



HAL
open science

Recognition of chordal graphs and cographs which are Cover-Incomparability graphs

Arun Anil, Manoj Changat

► **To cite this version:**

Arun Anil, Manoj Changat. Recognition of chordal graphs and cographs which are Cover-Incomparability graphs. 2023. ⟨hal-04169334⟩

HAL Id: hal-04169334

<https://hal.science/hal-04169334v1>

Preprint submitted on 24 Jul 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

Recognition of chordal graphs and cographs which are Cover-Incomparability graphs

Arun Anil¹ Manoj Changat¹

July 24, 2023

¹ Department of Futures Studies, University of Kerala, Thiruvananthapuram - 695581, India.

E-mails: arunanil93@gmail.com, mchangat@keralauniversity.ac.in

Abstract

Cover-Incomparability graphs (C-I graphs) are an interesting class of graphs from posets. A C-I graph is a graph from a poset $P = (V, \leq)$ with vertex set V , and the edge-set is the union of edge sets of the cover graph and the incomparability graph of the poset. The recognition of the C-I graphs is known to be NP-complete (Maxová et al., Order 26(3), 229–236(2009)). In this paper, we prove that chordal graphs having at most two independent simplicial vertices are exactly the chordal graphs which are also C-I graphs. A similar result is obtained for cographs as well. Using the structural results of these graphs, we derive linear time recognition algorithms for chordal graphs and cographs, which are C-I graphs.

Key Words: Cover-Incomparability graphs; Chordal graph; Cographs; Recognition Algorithm.

1 Introduction

Chordal graphs or triangulated graphs are graphs that do not have induced cycles of length greater than three, and cographs which are P_4 -free graphs form two well-studied graphs that have wider applications beyond mathematics. Special classes of these two graphs are also studied in many contexts, including theoretical, practical, and algorithmic interests. A vertex v of a graph G is a simplicial vertex if v together with all its adjacent vertices form a clique (a complete subgraph) in G . Simplicial vertices and simplicial or perfect elimination play a crucial role in chordal graphs. It is a well-known fact that every chordal graph contains at least one simplicial vertex, and a noncomplete chordal graph contains at least two nonadjacent simplicial vertices. Chordal graphs that contain exactly two non-adjacent simplicial vertices form a special class of chordal graphs. In a similar manner, cographs that contain exactly two non-adjacent simplicial vertices are a proper subclass of

35 the above described family of chordal graphs and the join of cographs having ex-
 36 actly two non-adjacent simplicial vertices form a special class of cographs. In this
 37 paper, we identify chordal graphs and cographs having exactly two non-adjacent
 38 simplicial vertices and cographs obtained by join of chordal cographs having ex-
 39 actly two simplicial vertices. Interestingly, these two special classes of graphs have
 40 been found to belong to a particular class of graphs that can be derivable from a
 41 partially ordered set or poset. This class of graphs obtained from posets is known
 42 as the cover-incomparability graphs of posets, or shortly C-I graphs. These graphs
 43 were introduced in [6] as the underlying graphs of the so-called standard transit
 44 function of posets. The C-I graphs are precisely the graphs whose edge set is the
 45 union of edge sets of the cover graph and the incomparability graph (complement of
 46 a comparability graph) of a poset. Graph-theoretic characterization of C-I graphs
 47 has not been known until now and, moreover, the recognition complexity of C-I
 48 graphs is NP-complete (Maxová et al. [24]). Hence, problems in C-I graphs are fo-
 49 cused on identifying the structure and characterization of well-known graph families,
 50 which are C-I graphs. Such C-I graphs studied include the family of split graphs,
 51 block graphs [7], cographs [8], Ptolemaic graphs [22], distance hereditary graphs
 52 [22], and k -trees [23]. The C-I graphs were identified among the planar and chordal
 53 graphs along with new characterizations of the Ptolemaic graphs, respectively, in [4]
 54 and [3]. It is also interesting to note that every C-I graph has a Ptolemaic C-I graph
 55 as a spanning subgraph [4]. C-I graphs are also studied among the comparability
 56 graphs [2]. The effect of composition operation, lexicographic, and strong products
 57 of C-I graphs was studied in a recent paper [1].

58 In this paper, we obtain that chordal graphs having at most two independent
 59 simplicial vertices are exactly the chordal graphs which are also C-I graphs. A similar
 60 result is obtained for cographs as those cographs which are join of chordal cographs
 61 having exactly two independent simplicial vertices are precisely cographs which are
 62 also C-I graphs. Using the structural results of these graphs, we derive linear time
 63 recognition algorithms for have for both classes. The recognition algorithm for an
 64 arbitrary chordal graph [29, 26] and a cograph [13] is well known and has linear-time
 65 complexity.

66 Chordal graphs are precisely the intersection graphs of subtrees of a tree [19].
 67 Most of the information contained in a chordal graph is captured in its clique tree
 68 representation, which is useful for its algorithmic applications [19, 28].

69 Cographs are exactly the P_4 free graphs and the class of cographs has been in-
 70 tensively studied since its definition by Seinsche [27]. The cographs appear as com-
 71 parability graphs of series-parallel partial orders [21], and can be generated from the
 72 single-vertex graph K_1 by complementation and disjoint union operations. It is well
 73 known that any cograph has a canonical tree representation called a cotree. This
 74 tree decomposition scheme for cographs is a particular case of modular decomposi-
 75 tion [18] that applies to arbitrary graphs. Indeed, the algorithm that computes the
 76 modular tree decomposition of an arbitrary graph in linear time can also recognize
 77 cographs in linear time. In 1994, linear time modular decomposition algorithms were
 78 designed independently by Cournier and Habib [14] and by McConnell and Spinrad
 79 [25]. In 2001, Dahlhaus et al. [15] proposed a simpler algorithm. Unfortunately,
 80 because they build the decomposition tree, all these algorithms are complicated or

81 need to maintain complicated data structures. In 2004, Habib and Paul [20] pro-
 82 posed a new algorithm, which is not incremental, and instead of building the cotree
 83 directly, it first computes a special ordering of the vertices, namely a factorizing per-
 84 mutation, using a very efficient partition refinement techniques via two elementary
 85 refinement rules.

