ELSEVIER

Discrete Applied Mathematics 99 (2000) 367–400

DISCRETE APPLIED MATHEMATICS

A linear time algorithm for minimum fill-in and treewidth for distance hereditary graphs

H.J. Broersma[a,*], E. Dahlhaus[b,1], T. Kloks[a]

[a] *University of Twente, Faculty of Mathematical Sciences, P.O.Box 217, 7500 AE Enschede, The Netherlands*

[b] *Department of Computer Science, University of Bonn, Bonn, Germany*

Received 30 September 1997; received in revised form 28 September 1998; accepted 9 March 1999

Abstract

A graph is distance hereditary if it preserves distances in all its connected induced subgraphs. 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 paper we show that both problems are solvable in linear time for distance hereditary graphs. © 2000 Elsevier Science B.V. All rights reserved.

Keywords: Distance hereditary graphs; Minimum fill-in; Treewidth; Chordal graphs; Linear time algorithm, Fragment tree; Tree representation

1. Introduction

A graph is called *distance hereditary* if it preserves distances in all connected vertex set induced subgraphs. They were introduced in [2]. They form a subclass of the well-studied class of perfect graphs. Moreover they belong to the subclass of weakly chordal graphs (no chordless cycle of length greater than four in the graph nor its complement). They are also interesting because they can be represented by a structure of size O(n), where n is the number of vertices of the graph. Note that distance hereditary graphs can all be generated by starting at a single vertex, appending a leaf at any vertex, creating a nonadjacent "false" twin of a vertex, and creating an adjacent "right" twin [2]. On the other hand, distance hereditary graphs contain the cographs as

* Corresponding author.

E-mail address: broersma@math.utwente.nl (H.J. Broersma)

[1] Part of the work has been done while the second author visited the Faculty of Mathematical Sciences of the University of Twente.

0166-218X/00/$ - see front matter
PII: S0166-218X(99)00146-8

a subclass. Many problems that are NP-complete in general can be solved efficiently for distance hereditary graphs. One example is the Hamilton cycle problem [19].

Much attention has been drawn to triangulation problems because of a large number of applications (e.g. Gauss elimination of sparse positive-definite matrices [21]). A *triangulation* of a graph G is an extension G' of G with the same vertex set that is chordal. A graph is *chordal* if it has no chordless cycle of length greater than three. One problem is MINIMUM FILL-IN, i.e. finding a triangulation with a minimum number of additional edges. The other problem we consider in this paper is TREEWIDTH. The objective is to find a triangulation that has a maximum clique size (clique number) that is as small as possible (the treewidth is the minimum clique number of a triangulation minus one). Note that a tree on at least two vertices has treewidth one. Both problems are NP-hard in general [23,1], but polynomial-time algorithms exist for many special graph classes such as cographs, circle graphs and circular arc graphs, permutation graphs and, more generally, cocomparability graphs with bounded dimension, chordal bipartite graphs, etc. [5,14,18,17,3,22,4,15,8,13,20].

For cographs, the treewidth problem can be solved in linear time [5]. With the same approach the minimum fill-in problem can be solved in linear time for cographs. In [4] both problems are solved in linear time for the larger class of permutation graphs. On the other hand, existing polynomial time minimum fill-in and treewidth algorithms for circle graphs and circular arc graphs are not linear. Note that distance hereditary graphs form a subclass of circle graphs (see for example [6]). Therefore, one could ask whether one could extend the methods in [5] to distance hereditary graphs. The linear time fill-in and treewidth algorithms for cographs are based on the so-called cotree representation of cographs. In our paper, we generalize the notion of a cotree to a similar notion called a *fragment tree* for distance hereditary graphs. Using the structure of the fragment tree, we show that the TREEWIDTH and the MINIMUM FILL-IN problem are solvable in linear time for distance hereditary graphs. Moreover, if the fragment tree of a distance hereditary graph is known, we can show that an elimination ordering and the size of a minimum fill-in and the treewidth can be determined in linear time with respect to the number of vertices.

In Section 2 we introduce some notation and terminology, and give some preliminary results. In Section 3 we discuss some structural properties of distance hereditary graphs. In Section 4 we present an algorithm to solve the minimum fill-in problem for distance hereditary graphs, and we discuss the computational complexity of the algorithm. In Section 5 we give a similar approach for solving the treewidth problem for distance hereditary graphs. In Section 6 we give a brief further outlook.

2. Preliminaries

Throughout this section, let $G=(V,E)$ denote a graph with vertex set V and edge set E.

We denote the number of vertices of G by n and the number of edges of G by m.

For a vertex $x \in V$, $N(x)$ is the neigborhood of x in G and $N[x] = \{x\} \cup N(x)$ is the closed neighborhood of x in G.

If Ω is a set and $x \in \Omega$, then we write $\Omega - x$ instead of $\Omega \backslash \{x\}$; if $A \subseteq \Omega$, we write $\Omega - A$ instead of $\Omega \backslash A$. For a nonempty subset Q of V, we write $G[Q]$ for the subgraph of G induced by the vertices of Q. For a vertex $x \in V$, we write $G - x$ instead of $G[V - x]$ if $V - x \neq \emptyset$; for a proper subset W of V we write $G - W$ for the graph $G[V - W]$.

We call a set C of vertices *connected* if its induced subgraph $G[C]$ is connected. If G is disconnected, we call a connected set $C \subseteq V$ a *c-set* and the subgraph $G[C]$ of G induced by C a *component* of G if $G[C]$ is connected and, for no proper superset $C' \supset C$, $G[C']$ is also connected.

Definition 1. A graph is *chordal* if it does not contain an induced cycle of length more than three.

Definition 2. An ordering $<$ on the vertex set V of $G = (V, E)$ is called a *perfect elimination ordering* if with $xy \in E$, $xz \in E$, $x < y$, and $x < z$, $yz \in E$ (i.e. the "larger" neighbors of any vertex induce a complete subgraph).

Lemma 3 (Fulkerson and Gross [11]). *A graph is chordal if and only if it has a perfect elimination ordering.*

Definition 4. A *triangulation* of G is a graph H with the same vertex set as G such that G is a (spanning) subgraph of H and H is chordal.

A triangulation H of G is *minimal* if no proper (spanning) subgraph of H is also a triangulation of G.

Definition 5. The *minimum fill-in* of G, denoted by $\mathrm{mfi}(G)$, is the minimum number of edges which are not edges of G, of a triangulation of G. A *minimum elimination ordering* is a perfect elimination ordering of a triangulation of G with a minimum number of edges. We write $\mathrm{mfi}^*(G) = m + \mathrm{mfi}(G)$ for the number of edges in a triangulation realizing the minimum fill-in. The *treewidth* of G, denoted by $\mathrm{tw}(G)$, 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.

If G is connected, then the *distance* between two vertices x, y of G is the length (number of edges) of a shortest path from x to y in G. It is denoted by $d_G(x, y)$ or $d(x, y)$.

Definition 6. G is called *distance hereditary* if for each pair of vertices x and y of G, and each induced connected subgraph G' of G containing x and y, $d_{G'}(x, y) = d_G(x, y)$.

We use C_k to denote a cycle on $k\geqslant 3$ vertices. A *house* is obtained from a C_5 by adding one extra edge (a chord) between two nonadjacent vertices of the C_5. A *domino* is obtained from a C_6 by adding a chord between two vertices at distance three in the C_6. A *hole* is a cycle of length at least 5. A *gem* is obtained from a house by adding one edge between two nonadjacent vertices of degree two and three, respectively.

As observed in [2] distance hereditary graphs are exactly those graphs that do not contain a house, a domino, a hole or a gem as an induced subgraph.

From the forbidden subgraph characterization it is clear that the class of distance hereditary graphs is properly contained in the class of HHD-free graphs (i.e. graphs that do not contain a house, a hole or a domino as an induced subgraph). In [7] the authors use a dynamic programming approach to show that the treewidth problem and the minimum fill-in problem can be solved in time $O(n^6)$ for HHD-free graphs.

If T is a tree, we speak about *nodes* instead of vertices, and we use the term *leaf set* to denote the set of nodes with degree one in T, and the term *inner node* for a node with degree two or more. If the tree is rooted, we use the terms *parent*, *child*, *ancestor*, and *descendant* in the usual sense.

A graph $G=(V,E)$ is called a *cograph* if there is a tree T with V as its leaf set, such that the inner nodes of T have label 0 or 1, and $xy\in E$ for x and y in V if and only if the least common ancestor of x and y in T has label 1. T is also called a *cotree* for G. Cographs are exactly those graphs that do not have a path of length three (a P_4) as an induced subgraph (see, e.g. [6]). Moreover, each cograph is a distance hereditary graph but not vice versa.

In the sequel we assume that 0's and 1's alternate in the cotree of a cograph. This is no restriction: if two inner nodes with equal labels, 0 or 1, would be adjacent in the cotree, then we can contract them to one inner node with the same label, and the resulting tree is still a cotree of the same graph. By similar arguments we will assume that every inner node of the cotree, except possibly for the root, has degree at least three, i.e. has at least two children.

3. The structure of distance hereditary graphs

Throughout this section we assume that $G=(V,E)$ is a connected distance hereditary graph. For disconnected distance hereditary graphs the following structural results apply to the components separately. We fix a vertex u of G and let L_i be the set of vertices that have distance i from u in G, i.e. $L_i=\{x\mid d_G(u,x)=i\}$. We also refer to L_i as the *ith level.*

Following [12] we get the following structural properties concerning the c-sets of the levels.

Proposition 7. *Let G and L_i be as above.*
1. *Let C be a c-set of L_i. Then $G[C]$ is a cograph.*

2. *Let C be a c-set of L_i ($i \geqslant 1$). Then all vertices in C have the same neighbors in L_{i-1}.*
3. *Let x be a vertex in L_{i+1} for some $i \geqslant 1$, and let C_1 and C_2 be two c-sets of L_i containing neighbors of x. Then all vertices of C_1 and of C_2 are in the neighborhood of x and all vertices in $C_1 \cup C_2$ have the same neighbors in L_{i-1}.*
4. *The neighborhoods of c-sets of L_{i+1} in L_i can be tree-like ordered with respect to the subset relation, i.e. if C_1 and C_2 are distinct c-sets of L_{i+1}, then $N(C_1) \cap N(C_2) \cap L_i = \emptyset$ or $N(C_1) \cap L_i$ and $N(C_2) \cap L_i$ are comparable with respect to $\subseteq$.*
5. *Let C be a c-set of L_{i+1}. Then all vertices in $N(C) \cap L_i$ have the same neighbors in $L_i - N(C)$.*

We call a maximal union of c-sets of L_i with at least one common neighbor in L_{i+1} (if $L_{i+1} \neq \emptyset$) an *m-set* of L_i. For technical reasons we also call the c-sets of the highest level m-sets.

Lemma 8. *All m-sets of L_i are pairwise disjoint.*

Proof. For the highest level this is obvious. For the other levels this is a consequence of the fact that neighborhoods $N(v) \cap L_i$, $v \in L_{i+1}$ can be tree-like ordered with respect to the subset relation. □

Corollary 9. *All vertices that appear in the same m-set of L_i ($i \geqslant 1$) have the same neighbors in L_{i-1}.*

Clearly, since the c-sets of the levels induce cographs, the m-sets of the levels induce cographs too.

Consider for each node t of the cotree T_C of the m-set C, the set F_t of descendants of t that are leaves (i.e. vertices of C). We call F_t a *cotree fragment*. The cotree fragments are ordered in a tree-like manner, i.e. $F_{t_1} \subseteq F_{t_2}$ if t_1 is a descendant of t_2, and $F_{t_1} \cap F_{t_2} = \emptyset$ if neither t_1 is a descendant of t_2 nor t_2 is a descendant of t_1. Let C be an m-set of L_i and C' be an m-set of L_{i+1}. Suppose $N(C') \cap L_i$ intersects C. Then C is the only m-set of L_i that has a nonempty intersection with $N(C')$. This follows from the following result.

