

On co-distance hereditary graphs

S. Dubois^a V. Giakoumakis^{a,*} and C.B. Ould El Mounir^a

^aMIS, Université d'Amiens, France

Abstract

We present a linear time recognition algorithm as well as a 4-expression for calculating the clique-width for the co-distance hereditary graphs which is the complementary class of the well known family of distance hereditary graphs.

Key words: Distance and co-distance hereditary graphs, split decomposition, clique-width, linear recognition algorithm

1 On distance hereditary graphs

For terms not defined here the reader is referred to [1]. Given a graph $G = (V, E)$, V will denote its vertex set, E its edge set and $N(x)$ the neighborhood of $x \in V$. A vertex having exactly one neighbor is called a *pendant* vertex. Two vertices x and y are *twins* iff $N(x) = N(y)$, they are *true twins* iff $(xy) \in E$ and *false twins* iff $(xy) \notin E$. The *distance* between two vertices x and y , denoted by $d_G(x, y)$, is the length of a shortest path between x and y . The class of Distance Hereditary (DH for shortly) graphs have been widely studied and many results have been obtained for these graphs (see [1]). Among them we recall that DH graphs are totally decomposable using *split decomposition*. We recall also that DH graphs are also known as HHGD-free graphs since they can be characterized by four forbidden configurations: the House (i.e. the complementary graph of a chordless chain of 5 vertices or P_5), the Hole (i.e. a chordless cycle of at least five vertices), the Domino (i.e. a cycle of 6 vertices $abcdef$ having exactly one chord cf) and the Gem (i.e. the graph formed by a $P_4 = abcd$ and a universal vertex w.r.t. this P_4). Finally, a graph G is distance hereditary iff for any connected subgraph H of G $d_G(x, y) = d_H(x, y)$ holds for every pair of vertices of H . A *pruning sequence* (S, σ) of G is a total ordering $\sigma[x_1, \dots, x_n]$ of its vertex set and a sequence $S[s_1, \dots, s_n]$ of triples, such that for $1 \leq i \leq n - 1$ and $i < j$, in the induced subgraph G_i of $G[V \setminus \{x_1, \dots, x_{i-1}\}]$, s_i is one of the following words: (x_i, P, x_j) , if $N(x_i) = \{x_j\}$ (x_j is a pendant vertex) or (x_i, F, x_j) (x_i and x_j are false twins) or (x_i, T, x_j) (x_i and x_j are true twins). The pruning sequence is used for the recognition of a DH graph G : starting from a vertex of G we construct successively subgraphs of G by adding true twins, false twins or pendant vertices. In [5] was

* Corresponding author.

proposed a linear time recognition algorithm for DH graphs by constructing the corresponding pruning sequence. But this algorithm recognizes also the domino and the house as DH graphs, problem that was resolved in [3]. The recognition algorithm of DH graphs presented in [3] will be for us the framework for the recognition of a co-distance hereditary graph G . We shall show in the next section that for testing if \overline{G} is distance hereditary we do not need to compute \overline{G} but we can work on G and make the necessary transformations to this algorithm in order to remain in linear time on the size of G . We give below the recognition algorithm in [3] and we enumerate the 6 steps needed these transformations.

Algorithm 1 ([3]). The pruning sequence of a connected DH graph

Build the distance levels $L_v = L_1, \dots, L_k$ from a vertex v of G **(1)**; $j \leftarrow 1$;
For $i = k$ **downto** 1 **do**
 For every connected component C of $G[L_i]$ **(2) Do**
 $z \leftarrow \text{Prune-cograph}(C, j)$ **(3)**; $j \leftarrow j + |C| - 1$; **End_For**
 Sort the vertices of L_i by increasing inner degree **(4)**
 For every vertex x of L_i having exactly one neighbor y **(5) do**
 $\sigma(j) \leftarrow y$ and $s_j \leftarrow (xPy)$; $j \leftarrow j + 1$; **End_For**
 For every vertex $x \in L_i$ taken in increasing inner degree order **Do**;
 $y \leftarrow \text{Prune-cograph}(G[N_{i-1}(x)], j)$ **(6)**; $j \leftarrow j + |N_{i-1}(x)| - 1$; $\sigma(j) \leftarrow x$
 and $s_j \leftarrow (xPy)$; $j \leftarrow j + 1$; **End_For**
End_For