86 A search of a graph visits all vertices and edges of the graph and will visit a new
 87 vertex only if it is adjacent to some previously visited vertex. The two fundamental
 88 search strategies are Breadth-First Search (BFS) and Depth-First Search (DFS). As
 89 the names indicate, BFS visits all previously unvisited neighbors of the currently
 90 visited vertex before visiting the previously unvisited non-neighbors. Several greedy
 91 recognition algorithms for chordal graphs are known. The most famous is Lex-BFS
 92 [26], a variant of *BFS*, introduced by Rose et al. in [26] and Maximum Cardinality
 93 Search (MCS for short) [29]. Both algorithms are linear. The recognition of chordal
 94 graphs involves two distinct phases: the execution of MCS or Lex-BFS in order to
 95 compute an elimination ordering and a checking procedure to decide whether this
 96 elimination ordering is perfect (PEO). By employing Lex-BFS as a basic method
 97 to check the chordality and to find the perfect elimination ordering, the recognition
 98 algorithm for the chordal graphs can be done in linear time for chordal graphs which
 99 are C-I graphs. For cographs, using the BFS, we can check whether a given rooted
 100 tree is the cotree of the C-I cograph in linear time because of the special structure
 101 of the cotrees of these graphs.

102 The rest of this section is organized as follows. In Section 2, we begin with
 103 some preliminaries. In Section 3, we characterize chordal C-I graphs in terms of
 104 the number of independent simplicial vertices and present a linear-time algorithm
 105 for recognizing chordal C-I graphs based on this characterization. In Section 4, we
 106 present a structure of the C-I cograph and characterize the structure of the cotree
 107 of C-I cographs and, using this, present a linear time recognition algorithm for C-I
 108 cographs.

109 2 Preliminaries

110 Some preliminary definitions and results that were used in this paper are discussed
 111 in this section.

112 A *partially ordered set* or *poset* $P = (V, \leq)$ consists of a nonempty set V and a
 113 reflexive, antisymmetric, transitive relation \leq on V , denoted as $P = (V, \leq)$, we call
 114 $u \in V$ an element of P . If $u \leq v$ or $v \leq u$ in P , we say u and v are *comparable*,
 115 otherwise *incomparable*. If $u \leq v$ but $u \neq v$, then we write $u < v$. If u and v are in
 116 V , then v *covers* u in P if $u < v$ and there is no w in V with $u < w < v$, denoted
 117 by $u \triangleleft v$. We write $u \triangleleft \triangleleft v$ if $u < v$ but not $u \triangleleft v$. By $u || v$, we mean that u and
 118 v are incomparable elements of P . Let $V' \subseteq V$ and $Q = (V', \leq')$ be a poset, Q is
 119 called a *subposet* of P , if $u \leq' v$ if and only if $u \leq v$, for any $u, v \in V'$. The subposet
 120 $Q = (V', \leq)$ is a *chain* (*antichain*) in P , if every pair of elements of V' is comparable
 121 (incomparable) in P . A chain of maximum cardinality is named as the *height* of P
 122 denoted as $h(P)$. An element u in P is a *minimal* (*maximal*) if there is no $x \in V$
 123 such that $x \leq u$ ($x \geq u$) in P . A finite *ranked poset* (also known as *graded poset* [10])
 124 is a poset $P = (V, \leq)$ that is equipped with a rank function $\rho : V \rightarrow \mathbb{Z}$ satisfying:

- 125 • ρ has value 0 on all minimal elements of P , and
- 126 • ρ preserves covering relations: if $a \triangleleft b$ then $\rho(b) = \rho(a) + 1$.

127 A ranked poset P is said to be *complete* if for every i , every element of rank i covers
 128 all elements of rank $i - 1$. For a completely ranked poset $P = (V, \leq)$ we say that the
 129 element $v \in V$ is at height i if $\rho(v) = i - 1$. We refer to [10], for notions of posets.

130 Let $G = (V, E)$ be a connected graph, vertex set and edge set of G denoted as
 131 $V(G)$ and $E(G)$ respectively, the complement of G is denoted as \overline{G} . For a vertex
 132 $v \in V(G)$, the set of all vertices adjacent to v is called the *open neighborhood* of
 133 v and is denoted by $N(v)$. The set consisting of the open neighborhood and the
 134 vertex v is the *closed neighborhood* of v and is denoted by $N[v]$. A *tree* is a graph in
 135 which two vertices are connected by exactly one path. Let T be a rooted tree and
 136 two vertices x and y in T , we say that x is an ancestor of y and y is a descendant
 137 of x if x lies on the path from y to the root of T . For a set of leaves S of T , we say
 138 that the *lowest common ancestor (LCA)* of S is the internal node v of T such that
 139 v is the root of the smallest rooted subtree of T containing S . A graph H is said to
 140 be a *subgraph* of G if $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$. H is an *induced subgraph*
 141 of G if for $u, v \in V(H)$ and $uv \in E(G)$ implies $uv \in E(H)$. A graph G is said to be
 142 *H-free* if G has no induced subgraph isomorphic to H . A *complete graph* is a graph
 143 whose vertices are pairwise adjacent, denoted K_n , a set $S \subseteq V(G)$ is a *clique* if the
 144 subgraph of G induced by S is a complete graph, and a *maximum clique* is a clique
 145 that is not contained by any other clique. A vertex v is called *simplicial vertex* if
 146 its neighborhood induces a complete subgraph. An *independent set* in a graph is
 147 a set of pairwise non-adjacent vertices. If graphs G_1 and G_2 have disjoint vertex
 148 set V_1 and V_2 and edge set E_1 and E_2 respectively, then their *union* $G = G_1 \cup G_2$
 149 has $V = V_1 \cup V_2$ and $E = E_1 \cup E_2$ and their *join*, denoted by $G_1 \vee G_2$, consists of
 150 $G = G_1 \cup G_2$ and all edges joining V_1 with V_2 .

151 A graph G is *chordal* if it contains no induced cycles of length more than 3, it is
 152 *distance-hereditary* if every induced path is also the shortest path in G . A graph G
 153 is *Ptolemaic* if it is distance-hereditary and chordal. Equivalently, G is Ptolemaic if
 154 and only if it is a 3-fan-free chordal graph. P_4 -free graphs are called *cographs*. A
 155 graph G that is both chordal and cograph is called *chordal cograph*. Finally, the
 156 cover-incomparability graph (C-I graph) of a poset $P = (V, \leq)$ denoted as G_P is the
 157 graph $G = (V, E)$, where $uv \in E(G)$, if $u \triangleleft v$ or $v \triangleleft u$ or $u \parallel v$ in P . A graph is a C-I
 158 graph if it is the C-I graph of some poset P . We call a graph that is both a chordal
 159 and a C-I graph as a *chordal C-I graph*. Similarly, we use the term *ptolemaic*
 160 *C-I graph*, *C-I cograph*, *chordal C-I cograph*, etc. to denote the C-I graph which is
 161 Ptolemaic, cograph, chordal cograph, etc.

