

Algorithms for the Treewidth and Minimum Fill-in of HHD-Free Graphs

H. J. Broersma¹, E. Dahlhaus² and T. Kloks^{1*}

¹ University of Twente
Faculty of Applied Mathematics
P.O.Box 217
7500 AE Enschede, the Netherlands
² Department of Computer Science
University of Bonn
Bonn, Germany

Abstract. A graph is HHD-free if it does not contain a house (i.e., the complement of P_5), a hole (a cycle of length at least 5) or a domino (the graph obtained from two C_4 by identifying an edge in one C_4 with an edge in the other C_4) as an induced subgraph. The MINIMUM FILL-IN problem is the problem of finding a chordal supergraph with the smallest possible number of edges. The TREewidth problem is the problem of finding a chordal embedding of the graph with the smallest possible clique number. In this note we show that both problems are solvable in polynomial time for HHD-free graphs.

Keywords: graphs, algorithms, HHD-free graphs, treewidth, minimum fill-in.
MSC: 68R10.

1 Introduction

A graph is HHD-free if it does not contain a house (i.e., the complement of P_5), a hole (C_k for $k \geq 5$) or a domino (see Figure 1).

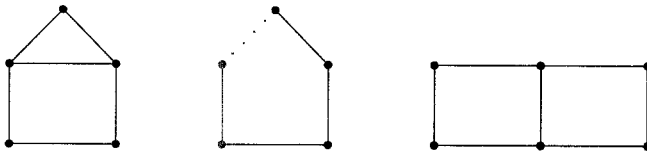


Fig. 1. 'House' (left), 'hole' (middle) and 'domino' (right)

* kloks@math.utwente.nl

Elimination and structural properties for HHD-free graphs were obtained in [8, 10]. For more information, the reader is referred to [17, 4].

A graph is chordal if it does not contain a chordless cycle of length at least four as an induced subgraph. A triangulation of a graph is a chordal supergraph with the same vertex set. Two triangulation problems have drawn much attention because of the large number of applications. The first is to find a triangulation of the graph such that the number of edges is minimum. This is called the MINIMUM FILL-IN problem. This problem is strongly related to Gaussian elimination of matrices. The second is called the TREEWIDTH problem. The objective in this case is to find a triangulation of a graph such that the clique number is as small as possible (the treewidth of the graph is the minimum clique number over all triangulations minus one). Both problems are NP-complete in general [21, 1], but polynomial time algorithms exist for many special graph classes such as cographs, circle and circular arc graphs, permutation graphs and, more generally, cocomparability graphs with bounded dimension, chordal bipartite graphs etc. [3, 12, 16, 15, 2, 20, 13, 5, 11, 18].

In this paper we show that the TREEWIDTH and the MINIMUM FILL-IN problem are solvable for HHD-free graphs.

Notice that adding an edge between two non adjacent vertices of a C_4 in an HHD-free graph, may introduce a new chordless cycle, and hence the resulting graph may no longer be HHD-free. This is illustrated for example by a graph consisting of a path and two non adjacent vertices that are adjacent to all vertices of the path. Joining the end vertices of the path by an edge would destroy the outer cycle (that was a C_4). But we get a cycle that consists of the path and the new edge. When we make the path long enough we get a cycle of length at least five. Hence, it is not clear whether a ‘minimum C_4 destroying set of chords’ leads to a chordal graph (note that all chordal graphs are HHD-free). If true, this could lead to a possible solution for the minimum fill-in problem by finding a minimum vertex cover in an auxiliary graph defined on the chords of the C_4 ’s (if the VERTEX COVER problem can be solved for this auxiliary graph) (see [19, 5]).

Instead of taking this approach we only make some fairly easy observations for the minimal separators of an HHD-free graph, which enable us to use a ‘standard’ dynamic programming technique to solve the problem.

2 Preliminaries

We denote the number of vertices of a graph $G = (V, E)$ by n and the number of edges by m . For a vertex $x \in V$, $N(x)$ is the neighborhood of x and $N[x] = \{x\} \cup N(x)$ is the closed neighborhood of x .

If Ω is a set and $x \in \Omega$, then we write $\Omega - x$ instead of $\Omega \setminus \{x\}$. For a subset Q of vertices, we write $G[Q]$ for the graph induced by the vertices of Q . For a vertex x , we write $G - x$ instead of $G[V - x]$, and for a subset W of vertices we write $G - W$ for the graph $G[V - W]$.

A *hole* is an induced cycle of length at least five. The house, hole and domino are depicted in Figure 1.

Definition 1. A graph is *HHD-free* if it does not contain a house, hole or domino as an induced subgraph.