Lemma 10. *Let C' be an m-set of L_{i+1} and C be an m-set of L_i. Then either $N(C') \cap C = \emptyset$ or $N(C') \cap L_i \subseteq C$.*

Proof. It is sufficient to show that $N(C')$ can intersect only one m-set of L_i. If two distinct c-sets C_1 and C_2 of L_i have a common neighbor $v \in L_{i+1}$, then by Proposition 7 all vertices in the c-set of L_{i+1} containing v have all vertices of C_1 and C_2 as its neighbors. By the maximality of the m-sets this implies that all neighbors in L_i of an m-set of L_{i+1} are in one m-set of L_i. □

We call $F_{C'} = N(C') \cap L_i$ a *neighborhood fragment*.

Lemma 11. *If F_1 is a cotree fragment and F_2 is a neighborhood fragment, then $F_1 \cap F_2 = \emptyset$ or $F_1 \subseteq F_2$ or $F_2 \subseteq F_1$.*

Proof. Note that by Proposition 7, all neighborhood fragments F_2 that are subsets of the m-set C have the property that all vertices in F_2 have the same neighbors in $C - F_2$. Assume F_1 is a maximal cotree fragment that intersects the neighborhood fragment F_2 but neither $F_1 \subset F_2$ nor $F_2 \subset F_1$ is true. Let $F_1 = F_t$ and t' be the parent of t in T_C. Then $F_2 \subset F_{t'}$. W.l.o.g. we assume that t is a 0-node and therefore t' is a 1-node. Pick a vertex $x \in F_2 \cap F_t$, a vertex $y \in F_2 - F_t$, and a vertex $z \in F_t - F_2$. Then $xz \notin E$ and $yz \in E$. This is a contradiction to the fact that all vertices of F_2 have the same neighbors in $C - F_2$. □

Define a *fragment* as a neighborhood fragment or a cotree fragment or a single vertex.

Lemma 12. *The fragments of an m-set C are ordered in a tree-like manner, i.e. if F_1 and F_2 are fragments of C, then either they are disjoint or comparable with respect to the subset relation.*

Proof. This follows immediately from Lemma 11 and Proposition 7. □

The introduction of fragments allows us now to define a *fragment tree*.

Extension of T_C. The *extended cotree T'_C* consists of the fragments of C and the singletons containing one vertex of C as nodes. The parent of any vertex of fragment F of C is the smallest fragment F' properly containing F. The labelling of T'_C with 0, 1, or "single vertex fragment" is done as follows.

1. A cotree fragment is labelled with 0 or 1 as in the cotree T_C.
2. A vertex fragment is labelled as a *vertex fragment*.
3. A neighborhood fragment F that does not coincide with a cotree fragment is labelled with the same label as the next greater cotree fragment that contains F.

Still we can show the following.

Lemma 13. *For vertices v and w in C, $vw \in E$ if and only if the least common ancestor of $\{v\}$ and $\{w\}$* in T'_C has label 1.

Proof. If the least common ancestor of $\{v\}$ and $\{w\}$ is a cotree fragment, then it is also the least common ancestor of v and w in T_C and we are done. Otherwise, the least common ancestor of v and w in T_C is the node t in T_C, such that F_t is the smallest fragment that contains the least common ancestor fragment F of $\{v\}$ and $\{w\}$ in T'_C. Note that F is a neighborhood fragment that does not coincide with a cotree fragment. This fragment F has the same label as F_t. □

One difference between the cotree and the extended cotree of C is that the labels of the extended cotree are not necessarily alternating.

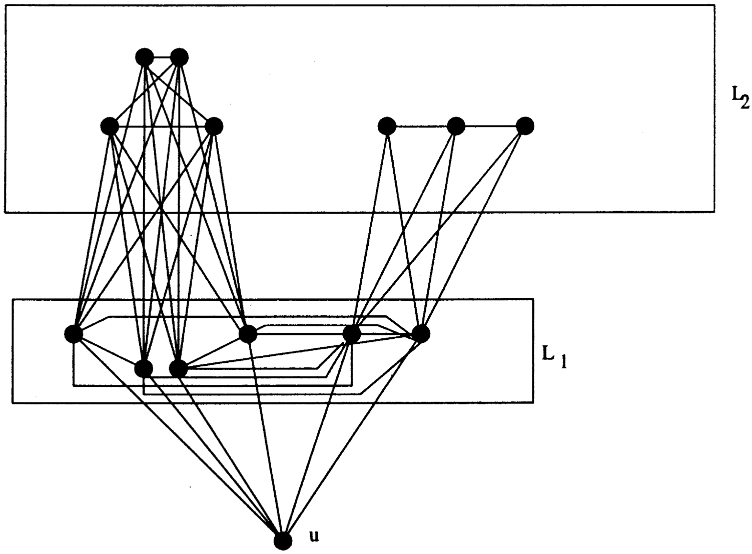

Fig. 1. A distance hereditary graph and its levels L_1 and L_2.

Edges between different m-sets: Note that each m-set C is a fragment. They represent the roots of the extended cotrees T'_C. Let C be a fragment of L_{i+1}. Then parent(C) of C is the fragment $N(C) \cap L_i$.

The whole fragment tree: The fragment tree T_G of the distance hereditary graph G consists of the fragments as nodes. If a fragment F is not an m-set and a fragment of an m-set C, then parent(F) of F is the parent of F in the extended cotree T'_C of C. If F is an m-set, then parent(F) is defined as in the last item. We label fragments with 0, 1, and "vertex fragment" as they were labelled as nodes of the extended cotree and label additionally m-sets with M.

Fig. 1 shows a distance hereditary graph G. Fig. 2 is an extension of Fig. 1 with the cotree fragments of G (broken lines). In Fig. 3 we added also the neighborhood fragments that are not cotree fragments. Finally it is easily checked that the tree shown in Fig. 4 is a fragment tree of G.

Proposition 14. *The distance hereditary graph G is uniquely determined by its fragment tree T_G, i.e. if T_G is known with all its labels, then G can uniquely be reconstructed from T_G.*

Proof. We first can determine the level of each m-set D by counting the number of m-sets that are ancestors of D in T_G.

Next we can determine the edges in each level by looking, for each m-set, at the maximal subtree of T_G rooted at the m-set vertex and containing no other m-set vertices. This subtree corresponds to the cotree of the m-set if we suppress the unlabelled neighborhood vertices of the subtree.

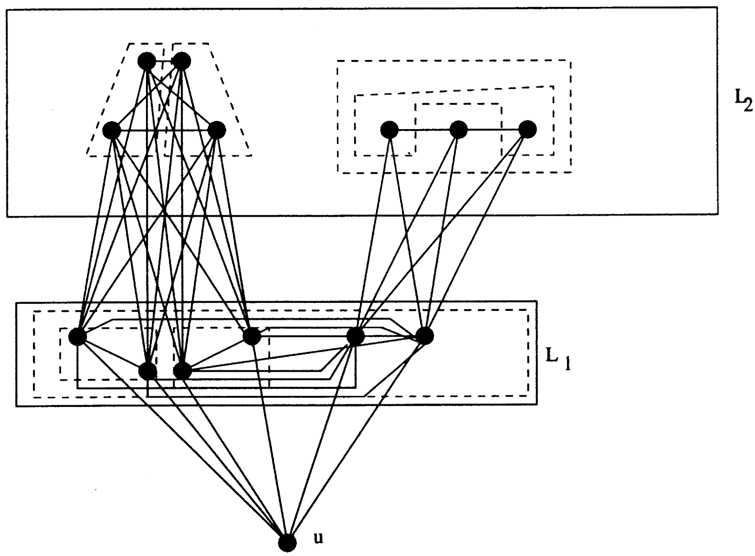

Fig. 2. The cotree fragments (broken lines).

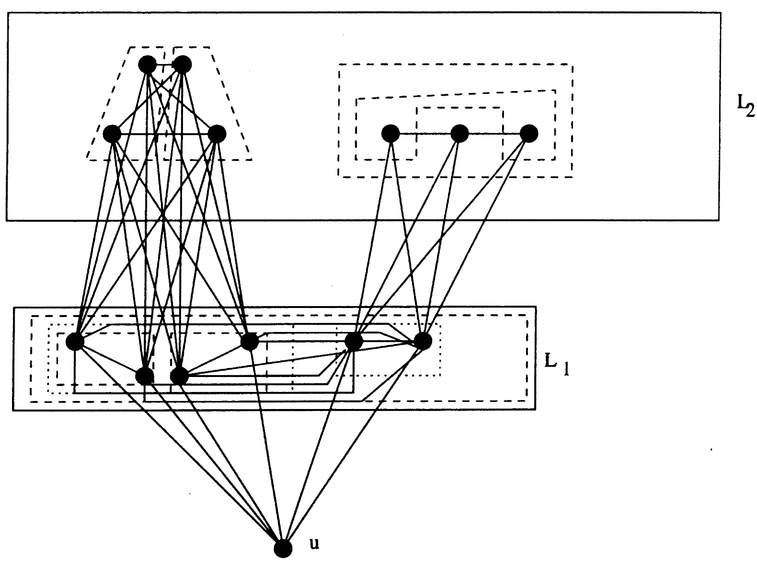

Fig. 3. Adding the neighborhood fragments (dotted lines).

Finally we can determine the edges between different levels, i.e. between L_i and L_{i+1} by looking at two subtrees of T_G. Consider, for each m-set C, the maximal subtree T_1 rooted at $\mathrm{Par}(C)$ and containing no other m-sets, and the maximal subtree T_2 rooted at C and containing no other m-sets. Join the vertices of degree zero or one in T_1 with the vertices of degree zero or one in T_2. $\square$

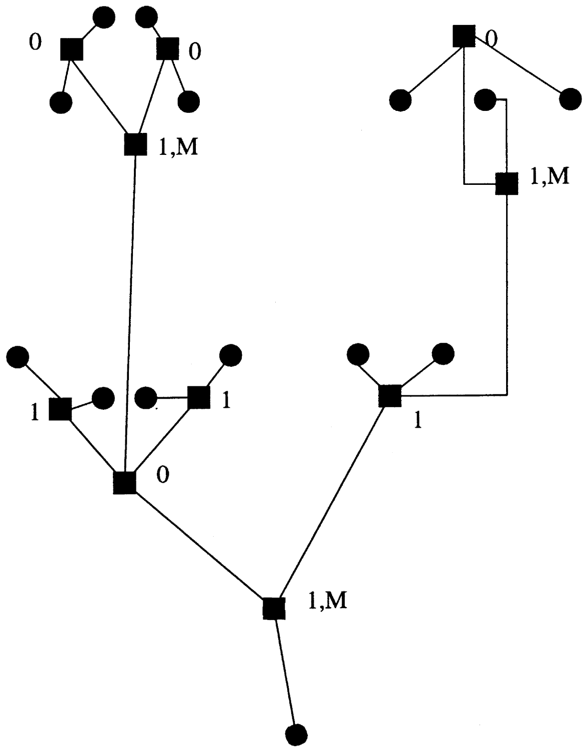

Fig. 4. The fragment tree.

Finally we would like to show that the fragment tree of a distance hereditary graph can be determined efficiently.

Theorem 15. For a distance hereditary graph G, a fragment tree T_G of G can be constructed in $O(n+m)$ time.

Proof. The levels L_i can be computed in $O(n+m)$ time (see [12]). Then the c-sets of each level L_i can be computed in linear time $O(|L_i| + |E(L_i)|)$. Also the equivalence classes of m-sets can be determined in linear time by selecting a neighbor in L_{i+1} with a maximum number of neighbors in L_i (compare [12]). Knowing the cotrees of all m-sets and the neighborhoods $N(C) \cap L_i$ of all m-sets of L_{i+1}, we get the tree T_G restricted to fragments in L_i as follows. For each fragment C, we select a vertex $v \in C$ and determine the next larger fragment C' that contains v. This can be done in linear time, because the number of pairs (x, C) with $x \in C$ and C is a fragment is bounded by $O(n+m)$: we need only one representing vertex for each fragment, there are at most as many neighborhoods of L_{i+1} in L_i as there are edges between these levels, and the number of distinct 0-fragments and 1-fragments that are descendants of a 1-fragment in a cotree of L_i correspond to at most twice the number of edges in L_i. To get the parents of m-sets in linear time is obvious. □

3.1. The structure of minimal triangulations

In this section we will show that any minimal triangulation of a distance hereditary graph is distance hereditary, characterize distance hereditary chordal graphs, and show how the fragment tree of a minimal triangulation of a distance hereditary graph looks like.

