that cross with $A_2 \subseteq A_1$ and $B_1 \subseteq B_2$, not both A_1 and B_2 are in $\mathcal{C}$.*

We want to proof in this section that if $\mathcal{C}$ is feasible and we complete each $C \in \mathcal{C}$, then the resulting graph is triangulated.

Theorem 3.1 *Let $\mathcal{C}$ be a feasible set of color classes of a chordal bipartite graph G. Let H be the graph obtained by making each $C \in \mathcal{C}$ complete. Then H is chordal.*

Proof. Assume G is a *minimal* counterexample; i.e. for every induced subgraph the theorem is true. Assume H is not chordal and let S be a chordless cycle of length greater than three in H. Let $G = (X, Y, E)$. We call the vertices of X red and the vertices of Y black. An edge of S is called red (black), if *both* its endvertices are red (black). The red and black edges of S are called colored edges. Notice that there must exist at least one colored edge, otherwise the cycle would also be a chordless cycle in G, and since G is chordal bipartite, S must be a square. But a square is contained in some maximal complete bipartite subgraph. Hence at least one of the color classes must have been completed. Let $\mathcal{R}$ be the set of all red edges of S and $\mathcal{B}$ be the set of all black edges of S. Without loss of generality we may assume that $\mathcal{R} \neq \emptyset$.

Consider a red vertex $x \notin S$, and let $e \in \mathcal{B}$ be a black edge of S. We say that x *creates* e, if there is a maximal complete bipartite subgraph (A, B) with $e \subseteq B$, $x \in A$, and $B \in \mathcal{C}$. A red vertex r is called *redundant*, if $r \notin S$ and it does not create any black edge. A black vertex $b \notin S$ is called redundant, if it does not create a red edge.

Lemma 3.2 *There are no redundant vertices.*

Proof. Suppose there is a redundant red vertex x. Consider the induced subgraph $G' = G[V \setminus \{x\}]$. Clearly, also G' is chordal bipartite. We create a feasible set of color classes $\mathcal{C}'$ as follows. Consider a maximal complete bipartite subgraph $M = (A, B)$ of G'. If M is also maximal in G, we put $C(M)$ in $\mathcal{C}'$. If M is not maximal in G then $N = (A \cup \{x\}, B)$ must be a maximal complete bipartite subgraph of G. In this case we put $C(N) \setminus \{x\}$ in $\mathcal{C}'$. It is easy to see that $\mathcal{C}'$ is a feasible set of color classes for G'.

Let H' be the graph obtained from G' by completing each set of $\mathcal{C}'$. We claim that S is also a chordless cycle in H'.

Let e be a colored red edge of S. There is a maximal complete bipartite subgraph (A, B), with $e \subseteq A$ and $A \in \mathcal{C}$. If $x \notin A$, then (A, B) is a maximal complete bipartite subgraph of G'. Hence $A \in \mathcal{C}'$. Now assume $x \in A$. Then $|A| \geq 3$. Take B' such that $B \subseteq B'$ and $(A \setminus \{x\}, B')$ is a maximal complete bipartite subgraph of G' (i.e. B' is the set of common neighbors of $A \setminus \{x\}$). If (A, B') is maximal in G, then $B' = B$ and $A \setminus \{x\} \in \mathcal{C}'$. Otherwise $(A \setminus \{x\}, B')$ is maximal in G. Notice that (A, B) and $(A \setminus \{x\}, B')$ cross. Hence $A \setminus \{x\}$ is in $\mathcal{C}$ and hence also in $\mathcal{C}'$.

Now let e be a black edge, and let (A, B) be a maximal complete bipartite subgraph with $e \subseteq B$ and $B \in \mathcal{C}$. Notice that since x is redundant, $x \notin A$. Hence (A, B) is also maximal in G' and $B \in \mathcal{C}'$. This shows that S is also a cycle in G'.