Definition 2. A graph is *chordal* if it does not contain an induced cycle of length more than three.

Definition 3. Let a and b be non adjacent vertices. A set S of vertices is a minimal a, b -separator if a and b are in different connected components of $G - S$ and there is no proper subset of S with the same property. A minimal separator is a set S of vertices for which there exist non adjacent vertices a and b such that S is a minimal a, b -separator.

For the following lemma, we refer to [7].

Lemma 4. *A set S of vertices is a minimal separator if and only if there exist two connected components C_1 and C_2 in $G - S$ such that every vertex of S has at least one neighbor in C_1 and at least one neighbor in C_2 .*

Definition 5. Let S be a minimal separator and C a connected component of $G - S$ such that every vertex of S has a neighbor in C . Then S is *close to C* .

There exist many characterizations of chordal graphs. We use the characterization given by Dirac [6] using minimal separators.

Lemma 6. *A graph G is triangulated if and only if every minimal vertex separator induces a complete subgraph of G .*

Definition 7. A *triangulation* of a graph G is a graph H with the same vertex set as G such that G is a subgraph of H and H is chordal. A triangulation H of G is *minimal* if no proper subgraph of H is also a triangulation of G .

Definition 8. The *minimum fill-in* of a graph G , denoted by $\text{mfi}(G)$, is the minimum number of edges which are not edges of G , of a triangulation of G . We write $\text{mfi}^*(G) = m + \text{mfi}(G)$ for the number of edges in a triangulation realizing the minimum fill-in. The *treewidth* of a graph G , $\text{tw}(G)$, is the minimum clique number of a triangulation of G minus one.

Remark. Notice that for the treewidth and minimum fill-in problem we only have to consider triangulations that are minimal.

For a proof of the following, see, e.g., [14].

Lemma 9. *Let H be a minimal triangulation of a graph G and let S be a minimal a, b -separator of H for non adjacent vertices a and b in H . Then S is also a minimal a, b -separator in G , and if C is the vertex set of a connected component of $H - S$ then C induces also a connected component in $G - S$.*

For a proof of the following corollary of Lemma 6 and Lemma 9, we refer to [15].

Corollary 10. *If G is a clique then the treewidth equals the number of vertices minus one. The minimum fill-in of a clique is zero. Assume G is not a clique. Then*

$$tw(G) = \min_S \max_C tw(H(S, C))$$

where the minimum is taken over all minimal separators S in G and the maximum is taken over all connected components C of $G - S$. For the minimum fill-in, we have:

$$mfi^*(G) = \min_S \binom{|S|}{2} + \sum_C \left(mfi^*(H(S, C)) - \binom{|S|}{2} \right)$$

2.1 Minimal separators in HHD-free graphs

Lemma 11. *Let S be a minimal separator in an HHD-free graph G . Let C be a connected component of $G - S$ such that S is close to C . For every pair of vertices x and y in S , there exists a vertex p in C adjacent to x and y .*

Proof. Assume x and y do not have a common neighbor in C . By lemma 4, there exists at least one other component C' of $G - S$ such that x and y have a neighbor in C' . Then either a house, a hole, or a domino must exist. \square

Theorem 12. *Let S be a minimal separator in a HHD-free graph and let C be a connected component of $G - S$ such that S is close to C . Then there is a vertex p in C adjacent to all vertices of S .*

Proof. Consider two adjacent vertices p and q in C and assume they have private neighbors in S , p' and q' respectively (i.e., p' is not adjacent to q and q' is not adjacent to p). By Lemma 11 p' and q' have a common neighbor in some other connected component C' . This gives a house or an induced 5-cycle. Hence, since C is connected, there is a linear ordering by inclusion of $N(x) \cap S$ for the vertices x in C . Since every vertex of S has a neighbor in C a maximal element in this ordering must be adjacent to all vertices of S . \square

Corollary 13. *Let G be an HHD-free graph and let S be a minimal separator in G . Then there exist non adjacent vertices p and q such that $S = N(p) \cap N(q)$. Hence, if G is HHD-free, then there are at most $O(n^2)$ different minimal separators in G .*

Proof. Since S is a minimal separator there exist connected components C and C' of $G - S$ such that S is close to both. By Lemma 12 there are vertices p and q in C and C' respectively such that $S \subseteq N(p) \cap N(q)$. The neighbors of p are contained in $S \cup C$ and the neighbors of q are contained in $S \cup C'$. This proves the corollary. \square

3 From lumps to smaller lumps

Definition 14. Let S be a minimal separator and C a connected component of $G - S$. The pair (S, C) is called a *lump*. We write $H = H(S, C)$ for the graph obtained from $G[C \cup S]$ by adding edges such that S becomes a clique. The graph H is called a *realizer* of the lump.