162 Now we recall some basic properties of posets and their C-I graphs.

163 **Lemma 1.** [6] *Let P be a poset. Then*

- 164 (i) *the C-I graph of P is connected;*
- 165 (ii) *points of P that are independent in the C-I graph of P lie on a common chain;*
- 166 (iii) *an antichain of P corresponds to a complete subgraph in the C-I graph of P ;*

167 (iv) the C-I graph of P contains no induced cycles of length greater than 4.

168 **Lemma 2.** [22] *If G is a C-I graph, then G does not contain 3 independent simplicial*
169 *vertices.*

170 **Lemma 3.** [22] *Let P be a poset and G its C-I graph. If v is a simplicial vertex in*
171 *G , then v is a maximal or a minimal element of P .*

172 **Theorem 1.** [23] *Let $G = (V, E)$ be a C-I graph of a poset P and $v \in V$ a minimal*
173 *or maximal element in P . Then $G \setminus v$ is also a C-I graph.*

174 In the following sections, we discuss algorithms for recognizing chordal C-I graphs
175 and C-I cographs.

176 3 Structure of chordal C-I graphs

177 In this section, we present an algorithm to recognize chordal C-I graphs. For that,
178 first, we prove a characterization of a chordal graph that is a C-I graph.

179 If v is a simplicial vertex of a chordal graph G , then the closed neighborhood
180 $N[v]$ is a clique in G . So removal of v does not affect the chordal property of $G \setminus \{v\}$.
181 This is stated in the following remark.

182 **Remark 1.** *If G is a chordal graph and v is a simplicial vertex, then $G \setminus v$ is chordal.*

183 **Lemma 4.** *Let G be a chordal graph, and v be a simplicial vertex of G . If G has*
184 *exactly two independent simplicial vertices, then either $G \setminus v$ is chordal and has*
185 *exactly two independent simplicial vertices or $G \setminus v$ is a complete graph.*

186 *Proof.* Let G be a chordal graph with exactly two independent simplicial vertices
187 and v be a simplicial vertex. Then by Remark 1, $G \setminus v$ is a chordal graph. Since
188 v is a simplicial vertex, the removal of v from G does not increase the number of
189 independent simplicial vertices in $G \setminus v$. And if $G \setminus v$ is not a complete graph, then
190 removal of v from G does not decrease the number of independent simplicial vertex
191 in $G \setminus v$. That is, $G \setminus v$ has exactly two independent simplicial vertices, otherwise
192 $G \setminus v$ is a complete graph. Hence the lemma. \square

193 From Lemmas 3-4 and Theorem 1, we get the following corollary.

194 **Corollary 1.** *Let G be a chordal C-I graph and v be a simplicial vertex of G . If*
195 *G has exactly two independent simplicial vertices, then $G \setminus v$ is a chordal C-I graph*
196 *and has exactly two independent simplicial vertices or $G \setminus v$ is a complete graph.*

197 A *pendant clique* of a graph G is a clique that contains a clique separator C such
198 that one of the components obtained after removing C is a single vertex. Observe
199 that if G is a chordal graph, then G has a pendant clique.

200 The following remark is immediate.

201 **Remark 2.** Let \mathcal{G} be the family of graphs obtained by adding a vertex v to a chordal
 202 graph G that has at most two independent simplicial vertices, in such a way that v
 203 is adjacent to some or all vertices in a pendant clique of G . Let G' be a chordal
 204 graph having exactly two independent simplicial vertices. If G is the graph obtained
 205 by adding a simplicial vertex v to G' such that the resulting graph has exactly two
 206 independent simplicial vertices, then G belongs to the family \mathcal{G} .

207 **Lemma 5.** Let G' be a chordal C-I graph having exactly two independent simplicial
 208 vertices, and let G be a graph obtained by adding a simplicial vertex v to G' . If G
 209 has exactly two independent simplicial vertices, then G is a chordal C-I graph.

210 *Proof.* Let G be a chordal graph obtained by adding a simplicial vertex v to a chordal
 211 G' that has exactly two independent simplicial vertices such that G also has exactly
 212 two independent simplicial vertices. Then by Remark 2, the vertex v is such that v
 213 is adjacent to some or all vertices in a pendant clique in G' .

214 Now we need to prove that G is a chordal C-I graph. Since G' is both a C-I graph
 215 and chordal, let P' be a poset such that $G_{P'} \cong G'$. Since the vertex v is added to
 216 some or all vertices in a pendant clique in G' , the graph G is chordal. Since G'_P
 217 is chordal and has exactly two independent simplicial vertices, G'_P has exactly two
 218 pendant cliques, and the two pendent cliques are formed, respectively, by S and S' ,
 219 where S and S' are defined as follows.

220 $M = \{u \in P' \mid u \text{ is a maximal element of } P'\}$, $S_1 = \{w \in P' \mid w \triangleleft u, u \in M\}$,
 221 $M' = \{u \in P' \mid u \text{ is a minimal element of } P'\}$, $S'_1 = \{w \in P' \mid u \triangleleft w, u \in M'\}$
 222 $S = M \cup S_1 \setminus \{u \in S_1 \mid \exists w \in S_1, u \triangleleft \triangleleft w \text{ in } P'\}$ and
 223 $S' = M' \cup S'_1 \setminus \{u \in S'_1 \mid \exists w \in S'_1, w \triangleleft \triangleleft u \text{ in } P'\}$. Now v is adjacent to some or
 224 all vertices of the set S or S' in G . If v is adjacent to some vertices of S , then v
 225 must be adjacent to all the elements of M , as otherwise there will be more than
 226 two independent simplicial vertices. Similarly, if v is adjacent to some vertices of S' ,
 227 then v must be adjacent to all elements of M' . Now, we construct a poset P from
 228 P' as follows.

229 If v adjacent to some vertices of S :

- 230 • If v is adjacent to only the elements in M , then P is constructed from P' with
 231 the covering relation defined as $u \triangleleft v$ for all $u \in M$.
- 232 • If v is adjacent to the elements of S_0 , where $M \subset S_0 \subset S$, then P is constructed
 233 from P' with the covering relation as $u' \triangleleft v$ for all $u' \in S_0 \setminus M$ and $u'' \triangleleft u \triangleleft v$
 234 for all $u'' \in S \setminus S_0$ for some $u \in M$.
- 235 • If v is adjacent to all elements of S , then P is constructed from P' with the
 236 covering relation defined as $u \triangleleft v$ for all $u \in S \setminus M$.