Since G was a minimal counterexample, there must be a chord in G' between two vertices p and q of S. Without loss of generality assume p and q are both red (the case where they are both black is similar). There exists a maximal complete bipartite subgraph (A, B) in G' such that $p, q \in A$ and $A \in \mathcal{C}'$. By definition either $A \cup \{x\} \in \mathcal{C}$ or $A \in \mathcal{C}$. Hence (p, q) is also an edge in G. This proves that there are no redundant red vertices. A similar argument show that there are no redundant black vertices either. $\qquad\square$

Consider the graph $G^* = split(G)$ obtained by making Y complete. Since G is chordal bipartite, we know that G^* is strongly chordal. Hence we know that G^* has a simple vertex. Let s be the simple vertex. The next lemma shows that s must be a red vertex.

Lemma 3.3 *A simple vertex s of G^* is red.*

Proof. Assume $s \in Y$ is simple. If s is not an element of the cycle, it must be in some maximal complete bipartite subgraph (A, B) creating some edge, otherwise s would be redundant. But then it has at least two nonadjacent neighbors, which is a contradiction. Assume s is a black vertex of the cycle S. Clearly s can not have two red neighbors, because red vertices are not adjacent in G. Hence s is incident with at least one edge of $\mathcal{B}$. Consider a maximal path of S, which contains only black vertices, and which contains s. Let p and q, be the endvertices of this path. We know that $p \neq q$. Since s is simple, and p and q are neighbors of s in G^*, either $N[p] \subseteq N[q]$ or $N[q] \subseteq N[p]$. Since p and q are both incident with a red vertex in S, it follows that S contains only one red vertex. This is a contradiction, because we assumed that $\mathcal{R} \neq \emptyset$. $\qquad\square$

Lemma 3.4 *A simple vertex s is an element of S.*

Proof. We know s is red. Assume s is not in S. Then we know that s creates some black edge $e = (p, q)$. Without loss of generality assume $N[p] \subseteq N[q]$. If p has a red neighbor in S, then this would also be a neighbor of q, and there would be a chord in S. Hence p has another black neighbor r in S. Consider the maximal complete bipartite subgraph (K, L) with $p, r \in L$ and $L \in \mathcal{C}$. Then p is adjacent to every vertex of K. Hence also q is adjacent to every vertex of K. It follows that $q \in L$. But $L \in \mathcal{C}$. Hence q and r are adjacent in H. This is a chord in S. $\qquad\square$

We have to consider three more cases. First we show that s is incident with at least one edge of $\mathcal{R}$.

Lemma 3.5 *A simple vertex s is incident with at least one red edge of S.*

Proof. We know s is a red vertex of S. Assume it has two black neighbors in S, x and y. We may assume $N[x] \subseteq N[y]$. If in the cycle x has two red neighbors,

then also y is adjacent to both red neighbors, which is a contradiction. Hence x is adjacent to a black vertex z in S. Consider a maximal complete bipartite subgraph (K, L) creating the edge (x, z). Then x is adjacent to every vertex of K, and hence this also holds for y. But then $y \in L$. $L \in C$ hence x and y are adjacent in H. This is a chord in S. $\square$

Lemma 3.6 *A simple vertex s is incident with exactly one red edge of S.*

Proof. We know that $s \in S$ and s has at least one red neighbor in S. Asume s has *two* red neighbors x and y in S. Consider the maximal complete bipartite subgraphs (K_x, L_x) creating the edge (s, x) and (K_y, L_y) creating (s, y). Now there must exist $p \in L_x \backslash L_y$ which is not adjacent to y, otherwise $(K_x \cup \{y\}, L_x)$ is complete bipartite. Also there must exist $q \in L_y \backslash L_x$ which is not adjacent to x. Without loss of generality we may assume $N[p] \subseteq N[q]$. But then we have a contradiction since $x \in N[p] \subseteq N[q]$. $\square$