Our algorithms for treewidth and minimum fill-in use dynamic programming on lumps. In this section we describe in detail how the minimum fill-in and treewidth of the realizer of a lump are expressed in the treewidth and minimum fill-in of smaller realizers.

In the first stage of the algorithm a list is made of all lumps (S, C) and this list is sorted according to $|S| + |C|$, the number of vertices of $G[S \cup C]$. For each lump (S, C) , the treewidth and minimum fill-in of the realizer $H = H(S, C)$ is computed (in a way described hereafter). When this is completed the treewidth and minimum fill-in of G can be obtained using Corollary 10.

We describe in detail how the treewidth and minimum fill-in of a realizer is expressed in the treewidth and minimum fill-in of smaller realizers in the rest of this section.

Throughout this section, let S be a minimal separator of G , let C be a connected component of $G \setminus S$, and let $H = H(S, C)$ be the realizer of the lump (S, C) .

Lemma 15. *Let $S^* \subseteq S$ be the set of vertices of S with a neighbor in C . Then S^* is a minimal separator in G close to C .*

For the treewidth of H , we have $tw(H) = \max(|S| - 1, tw(H^))$ where $H^* = H(S^*, C)$, and for the minimum fill-in: $mft^*(H) = \binom{|S|}{2} - \binom{|S^*|}{2} + mft^*(H^*)$.*

Proof. If $S^* = S$ there is nothing to prove.

We show that S^* is a minimal separator in G . C is a connected component of $G - S^*$ and every vertex of S^* has a neighbor in C . There exists at least one connected component C' different from C in $G - S$ such that every vertex of S has a neighbor in C' . Since $S^* \subseteq S$, C' is contained in connected component different from C in $G - S^*$. Using Lemma 4 this proves the lemma. \square

Let Ω be the set of vertices in C which are adjacent to all vertices of S . Hence if S is close to C , by Theorem 12, $\Omega \neq \emptyset$.

We first consider minimal triangulations Q of H in which Ω is *not* a clique.

Lemma 16. *Assume $\Omega \neq \emptyset$. Let Q be a minimal triangulation of H such that Ω is not a clique in Q . Then there is a minimal separator S' of Q which is also a minimal separator of G with $S \subset S' \subseteq S \cup C$.*

Let C_1, \dots, C_t be the connected components of $H - S'$. If Q realizes the treewidth of H , then $tw(H) = \max_i tw(H_i)$ where $H_i = H(S', C_i)$. If Q realizes the minimum fill-in we have $mft^(H) = \binom{|S'|}{2} + \sum_i (mft^*(H_i) - \binom{|S'|}{2})$.*

Proof. Let $p, q \in \Omega$ be non adjacent in Q . Clearly every minimal p, q -separator S' in Q contains S since p and q are in Ω . Since S' is a minimal separator in Q for vertices p and q in C , and since S is a clique in Q , $S' \subset S \cup C$. Since Q is a minimal triangulation of H , S' is a minimal p, q -separator in H . Since $S \subset S'$, G and H have the same set of edges, except between vertices which are both in S' . Hence S' is a minimal p, q -separator of G . The formulae follow from Corollary 10. \square

Remark. Notice that the the number of vertices in each lump H_i in Lemma 16 is strictly less than the number of vertices in H , since p and q are not both contained in the same lump.

We now consider minimal triangulations of H in which Ω is a clique.

Lemma 17. *Assume $\Omega = C$ and let Q be a minimal triangulation Q of H such that Ω is a clique in Q .*

If Q realizes the treewidth of H , then $tw(H) = |C| + |S| - 1$. If Q realizes the minimum fill-in: $mft^(H) = \binom{|C \cup S|}{2}$.*

Proof. Obvious, since Q is a clique. \square

Before we continue we need the following crucial observation.