237 Similarly, if v adjacent to some vertices of S' :

- 238 • if v is adjacent to only the elements in M' then P is constructed from P' with
 239 the covering relation defined as $v \triangleleft u$ for all $u \in M'$.
- 240 • if v is adjacent to the elements of S'_0 , where $M' \subset S'_0 \subset S'$, then P is con-
 241 structed with the covering relation defined as $v \triangleleft u'$ for all $u' \in S'_0 \setminus M'$ and
 242 $v \triangleleft u \triangleleft u''$ for all $u'' \in S' \setminus S'_0$ and some $u \in M'$.

243 • If v is adjacent to all elements of S' , then P is constructed from P' with the
 244 covering relation $v \triangleleft u$ for all $u \in S' \setminus M'$.

245 It follows from construction that P is a well-defined poset and that G is isomorphic
 246 to G_P . □

247 A *perfect elimination ordering (PEO)* is an ordering $\pi = v_1, \dots, v_n$ of vertices
 248 in G such that the neighborhood $N[v_i]$ of v_i is a clique of the subgraph $G_{\{v_i, \dots, v_n\}}$
 249 induced by the vertices $\{v_i, \dots, v_n\}$ of G . The following characterizations of chordal
 250 graphs are well known.

251 **Theorem 2.** [17] *A graph G is chordal if and only if G has a perfect elimination*
 252 *ordering.*

253 **Lemma 6.** [16] *Every chordal graph G has a simplicial vertex. If G is not complete,*
 254 *then it has two non-adjacent simplicial vertices.*

255 In the following, we prove an interesting characterization of C-I chordal graphs
 256 as precisely those chordal graphs having exactly two independent simplicial vertices.

257 **Theorem 3.** *Let G be a chordal graph. Then G is a C-I graph if and only if G is*
 258 *a complete graph or G has exactly two independent simplicial vertices.*

259 *Proof.* If G is a chordal and CI graph, then G is a complete graph or G contains two
 260 independent simplicial vertices by Lemma 6. Also, every complete graph is a C-I
 261 graph. It remains to prove that if G is chordal and G has exactly two independent
 262 simplicial vertices, then G is a C-I graph.

263 Let G be the chordal graph with the vertex set $V(G) = \{v_1, v_2, v_3, \dots, v_n\}$. Since
 264 G is chordal, there is a perfect elimination ordering (PEO), let $v_n, v_{n-1}, \dots, v_2, v_1$ be a
 265 PEO in G . By the definition of PEO on a chordal graph, the neighborhood $N(v_{n-i})$
 266 of v_{n-i} is a clique in the subgraph $G_{\{v_{n-i+1}, \dots, v_1\}}$, for $i = 0, 1, \dots, n-1$ and also the
 267 subgraphs $G_{\{v_{n-(i+1)}, \dots, v_1\}}$ are chordal for $i = 0, 1, \dots, n-1$.

268 We eliminate the simplicial vertices until we get the smallest non-trivial chordal
 269 graph containing exactly two independent simplicial vertices, say G_k . The removal of
 270 a simplicial vertex from G_k results in a complete graph. That is, G_{k-1} is a complete
 271 graph with $G_{k-1} = G_k \setminus v_k$, where v_k is a simplicial vertex in G_k . Clearly, G_{k-1} is a
 272 C-I graph.

273 Now, we add the simplicial vertices to the graphs $G_{k-1}, G_k, \dots, G_{n-1}$, in reverse
 274 order in the PEO; that is, we add vertices in order v_k, v_{k+1}, \dots, v_n and consider all
 275 the possible ways of adding edges to build G . We add edges so that the resulting
 276 graph is obtained in each stage such that the graphs contain only two independent
 277 simplicial vertices.

278 In particular, the graph G_k is obtained by adding the simplicial vertex v_k to
 279 G_{k-1} . The vertex v_k is adjacent to some vertices of G_{k-1} . Let $C_{k-1} \subset V(G_{k-1})$ and
 280 $C'_{k-1} = V(G_{k-1}) \setminus C_{k-1}$ be such that v_k is adjacent to every element of C_{k-1} and v_k
 281 is not adjacent to any element of C'_{k-1} . It follows by construction that the graph
 282 G_k is a Ptolemaic C-I graph with height 3 so that G_k is a chordal C-I graph having
 283 exactly two simplicial vertices.

284 Now, the graph G_{k+1} is obtained by adding the simplicial vertex v_{k+1} to G_k
 285 and G_{k+2} is obtained by adding the simplicial vertex v_{k+2} to G_{k+1} and continuing
 286 as this, we obtain G_n by adding the simplicial vertex v_n to G_{n-1} . By the lemma
 287 5 and Corollary 1, $G_{k+1}, G_{k+2}, \dots, G_{n-1}$ are chordal C-I graphs with exactly two
 288 independent simplicial vertices.

289 Finally, we see that the graph G_n is such that $G \cong G_n$ and is a C-I graph.

290

□

291 It is well known that a chordal graph G can be characterized as the intersection
 292 graph of sub-trees of some tree T . Such a tree is known as the clique tree of G .
 293 This well-known theorem is proved independently by Buneman [11] and Gavril [19].
 294 A tree representation of a chordal graph G is a pair (T, \mathcal{F}) where T is a tree and
 295 \mathcal{F} is a family of subtrees of T such that the intersection graph of \mathcal{F} is isomorphic
 296 to G . Further Gavril [19] has shown that given a chordal graph G , it is possible to
 297 construct a tree T with vertex set $K = \{q_1, q_2, \dots, q_r\}$ where q_i corresponds to the
 298 maximal clique Q_i of G , such that $(T, \{R_{v_1}, R_{v_2}, \dots, R_{v_n}\})$ is a tree representation of
 299 G . Here, each R_{v_i} , $1 \leq i \leq n$, is the set of maximal cliques that contain the vertex
 300 v_i , that is, $R_{v_i} = \{q_j \mid v_i \in Q_j\}$. Such a tree representation of G is called a *clique*
 301 *tree* of G . The clique tree T of G need not be unique, for the sets R_i determine the
 302 edges of T , not necessarily in a unique manner. Fig.1 shows a chordal graph and
 303 two of its clique trees.