Hence we know that a simple vertex s is a red vertex of the cycle S with one red neighbor x and one black neighbor y. Let (K_x, L_x) be the maximal complete bipartite subgraph that creates (s, x). Since s is simple and y is not adjacent to x, we must have $N[y] \cap X \subseteq K_x$. Now assume that in the cycle y is adjacent to another red vertex u. Then $u \in N[y] \cap X \subseteq K_x$. But then u, x and s would form a triangle, which is a contradiction. Hence y is adjacent to a black vertex v in the cycle. Let (K_v, L_v) be the maximal complete bipartite subgraph creating (y, v). Notice that $K_v \subseteq N[y] \cap X \subseteq K_x$. It follows that $L_x \subseteq L_v$, and hence (K_x, L_x) and (K_v, L_v) cross. But L_v and K_x are completed, hence C is not feasible. This completes the proof of theorem 3.1. $\square$

In the next section we show that there exists a feasible set of color classes of G such that the triangulated graph H is optimal, i.e. has the smallest possible maximum cliquesize.

4 Treewidth

Let $G = (X, Y, E)$ be a chordal bipartite graph. If C is a feasible set of color classes, we denote by H_C the chordal graph obtained from G, by completing each $C \in C$. In this section we show the following. There is a feasible set of color classes C, such that the triangulation H_C minimizes the cliquesize.

Theorem 4.1 *Let G be a chordal bipartite graph with treewidth $\leq k$. Then there exists a feasible set of color classes C such that H_C has cliquesize $\leq k + 1$.*

Proof. Let H be any triangulation of G. We show that there is a feasible set of color classes C such that H_C is a subgraph of H. Let $\mathcal{M}$ be the set of all maximal complete

bipartite subgraphs of G. Suppose we list the elements of $\mathcal{M}$ one by one, creating a feasible set of color classes $\mathcal{C}$ as follows. Start with $\mathcal{C} = \emptyset$. For each $M \in \mathcal{M}$ at least one of the color classes is a complete subgraph in H. Let $M = (A, B)$ and assume A is complete in H. Check the list $\mathcal{C}$ created thus far, if there is a maximal complete subgraph (C, D) which crosses with (A, B), such that $C \subseteq A$ and $D \in \mathcal{C}$. If not, then put A in $\mathcal{C}$. Otherwise put B in $\mathcal{C}$. Notice that there cannot be a maximal complete bipartite subgraph (K, L) crossing with (A, B), such that $L \subseteq B$ and $K \in \mathcal{C}$. Otherwise (K, L) and (C, D) would also cross with $C \subseteq K$ and $K, D \in \mathcal{C}$. This shows that in this way we create a feasible set of color classes. Now notice that $H_{\mathcal{C}}$ is a subgraph of H, hence the cliquesize of $H_{\mathcal{C}}$ does not exceed the cliquesize of H. $\qquad\square$

5 Cliques

Let $G = (X, Y, E)$ be a chordal bipartite graph and let $\mathcal{C}$ be a feasible set of color classes. In this section we analyse the structure of the maximal cliques in $H_{\mathcal{C}}$. We first look at a clique B with vertices only in one color class.

Definition 5.1 *Let B be a clique with all vertices in Y. We say that a set $\mathcal{A}$ of maximal complete bipartite subgraphs is a cover for B, if the following two conditions are satisfied:*

1. *For each $(A', B') \in \mathcal{A} : B' \in \mathcal{C}$.*

2. *for every pair $x, y \in B$ there is a maximal complete bipartite subgraph $(A', B') \in \mathcal{A}$ such that $x, y \in B'$.*

Theorem 5.1 *Let B be a clique of $H_{\mathcal{C}}$ with all vertices in Y. Assume $|B| \geq 2$. A minimal cover for B has only one element.*

Proof. Let $\mathcal{A} = \{(A_1, B_1), \ldots (A_t, B_t)\}$ be a minimal cover for B. Assume $t \geq 2$. Let $A = \bigcup_i A_i$.