Let us explain some terms used in the above algorithm. Let G be a connected graph and v be a vertex of G . A distance level L_v in G is the set L_1, \dots, L_k of vertices of G such that $x \in L_i$ if $d_G(v, x) = i$. For every vertex x of G and for every integer i such that $1 \leq i \leq k$, we denote by $N_i(x) = N(x) \cap L_i$. The inner degree of x is the cardinality of $N_i(x)$. The algorithm *Prune-cograph* (C, j) constructs the pruning sequence (S, σ) of the cograph corresponding to the connected component C and contracts C to the last vertex z of σ . We must point out that *Prune-cograph* (C, j) works on the cotree $T(C)$ corresponding to the cograph C and by [2] a cotree can be obtained in linear time on the size of the considered cograph.

2 Linear recognition of co-distance hereditary graphs

We shall show now how the recognition algorithm of DH graphs presented in previous section can be transformed in order to recognize a co-distance hereditary (co-DH for shortly) graph G in linear time on the size of G . We assume that G is a connected graph as well as \overline{G} . If \overline{G} is not connected we shall work in each connected component of \overline{G} . Let us explain now how we can process the 6 steps in algorithm 1 in \overline{G} using the edges of the graph G .

Step 1. Algorithm 2 : constructing all distance levels L_v of a connected component of \overline{G}

Input : A graph G with n vertices, a list $L = \{1, \dots, n\}$ of all vertices of G and an array $\text{index}[1..n]$ such that $\text{index}[i] = 0$ for all i .

Output : The set $L_v = \{L_1, \dots, L_k\}$ of distance levels from a vertex v of a connected component of \overline{G} .

$i := 0$; Pick an arbitrary vertex x of L , $L_i := x$ and delete it from L .

While L is non empty **Do**

[For every vertex $y \in L_i$, and for every vertex $z \in N(y) \cap L$ increase by 1 $\text{index}[z]$. If $\text{index}[z] = |L_i|$ move z from L to L_{i+2} and put $\text{index}[z] = 0$];
 If $L = \emptyset$ then **exit**; $\{L_0 \cup \dots \cup L_i$ is a connected component of \overline{G} $\}$
 $L_{i+1} := L$; $i := i + 1$; $L := L_{i+1}$

End_While

Complexity of Step 1. When constructing L we construct also an array $A[1..n]$ such that $A[i]$ contains a direct access in the list containing i during the execution of the above algorithm. Hence, we can find in constant time the neighbors of every vertex $y \in L_i$. It is easy to see that the complexity of the above algorithm is linear on the size of G .

Steps 2,3. Since $G[L_i]$ must be a cograph we check this by obtaining in linear time the corresponding cotree T by using the algorithm in [2]. Then we obtain the corresponding cotree as well as the connected components of $\overline{G[L_i]}$ by changing the 0-nodes of T into 1-nodes and its 1-nodes into 0-nodes.

Steps 4,5. We shall sort the vertices of L_i by decreasing inner degree, y will be the vertex whose inner degree will be $|L_{i-1}| - 1$.

Steps 6. Once the vertices of L_i have been sorted by decreasing inner degree, using the array A we find first the non-neighborhood in L_{i-1} of each vertex x of L_i within $O(\text{degree}(x))$ complexity and then proceed in an analogous manner described on Steps 2 and 3 above.

It is clear now that we can apply the Algorithm 1 in \overline{G} within linear time complexity on the size of G . It remains a last verification presented in [3] that consists to check if the obtained pruning sequence (S, σ) corresponds to an HD graph. Due to the space limitations of this extended abstract, we leave to the reader to verify that this can be done in linear time on the size of G .