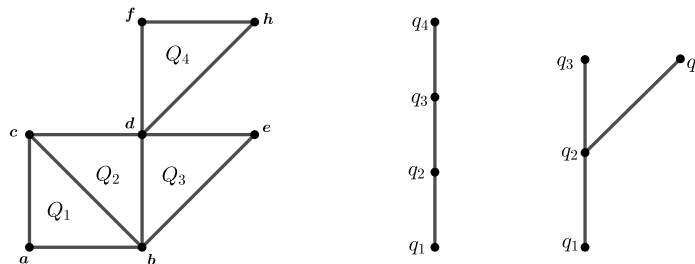


Fig. 1: A chordal graph and two of its clique trees

304 **Theorem 4** (Gavril [19] and Buneman [11]). *The following propositions are equiv-*
 305 *alent:*

- 306 1. G is a chordal graph.
- 307 2. G is the intersection graph of a family of subtrees of a tree.
- 308 3. There exists a tree T (called *Clique tree*) with vertex set $\{q_1, q_2, \dots, q_r\}$ such
 309 that for each vertex v , the set $R_v = \{q_i \mid v \in Q_i\}$ induces a subtree of T . Here
 310 q_i represents the maximal clique Q_i .

311 From the above description, we have that a graph G is a chordal graph if and
 312 only if there exists a clique tree T with maximal cliques of G as the vertices of T and
 313 any two cliques containing $v \in V(G)$ are either adjacent in T or connected by a path
 314 of cliques that contain v . The Lemma below is also very useful in understanding
 315 the structure of chordal graphs and their simplicial vertices.

316 **Lemma 7.** [5] *A vertex is simplicial if and only if it belongs to precisely one maximal*
 317 *clique.*

318 The following result can be proved for establishing the recognition of chordal
 319 graphs which are C-I graphs.

320 **Lemma 8.** *Every leaf node of a clique tree of a chordal graph contains a simplicial*
 321 *vertex.*

322 *Proof.* Let T be a clique tree of a chordal graph G . Let C_k be a leaf node of T .
 323 Since C_k is a leaf node of T there is a node $C_{k'}$ in T such that $C_k C_{k'} \in E(T)$. Since
 324 T is a clique tree of G every node of T is a maximal clique of G and hence there is a
 325 vertex v of G in C_k but not in $C_{k'}$. From the definition of the clique tree, it is clear
 326 that v is not in any other nodes of T (cliques). Then it follows that v is a simplicial
 327 vertex by Lemma 7. \square

328 **Lemma 9.** *Let G be a chordal graph with the clique tree T being a path. Let C_i be*
 329 *an internal node of T , C_{i+1} , the parent node of C_i , and C_{i-1} the child node of C_i*
 330 *in T . If v is vertex of G such that $v \in C_i \setminus C_{i+1}$ and $v \in C_i \setminus C_{i-1}$, then v is a*
 331 *simplicial vertex in G .*

332 *Proof.* Since $v \in C_i \setminus C_{i+1}$ and $v \in C_i \setminus C_{i-1}$ and T is a path, it follows from the
 333 definition of a clique tree that v is not in any other cliques in T . Then by Lemma 7,
 334 v is a simplicial vertex. \square

335 Based on Theorem 3 and Lemma 8 and 9, we formulate the following algorithm
 336 for recognizing a chordal C-I graph G . The correctness of the algorithm follows from
 337 these results.

Algorithm 1: Algorithm for recognizing given graph G is chordal C-I
 graph G or not.

Input : G be a connected graph with $|V(G)| = n$ and $|E(G)| = m$

Output: G is chordal C-I graph or not

Step 1: Apply the perfect elimination ordering (PEO) on G and check whether G is a
 chordal graph or not. If there is no PEO then return G not chordal and stop.
 Otherwise, go to Step 2.

338 Step 2: Using the PEO find the clique tree T of the given graph G . If the clique tree has
 more than 2 leaf nodes, then return G , not chordal C-I graph, and stop.
 Otherwise, go to Step 3. (Now the clique tree is a path, and the vertices are in
 order C_1, C_2, \dots, C_k)

Step 3: Check whether there is an element in C_i that is not in C_{i+1} and C_{i-1} for
 $i = 2, 3, \dots, k - 1$. If such an element exists at any stage, then stop and return G
 is not a chordal C-I graph. Otherwise, return G is a chordal C-I graph.

339 The time complexity of Algorithm 1 can be analyzed as follows.

340 Using the Lex-BFS or *MCS*, the PEO can be done in $O(n+m)$ time. For Step 2,
 341 using PEO, the optimum clique tree T of G can be determined as follows. For each
 342 vertex v in the PEO, let S be the set of its neighbors that come later in the PEO.

343 Find the maximal cliques in the induced subgraph of G formed by S and v . For each
 344 maximal clique C , create a node in T corresponding to C . For each pair of maximal
 345 cliques C_1 and C_2 that share a vertex, add an edge between their corresponding
 346 nodes in T so that the resulting tree is the clique tree of G . Because each maximal
 347 clique contains all the vertices of its smaller cliques, the resulting tree is guaranteed
 348 to be a clique tree of the chordal graph G . The time complexity of this procedure
 349 is $O(n + m)$, where n is the number of vertices and m is the number of edges in
 350 the chordal graph G . This is because computing a PEO can be done in linear time,
 351 and the time to find maximal cliques is $O(n + m)$ using a standard algorithm such
 352 as Bron-Kerbosch, and the number of maximal cliques is $O(m)$. Checking whether
 353 there are more than 2 leaf nodes can be done in constant time.

354 In Step 3, let k be the length of the clique tree which is a path P . Then, in
 355 the worst case, the size of the sets C_i is of the order $O(n/k)$. To check whether an
 356 element $v \in C_i \setminus C_{i+1}$ and $v \in C_i \setminus C_{i-1}$ takes $O(3n/k)$ time by using a hash set
 357 operation. Since there are at most $k - 2$ internal nodes, the total time for Step 3 is
 358 $O(n)$. Now, the complexity of Algorithm 1 is $O(n + m)$.

359 4 Structure of C-I cographs

360 In this section, we present an algorithm for recognizing C-I cograph.

361 **Theorem 5.** [8] *Let G be a chordal cograph. Then G is a C-I graph if and only if*
 362 *G is a connected graph that contains at most two maximal cliques.*

363 From Theorem 5, we get the following remark

364 **Remark 3.** *A graph G is chordal C-I cograph if and only if there exists three pairwise*
 365 *disjoint sets C_1, C_2 and C_3 such that $V(G) = C_1 \cup C_2 \cup C_3$, and $x, y \in V(G)$ are*
 366 *adjacent in G if and only if $x, y \in C_1 \cup C_2$ or $x, y \in C_2 \cup C_3$.*

367 That is, $C_1, C_2, C_3, C_1 \cup C_2$ and $C_2 \cup C_3$ form cliques and the graph has no other
 368 edges. So in the C-I chordal cograph, if $xy, xz \notin E(G)$ then $yz \in E(G)$. It is
 369 clear that a chordal C-I cograph is a Ptolemaic C-I graph as it is a C-I graph of
 370 a completely ranked poset of height 3. The structure of an arbitrary chordal C-I
 cograph is shown in Fig. 2.