Lemma 5.1 *Every vertex of A has at least two neighbors in B.*

Proof. Assume there is a vertex $a \in A$ with only one neighbor b in B. There is a complete bipartite subgraph (A_i, B_i) containing a. We claim that B is covered by $\mathcal{A} \setminus \{(A_i, B_i)\}$. For every other vertex $b' \in B$ the edge (b, b') must be in some (A_j, B_j) with $j \neq i$. This proves the lemma. $\qquad\square$

Lemma 5.2 *Every vertex of B has at least two neighbors in A.*

9

Proof. Let $b \in B$. Take another vertex $b' \in B$. The edge (b, b') must be in some maximal complete bipartite subgraph (A_i, B_i). □

Take the subgraph H of G induced by $A \cup B$. Let $W = split(H)$ obtained by making A a complete graph. W is strongly chordal. Each vertex of A has two nonadjacent neighbors, hence A can not contain a simple vertex. Let $b \in B$ be a simple vertex. Let $a_1, \ldots, a_k$ be the neighbors of b in A, and let $N[a_1] \subseteq N[a_2] \subseteq \ldots \subseteq N[a_k]$. Notice that $k \geq 2$. Since A is a clique in W, we have that $N[a_1] \cap B \subseteq \ldots \subseteq N[a_k] \cap B$. We claim that not all a_i's can be contained in one color class A_i.

Lemma 5.3 $\forall_i \exists_j [a_j \notin A_i]$

Proof. Assume that $\{a_1, \ldots, a_k\} \subseteq A_i$ for some color class A_i. Take $b' \notin B_i$. There exists a maximal complete bipartite subgraph (A_j, B_j) such that $\{b, b'\} \subseteq B_j$. Clearly $A_j \subseteq \{a_1, \ldots, a_k\} \subseteq A_i$. This implies $B_i \subseteq B_j$. This is a contradiction since $\mathcal{A}$ is minimal. □

Now we can finish the proof of theorem 5.1. Let (A_1, B_1) be a maximal complete bipartite subgraph containing a_1. Let $i > 1$. Notice $B_1 \subseteq N[a_1] \cap B \subseteq N[a_i] \cap B$. Hence $(A_1 \cup \{a_i\}, B_1)$ is a complete bipartite subgraph. This implies that $a_i \in A_1$ for all $1 \leq i \leq k$, which is a contradiction. This proves theorem 5.1. □

The following theorem shows that the structure of the maximal cliques in H_C is very simple.

Theorem 5.2 *Let K be a maximal clique in H_C with $|K| > 2$. Let $K_x = K \cap X$ and $K_y = K \cap Y$. Assume $|K_x| \geq 2$. Then one of the following two cases holds:*

1. *$|K_y| = 1$ and there exists a maximal complete bipartite subgraph (A, B) such that $K_x = A$, $y \in B$ and $A \in \mathcal{C}$.*

2. *$|K_y| > 1$ and there exist maximal complete bipartite subgraphs (A_1, B_1) and (A_2, B_2), with $A_1 \in \mathcal{C}$ and $B_2 \in \mathcal{C}$ such that $K_x \subseteq A_1$ and $K_y \subseteq B_2$.*

Proof. By theorem 5.1 there exists a maximal complete bipartite subgraph (A_1, B_1) such that $K_x \subseteq A_1$ and $A_1 \in \mathcal{C}$. Assume $K_y = \emptyset$. Then clearly, K cannot be maximal, since for any vertex $y \in B_1$, $A_1 \cup \{y\}$ is a clique. Assume $|K_y| = 1$. Let $K_y = \{y\}$. Notice that $(K_x, \{y\})$ is contained in a complete bipartite subgraph $(K_x, B_1 \cup \{y\})$ in G. Hence $(K_x, B_1 \cup \{y\})$ is contained in a maximal complete bipartite subgraph (A_2, B_2) in G. Since $B_1 \subseteq B_2$, we have $A_2 \subseteq A_1$. Since $\mathcal{C}$ is feasible, and $A_1 \in \mathcal{C}$, we must have $A_2 \in \mathcal{C}$. Hence the first case holds. Now assume $|K_y| \geq 2$. By theorem 5.1 there exists a maximal complete bipartite subgraph (A_2, B_2) with $K_y \subseteq B_2$ and $B_2 \in \mathcal{C}$. □