First we will start with some general results on minimal triangulations.

Definition 16. Let a and b be distinct nonadjacent vertices of G. A set $S \subset V$ is a *minimal* a,b*-separator* of G if a and b are in different components of $G-S$ and there is no proper subset of S with the same property. A *minimal separator* of G is a set $S \subset V$ for which there exist distinct nonadjacent vertices a and b such that S is a minimal a,b-separator of G.

Definition 17. Let S be a minimal separator and C a c-set of $G-S$ such that every vertex of S has a neighbor in C. Then we say that S is *close to* C.

There exist many characterizations of chordal graphs. We use the characterization given by Dirac [10] using minimal separators.

Lemma 18. *G is chordal if and only if every minimal separator induces a complete subgraph of G.*

For a proof of the following, see, e.g. [16].

Lemma 19. *Let H be a minimal triangulation of G and let S be a minimal a,b-separator of H for distinct nonadjacent vertices a and b in H. Then S is also a minimal a,b-separator in G, and if C is a c-set of $H-S$, then C induces also a component in $G-S$.*

This lemma shows that we always get a minimal triangulation in that way that we construct a sequence $(G_i)_{i=0}^{k}$, such that starting with $G_0 = G$, G_i arises from G_{i-1} by making a minimal separator of G_{i-1} complete, and the minimal triangulation H is G_k. One also can prove that each H that is constructed in that way is a minimal triangulation [20]. This follows from the following result.

Lemma 20 (Parra and Scheffler [20]). *If S is a minimal separator of G, then there is a minimal triangulation H of G, such that S is also a minimal separator of H.*

For the rest of this section, we assume that the graph G is distance hereditary.

The following lemma characterizes minimal separators of distance hereditary graphs as nonempty joint neighborhoods of nonadjacent vertices.

Lemma 21. *Assume x and y are distinct nonadjacent vertices of G and $C = N(x) \cap N(y) \neq \emptyset$. Then C is a (minimal) separator. Conversely, if C is a minimal separator, then there are x and y that are separated by C and $C = N(x) \cap N(y)$.*

Proof. The first statement follows directly from the fact that G is distance hereditary (all chordless paths from x to y have length two).

The second statement is also true for HHD-free graphs that contain the distance hereditary graphs (see for example [7]). □

Lemma 22. *Assume x and y have distance two in G. Then in a (minimal) triangulation of G, xy is an edge or $C = N(x) \cap N(y)$ induces a complete graph.*

Proof. Let G' be a (minimal) triangulation of G, and suppose that xy is not an edge of G', and $G[C]$ is not complete. Then for a nonadjacent pair $z, w \in C$ of G', x, z, y, w induces a chordless cycle of length four, a contradiction. □

Lemma 23. *Any minimal triangulation of a distance hereditary graph is distance hereditary.*

Proof. Due to Lemma 20, we only have to show that when a minimal separator S of G is made complete, the resulting graph G' is still distance hereditary. By Lemma 21, S is the joint neighborhood of two nonadjacent vertices, say x and y. As shown in [7], G' is HHD-free (note that G is HHD-free), because a graph remains HHD-free if we make a minimal separator complete. Following the characterization of distance hereditary graphs by forbidden introduced subgraphs, it remains to show that G' does not have a gem. Let S be the joint neighborhood of the nonadjacent vertices x and y. S separates x and y. Moreover, for all c-sets C of $G - S$, all $z \in C$ that have neighbors in S have the same neighbors in S and for all different c-sets C_1 and C_2 of $G - S$, $N(C_1) \cap S$ and $N(C_2) \cap S$ are disjoint or comparable with respect to the subset relation. Suppose we create a gem. Then either two adjacent vertices or three vertices belonging to a common triangle belong to S. It is easily checked that in all combinations, one gets a contradiction to the statements of the last sentence.

This proves that G' is distance hereditary. □

Since any minimal triangulation of a distance hereditary graph is distance hereditary, we now take a further look at distance hereditary chordal graphs.

Lemma 24. *A fragment tree T is a fragment tree of a distance hereditary chordal graph if and only if*

1. *every neighborhood fragment is complete, i.e. every fragment F that is the parent of an M-labelled node is either a single vertex fragment or all its descendants in the extended cotree it belongs to are* 1*-labelled or single vertex fragments and*
2. *for every 1-labelled fragment, all but one child fragment is complete, i.e. if F is a 1-labelled fragment of the m-set C, then for all but one of the children F' of*

F, F' and all descendants of F' in the extended cotree T'_C are 1*-labelled or single vertex fragments.*

Proof. Suppose T is a fragment tree of a chordal distance hereditary graph, i.e. T is a tree such that the vertices are labelled with $0, 1$, or "single vertex" and some vertices are also labelled with M, and the distance hereditary graph defined by the labelled tree is chordal.

Note that each node t of T belongs to the m-set M_t that is associated with the next ancestor m_t of t that is labelled with M. The vertices of M_t are the descendants v of m_t that are labelled with "single vertex", such that m_t is the next ancestor of v that is M-labelled.

Let F_t be the fragment associated with t and let F_t be a neighborhood fragment. That means there is an M-labelled child t' of t. We show that F_t is complete. Assume F_t is not complete. Then F_t has more than one element. Then also the m-set M_t containing F_t has more than one element. Therefore M_t is not in the level L_0 and has therefore a parent, say D. If $M_t \subseteq L_i$, then D is the neighborhood of M_t in L_{i-1}, and the m-set $M_{t'}$ associated with t' is in L_{i+1}. Note that no vertex in D is adjacent to any vertex in $M_{t'}$. We pick two nonadjacent vertices in F_t, say x and y, a vertex $z \in D$, and a vertex $w \in M_{t'}$. These four vertices induce a cycle of length four. This is a contradiction.

Next we show that for each 1-labelled fragment F, at most one child fragment F' is not complete. We may assume that F is not complete. Therefore F is not a neighborhood fragment. That means that all child fragments of F are in the same m-set as F. Assume F has two noncomplete child fragments F_1 and F_2. Then we may pick two nonadjacent vertices in F_1 and two nonadjacent vertices in F_2, and we get an induced cycle of length four. This is a contradiction.

Vice versa, assume that the fragment tree T satisfies conditions 1 and 2 of the lemma. We construct a perfect elimination ordering as follows. We sort the vertices of the graph G (they are the "single vertex" labelled nodes) in the first priority in descending order with respect to the number of ancestor fragments that are m-sets and in the second priority with respect to the number of 0-labelled ancestor fragments. If the numbers of ancestors that are m-sets and the number of 0-labelled ancestors of vertices are equal, then they are sorted in any way. Let $<$ be the ordering defined by this sorting. Assume y and z are neighbors of x and $x < y$ and $x < z$. We have to show that y and z are neighbors. Assume $x \in L_i$. Then y and z are in L_i or L_{i-1}, because the vertices are sorted in descending order with respect to the number of ancestors that are m-sets and neighborhoods can be either in the same level L_i or between adjacent levels.

First we assume that $y, z \in L_{i-1}$. Then y and z are in the neighborhood of the m-set M in L_{i-1} that contains x and therefore in a common neighborhood fragment. Since each neighborhood fragment is complete, y and z are neighbors.

Next we assume that $y \in L_i$ and $z \in L_{i-1}$. Since xy is an edge, y and x are in the same m-set, say M. z is in the neighborhood of M in L_{i-1}. Since all vertices of M have the same neighbors in L_{i-1}, yz is an edge.

Finally we assume that y and z are in L_i. Since xy and xz are edges, the least common ancestors F_{xy} and F_{xz} of x and y and of x and z, respectively, are 1-labelled. If $F_{xy} \neq F_{xz}$, then the least common ancestor F_{yz} is one of F_{xy} or F_{xz} and therefore yz is an edge. Otherwise if $F = F_{xy} = F_{xz}$, we assume that the least common ancestor F_{yz} is 0-labelled. Then F_{yz} is a descendant of F. The child fragment F' of F containing x and the child fragment F'' of F containing the fragment F_{yz} are different. F'' is not complete and therefore, since only one child fragment of F is not complete, F' is a complete fragment. F' and each descendant of F' is not 0-labelled. Therefore x has at least one 0-labelled ancestor less than y and z. This is a contradiction to the assumption that vertices that have the same number of m-set ancestors are sorted in descending order with respect to the number of 0-labelled ancestors. That means the least common ancestor of y and z is 1-labelled and therefore yz is an edge. □

Next we discuss the fragment tree structure of minimal triangulations of distance hereditary graphs.

Let Compl be the set of all fragments of G that are complete in the minimal triangulation G' of G. Then we immediately observe the following.

Lemma 25.
1. *Each "single vertex" labelled fragment is in* Compl.
2. *If F is a fragment of the m-set C and $F \notin$* Compl, *then all ancestors of F in the extended cotree T'_C (i.e. all ancestors of F that are descendants of C including C) are not in* Compl.
3. *If an m-set C is not in* Compl, *then its parent fragment (that is one level L_i lower) is in* Compl.
4. *If F is a* 1*-labelled fragment of the m-set C, then there is at most one child of F in the extended cotree T'_C of C that is not in* Compl.

Proof. Statements 1 and 2 are trivial, and 3 can be proved as follows. Note that the vertices of the parent of C belong to the joint neighborhood of the vertices of C. Since C is not complete, the vertices of the parent of C belong to the joint neighborhood of two nonadjacent vertices. Therefore C must be complete.

Statement 4 is shown as follows. Suppose F' is a child of the 1-labelled fragment F in C. Then the vertices of each other child F'' in C belong to the joint neighborhood of all vertices in F'. If F' is not complete, then the vertices of F'' are in the joint neighborhood of two nonadjacent vertices. Therefore F'' is complete and in Compl. □

We call a set Compl that satisfies the requirements of the last lemma a *triangulation representative*.

We can reformulate the definition of a triangulation representative as follows.

Lemma 26. Compl *is a triangulation representative if and only if*
1. *All "single vertex"-labelled fragments are in* Compl,

2. *If a fragment* F *is in* Compl, *then all its nonm-set children of* F *are in* Compl,
3. *If* F *is not in* Compl, *then all its m-set children are in* Compl *and*
4. *If* F *is a* 1*-labelled fragment, then at most one nonm-set child is not in* Compl.

Proof. Note that if F is a fragment of the m-set C, then the nonm-set children of F are exactly the children of F in the extended cotree T'_C. The fact that all ancestors of F in T'_C are not in Compl if F is not in Compl translates into the fact that if F is in Compl, then all descendants of F in T'_C are in Compl, and this translates again into the fact that if F is in Compl, then all nonm-set children are in Compl.

The fact that the parent of an m-set C that is not in Compl is in Compl translates into the fact that the m-set children of a fragment F that is not in Compl are in Compl.

This proves that the second and the third items of Lemma 25 and of Lemma 26 are equivalent. The remaining items of Lemmas 25 and 26 are identical. □

Let Compl be a triangulation representative. Then we can construct a fragment tree T_{Compl} as follows.

1. If F is a fragment of the m-set C and $F \in$ Compl, then F and all its 0-labelled descendants in the extended cotree T'_C of C are 1-labelled.
2. Suppose $F \notin$ Compl and F is a neighborhood fragment. Then we create a new fragment F' of F that becomes the parent of F, and such that the parent of F' is the original parent of F in T_G. For each child C of F that was an m-set in G, the parent of C in T_{Compl} is F'. Each child C of F that was an m-set in T_G (and therefore M-labelled in T_G) is not M-labelled in T_{Compl}. F' is a 1-labelled node. If F was an m-set in T_G, then F is not an m-set in T_{Compl} and F' becomes an m-set of T_{Compl}.

Lemma 27.

1. *For each triangulation representative* Compl, T_{Compl} *is a fragment tree of a chordal graph.*
2. *If* Compl *is the set of fragments of* G *that are complete in the minimal triangulation* G' *of* G, *then* T_{Compl} *is a fragment tree for* G'.