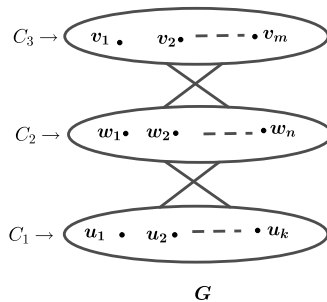


Fig. 2: General structure of a chordal C-I cograph G

372 **Theorem 6.** [8] A graph G is a C-I cograph if and only if G is the join of chordal
 373 C-I cographs.

374 From Theorem 6, we get the following remark.

375 **Remark 4.** Let G be a C-I cograph such that $G = G_1 \vee G_2 \vee \dots \vee G_k$, where G_i 's
 376 are chordal C-I cographs. By Remark 3, there exists a pairwise disjoint set C_1^i, C_2^i
 377 and C_3^i such that $V(G_i) = C_1^i \cup C_2^i \cup C_3^i$ and $xy \in E(G_i)$ if and only if $x, y \in C_1^i \cup C_2^i$
 378 or $x, y \in C_2^i \cup C_3^i$. Thus in G , $\bigcup_{i=1}^k C_2^i$ are universal vertices and for $u, v \in V(G)$ such
 379 that $uv \notin E(G)$ if and only if both u and v in $V(G_i)$ with $u \in C_1^i$ and $v \in C_3^i$, for
 380 some i . Thus, for any $u, v, w \in V(G)$ with $uv, uw \notin E(G)$, then $vw \in E(G)$.

The structure of an arbitrary C-I cograph is depicted in Fig. 3.

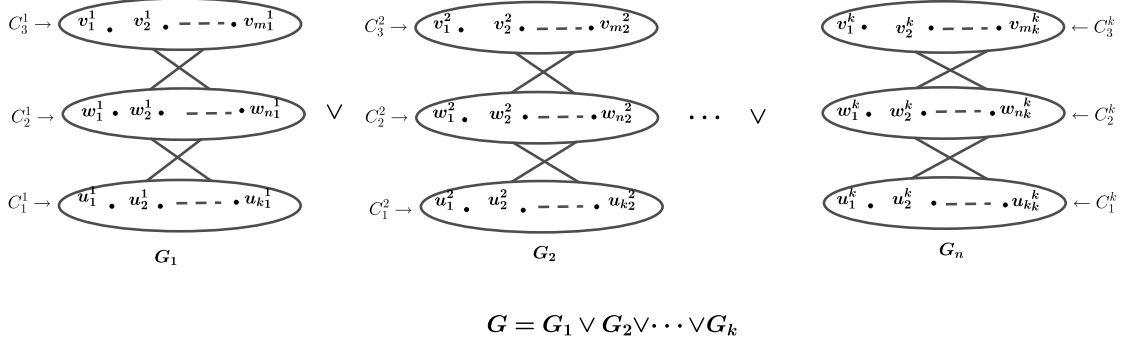


Fig. 3: General structure of a C-I cograph G

381

382 **Lemma 10.** Let G be a C-I cograph then there exists a poset $P = P_1 + P_2 + \dots + P_r$,
 383 where P_i 's are completely ranked poset of rank 2 such that $G \cong G_P$.

384 *Proof.* Let G be a C-I cograph. Then G is the join of chordal C-I cographs. That is,
 385 let $G = G_1 \vee G_2 \vee \dots \vee G_r$. Since G_i 's are chordal cographs, G_i 's are C-I graphs of the
 386 completely ranked poset P_i 's of rank at most 2. Hence $P = P_1 + P_2 + \dots + P_r$. \square

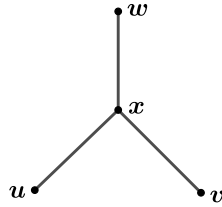


Fig. 4: Claw

387 **Lemma 11.** If G is a C-I cograph then G is a claw-free graph

388 *Proof.* Let G be a C-I cograph and $G = G_1 \vee G_2 \vee \dots \vee G_k$, where G_i 's are chordal
 389 C-I cograph for $i = 1, 2, \dots, k$. Suppose that G contains an induced claw by the
 390 vertices $\{u, v, w, x\} \subseteq V(G)$ with edges $ux, vx, wx \in E(G)$ and $uv, vw, uw \notin E(G)$

391 (see Fig.6). Since $G = G_1 \vee G_2 \vee \dots \vee G_k$, the non-adjacent vertices lie in the same
 392 G_j for some $j \in \{1, 2, \dots, k\}$. That is, $u, v, w \in V(G_j)$. Since G_j is a chordal C-I
 393 cograph, if $uv, uw \notin E(G_j)$ then $vw \in E(G_j)$, which is a contradiction. Hence, the
 394 C-I cograph is a claw-free graph. \square

395 A *cotree* is a tree in which the internal nodes are labelled with the numbers 0
 396 and 1. Every cotree T defines a cograph G having the leaves of T as vertices, and in
 397 which the subtree rooted at each node of T corresponds to the induced subgraph in
 398 G defined by the set of leaves descending from that node. A subtree rooted at a node
 399 labelled 0 corresponds to the union of the subgraphs defined by the children of that
 400 node and a node labelled 1 corresponds to the join of the subgraphs defined by the
 401 children of that node. The cotree satisfies the property that, on every root-to-leaf
 402 path, leaves are the vertices of the graph, the labels of the internal nodes alternate
 403 between 0 and 1, and every internal node has at least two children. The cotree can
 404 be easily obtained from any tree labelled with such 0/1 T by coalescing all pairs
 405 of child-parent nodes in T having the same label or where the parent has only one
 406 child. Vertices x and y of G are adjacent in G if and only if their lowest common
 407 ancestor(LCA) in the cotree is labelled 1. This representation is unique and every
 408 cograph can be represented in this way by a cotree [12]. Since G is a connected
 409 graph, the root of the cotree is a 1-node.