10

6 The algorithm

Let $G = (X, Y, E)$ be a chordal bipartite graph. In this section we show a polynomial time algorithm which determines whether the treewidth of G is at most some given integer k, and if so triangulates G such that the maximum cliquesize does not exceed $k+1$. We first describe the algorithm. In the method we use, we construct a digraph with vertex set $\mathcal{M}$. We direct an edge from $M_1 = (A_1, B_1)$ to $M_2 = (A_2, B_2)$, with A_1, $A_2 \in X$, if $A_1 \in \mathcal{C}$ then necessarily also $A_2 \in \mathcal{C}$. Next we color some of the vertices of this graph red or black. A vertex (A, B) is colored red if necessarily $A \in \mathcal{C}$ and black if necessarily $B \in \mathcal{C}$. In the next step we try to extend this coloring to the whole graph.

step 1 If $k \leq 3$ then use the algorithm described in [Mat 91] to decide whether the treewidth of G is at most k and if so find a suitable triangulation. If $k > 3$ perform the following steps.

step 2 First make a list $\mathcal{M}$ of all maximal complete bipartite subgraphs of G.

step 3 Make a directed graph W with vertex set $\mathcal{M}$ as follows. Let $M_1 = (A_1, B_1)$ and $M_2 = (A_2, B_2)$ be two different elements of $\mathcal{M}$. We direct an edge from M_1 to M_2 if one of the following cases holds:

- Compute the maximal number of vertices of a maximal complete bipartite subgraph in the induced subgraph $G[A_1 \cup B_2]$. (M_1, M_2) is a directed edge if this size exceeds $k + 1$.

- If M_1 and M_2 cross, with $A_2 \subseteq A_1$ then (M_1, M_2) is a directed edge.

step 4 Color some of the vertices of W as follows. If $M = (A, B)$ is a maximal complete bipartite subgraph with $|A| > k$ and $|B| > k$ then output that the treewidth of G exceeds k. Otherwise, if $|B| > k$, then we color M red and if $|A| > k$, then we color M black. In case $|A| \leq k$ and $|B| \leq k$, then we do not color M yet.

step 5 While there is some arc (M_1, M_2) with M_1 colored red and M_2 not colored red then consider the following cases:

- If M_2 is black then output that the treewidth of G exceeds k

- If M_2 is not colored, then color M_2 red.

step 6 All vertices which do not yet have a color, are colored black.

step 7 For all elements $M = (A, B)$ of $\mathcal{M}$: If the color of M is red, then complete A, and if the color of M is black then complete B.

Theorem 6.1 *Let G be chordal bipartite and let k be an integer. The treewidth of $G \leq k$ if and only if the algorithm produces a triangulation with maximum cliquesize at most $k + 1$.*

Proof. Clearly, the theorem is true when $k \leq 3$. Assume $k > 3$. Clearly, if the algorithm does not produces a triangulation, then some necessary condition is not satisfied and hence the treewidth is more than k. Assume each $M \in \mathcal{M}$ is colored. Consider the set $\mathcal{C}$ of color classes, defined as follows. If $M = (A, B)$ is red, then $A \in \mathcal{C}$, otherwise $B \in \mathcal{C}$. It is easily checked that $\mathcal{C}$ is a feasible set of color classes. It follows from theorem 3.1 that the algorithm produces a triangulation H. Assume H has a maximal clique K with more than $k + 1$ vertices. By theorem 5.2 there are two cases to consider. First consider the case $|K_y| = 1$, let $K_y = \{y\}$. Then $|K_x| > k$. There is a maximal complete bipartite subgraph $M = (A, B)$, with $K_x \subseteq A \in \mathcal{C}$, and $y \in B$. However, the algorithm can not color M red (step 4) hence A cannot be in $\mathcal{C}$. A similar argument shows that $|K_x| = 1$ is also not possible. Now assume $|K_x| \geq 2$ and $|K_y| \geq 2$. By theorem 5.2, there exist complete maximal bipartite subgraphs, $M_1 = (A_1, B_1)$ and $M_2 = (A_2, B_2)$, such that $K_x \subseteq A_1$ and $K_y \subseteq B_2$, with $A_1 \in \mathcal{C}$ and $B_2 \in \mathcal{C}$. This means that M_1 is colored red and M_2 is colored black. But (K_x, K_y) is a complete bipartite subgraph in the induced subgraph $G[A_1 \cup B_2]$, hence there is an arc (M_1, M_2) (step 3). This is a contradiction. $\square$