Proof. To prove the first statement, we proceed as follows. If F is a neighborhood fragment that is not in Compl, then F' is not a neighborhood fragment in T_{Compl}. No child of F' in T_{Compl} is labelled with M. The only case that a new m-set is created is that F was an m-set and a neighborhood fragment in T_G and F' becomes an m-set in T_{Compl}. In T_G, the parent of F is a neighborhood fragment $F_1 \in$ Compl and in T_{Compl}, F_1 is the parent of F'. Therefore each neighborhood fragment in T_{Compl} is a neighborhood fragment in T_G. Since all neighborhood fragments of T_G that are not in Compl lose their m-set children in T_{Compl} and these children lose their M-label, the remaining neighborhood fragments of T_{Compl} are all in Compl, and their corresponding vertex sets are complete. This proves that all neighborhood fragments of T_{Compl} are complete sets.

It remains to show that each 1-labelled fragment of an m-set C in T_{Compl} has only one noncomplete child fragment of the same m-set C. We first take closer look at the structure of the m-sets of T_{Compl}. Suppose C is an m-set. Note that the vertices and fragments of C are those F that can be reached from C in T_G or T_{Compl} without passing an m-set. We denote the set of fragments F belonging to C in T_G by T_C^G and the set of fragments belonging to C in T_{Compl} by T_C^{Compl}. Note that if C is an m-set of T_G not in Compl, then either C is an m-set in T_{Compl} (if C is not a neighborhood fragment) or the fragment C' consisting of C and all m-set children of C in T_G, as children become an m-set of T_{Compl}. In that case we say that C *remains an m-set*. If an m-set C is in Compl, then either C remains an m-set in T_{Compl} (if also its parent is in Compl) and the set of fragments belonging to C is not changed, or C and all its descendants in T_C^G become members of T_D^{Compl}, where D is the m-set that contains the parent of C in T_G. Note that C becomes a member of $T_{D'}^{\text{Compl}}$ if D is a neighborhood fragment. In that case, we say that C is *amalgamated into* D. Note that only m-sets C that are in Compl are amalgamated into the m-set D. Therefore also after amalgamation, each 1-labelled fragment has only one nonm-set child that is not in Compl. This proves that in T_{Compl}, each 1-labelled fragment F has at most one child F' that is not complete, i.e. not all descendants F'' of F' including F that belong to the m-set C that contains F are 1- or "single vertex"-labelled.

We continue with the proof of the second part of the lemma. Let Compl be the set of fragments that are made complete by a minimal fill-in G' of G. We have to show that each edge of the graph that is represented by T_{Compl} is an edge in G'. There are two types of edges xy: x and y are in the same m-set and the least common ancestor of x and y is 1-labelled, or x is in an m-set C and y is in the parent fragment of C.

Suppose x and y are in the same m-set C of T_G. Then the least common ancestors of x and y in T_G and T_{Compl} are the same. Note that for fragments F_1 and F_2 of the m-set C of T_G, F_1 is an ancestor of F_2 in T_G if and only if F_1 is an ancestor of F_2 in T_{Compl}, because any new element F' as parent of the neighborhood c-set F is inserted between F and the parent of F and the insertion of F' does not affect the ancestorship of F_1 and F_2 and any other change of the parent of a fragment F is only possible if F is an m-set. Note that x and y are joint by an edge in G' if and only the least common ancestor of x and y is 1-labelled in T_{Compl}, i.e. it is 1-labelled in T_G and therefore xy is an edge in G or it is in Compl and therefore xy is an edge in G'.

Now we assume that x and y are in different m-sets of T_G but in the same m-set of T_{Compl}. Then at least one of the vertices x and y appears in an m-set C of T_G whose parent fragment is not complete. Let C_x and C_y be the m-sets containing x and y. The first case is that the parent of C_x is a neighborhood fragment $F_x \notin \text{Compl}$ of $D = C_y$. In T_{Compl}, C_x and F_x become children of F_x'. In T_{Compl}, the least common ancestor F of x and y is therefore the least common ancestor of F_x' and y. If $F \neq F_x'$, then F is the least common ancestor of F_x and y in T_{Compl}, and since F_x and y are fragments of the same m-set, F is the least caommon ancestor of F_x and y in T_G. Note that $F \notin \text{Compl}$, because the descendant F_x of F is not in Compl. If $F \neq F_x'$ and xy is an edge of G', then F is 1-labelled also in T_G and therefore, since F is the least common ancestor of

F_x and y, y is adjacent to all vertices that belong to the fragment F_x. Note that also x is adjacent to all vertices in F_x. Since F_x is not in Compl, there are two vertices u and v belonging to F_x that are not adjacent in G'. Therefore xy must be an edge in G'. If $F = F'_x$, then y belongs to the fragment F_x and is therefore in the neighborhood of C_x in G and therefore xy is also an edge in G.

The second case is that C_x and C_y are in Compl and their parents F_x and F_y are not in Compl and in the same m-set D. Note that F_x is an ancestor of F_y or vice versa or the least common ancestor of F_x and F_y in T_G is 0-labelled; otherwise Compl is not a triangulation representative. In the last case, also the least common ancestor of F'_x and F'_y in T_{Compl} is the same and therefore 0-labelled. Assume xy is an edge of G'. W.l.o.g. we may assume that F_x is an ancestor of F_y. Then each vertex belonging to F_y belongs also to F_x. Therefore also x is adjacent to all vertices in F_y. Since F_y is not in Compl, one again finds two vertices u and v that belong to F_y and that are not adjacent in G'. Therefore x and y must be adjacent in G' as common neighbors of u and v.

Finally we assume that x and y are in different m-sets D_1 and D_2 of T_{Compl}. Then we may assume that y is in the parent fragment F_y of D_1. In any case, F_y is also a fragment of T_G and in Compl, because it is a neighborhood fragment in T_{Compl}. If in T_G the parent fragment of the m-set C_x containing x is F_y, then we are done. Otherwise C_x is amalgamated into D_1 and therefore the parent of C_x is a neighborhood fragment F_x that is not in Compl and that is also a fragment of the m-set D_1. C_x and F_y, and therefore x and y belong to the neighborhood of F_x, and since F_x is not in Compl, to the common neighborhood of two vertices that are not adjacent in G'. Therefore xy is an edge in G'. □

4. The algorithm for minimum fill-in

Theorem 28. *If a fragment tree of a distance hereditary graph G is known, then a minimum elimination ordering (perfect elimination ordering of a minimum fill-in) can be determined in* $O(n)$ *time.*

We develop an algorithm to compute an elimination ordering with a minimum fill-in or to compute a fill-in of minimum maximum clique size (for the purpose of computing the treewidth) as follows.

1. We first compute a triangulation representative Compl that creates a minimum number of fill-in edges. We do not compute the set of fill-in edges, because it would require a time of $O(n^2)$. But with the knowledge of the set Compl of fragments that are made complete, we will compute the number of fill-in edges in $O(n)$ time.
2. With the knowledge of the fragments that have to be made complete, we also can compute a perfect elimination ordering of the minimum fill-in.

4.1. Counting fill-in edges to fragments

First we count the number of additional edges that are created by the graph that is represented by T_{Compl}. We count fill-in edges to fragments.

Lemma 29. *Let G be a distance hereditary graph and let T_G be its fragment tree. Let* Compl *be a triangulation representative and T_{Compl} its fragment tree. Let G' be the graph that is represented by T_{Compl}. Then for an edge xy of G' that is not in G, if x and y belong to the same m-set C, then the least common ancestor of x and y in T_G is in* Compl.

Proof. By construction of T_{Compl}, the least common ancestors of x and y in T_G and in T_{Compl} are the same. It is 1-labelled if and only if it is in Compl. □

Lemma 30. *Let G be a distance hereditary graph and let T_G be its fragment tree. Let* Compl *be a triangulation representative and T_{Compl} its fragment tree. Let G' be the graph that is represented by T_{Compl}. Then for an edge xy of G' that is not in G, if x and y do not belong to the same m-set C and C_x and C_y are the m-sets containing x and y respectively, then the parent of C_x is an ancestor of the parent of C_y or vice versa. If the parent of C_y is an ancestor of the parent of C_x, then the joint neighborhood of x and y in G is the parent of C_x, and the parent of C_x is not in* Compl.

Proof. We denote the parents of C_x and C_y by F_x and F_y respectively. We distinguish the cases that x and y are in the same m-set of T_{Compl} and that x and y are in different m-sets of T_{Compl}.

First consider the case that x and y are in the same m-set D of T_{Compl}. Consider the situation that C_x and C_y are amalgamated into D. Then for the same reasons as in the proof of Lemma 27, F_x is an ancestor of F_y or vice versa. W.l.o.g. we may assume that F_x is an ancestor of F_y. Then all vertices belonging to F_x belong also to F_y. Therefore the joint neighborhood of x and y is the set of vertices belonging to F_x. Consider the situation that F_x is a fragment that belongs to the m-set C_y. Then C_y and therefore also F_y is an ancestor of F_x. Since y is not a neighbor of x, y is not in F_x and the only common neighbors of x and y are in C_y. Therefore F_x is the joint neighborhood of x and y.

Finally consider the case that x and y are in different m-sets of T_{Compl}. Let C'_x and C'_y be the m-sets of T_{Compl} containing x and y respectively. Then the parent F''_x of C'_x is a fragment of C'_y or vice versa. W.l.o.g. we assume that F''_x is a fragment of C'_y. Note that F''_x is a fragment of T_G that is in Compl and a neighborhood fragment in T_G. Moreover, F''_x is a fragment belonging to C_y. Since xy is not an edge in G, C_x is amalgamated into C'_x by the fragment F_x. Clearly C_y, and therefore also F_y is an ancestor of F_x. The joint neighborhood of x and y in G is F_x. □

Let $G=(V,E)$ be a distance hereditary graph and let T_G be its fragment tree. Let T'_C be the extended cotree of the m-set C. Then for each fragment F belonging to C,

$$\mathrm{Co}(F) := \{xy \notin E \mid x \in C,\ y \in C,\ \text{the least common ancestor of } x \text{ and } y \text{ is } F\}$$

and

$$\mathrm{NC}(F) := \{xy \notin E \mid x \text{ and } y \text{ belong to different m-sets of } T_G \text{ and the joint neighborhood of } x \text{ and } y \text{ is } F\}.$$

From the previous lemma we get immediately the following.

Corollary 31. *Let* Compl *be a triangulation representative of* T_G *and* T_{Compl} *be defined as before. Let* $G'=(V,E')$ *be the triangulation of* G *represented by* T_{Compl}. *Then*

$$E' - E = \bigcup_{F \notin \mathrm{Compl}} \mathrm{NC}(F) \cup \bigcup_{F \in \mathrm{Compl}} \mathrm{Co}(F).$$

Note that the sets NC(F) and Co(F) are pairwise disjoint. Therefore we also get the following.

Corollary 32. *Let* Compl *be a triangulation representative of* T_G *and* T_{Compl} *be defined as before. Let* $G'=(V,E')$ *be the triangulation of* G *represented by* T_{Compl}. *Then*

$$|E' - E| = \sum_{F \notin \mathrm{Compl}} |\mathrm{NC}(F)| + \sum_{F \in \mathrm{Compl}} |\mathrm{Co}(F)|.$$

This yields the following expression for the size of a minimum fill-in.

Corollary 33. *The size of a minimum fill-in is determined by*

$$\min_{\mathrm{Compl}} \sum_{F \notin \mathrm{Compl}} |\mathrm{NC}(F)| + \sum_{F \in \mathrm{Compl}} |\mathrm{Co}(F)|,$$

where the minimum is taken over all triangulation representatives.

We say that a fill-in edge is *counted* to a fragment F if it is in NC(F) or in Co(F).

The strategy of the algorithm is that we determine, recursively by a bottom up strategy on the fragment tree, for each fragment F, the minimum number $\mathrm{NC}^*(F)$ of fill-in edges of a fill-in making F not complete that are counted to F or a descendant of F, and the minimum number $\mathrm{Co}^*(F)$ of fill-in edges of a fill-in making F complete that are counted to F or a descendant of F.

First we show that we can determine the sizes of NC(F) and Co(F) in O(n) time. Then, with the knowledge of $|\mathrm{NC}(F)|$ and $|\mathrm{Co}(F)|$, we will determine $\mathrm{NC}^*(F)$ and $\mathrm{Co}^*(F)$, for all fragments F, in O(n) time. Note that we will not determine the edge sets contributing to $\mathrm{NC}^*(F)$ or $\mathrm{Co}^*(F)$ explicitly.