2 Clique width of co-distance hereditary graphs

The well known notion of *clique-width* of a graph G denoted $cwd(G)$, is the minimum number of labels needed for constructing G using four graph operations: labeling by i a new vertex v (denoted $i(v)$), disjoint union of H and H' denoted $H \oplus H'$, $\eta_{i,j}(G)$, $i \neq j$, is the graph obtained by connecting all the vertices labeled i to all the vertices labeled j in G and $\rho_{i,j}(G)$ the graph obtained by renaming i into j in G . An expression obtained from the above four operations using k labels is called a k -*expression*. We denote by $G(t)$ a graph defined by the expression t . In [4], it is proved that every distance hereditary graph, has clique-width at most 3 and a 3-expression defining it can be obtained in linear time. This expression is constructed as follows: from the pruning sequence (S, σ) associated with a DH graph G we construct a special tree $T(G)$, the *pruning tree*, whose vertices are the vertices of G and whose edges $\{x, y\}$ are labeled l , F or T if there exists s_i in σ such that s_i is (xlf) , (xFy) or (xTy) respectively. Let α be a node of $T(G)$, T_α is the set of vertices of G of the sub-tree rooted at α and $G(T_\alpha)$ the subgraph of G induced by the vertices of T_α . Let u and v be two vertices of T_α , then u is a twin descendant of v if all the edges connected u to v are labeled with true or false. Let $\alpha_1, \dots, \alpha_l$ be the sons of α ordered from left to right. In [4] it is proved that for every α_i the set of edges relying the vertices of $G(T_{\alpha_i})$ and $G(\alpha \cup T_{\alpha_{i+1}} \cup \dots \cup T_{\alpha_l})$ is empty whenever α_i is a false twin son of α and it is formed by all $\{u, v\}$ where

u is a twin descendant of α_i in T_{α_i} and v is either α or a twin descendant of α in $T_{\alpha_{i+1}} \cup \dots \cup T_{\alpha_1}$. This result allowed to obtain a 3 – expression for a distance hereditary graph by labeling the twin descendants of any node β in T_β with 2 and by 1 all the other vertices of T_β . Using this labeling for the vertices of G and using the pruning tree of \overline{G} we can obtain a 4 – expression for G in linear time. For this we shall calculate the expression e_i associated with $G(T_{\alpha_i} \cup T_{\alpha_{i+1}} \cup \dots \cup T_{\alpha_1})$ by assuming that we know the 3 – expression t_{α_i} associated with $G(T_{\alpha_i})$ and the 3 – expression e_{i+1} associated with $G(T_{\alpha_{i+1}} \cup \dots \cup T_{\alpha_1})$. We then have:

1. If a_i is a leaf son of α then

$$e_i = \rho_{4 \rightarrow 1}(\rho_{3 \rightarrow 1}(\eta_{1,4}(\eta_{1,3}(\eta_{2,3}(e_{i+1} \oplus (\rho_{2 \rightarrow 4}(\rho_{1 \rightarrow 3}(t_{\alpha_i}))))))))))$$

2. If a_i is a true twin son of α then $e_i = \rho_{4 \rightarrow 2}(\rho_{3 \rightarrow 1}(\eta_{1,4}(\eta_{1,3}(\eta_{2,3}(e_{i+1} \oplus (\rho_{2 \rightarrow 4}(\rho_{1 \rightarrow 3}(t_{\alpha_i}))))))))))$

3. If a_i is a false twin son of α then $e_i = \rho_{4 \rightarrow 2}(\rho_{3 \rightarrow 1}(\eta_{1,4}(\eta_{1,3}(\eta_{2,4}(\eta_{2,3}(e_{i+1} \oplus (\rho_{2 \rightarrow 4}(\rho_{1 \rightarrow 3}(t_{\alpha_i}))))))))))$.

It is now to see how we can obtain a 4 – expression for G in linear time on the size of G . It follows that many optimization problems have linear solution for co-DH graphs (see [4]).

References

- [1] A. Brandstädt, V.B. Le and J. Spinrad, Graph classes: a survey, *SIAM Monographs on Disc. Math. and Applications*, (1999).
- [2] D.G. Corneil, Y. Perl, L.K. Stewart, A linear recognition for cographs, *SIAM J. comput.* 14(4), (1985), 926-934.
- [3] G. Damiand, M. Habib, C. Paul, A simple paradigm for graph recognition : Application to cographs and distance hereditary graphs. *Theoretical Computer Sciences*, 263, (2001), 99-111.
- [4] M.C. Golumbic, U. Rotics : On the clique-width of some perfect graph classes, *Int. J. Found. Comput. Sci.* 11(3), (2000), 423-443 .
- [5] P.L. Hammer, F. Maffray, Completely separable graphs, *Discrete Appl. Math.* 27, (1990), 85-99.