In the last part of this section we discuss the running time of the algorithm and we show it is polynomial. Consider the time it takes to find all maximal complete bipartite subgraphs. Corollary 2.1 shows we can find a perfect edge without vertex elimination ordering in time $O(n^4)$. Each bisimplicial edge gives a complete bipartite subgraph. It follows that we can find a list with at most e complete bipartite subgraph in time $O(e^2)$. From this list we can get the maximal complete bipartite subgraphs in time $O(e^2 n)$. It follows that step 2 can be performed in $O(e^2 n)$. Now clearly, the time complexity is dominated by step 3 of the algorithm. For each pair $M_1 = (A_1, B_1)$ and $M_2 = (A_2, B_2)$ we have to find the maximum size of a complete bipartite subgraph in $G[A_1 \cup B_2]$. We can do this as follows. First we compute for each pair (A_1, B_1) and $(A_2, B_2$ the number of vertices in $A_1 \cap A_2$ and in $B_1 \cap B_2$. Then for each $M \in \mathcal{M}$ we can compute the number of vertices in $A_1 \cup B_2$ in time $O(n \log n)$. It follows that for each pair M_1, M_2 we can decide if there is an arc (M_1, M_2) in time $O(e)$ (checking whether they cross can be done in $O(n \log n)$). It follows that step 3 of the algorithm can be performed in $O(e^3)$. This is also the total time complexity of the algorithm which can be easily checked. This proves the following theorem.

Theorem 6.2 *Let G be chordal bipartite and let k be an integer. Then there is a polynomial time algorithm to decide whether the treewidth of G is at most k. If so, the algorithm returns a chordal embedding of G with cliquesize at most $k + 1$.*

12

7 Conclusions

In this paper we showed that computing the treewidth of a chordal bipartite graph can be done in polynomial time. There are other classes of graphs for which the treewidth can be computed in polynomial time. For example chordal graphs, cographs ([Bod 90]) and circular arc graphs ([Sun 91]). It would be of interest to know if there are other large classes of graphs for which the treewidth can be computed in polynomial time, for example permutation graphs. Chordal bipartite graphs are properly contained in perfect elimination bipartite graphs. It would be interesting to know if treewidth is solvable in polynomial time for this larger class of graphs. There might be faster algorithms for subclasses of chordal bipartite graphs, like convex bipartite graphs. Another interesting question is of course whether the time bound of this algorithm can be improved. Computing the pathwidth is usually a more complicated matter. For example, for triangulated graphs, computing the pathwidth is already NP-complete. Is the pathwidth problem NP-complete for chordal bipartite graphs?

Notice that there exist fast algorithms that find good approximations of the treewidth for many classes of perfect graphs like permutation graphs, cotriangulated graphs, convex bipartite graphs and cocomparability graphs ([Klo 92]). Does there exist a fast algorithm to approximate treewidth within a constant factor for chordal bipartite graphs? Finally, for many subclasses $\mathcal{G}$ of perfect graphs holds: There is a polynomial p, such that for every graph $G \in \mathcal{G}$ with treewidth $\leq k$ the pathwidth is bounded by $p(k)$. Can such a polynomial be given for chordal bipartite graphs?