4.2. Computing the sizes of NC(F) and Co(F)

Lemma 34. *For all fragments F simultaneously, $|\mathrm{NC}(F)|$ and $|\mathrm{Co}(F)|$ can be computed in $\mathrm{O}(n)$ time, provided the fragment tree of G is known.*

Proof. Note that we can determine the size $|F|$ of all fragments F in $\mathrm{O}(n)$ time, because we only have to compute, for each m-set C, for each fragment belonging to C, the number of leaf descendants, by recursively adding the sizes of the child fragments of the same m-set C.

We first determine $|\mathrm{Co}(F)|$. If F is a 1-fragment, then $\mathrm{Co}(F)=\emptyset$ and therefore $|\mathrm{Co}(F)|=0$. If F is a 0-fragment with nonm-set children $D_1,\dots,D_k$, then

$$|\mathrm{Co}(F)| = \sum_{\mu<\nu} |D_\mu|\,|D_\nu| = \frac{1}{2}\left(\left(\sum_{\mu=1}^{k} |D_\mu|\right)^2 - \sum_{\mu=1}^{k} |D_\mu|^2\right).$$

Note that $|\mathrm{Co}(F)|$ can be computed in $\mathrm{O}(k)$ time, and therefore for all F simultaneously in $\mathrm{O}(n)$ time.

Next we determine $|\mathrm{NC}(F)|$. We denote the parent of a fragment F in the fragment tree T_G by Parent(F). For any vertex x, we denote the m-set that contains x by C_x and the parent fragment of C_x by F_x. Note that if xy is in NC(F), then $F_x=F$ or $F_y=F$. W.l.o.g. we assume that $F=F_x$. Then one of the next three cases can occur.

1. F_x and F_y belong to the same m-set and $F_x\subseteq F_y$ (x and y are in the same level L_i and therefore F_x and F_y are in L_{i-1}). If both $F_x=F_y=F$, then xy is put into $\mathrm{NC}_0(F)$; otherwise, we put xy into $\mathrm{NC}_1(F)$.
2. $x\in L_{i+1}$ and $y\in L_i$. Then y belongs to the same m-set as F, y does not belong to F and the smallest fragment containing y and F is a 1-fragment. In this case xy is put into $\mathrm{NC}_2(F)$.
3. $x\in L_{i+1}$ and $y\in L_i$. If C is the m-set containing F, then $y\in \mathrm{Parent}(C)$. In that case, xy is put into $\mathrm{NC}_3(F)$.

We determine the sizes of these sets $\mathrm{NC}_0(F),\dots,\mathrm{NC}_3(F)$.

Let $D_1,\dots,D_k$ be the m-sets with $\mathrm{Parent}(D_\mu)=F$. Then

$$|\mathrm{NC}_0(F)| = \sum_{\mu<\nu} |D_\mu|\,|D_\nu|.$$

By the same argument as in the computation of $|\mathrm{Co}(F)|$, for a 0-labelled fragment C, we can compute $|\mathrm{NC}_0(F)|$, for all C simultaneously, in $\mathrm{O}(n)$ time.

To determine $|\mathrm{NC}_1(F)|$, we first determine, for each fragment D, the number

$$x_D = \sum (|D'|\colon D' \text{ is an m-set and } \mathrm{Parent}(D')=D).$$

This can be done in $\mathrm{O}(n)$ time, for all D simultaneously. Then for any fragment C in L_i,

$$|\mathrm{NC}_1(F)| = x_F\left(\sum_{D \text{ proper ancestor of } F \text{ in } L_i} x_D\right).$$

We determine the sums $u_F := \sum_{D \text{ proper ancestor of } F \text{ in } L_i} x_D$ going top down. If F is an m-set, then $u_C = 0$. If C is not an m-set, then $u_F = u_{\text{Parent}(F)} + x_{\text{Parent}(F)}$. We get all u_F's in $O(n)$ time and therefore also all $|\text{NC}_1(F)|$ in $O(n)$ time.

To determine $|\text{NC}_2(F)|$, we determine the number a_F of vertices in the same m-set C as F that are not in F but adjacent to all vertices in F. Note that $|\text{NC}_2(F)| = x_F a_F$.

To get a_F, we determine, for each fragment D that is not an m-set and with a 1-fragment D' as parent, the number y_D of vertices u with D' as smallest fragment containing u and D. Note that $y_D = \sum_{D'':\ \text{Parent}(D'')=\text{Parent}(D), D'' \neq D} |D''| = \sum_{D'':\text{Parent}(D'')=\text{Parent}(D)} |D''| - |D|$. If D has k children, y_D can be computed in $O(k)$ time. Therefore, for all D simultaneously, y_D can be computed in $O(n)$ time.

Now if F is in the m-set C, then the number of vertices in C that are not in F and that are adjacent to all vertices in F, say a_F, is the sum over all y_D, such that D is in C, Parent(D) is 1-labelled, and D is an ancestor of F. Let $y_D = 0$ if Parent(D) is not 1-labelled. Then we can determine all a_F's in $O(n)$ time in the same way as we determined the u_F's (going top down).

Therefore $|\text{NC}_2(F)|$ can be determined in $O(n)$ time, for all F simultaneously.

Let C be the m-set containing F. Then $|\text{NC}_3(F)| = x_F|\text{Parent}(C)|$. $|\text{NC}_3(F)|$ can clearly be determined in $O(n)$ time, for all F simultaneously. □

4.3. Computing the size of a minimum fill-in

Before we compute the size of a minimum fill-in, we introduce the notion of a *partial triangulation representative*. Let F be a fragment of T_G. Then T_F is the subtree of T_G consisting of F and all its descendants. A subset Compl of the fragments of T_F is called a *partial triangulation representative* of F if it satisfies the requirements of a triangulation representative for the fragments in T_F, i.e.

1. each "single vertex" labelled fragment of T_F is in Compl.
2. If F' is a fragment in T_F and $F' \in$ Compl, then all its nonm-set children are in Compl.
3. If F' is a fragment in T_F and $F' \notin$ Compl, then all its m-set children are in Compl.
4. If F' is a 1-labelled fragment of T_F, then there is at most one 0-labelled child of F' that is not in Compl and that is not an m-set.

Note that a triangulation representative is a partial triangulation representative of the root fragment L_0.

Lemma 35. *Recursively, we can define partial triangulation representatives as follows.*

1. *If F is a leaf, then $\{F\}$ is a partial triangulation representative of F. (Note that leaves are all "single vertex"-labelled fragments.)*
2. *Let $C_1,\ldots,C_k$ denote the children of F that are m-sets and $D_1,\ldots,D_l$ denote the children of F that are not m-sets. Assume* $\text{Compl}_1,\ldots,\text{Compl}_k$ *are partial triangulation representatives of $C_1,\ldots,C_k$ and* $\text{Compl}'_1,\ldots,\text{Compl}'_l$ *are partial triangulation representatives of $D_1,\ldots,D_l$.*

(*a*) If $\mathrm{Compl}'_1,\dots,\mathrm{Compl}'_l$ *contain* $D_1,\dots,D_l$ *respectively*, *then*

$$\mathrm{Compl} := \bigcup_{i=1}^{k} \mathrm{Compl}_i \cup \bigcup_{i=1}^{l} \mathrm{Compl}'_i \cup \{F\}$$

is a partial triangulation representative of F (*this includes "single vertex"-labelled fragments that are not leaves*).

(*b*) *If* $\mathrm{Compl}_1,\dots,\mathrm{Compl}_k$ *contain* $C_1,\dots,C_k$, *respectively*, *and* F *is* 0-*labelled*, *then*

$$\mathrm{Compl} := \bigcup_{i=1}^{k} \mathrm{Compl}_i \cup \bigcup_{i=1}^{l} \mathrm{Compl}'_i$$

is a partial triangulation representative of F.

(*c*) *If* $\mathrm{Compl}_1,\dots,\mathrm{Compl}_k$ *contain* $C_1,\dots,C_k$, *respectively*, F *is* 1-*labelled and at most one* D_i *is not in* Compl'_i, *then*

$$\mathrm{Compl} := \bigcup_{i=1}^{k} \mathrm{Compl}_i \cup \bigcup_{i=1}^{l} \mathrm{Compl}'_i$$

is a partial triangulation representative of F.

3. *All partial triangulation representatives can be created in this way*.

Proof. First note that if F is a fragment, F' is a child of F, and if Compl' is the restriction of the partial triangulation representative Compl of F to $T_{F'}$, then Compl' is a partial triangulation representative of F'.

It is easily checked that in all cases, if Compl_i, for $i=1,\dots,k$ and Compl'_j, for $j=1,\dots,l$ are partial triangulation representatives of $C_1,\dots,C_k$ and $D_1,\dots,D_l$, then Compl is a partial triangulation representative of F.

Conversely, if Compl is a partial triangulation representative of F, then let Compl_i be Compl restricted to T_{C_i} and Compl'_j be Compl restricted to T_{D_j}. It is easily checked that Compl can be created by the above recursion rules from $\mathrm{Compl}_1,\dots,\mathrm{Compl}_k$ and $\mathrm{Compl}'_1,\dots,\mathrm{Compl}'_l$. □

For a partial triangulation representative Compl of F, the *weight* of Compl is defined as

$$W(\mathrm{Compl}) := \sum_{F'\in\mathrm{Compl}} |\mathrm{Co}(F')| + \sum_{F'\in T_F-\mathrm{Compl}} |\mathrm{NC}(F')|.$$

Note that if R is the root fragment (this is L_0), then the size of the fill-in of the triangulation representative Compl is $W(\mathrm{Compl})$.

Let $\mathrm{Co}^*(F)$ be the minimum weight of a partial triangulation representative of F containing F, $\mathrm{NC}^*(F)$ be the minimum weight of a partial triangulation representative of F not containing F, and $M(F)=\min(\mathrm{NC}^*(F),\mathrm{Co}^*(F))$ be the minimum weight of a partial triangulation representative of F. Then for the root fragment R, $M(R)$ is the minimum size of a fill-in.

Theorem 36. *Let F be a fragment with m-set children $C_1,\dots,C_k$ and nonm-set children $D_1,\dots,D_l$. Then* $\mathrm{NC}^*(F)$ *and* $\mathrm{Co}^*(F)$ *are recursively determined as follows.*

1. *If F is a leaf, then*

$$\mathrm{Co}^*(F)=0.$$

2. *If F is a "single vertex"-labelled fragment, then* $\mathrm{NC}^*(F)$ *is not defined, i.e. is set* ∞.
3. *For any fragment F,*

$$\mathrm{Co}^*(F)=|\mathrm{Co}(F)|+\sum_{i=1}^{k}M(C_i)+\sum_{j=1}^{l}\mathrm{Co}^*(D_j).$$

4. *If F is* 0*-labelled, then*

$$\mathrm{NC}^*(F)=|\mathrm{NC}(F)|+\sum_{i=1}^{k}\mathrm{Co}^*(C_i)+\sum_{j=1}^{l}M(D_j).$$

5. *If F is* 1*-labelled, then*

$$\mathrm{NC}^*(F)=|\mathrm{NC}(F)|+\sum_{i=1}^{k}\mathrm{Co}^*(C_i)+\min_{j=1}^{l}\,(M(D_j)+\sum_{i\neq j}\mathrm{Co}^*(D_i)).$$

For a fragment F, $M(F):=\min(\mathrm{Co}^*(F),\mathrm{NC}^*(F))$.

Proof. Note that a leaf fragment is always a "single vertex"-labelled fragment. Therefore $\mathrm{Co}^*(F)=0$ for any leaf fragment F. If F is a single vertex fragment, then $\mathrm{NC}^*(F)$ is not defined, because F belongs to any partial triangulation representative of any ancestor F' of F (including F). This realizes 1 and 2.

To show 3 one has to consider $2a$ of the previous lemma. This is the only case where F is in the partial triangulation representative Compl. The m-set children can be in Compl or not. The nonm-set children must be in Compl. To get $\mathrm{Co}^*(F)$, we have therefore to sum up the minimum weights $M(C_i)$ of the m-set children C_i and the minimum weights $\mathrm{Co}^*(D_j)$ of partial triangulation representatives of D_j containing D_j and $\mathrm{Co}^*(F)$.