Theorem 18. *Assume $\Omega \neq \emptyset$ and $\Omega \neq C$. Let C^* be a connected component of $G[C - \Omega]$. Then at least one vertex of S does not have a neighbor in C^* .*

Proof. Consider the graph $G^* = G - \Omega$. Then G^* is HHD-free and C^* is a connected component of $G^* - S$. If every vertex of S has a neighbor in C^* then S would be a minimal separator of G^* close to C^* . But then, by Theorem 12, there is a vertex in C^* adjacent to all vertices of S , which is a contradiction. \square

Lemma 19. *Assume $\Omega \neq \emptyset$ and $\Omega \neq C$. Let Q be a minimal triangulation of H such that Ω is a clique in Q . Let C_1, \dots, C_t be the connected components of $G[C - \Omega]$. Let S_i be the set of vertices in $S \cup \Omega$ with a neighbor in C_i in G . Then S_i is a minimal separator in G .*

If Q realizes the treewidth of H , $tw(H) = \max(|S \cup \Omega| - 1, \max_i tw(H_i))$, where $H_i = H(S_i, C_i)$. If Q realizes the minimum fill-in we find: $mft^(H) = \binom{|S \cup \Omega|}{2} + \sum_i (mft^*(H_i) - \binom{|S_i|}{2})$.*

Proof. Since Q is a minimal triangulation C_i ($i = 1, \dots, t$) are the connected components of $Q[C - \Omega]$ and S_i is the set of vertices with a neighbor in C_i in Q . By Theorem 18 there exists a vertex $p_i \in S$ without a neighbor in C_i . Then for any vertex $q_i \in C_i$ the set S_i is a minimal p_i, q_i -separator in Q and hence also in H . We show that S_i is also a minimal separator in G . There exists a connected component C' other than C such that S is close to C' . The component C' is contained in a connected component C^* of $G - S_i$ different from C_i . Then $p_i \in C^*$. Hence every vertex of $S_i \cap \Omega$ is a neighbor of $p_i \in C^*$ and every vertex of $S_i \cap S$ has a neighbor in $C' \subset C^*$. It follows that H_i is a lump. \square

Remark. Notice that the number of vertices of each realizer H_i is strictly less than the number of vertices in H since at least one vertex of S is not a vertex of H_i .

4 Algorithms for the treewidth and minimum fill-in

The algorithms we propose use dynamic programming on lumps to compute the treewidth and minimum fill-in of the realizers of these lumps. First a list of all lumps (S, C) is made and this list is sorted according to the number of vertices of $S \cup C$.

We use the adjacency matrix to test for adjacencies. Creating a list of all minimal separators can be performed in $O(n^3)$ time using Corollary 13. For a minimal separator S , the vertex set of each connected component of $G - S$ can be computed in $O(n + m)$ time. Hence, creating a list of all lumps takes $O(n^2(n + m))$ time. Furthermore, within the same time bound, for each lump (S, C) a minimal separator $S^* \subseteq S$ can be determined such that S^* is close to C .

Notice that the total size of the list of lumps is $O(n^3)$. Sorting this list according to the number of vertices in the lumps using a linear time sorting algorithm takes $O(n^3)$ time.

If for a lump (S, C) the separator S is not close to C , the treewidth and minimum fill-in are given by Lemma 15. Since S^* is known, the treewidth and minimum fill-in of $H(S, C)$ can be determined in constant time in this case.

Now consider a lump (S, C) such that S is close to C . Clearly a possible triangulation of the realizer $H = H(S, C)$ is to make a clique of $S \cup C$. In that case the treewidth and minimum fill-in of that triangulation can easily be determined.

Determining the set Ω takes linear time (for each x , one only has to count the number of neighbors in C , and if this number is equal to the size of C then x belongs to Ω).

We first consider triangulations Q of H such that Ω is not a clique in Q . We use Lemma 16 to determine the minimum fill-in and treewidth in that case. Creating a list of minimal separator S^* with $S \subset S^* \subseteq S \cup C$ can be done in $O(n^3)$ time. For each such S^* , we determine the components C_1, \dots, C_t that partition $G[C - (S^* \setminus S)]$ in time $O(n)$ time. If this partition contains only one component we can ignore the choice of S^* since in that case S^* cannot be a minimal separator for two vertices in Ω . Otherwise update the minima for the treewidth and minimum fill-in of H according to the formulae given in Lemma 16.

Now we consider triangulations Q where Ω is a clique in Q . If $\Omega = C$, then updating the current minima for the minimum fill-in and treewidth is trivial according to Lemma 17. If $\Omega \neq C$ we use Lemma 19. In that case the components C_1, \dots, C_t of $G[C - \Omega]$ can be computed in linear time. It is easy to see that the sets $S_i \subset S \cup \Omega$ which have a neighbor in C_i can be determined in $O(n + m)$

time. Hence updating the minima for the treewidth and minimum fill-in of H in this case takes $O(n + m)$ time.

Theorem 20. *There exists an $O(n^6)$ time algorithm computing the treewidth and minimum fill-in of an HHD-free graph.*

5 Conclusions

In this note we presented a polynomial time algorithm to compute the treewidth and minimum fill-in for HHD-free graphs. We do not claim that our algorithm is a very practical one. Indeed we feel that it is possible to improve the time bounds for these algorithms by analyzing the structure of the minimal separators or, equivalently, the structure of the C_4 's in more detail.