8 Acknowledgements

References

[Arn 87] S. Arnborg, D.G. Corneil and A. Proskurowski, Complexity of finding embeddings in a k-tree, *SIAM J. Alg. Disc. Meth.* **8**, 277 − 284, 1987.

[Arn 91] S. Arnborg, J. Lagergren and D. Seese, Easy problems for tree-decomposable graphs, *J. Algorithms* **12**, 308 − 340, 1991.

[Bod 90] H. Bodlaender and R.H. Möhring, The pathwidth and treewidth of cographs, In *Proceedings 2nd Scandinavian Workshop on Algorithm Theory*, 301 − 309, Springer Verlag, Lecture Notes in Computer Science vol. 447, 1990.

[Bod 91] H. Bodlaender, J. Gilbert, H. Hafsteinsson and T. Kloks, Approximating treewidth, pathwidth and minimum elimination tree height, In G. Schmidt and R. Berghammer, editors, *proceedings 17th International Workshop on Graph-Theoretic Concepts in Computer Science WG '91*, 1 − 12, Springer Verlag, Lecture Notes in Computer Science, vol. 570 1991.

[Bod 92/1] H.L. Bodlaender, A tourist guide through treewidth, Technical report RUU-CS-92-12, Department of computer science, Utrecht University, Utrecht, The Netherlands, 1992. To appear in: *Proceedings 7th International Meeting of Young Computer Scientists*, Springer Verlag, Lecture Notes in Computer Science.

[Bod 92/2] H. Bodlaender, T. Kloks and D. Kratsch, Treewidth and pathwidth of permutation graphs. Manuscript 1992.

[Bran 91] A. Brandstädt, Special Graph Classes—A Survey, Schriftenreihe des Fachbereichs Mathematik, SM-DU-199, 1991, Universität Duisburg Gesamthochschule.

[Dah 89] E. Dahlhaus, Chordale Graphen im besonderen Hinblick auf parallele Algorithmen, Habilitationsschrift, Bonn, 1989.

[Dam] P. Damaschke and H. Müller, Hamiltonian circuits in convex and chordal bipartite graphs, submitted to Discrete Mathematics.

[Far 83] M. Farber, Characterizations of strongly chordal graphs, *Discrete Mathematics* **43** 173 − 189, 1983.

[Goh 82] L. Goh and D. Rotem, Recognition of perfect elimination bipartite graphs, *Information Processing Letters*, **15**, No. 4, 179 − 182, 1982.

[Gol 80] M.C. Golumbic, *Algorithmic Graph Theory and Perfect Graphs*, Academic Press, New York, 1980.

[Klo 92] T. Kloks, Approximating treewidth and pathwidth of some classes of perfect graphs. To appear in *3th Annual International Symposium on Algorithms and Computation (ISAAC '92)*.

[Lub 87] A. Lubiw, Doubly lexical orderings of matrices, *SIAM J. Comput.* **16**, No. 5, 854 − 879, 1987.

[Mat 91] J. Matoušek and R. Thomas, Algorithms Finding Tree-Decompositions of Graphs, *Journal of Algorithms* **12**, 1 − 22, 1991.

[Mül 87] H. Müller and A. Brandstädt, The NP-completeness of Steiner tree and dominating set for chordal bipartite graphs, *Theoretical Computer Science*, **53**, 257 − 265, 1987.

[Ree 92] B. Reed, Finding approximate separators and computing treewidth quickly, To appear in: Proceedings STOC '92 1992.

[Rob 85] N. Robertson and P.D. Seymour, Graph minors—A survey. In I. Anderson, editor, *Surveys in Combinatorics*, 153 − 171. Cambridge Univ. Press 1985.

[Sun 91] R. Sundaram, K. Sher Singh and C. Pandu Rangan, Treewidth of circular arc graphs, Manuscript 1991.