410 It is known that cographs have a unique tree representation, called a cotree.
 411 Using the cotree, it is possible to design very fast polynomial-time algorithms for
 412 problems that are intractable for graphs in general. Such problems include chro-
 413 matic number, clique determination, clustering, minimum weight domination, iso-
 414 morphism, minimum fill-in, and Hamiltonicity. A linear-time cograph recognition
 415 algorithm, such as those in [9, 13, 20], which also builds a cotree, a data structure
 416 that fully encodes the cograph.

417 The Fig. 5 illustrates a cograph G and an embedding of the corresponding cotree
 418 T_G . The leaves of T_G represent the set of vertex $V(G)$, and each internal node
 419 signifies the union (0) or the joining (1) operations in the children. The significance
 420 of the 0(1) nodes is captured by the fact that $xy \in E(G)$ if and only if $LCA(x, y)$ is
 421 a 1 node, as shown in Fig. 5.

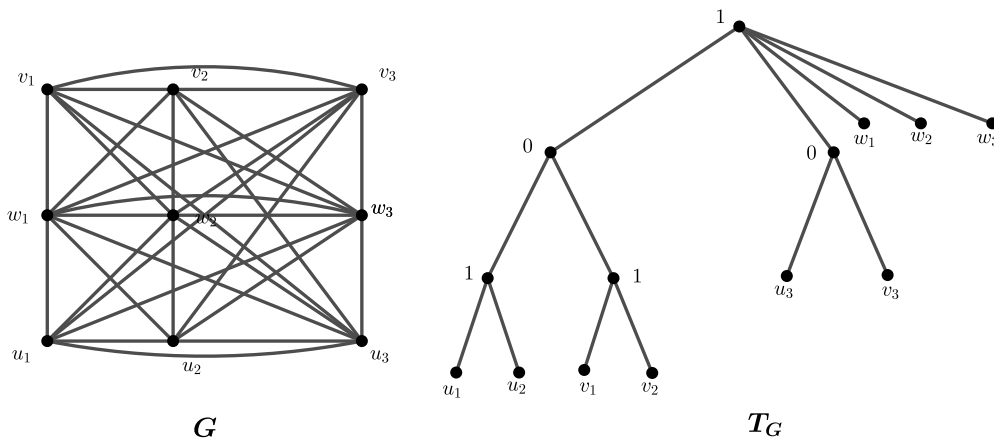


Fig. 5: G be a cograph and T_G be the cotree of G

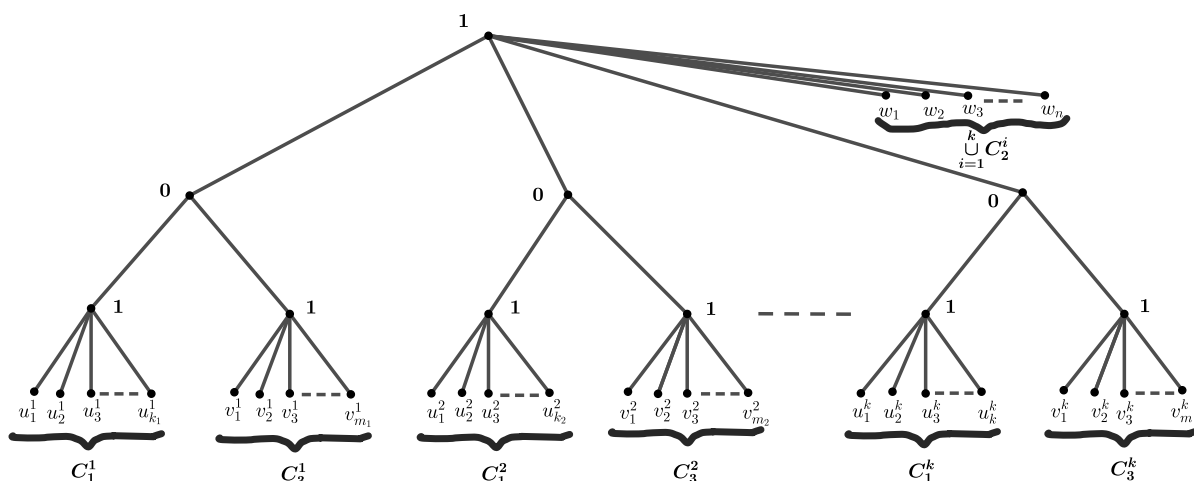


Fig. 6: Cotree for C-I cograph (\mathcal{T}_C)

422 The general structure of the cotree of C-I cographs is shown in Fig. 6, and
423 we denote the family of cotree of C-I cograph as \mathcal{T}_C . Let G be a C-I cograph,
424 $G = G_1 \vee G_2 \vee \dots \vee G_k$, where G_i 's are chordal C-I cographs. Then by Remark 4,
425 two vertices x and y are not adjacent in G if and only if $x \in C_1^i$ and $y \in C_3^i$ for
426 $i = 1, 2, \dots, k$. Therefore, in the cotree, the lowest common ancestor(LCA) of x and
427 y is 0-node. That is, $LCA(x, y) = 0$ for $x \in C_1^i$ and $y \in C_3^i$. In other cases, LCA is
428 a 1-node. That is, $LCA(x, y) = 1$ for $x \notin C_1^i$ and $y \notin C_3^i$.

429 **Lemma 12.** *The cotree of C-I cograph which is not a complete graph satisfies the*
430 *following properties*

- 431 (i) *The root node (which is a 1-node) of \mathcal{T}_C has leaf nodes and 0-nodes as children.*
432 *the number of 0-nodes (which are children of the root node) is always less than*
433 *or equal to the number of leaf nodes (which are children of the root node).*
- 434 (ii) *Every 0-node has exactly two children (both the children are 1-nodes or both*
435 *the children are leaves nodes or one child is a leaf node and another child is*
436 *1-node).*
- 437 (iii) *The 1-nodes other than the root node have only leaf nodes as children.*
- 438 (iv) *The cotree of G belongs to the family of \mathcal{T}_C .*

439 *Proof.* Let G be a C-I cograph which is not a complete graph.

- 440 (i) Since G is not a complete C-I graph, it is either a chordal cograph or joins
441 of chordal cographs. Since there are non-adjacent vertices, there exist some
442 0-nodes as children of the root node. From the structure of the C-I cograph,
443 there exist universal vertices, which should be leaf nodes of the root node.
444 From Theorem 6, $G = G_1 \vee G_2 \vee \dots \vee G_k$, where G_i are chordal C-I cographs.
445 The leaf nodes of the subtree rooted by each 0-node contain the non-universal
446 vertices of each G_i . That is, if there are k 0-nodes, then since the universal
447 vertices of each G_i , are also universal vertices of G , and since each G_i has at

448 least one universal vertex, the number of 0-nodes of the root node must be at
 449 most the number of leaf nodes as a child nodes. Hence (i) follows.