To show 4, one has to consider part 2b of the previous lemma. This covers the case that F is 0-labelled and not in Compl. Here the m-set children are in Compl and the nonm-set children can be in Compl or not. Therefore we have to sum up over $|\mathrm{NC}(F)|$, the weights $\mathrm{Co}^*(C_i)$, and over the minimum weights $M(D_j)$.

To show 5, we refer to 2c of the previous lemma. This covers the case that F is 1-labelled and F is not in Compl. Again the m-set children are in Compl. Contrary to the case that F is 0-labelled, at most one nonm-set child is not in Compl. Therefore we have to minimize over all j the sum of $\mathrm{Co}^*(D_i)$ with $i \neq j$ and the minimum weight $M(D_j)$. We add to this minimum the sum over all $\mathrm{Co}^*(C_i)$ and $|\mathrm{NC}(F)|$. □

Note that Theorem 36 defines a recursive algorithm to determine the size of a minimum fill-in. It remains to check the complexity of the algorithm.

Clearly 1 and 2 can be done in O(1) time and 3 and 4 can be done in $O(k+l)$ time, i.e. in the order of the number of leaves of F.

Lemma 37. *Part 5 of Theorem 36 can be executed in* $O(k+l)$ *time.*

Proof. The difficult step is to compute $\min_{j=1}^{l}(M(D_j)+\sum_{i\neq j}\mathrm{Co}^*(D_i))$. Define $X_j := (M(D_j)+\sum_{i\neq j}\mathrm{Co}^*(D_i))$. We have to compute the minimum over all X_j. We compute $S := \sum_{i=1}^{l}\mathrm{Co}^*(D_i)$ in $O(l)$ time and get $X_j = S - \mathrm{Co}^*(D_j) + M(D_j)$. This can be done for each X_j separately in O(1) time, and therefore for all X_j simultaneously in $O(l)$ time. The minimization over all X_j can be done in $O(l)$ time. It remains to add all $\mathrm{Co}^*(C_i)$ and $|\mathrm{NC}(F)|$. This can be done in $O(l)$ time. Therefore we get an overall time of $O(k+l)$. □

The overall complexity of the algorithm to get the size of a minimum fill-in is the sum of the number of children over all fragments and the number of leaves of the fragment tree. Therefore we get the following.

Theorem 38. *The size of a minimum fill-in of a distance hereditary graph G can be computed in* $O(n+m)$ *time. If the fragment tree of G is known, then it can be computed in* $O(n)$ *time.*

4.4. Computing the triangulation representative of a minimum fill-in

To get the fragment tree of a minimum fill-in, we have to determine a triangulation representative Compl of minimum weight.

We assume that we have computed $\mathrm{NC}^*(F)$ and $\mathrm{Co}^*(F)$ for all fragments of G. Whenever $M(F)$ is called, we keep track whether it is $\mathrm{NC}^*(F)$ or $\mathrm{Co}(F)$. Denote the set of children of F by Child_F. Note that $\mathrm{NC}^*(F)$ and $\mathrm{Co}^*(F)$ are minimum sums $\sum_{F'\in P}\mathrm{Co}^*(F')+\sum_{F'\in \mathrm{Child}_F - P}M(F')$ such that P is a subset of Child_F satisfying certain requirements. For each F, we easily can compute these sets $P = P_{F,\mathrm{NC}}$ and $P = P_{F,\mathrm{Co}}$, such that

$$\mathrm{NC}^*(F) = \sum_{F'\in P_{F,\mathrm{NC}}}\mathrm{Co}^*(F') + \sum_{F'\in \mathrm{Child}_F - P_{F,\mathrm{NC}}} M(F')$$

and

$$\mathrm{Co}^*(F) = \sum_{F'\in P_{F,\mathrm{Co}}}\mathrm{Co}^*(F') + \sum_{F'\in \mathrm{Child}_F - P_{F,\mathrm{Co}}} M(F').$$

Note that $P_{F,\mathrm{NC}}$ and $P_{F,\mathrm{Co}}$ can be computed by the following extension of the recursive procedure that computes NC^* and Co^*.

1. If F is a leaf, then

$$\mathrm{Co}^*(F) := 0.$$

2. If F is a "single vertex"-labelled fragment, then $\mathrm{NC}^*(F)$ is not defined, i.e. is set ∞.

3. For any fragment F,

$$\mathrm{Co}^*(F) := |\mathrm{Co}(F)| + \sum_{i=1}^{k} M(C_i) + \sum_{j=1}^{l} \mathrm{Co}^*(D_j).$$

$P_{F,\mathrm{Co}} := \{D_1, \ldots, D_l\}.$

4. If F is 0-labelled, then

$$\mathrm{NC}^*(F) := |\mathrm{NC}(F)| + \sum_{i=1}^{k} \mathrm{Co}^*(C_i) + \sum_{j=1}^{l} M(D_j).$$

$P_{F,\mathrm{NC}} := \{C_1, \ldots, C_k\}.$

5. If F is 1-labelled, then select a j such that

$$\mathrm{NC}^*(F) = |\mathrm{NC}(F)| + \sum_{i=1}^{k} \mathrm{Co}^*(C_i) + \left(M(D_j) + \sum_{i \neq j} \mathrm{Co}^*(D_i)\right)$$

is minimum.

$P_{F,\mathrm{NC}} := \{C_1, \ldots, C_k\} \cup \{D_i \mid i \neq j\}.$

6. For a fragment F, $M(F) := \min(\mathrm{Co}^*(F), \mathrm{NC}^*(F))$.

This procedure has a time bound of $O(n)$.

We get a triangulation representative Compl that realizes a minimum fill-in by going top down along the fragment tree.

1. The root $R = \{v_0\}$ belongs to Compl.
2. If F is in Compl, then each child $F' \in P_{F,\mathrm{Co}}$ is set to be in Compl and each child $F' \notin P_{F,\mathrm{Co}}$ is set to be in Compl if and only if $M(F') = \mathrm{Co}^*(F')$. Otherwise F' is set not to be in Compl.
3. If F is set not to be in Compl, then each child $F' \in P_{F,\mathrm{NC}}$ is set to be in Compl and each child $F' \notin P_{F,\mathrm{NC}}$ is set to be in Compl if and only if $M(F') = \mathrm{Co}^*(F')$. Otherwise F' is set not to be in Compl.

As an overall result of this subsection, we get the following.

Theorem 39. *A triangulation representative of a minimum fill-in can be computed in* $O(n)$ *time if the fragment tree is known.*

4.5. Computing the fragment tree of the minimum fill-in and the minimum elimination ordering

Lemma 40. *A fragment tree of the minimum fill-in can be determined in* $O(n)$ *time.*

Proof. We know that the triangulation representative Compl can be computed in $O(n)$ time. Recall that the fragment tree T_{Compl} of the triangulation of Compl is determined as follows.

1. If F is a fragment of the m-set C and $F \in$ Compl, then F and all its 0-labelled descendants in the extended cotree T'_C of C are 1-labelled.

2. Suppose $F \notin \text{Compl}$ and F is a neighborhood fragment. Then we create a new fragment F' of F that becomes the parent of F, and the parent of F' is the original parent of F in T_G. For each child C of F that was an m-set in G, the parent of C in T_{Compl} is F'. Each child C of F that was an m-set in T_G (and therefore M-labelled in T_G) is not M-labelled in T_{Compl}. F' is a 1-labelled node. If F was an m-set in T_G, then F is not an m-set in T_{Compl} and F' becomes an m-set of T_{Compl}.

It is easily checked that this can be done in $O(n)$ time. □

Knowing the fragment tree of T_{Compl}, we can also obtain a perfect elimination ordering of the graph G' that is represented by T_{Compl} as follows (see the proof of Lemma 24).

1. We sort the vertices v of the graph G (they are the "single vertex" labelled nodes) in the first priority in descending order with respect to the number of ancestor fragments that are m-sets and in the second priority with respect to the number of 0-labelled ancestor fragments in the m-set that contains v.
2. If the number of ancestors that are m-sets and the number of 0-labelled ancestors of vertices are equal, then they are sorted in either way.

We show that this can be done in $O(n)$ *time*. Clearly the number of m-set ancestors and 0-labelled ancestors of all fragments can be computed in $O(n)$ time. Also the m-set containing a fragment F can be determined, for all F in $O(n)$ time (this is the next ancestor of F that is an m-set. Therefore for each fragment F, also the number of 0-ancestors in the same m-set can be determined in $O(n)$ time. Sorting can be done in $O(n)$ time by bucket sort as follows. We first create lists A_i containing the fragments with i m-set ancestors. Then for each A_i, we create the lists $A_{i,j}$ of fragments in A_i that have j 0-labelled ancestors in A_i. Note that there are only $O(n)$ lists $A_{i,j}$. Moreover, if A_i is a nonempty list and $i' < i$, then also $A_{i'}$ is nonempty, and if $A_{i,j}$ is a nonempty list and $j' < j$, then $A_{i,j'}$ is nonempty. We concatenate the lists $A_{i,j}$ and eliminate those fragments that are not "single vertex"-labelled. This defines a perfect elimination ordering.

Concluding this section, we get the following.

Theorem 41. *A minimum elimination ordering of a distance hereditary graph can be determined in* $O(n+m)$ *time, and in* $O(n)$ *time if a fragment tree is known.*

5. Computing the treewidth

To show how to determine the treewidth, we first discuss the structure of cliques, i.e. maximal complete sets, of distance hereditary chordal graphs.

5.1. Cliques of distance hereditary chordal graphs

A first observation is the following.

Lemma 42. *If $G=(V,E)$ is a chordal graph with a perfect elimination ordering $<$, then $Q_x := \{y > x \mid xy \in E\} \cup \{x\}$ induces a complete subgraph of G. Moreover, for each clique Q of G, there is a vertex $x = x_q$, such that $Q = Q_x$.*

The first statement of the lemma follows directly from the fact that all greater neighbors of any vertex are pairwise adjacent. To show the second statement, we only have to select the smallest vertex x_q of Q and clearly $Q \subseteq Q_{x_q}$, and since Q is a (maximal) clique, $Q = Q_x$.

In this subsection, we assume that G is a distance hereditary chordal graph with fragment tree T_G. Let C be an m-set and T'_C its extended cotree. Recall that a fragment F of C represents a complete set of vertices if and only if F and all its descendants in T'_C are 1-labelled. In that case, we say also that F is *complete*. Recall that any 1-labelled fragment of C has at most one noncomplete child in T'_C, i.e. at most one noncomplete nonm-set child.

Let V_F be the set of vertices of the fragment F.

Lemma 43. *For each clique Q of G, there is an m-set C, such that $Q = V_{\mathrm{Parent}(C)} \cup D$, where D is a clique of $G[V_C]$.*

Proof. Let Q be a clique of G. Let i be the maximum, such that $Q \cap L_i \neq \emptyset$. Note that all vertices of $Q \cap L_i$ are in the same c-set of L_i and therefore in the same m-set C. Vertices of $Q - L_i$ can only be in the neighborhood of C in L_{i-1}, i.e. the parent fragment Parent(C) of C. Since all vertices in V_C have the same neighbors in L_{i-1} and Q is a maximal complete subset, all vertices of $V_{\mathrm{Parent}(C)}$ are in Q. Therefore $Q - L_i = V_{\mathrm{Parent}(C)}$. Since all vertices of V_C have the same neighbors in $V_{\mathrm{Parent}(C)}$ and therefore the same neighbors in $Q - L_i$, $Q \cap L_i$ is a maximal complete set (clique) of $G[V_C]$. □

Next we have to consider the structure of cliques of $G_C = G[V_C]$. For any vertex $x \in V_C$, let B_x be the first ancestor fragment of x in T'_C that is 0-labelled and A_x be the child fragment of B_x that has x as an ancestor. If x has no 0-labelled ancestor in T'_C, then $A_x := C$.