Another question which is left open, is whether a minimum cover of all C_4 's by 'diagonals' gives a chordal graph. If this is the case, this could lead to a more efficient algorithm for the minimum fill-in of HHD-free graphs.

The class of HHD-free graphs is properly contained in that of the weakly chordal graphs. These are graphs without induced C_k or $\overline{C_k}$ for any $k \geq 5$. It is easy to see (by using the results on so called two-pairs [9]) that the number of minimal separators in this case is also at most $O(n^2)$. However, until now, the complexity of the treewidth and minimum fill-in problem for this graph class is unknown.

References

1. Arnborg, S., D. G. Corneil and A. Proskurowski, Complexity of finding embeddings in a k -tree, *SIAM J. Alg. Disc. Meth.* **8**, (1987), pp. 277–284.
2. Bodlaender, H., T. Kloks, D. Kratsch and H. Müller, Treewidth and minimum fill-in on d -trapezoid graphs, Technical report RUU-CS-1995-34, Utrecht University, The Netherlands, 1995.
3. Bodlaender, H. and R. Möhring, The pathwidth and treewidth of cographs, *SIAM Journal on Discrete Mathematics* **7** (1993), pp. 181–188.
4. Brandstädt, A., Special graph classes – A survey, Schriftenreihe des Fachbereichs Mathematik, SM-DU-199 (1991), Universität Duisburg Gesamthochschule.
5. Maw-Shang Chang, Algorithms for maximum matching and minimum fill-in on chordal bipartite graphs. ISAAC'96 (T. Asano et al. ed.), LNCS 1178, pp. 146–155.
6. Dirac, G. A., On rigid circuit graphs, *Abh. Math. Sem. Univ. Hamburg* **25**, (1961), pp. 71–76.
7. Golumbic M. C., *Algorithmic graph theory and perfect graphs*, Academic Press, New York, 1980.
8. Hammer, P. L. and F. Maffray, Completely separable graphs, *Discrete Applied Mathematics* **27**, (1990), pp. 85–99.
9. Hayward, R., C. T. Hoang and F. Maffray, Optimizing weakly triangulated graphs, *Graphs and combinatorics* **5**, (1989), pp. 339–349.
10. Jamison, B. and S. Olariu, On the semi-perfect elimination, *Advances in Applied Mathematics* **9**, (1988), pp. 364–376.

11. Kloks, T., *Treewidth – Computations and Approximations*, Springer Verlag, Lecture Notes in Computer Science 842, (1994).
12. Kloks, T., Treewidth of circle graphs, *International Journal of Foundations of Computer Science* **7**, (1996), pp. 111–120.
13. Kloks, T. and D. Kratsch, Treewidth of chordal bipartite graphs, *J. of Algorithms* **19**, (1995), pp. 266–281.
14. Kloks, T., D. Kratsch and H. Müller, Approximating the bandwidth for AT-free graphs, *Proceedings of the Third Annual European Symposium on Algorithms (ESA '95)*, Springer-Verlag, Lecture Notes in Computer Science 979, (1995), pp. 434–447.
15. Kloks, T., D. Kratsch and J. Spinrad, Treewidth and pathwidth of cocomparability graphs of bounded dimension, Computing Science Notes, 93/46, Eindhoven University of Technology, Eindhoven, The Netherlands, (1993), to appear in *Order*.
16. Kloks, T., D. Kratsch and C. K. Wong, Minimum fill-in of circle and circular arc graphs, *Proceedings of the 21th International Symposium on Automata, Languages and Programming (ICALP'96)*, Springer-Verlag Lecture Notes in Computer Science 1113, (1996), pp. 256–267.
17. Olariu, S., Results on perfect graphs, PhD thesis, School of Computer Science, McGill University, Montreal, 1986.
18. Parra, A., Scheffler, P., How to use minimal separators for its chordal triangulation, *ICALP'95*, LNCS 944, pp. 123–134.
19. Spinrad, J., A. Brandstädt and L. Stewart, Bipartite permutation graphs, *Discrete Applied mathematics* **18**, (1987), pp. 279–292.
20. Sundaram, R., K. Sher Singh and C. Pandu Rangan, Treewidth of circular arc graphs. To appear in *SIAM J. Disc. Math.*
21. Yannakakis, M., Computing the minimum fill-in is NP-complete, *SIAM J. Alg. Disc. Meth.* **2**, (1981), pp. 77–79.