Lemma 44. *For vertices x and y of G_C, $xy \in E$ if and only if A_x is an ancestor of A_y or vice versa.*

Proof. Note that $xy \in E$ if and only if the least common ancestor F of x and y in T'_C is 1-labelled. Suppose $xy \in E$. Let F_x and F_y be the children of F that have x and y as descendants. Since G_C is chordal, at least one of F_x and F_y is complete. We assume that F_x is complete. Then B_x is a proper ancestor of F, and therefore A_x is F or an ancestor of F. Therefore A_x is also an ancestor of A_y. □

Note that all A_x are 1-labelled fragments such that the parent B_x of A_x is 0-labelled if A_x is not an m-set. In general, a fragment A is called a 1–0 *jump* if A is 1-labelled or a single vertex and its parent is 0-labelled or if A is a 1-labelled or single vertex m-set. For a 1–0 jump A, let

$$Q_A := \{x \mid A_x = A \text{ or } A_x \text{ is an ancestor of } A \text{ in } T'_C\},$$

where C is the m-set containing A and T'_C is the extended cotree of C. Due to the previous lemma, all vertices in Q_A are pairwise adjacent (i.e. they form a complete set).

Lemma 45. *Q is a clique if and only if there is a complete 1–0-jump A such that $Q = Q_A$.*

Proof. Define as a *chain* a set Ch of fragments such that with $F, F' \in \mathrm{Ch}$, F is an ancestor of F' or vice versa. Note that there is a one-one correspondence between cliques and maximal chains of 1–0-jumps. Let Ch_q be the maximal chain of 1–0 jumps corresponding to Q and A the fragment in Ch_q that has the largest distance from the root C of T'_C in T'_C. Then A has no 0-labelled descendant in Ch (otherwise we could add any non 0-labelled child to Ch_q). That means that A is complete. Moreover, Ch_q is the set of ancestors of A that are 1–0 jumps. Therefore $Q = Q_A$.

Conversely, let A be a complete 1–0 jump. Then the set Ch_A of ancestors of A including A that are 1–0-jumps is a maximal chain of 1–0-jumps, and therefore Q_A is a clique. □

For a fragment F belonging to the m-set C, let $N(F)$ be the set of all vertices of the m-set C that are adjacent to all vertices belonging to F, i.e.

$$N(F) := \{y \mid y \text{ is a single vertex in } C \text{ and not a descendant of } F \text{ and the least common ancestor of } y \text{ and } F \text{ is 1-labelled}\}.$$

Then by Lemma 44,

$$Q_A = V_A \cup N(A).$$

By Lemma 43, we get the following.

Corollary 46. *Each clique of the distance hereditary chordal graph G is of the form $Q_A \cup \mathrm{Parent}(C)$, where A is a complete 1–0-jump and C is the m-set containing the fragment A.*

5.2. The cliques of a minimal fill-in of a distance hereditary graph

Now let G be any distance hereditary graph and G' its minimal triangulation. Let Compl be the set of fragments of G that are complete in G'. Then we can make the following observation.

Remark. If F is a 1-labelled fragment and all nonm-set children of F are in Compl, then also F is in Compl $(^*)$.

We call a triangulation representative satisfying (*) a *strong triangulation representative.*

Lemma 47. *Let* Compl *be a strong triangulation representative of* G, *and let* G' *be the triangulation of* G *corresponding to* Compl. *Let* V_F *be the set of vertices belonging to* F. *If* F *is not in* Compl, *then* $G'[F]$ *is not complete.*

Proof. Let C be the m-set of the fragment tree T_G of G containing F (C is the next m-set ancestor of F). Let T'_C be the extended fragment tree of C. Note that F has a 0-labelled descendant F' in T'_C that is not in Compl. Let T_{Compl} be the fragment tree of G'. Note that F' is also 0-labelled in T_{Compl}. Therefore $V_{F'}$ induces a noncomplete subgraph of G'. Note that $V_{F'}$ is a subset of V_F. □

Let G' be the triangulation of G that corresponds to the strong triangulation representative Compl, and let T_{Compl} be the fragment tree of G' constructed from the fragment tree T_G of G and Compl.

Lemma 48. *Let* A_q *be the complete* 1–0 *jump of the clique* Q *of* G' *and* B_q *be the parent of* A_q *in* T_{Compl}. *Then* A_q *and* B_q *are also fragments of* T_G, *and* B_q *is also the parent of* A_q *in* T_G.

Proof. The first case is that A_q is an m-set C of T_{Compl}. Then all fragments of the extended cotree T'_C of T_G are in Compl, and no m-set C' is amalgamated into C. Therefore C is also an m-set of T_G. The parent of C in T_G and in T_{Compl} are the same and the parent Parent(C) of C is in Compl.

The second case is that A_q is not an m-set. Then B_q is 0-labelled in T_{Compl} and therefore not the parent fragment F' in T_{Compl} of a neighborhood fragment F of T_G that is not in Compl. Also A_q is not such a fragment F', because it is in Compl and has no nonm-set children that are not in Compl. Clearly B_q is also the parent of A_q in T_G. □

By the above observations we can identify each clique with an edge A_qB_q of the fragment tree T_G of G.

Next we would like to determine the clique associated with A_qB_q in terms of the fragment tree T_G of G.

Let F be a fragment of T_G that belongs to the m-set C. Then V_F is the set of vertices that belong to F (i.e. the set of vertices, i.e. "single vertex"-labelled fragments that are descendants of F in T'_C).

Lemma 49.

1. *If* A_q *is an m-set of* T_G *and* B_q *is the parent of* A_q, *then the clique* Q *is* $V_{A_q} \cup V_{B_q}$.
2. *If* A_q *is not an m-set of* T_G, B_q *is the* 0*-labelled parent of* A_q *in* T_G, *and* A_q *and* B_q *belong to the m-set* C, *then the clique* Q *associated with* A_qB_q *consists of*
   (a) *the vertices in* V_{A_q},

(b) *the vertices of* $V_{\mathrm{Parent}(C)}$,
(c) *the vertices of* $V_{C'}$, *where* C' *is an m-set and* $\mathrm{Parent}(C')$ *is an ancestor in* T'_C *of* B_q, *or* $\mathrm{Parent}(C') = B_q$ *and* $\mathrm{Parent}(C')$ *belongs to the m-set* C *and*
(d) *the vertices* v *in* V_C, *such that the least common ancestor of* B_q *and* v *in* T_G (*i.e.* T'_C) *is* 1-*labelled.*

Proof. In general, $A_q \in \mathrm{Compl}$, because it is complete in G'.

First suppose A_q is an m-set in T_G. We claim that A_q is also an m-set in T_{Compl}. Note that no m-set child is amalgamated with the m-set $A_q = C$. Moreover the m-set A_q is not amalgamated into an m-set C' (otherwise the parent of A_q in T_{Compl} is 1-labelled). Since $C = A_q$ is in Compl, all fragments belonging to C are in Compl and therefore no m-set is amalgamated into C. Therefore the vertices belonging to C in T_G and the vertices belonging to C in T_{Compl} are the same. Moreover, the parent of $A_q = C$, say B_q, is in Compl (otherwise C would not remain an m-set in T_{Compl}). Note that also the vertices belonging to B_q in T_G and the vertices belonging to B_q in T_{Compl} are the same, because all descendants of B_q in the same m-set as B_q are in Compl and therefore no m-set is amalgamated with a fragment that is a descendant of B_q and that belongs to the same m-set as B_q. Therefore $Q = V_{A_q} \cup V_{B_q}$.

Suppose now that A_q is not an m-set in T_G. Recall that G' is the graph represented by T_{Compl}. Let C be the m-set of T_G the fragment A_q belongs to. Note that also B_q belongs to C. Since B_q is not in Compl, C does not belong to Compl. Therefore C is not amalgamated into another m-set. Let C' be the m-set of T_{Compl} the fragment A_q belongs to. Then $C = C'$ or C' is the 1-labelled parent of C in T_{Compl} (the latter is the case if C is a neighborhood fragment in T_G). Note that the parent $\mathrm{Parent}(C)$ of C in T_G is in Compl and therefore the vertices belonging to $\mathrm{Parent}(C)$ in T_G and the vertices belonging to $\mathrm{Parent}(C)$ in T_{Compl} are the same. Moreover $\mathrm{Parent}(C)$ is the parent of C' in T_{Compl}. Since A_q is in Compl, the vertices belonging to A_q in T_G and in T_{Compl} are the same. Let $N(F)$ be the set of vertices belonging to the m-set C' that are adjacent to all vertices of F in G' and that are not descendants of F. Note that the vertices of Q belonging to C' are the vertices belonging to A_q and the vertices of $N(A_q)$, i.e. those vertices x that do not belong to A_q and such that the least common ancestor of A_q and x in T_{Compl} is 1-labelled. Note that all 1-labelled ancestors of A_q are ancestors of B_q, since B_q is 0-labelled. There are two types of vertices in $N(A_q)$.

1. The least common ancestor of v and A_q is a 1-labelled fragment F of T_G. Then F is also the least common ancestor of v and B_q in T_G. Then v is in V_C and the least common ancestor of v and B_q in T_G is 1-labelled.
2. The least common ancestor of v and A_q is not a fragment of T_G, i.e. the parent fragment F' of a neighborhood fragment $F \notin \mathrm{Compl}$ in T_{Compl}. The children of F' in T_{Compl} are F and the m-set children of F in T_G. Since F is the only child of F' that belongs to the m-set C of T_G, F' is an ancestor of A_q in T_{Compl}, and A_q belongs to the m-set C of T_G, F is an ancestor of A_q. Since F is not in Compl, F is an ancestor of B_q or is B_q itself. Since the least common ancestor of v and A_q

(that is also the least common ancestor of v and B_q) in T_{Compl} is F', v is a vertex belonging to a child of F' that is an m-set of T_G, i.e. to a child of F in T_G that is an m-set of T_G. That means v is a vertex in an m-set D of T_G, such that the parent of D in T_G is an ancestor of B_q in T_G.

Note that the vertices not belonging to C that are in Q are the vertices that belong to the parent of C in T_G. □

For any fragment A of T_G that has a 0-labelled parent or that is an m-set, we define *the associated clique Q_A* as follows.

Let B be the parent of A. Then

1. If A is an m-set of T_G, then Q_A is $V_A \cup V_B$.
2. If A is not an m-set of T_G, and A and B belong to the m-set C, then Q_A associated with AB consists of
   (a) the vertices in V_A,
   (b) the vertices of $V_{\text{Parent}(C)}$,
   (c) the vertices of $V_{C'}$, where C' is an m-set and Parent(C') is an ancestor in T'_C of B, or Parent$(C') = B$ and Parent(C') belongs to the m-set C and
   (d) the vertices v in V_C, such that the least common ancestor of B and v in T_G (i.e. T'_C) is 1-labelled.

Note that each clique of G' is associated with a complete 1–0-jump of T_{Compl}, i.e. an $A \in$ Compl, such that either A is not an m-set and Parent(A) is 0-labelled and not in Compl, or A is an m-set and Parent(A) is in Compl. We also say in that case that A is a 1–0-*jump of* Compl. Note that A is not necessarily 1-labelled in T_G.

From the above observations, we obtain the following useful result.

Corollary 50. *The cliques of G' are exactly those Q_A, such that A is a* 1–0-*jump of* Compl.

Before we determine the minimum treewidth, we determine the sizes of the Q_A.

Lemma 51. *The sizes of the sets Q_A, such that A has a* 0-*labelled parent in T_C or A is an m-set can be determined in* O(n) *time.*

Proof. If A is an m-set with parent B, then $|Q_A| = |V_A| + |V_B|$. Note that the size $|V_F|$ of the set of all vertices belonging to a fragment F can be determined in O(n) time, for all fragments F simultaneously (by recursively summing up, for each fragment F, the sizes $|V_{F'}|$ of child fragments F' of F).

Now suppose that A is not an m-set. First, for all neighborhood fragments F, we can determine the sum x_F of the sizes of the m-set children of F in O(n) time. Then for all fragments F in the m-set C, we can determine the number a_F of vertices of C that are not vertices of F and that have a 1-labelled least common ancestor with F in O(n) time. These time bounds have been proved when we determined the number of fill-in edges associated to a fragment that is not in Compl. The size of Q_A

is now

$$|V_A| + \sum_{F \text{ ancestor of } A \text{ in } T'_C} x_F + a_B + |V_{\mathrm{Parent}(C)}|.$$

Therefore the sizes of all Q_A can be determined in $\mathrm{O}(n)$ time. □

5.3. Partial strong triangulation representatives and their clique sizes

A *partial strong triangulation representative* of a fragment F is a partial triangulation representative of F satisfying the requirements of a strong triangulation representative, i.e. a partial triangulation representative Compl of F, such that for each 1-labelled descendant F' of F, if all nonm-set children are in Compl, then also F' is in Compl. Recursively we can define partial strong triangulation representatives as follows.

We assume that the fragment F has the m-set children $C_1, \ldots, C_k$ and the nonm-set children $D_1, \ldots, D_l$.

1. If F is a leaf, then $\{F\}$ is a partial strong triangulation representatives of F.
2. Let $C_1, \ldots, C_k$ be the children of F that are m-sets and $D_1, \ldots, D_l$ be the children of F that are not m-sets. Assume $\mathrm{Compl}_1, \ldots, \mathrm{Compl}_k$ are partial strong triangulation representatives of $C_1, \ldots, C_k$ and $\mathrm{Compl}'_1, \ldots, \mathrm{Compl}'_l$ are partial strong triangulation representatives of $D_1, \ldots, D_l$.

   (a) If $\mathrm{Compl}'_1, \ldots, \mathrm{Compl}'_l$ contain $D_1, \ldots, D_l$, respectively, then

$$\mathrm{Compl} := \bigcup_{i=1}^{k} \mathrm{Compl}_i \cup \bigcup_{i=1}^{l} \mathrm{Compl}'_i \cup \{F\}$$

is a partial strong triangulation representative of F (this includes "single vertex"-labelled fragments that are not leaves).

   (b) If $\mathrm{Compl}_1, \ldots, \mathrm{Compl}_k$ contain $C_1, \ldots, C_k$, respectively, and F is 0-labelled, then

$$\mathrm{Compl} := \bigcup_{i=1}^{k} \mathrm{Compl}_i \cup \bigcup_{i=1}^{l} \mathrm{Compl}'_i$$

is a partial strong triangulation representative of F.

   (c) If $\mathrm{Compl}_1, \ldots, \mathrm{Compl}_k$ contain $C_1, \ldots, C_k$, respectively, F is 1-labelled and *exactly* one D_i is not in Compl'_i, then

$$\mathrm{Compl} := \bigcup_{i=1}^{k} \mathrm{Compl}_i \cup \bigcup_{i=1}^{l} \mathrm{Compl}'_i$$

is a partial strong triangulation representative of F.

The proof is the same as the proof of Lemma 35.

For a partial strong triangulation representative Compl of F, the *clique weight* Cl(Compl) *of* Compl is the maximum $|Q_A|$, such that A is a descendant of F, the parent B of A is F or a descendant of F, and $A \in$ Compl and A is not an m-set and B is 0-labelled, or A is an m-set and B is in Compl. Note that the clique weight of a triangulation representative is $\mathrm{Cl}(R)$, where R is the (one vertex) root fragment of the fragment tree T_G. Here $\mathrm{NC}^*(F)$ is the minimum clique weight Cl(Compl) of a partial strong triangulation representative Compl of F that does not contain F and $\mathrm{Co}^*(F)$ is the minimum clique weight Cl(Compl) of a partial strong triangulation representative Compl that contains F. Let $M(F)=\min(\mathrm{Co}^*(F),\mathrm{NC}^*(F))$ be the minimum clique weight of a partial strong triangulation representative of F.

$\mathrm{NC}^*(F)$ and $\mathrm{Co}^*(F)$ can be determined recursively as follows.

Lemma 52. *Let F be a fragment with nonm-set children $D_1,\dots,D_l$ and m-set children $C_1,\dots,C_k$. Then*
1. *If F is a 1-fragment, then*

$$\mathrm{NC}^*(F)=\max\left(\min_{i=1}^{l}\max_{j\neq i}(\mathrm{NC}^*(D_i),\mathrm{Co}^*(D_j)),\max_{i=1}^{k}\mathrm{Co}^*(C_i)\right).$$

2. *If F is a 0-fragment, then*

$$\mathrm{NC}^*(F)=\max\left(\max_{i=1}^{l}\min\left(\mathrm{NC}^*(D_i),\max(|Q_{D_i}|,\mathrm{Co}^*(D_i)),\max_{i=1}^{k}\mathrm{Co}^*(C_i)\right)\right).$$

3. *If F is a 0- or 1- or "single vertex"-fragment, then*

$$\mathrm{Co}^*(F)=\max\left(\max_{i=1}^{k}\left(\min\left(\mathrm{NC}^*(C_i),\max(|Q_{C_i}|,\mathrm{Co}^*(C_i)),\max_{i=1}^{l}\mathrm{Co}^*(D_i)\right)\right)\right).$$

Proof. Case 1 is covered by case 2c of the recursive definition of a partial strong triangulation representative. We have to minimize over all cases where D_i is the nonm-set child of F that is not in Compl. Since F is not in Compl, we have to consider the case that each C_i is in Compl and not a 1–0-jump of Compl and each D_j with $j\neq i$ is in Compl. This leads to the formula in 1.

Case 2 is covered by case 2b of the recursive definition of a partial strong triangulation representative. Each D_i might be in Compl or not. If D_i is in Compl, then D_i is a 1–0-jump of Compl, and we therefore have to maximize over $|Q_{D_i}|$ and $\mathrm{Co}^*(D_i)$. To cover both cases, $D_i\in$ Compl and $D_i\notin$ Compl, we have to minimize over $\mathrm{NC}^*(D_i)$ and the maximum of $|Q_{D_i}|$ and $\mathrm{Co}^*(D_i)$. As in case 1, all C_i belong to Compl and since F is not in Compl, the C_i are no 1–0-jumps.

Case 3 is covered by case 2c of the recursive definition of a partial strong triangulation representative. Necessarily all D_i are in Compl if F is in Compl, and therefore for the D_i we have to consider the clique weights. The m-sets C_i may be in Compl or not. If C_i is in Compl, then, since F is in Compl, C_i is a 1–0-jump of Compl. Therefore we have to consider $|Q_{C_i}|$ to cover the case that $C_i\in$ Compl. In a similar way as in case 1 we get the formula for 3. □

As in the section where we determined the size of the minimum fill-in, Lemma 52 defines a recursive procedure to determine $\mathrm{NC}^*(F)$ and $\mathrm{Co}^*(F)$. For leaves F, we set $\mathrm{Co}^*(F):=1$ and for any "single vertex"-fragment F, we set $\mathrm{NC}^*(F):=\infty$, because every clique is of size at least one and a "single vertex"-labelled fragment is always in Compl.

The following lemma shows that the recursive procedure as stated in the last lemma can be done in O(n) time.

Lemma 53. *All recursions of the previous lemma can be done in* O($k+l$) *time*.

Proof. The statements 2 and 3 can be trivially done within this time bound. It remains to show this time bound for statement 1. We have to show that

$$\min_i \max_{j\neq i}(\mathrm{NC}^*(D_i), \mathrm{Co}^*(D_j))$$

can be determined within this time bound. We determine an i' such that $\mathrm{Co}^*(D_{i'})$ is maximum. If $\mathrm{NC}^*(D_{i'}) < \mathrm{Co}^*(D_{i'})$, we take this $i=i'$ for the minimum. Otherwise we take an i, such that $\mathrm{NC}^*(D_i)$ is minimum, for the minimum. □

Starting with $\mathrm{Co}^*(x)=1$ and $\mathrm{NC}^*(x)=\infty$ for leaf fragments x, we get a recursion that can be done in O(n) time. Thus we have the following result.

Theorem 54. *The treewidth of a distance hereditary graph can be determined in* O($n+m$) *time*, *and in* O(n) *time if a fragment tree is known*.

6. A further outlook

We conjecture that we can parallelize the described algorithms for min fill-in and treewidth of distance hereditary graphs. In fact, we believe that, with the knowledge of the fragment tree, we need O($n/\log n$) processors and O($\log n$) time on an EREW-PRAM. Note that distance hereditary graphs can be recognized in O($\log^2 n$) time with a linear processor number by an EREW-PRAM [9]. The fragment tree of a distance hereditary graph can be determined within the same time bound.

References

[1] S. Arnborg, D.G. Corneil, A. Proskurowski, Complexity of finding embeddings in a k-tree, SIAM J. Algebra Discrete Math. 8 (1987) 277–284.

[2] H.J. Bandelt, H.M. Mulder, Distance-hereditary graphs, J. Combin. Theory (B) 41 (1986) 182–208.

[3] H. Bodlaender, T. Kloks, D. Kratsch, H. Müller, Treewidth and minimum fill-in on d-trapezoid graphs, Technical Report RUU-CS-1995-34, Utrecht University, The Netherlands, 1995.

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

[5] H. Bodlaender, R. Möhring, The pathwidth and treewidth of cographs, Proceedings of the second Scandinavian Workshop on Algorithm Theory, Lecture Notes in Computer Science, vol. 447, Springer, Berlin, 1990, pp. 301–309.

[6] A. Brandstädt, Special graph classes — A survey, Schriftenreihe des Fachbereichs Mathematik, SM-DU-199, Universität Duisburg Gesamthochschule, 1991.

[7] H.J. Broersma, E. Dahlhaus, T. Kloks, Algorithms for the treewidth and minimum fill-in of HHD-free graphs, Memorandum No. 1356, Faculty of Applied Mathematics, University of Twente, Enschede, The Netherlands, 1996.

[8] M.S. Chang, Algorithms for maximum matching and minimum fill-in on chordal bipartite graphs, proceedings ISAAC'96, Lecture Notes in Computer Sccience, Vol. 1178, Springer, Berlin, 1996, pp. 146–155.
[9] E. Dahlhaus, Efficient parallel recognition algorithms of cographs and distance hereditary graphs, Discrete Appl. Math. 57 (1995) 29–44.
[10] G.A. Dirac, On rigid circuit graphs, Abh. Math. Sem. Univ. Hamburg 25 (1961) 71–76.
[11] D. Fulkerson, O. Gross, Incidence matrices and interval graphs, Pacific J. Math. 15 (1965) 835–855.
[12] P.L. Hammer, F. Maffray, Completely separable graphs, Discrete Appl. Math. 27 (1990) 85–99.
[13] T. Kloks, Treewidth — Computations and Approximations, Lecture Notes in Computer Science, vol. 842, Springer, Berlin, 1994.
[14] T. Kloks, Treewidth of circle graphs, Int. J. Found. Comput. Sci. 7 (1996) 111–120.
[15] T. Kloks, D. Kratsch, Treewidth of chordal bipartite graphs, J. Algorithms 19 (1995) 266–281.
[16] T. Kloks, D. Kratsch, H. Müller, Approximating the bandwidth for AT-free graphs, Proceedings of the Third Annual European Symposium on Algorithms (ESA'95), Lecture Notes in Computer Science, vol. 979, Springer, Berlin, 1995, pp. 434–447.
[17] T. Kloks, D. Kratsch, J. Spinrad, Treewidth and pathwidth of cocomparability graphs of bounded dimension, Computing Science Notes 93/46, Eindhoven University of Technology, Eindhoven, The Netherlands, 1993.
[18] T. Kloks, D. Kratsch, C.K. Wong, Minimum fill-in of circle and circular arc graphs, Proceedings of the 21st International Symposium on Automata, Languages and Programming (ICALP'96), Lecture Notes in Computer Science, vol. 1113, Springer, Berlin, 1996, pp. 256–267.
[19] F. Nicolai, Hamilton Problems on Distance Hereditary Graphs, Technical Report TR SM-DU-264, University of Duisburg, 1994.
[20] A. Parra, P. Scheffler, How to use minimal separators for its chordal triangulation, Proceedings of the 20th International Symposium on Automata, Languages and Programming (ICALP'95), Lecture Notes in Computer Science, vol. 944, Springer, Berlin, 1995, pp. 123–134.
[21] D. Rose, Triangulated graphs and the elimination process, J. Math. Analysis and Appl. 32 (1970) 597–609.
[22] R. Sundaran, K. Sher Singh, C. Pandu Rangan, Treewidth of circular-arc graphs, Siam J. Disc. Math. 7 (1994) 647– 655.
[23] M. Yannakakis, Computing the minimum fill-in is NP-complete, Siam J. Alg. Disc. Math. 2 (1981) 77–79.