450 (ii) Suppose there are 0-nodes having more than two children. Let $0'$ be a 0-
 451 node having three child nodes. Let u, v, w be the three leaf nodes from the
 452 different branches of the subtree with the root node as the $0'$ -node. Then by
 453 the structure of the C-I cograph, we have that $uv, vw, uw \notin E(G)$. Let x be a
 454 leaf node of the root of the cotree \mathcal{T}_C . Then x is adjacent to u, v and w . That
 455 is, $\{u, v, w, x\}$ induce a claw in G which is a contradiction to Lemma 11. So
 456 every 0-node has no more than two children. In a cotree, every node has a
 457 minimum of two child nodes. Hence every 0-node has exactly two child nodes
 458 as 1-nodes. Hence (ii) follows.

459 (iii) Suppose that there is a 1-node other than the root node which has a 0-node,
 460 say $0'$. Consider a sub-tree T of \mathcal{C}_T with root as $0'$. Let u and v be the vertices
 461 of G such that the $LCA(u, v)$ is $0'$. The vertices u and v exist since the sub-tree
 462 T has exactly two branches. Now consider the sub-tree T' of \mathcal{C}_T containing T
 463 with root as a 0-node, say $0''$ different from $0'$. Let w be a vertex of G such
 464 that the $LCA(u, v, w)$ is $0''$ (the vertex w exists since there are two children
 465 from $0''$ of T'). Clearly, the vertices u, v, w are mutually non-adjacent. Now
 466 any vertex x of G which are leaves of the root node 1 of \mathcal{C}_T is adjacent to all
 467 of u, v, w so that u, v, w, x form an induced claw, a contradiction. Hence (iii)
 468 follows.

469 (iv) Follows from (i),(ii) and (iii).

470 □

471 **Theorem 7.** *A cograph G is a C-I cograph if and only if the cotree of G belongs to*
 472 *the family \mathcal{C}_T*

473 *Proof.* It is clear that when G is a complete graph if and only if the cotree is the
 474 tree with every vertex of G as a child of the root node and which is a part of \mathcal{C}_T .

475 Let G be a C-I cograph which is not a complete graph. Then by Lemma 12,
 476 cotree of G is of the form \mathcal{T}_C

477 Conversely, we need to prove that if \mathcal{T}_C is the cotree of a cograph G then G is a
 478 C-I graph. Consider the set of leaves of the co-tree \mathcal{T}_C (Refer Figure 6). It follows
 479 by the definition of co-tree of a cograph G that the sets C_i^j 's are cliques of G , for

480 $i = 1, 3$ and $j = 1, 2, \dots, k$ and $\bigcup_{i=1}^k C_2^i = \{w_1, w_2, \dots, w_m\}$ are universal vertices.

481 Now C_1^j is adjacent to every vertex in G except C_3^j and C_3^j is adjacent to every
 482 vertex in G except C_1^j for $j = 1, 2, \dots, k$. Then by Remark 4, G is a C-I cograph.

483 Hence proved.

484 □

485 In [13], Corneil et al. presented a linear time algorithm for recognizing cographs
 486 and constructing their cotree representation. Using that algorithm, we get the
 487 cotree.

488 **Theorem 8.** [13] *The family of cographs possesses a linear time recognition algo-*
489 *rithm by the construction of the cotrees in linear time.*

Algorithm 2: Algorithm for recognizing C-I cograph

Input : A connected graph G with $|V(G)| = n$

Output: G is a C-I cograph or not.

Step 0: Using the recognition algorithm in Theorem 8 determine the cotree if G is a cograph and go to Step 1. Otherwise, stop and return G is not cograph

Step 1: Perform BFS from the root node 1 of the cotree.

Step 2: If BFS completes in one step by reaching all the leaf nodes (if all the adjacent nodes of the root node are leaf nodes), then the graph is a complete graph and it is C-I cograph; stop, and we are done. Otherwise, go to Step 3.

490 Step 3: (that is in Step 2, we obtain 0-nodes and leaf nodes or only 0-nodes). If all the neighbours of the root nodes are only 0-nodes or the number of 0-nodes is greater than the number of leaf nodes of the root node, then G is not a C-I cograph and stop. Otherwise, go to Step 4.

Step 4: Continue the searching at level 2. If 0-nodes has more than two children as 1-nodes then G is not a C-I cograph and stop. Otherwise, go to Step 5.

Step 5: Continue searching at level 3. In level 3, if any 1-node has 0-node as a child, then G is not a C-I cograph and stop. If all the children of the 1-nodes are the leaf node then we are done. the resulting graph is a C-I cograph.

491 The correctness of the algorithm follows from the structure of cotree stated in
492 Lemma 10 and the facts mentioned above.

493 Now the size of the cotree of a C-I cograph can be estimated as follows. Let G
494 be a C-I cograph with n vertices and $G = G_1 \vee G_2 \vee \dots \vee G_k$. Then from the structure
495 of the cotree T of G , T contains leaf nodes(which are the n -vertices of G), 0-nodes,
496 and 1-nodes. Then T contains k 0-nodes, and each 0-node has exactly 2 children.
497 Therefore T has at most $1 + k + 2k + n$ vertices. The worst case for k is when $k = \frac{n}{3}$.
498 Therefore in the worst case, the total number of vertices in T is $1 + \frac{n}{3} + \frac{2n}{3} + 3$, which
499 is of the order of $O(n)$.

500 The time complexity of the recognition algorithm for cographs and finding its
501 cotree by Theorem 8 is linear. Now the complexity of Algorithm 2 is linear in the
502 size of the cotree, which is $O(n)$ in the worst case since the complexity of the BFS
503 can be performed on the cotree in linear time.

504

505

506 **Acknowledgment:** Anil, A. acknowledges the financial support from the University
507 of Kerala, for providing University Post Doctoral Fellowship (Ac EVII 5576/2023/UOK
508 dated 30/06/2023).

509 References

- 510 [1] Anil, A., Changat, M.: Composition and Product of Cover-Incomparability
511 Graphs. *The Art of Discrete and Applied Mathematics* 6, #P2.09 (2023)
512 <https://doi.org/10.26493/2590-9770.1515.af9>
- 513 [2] Anil, A., Changat, M.: Comparability Graphs Among Cover-Incomparability
514 Graphs., N. Balachandran and R. Inkulu (Eds.): CALDAM 2022, Algorithms
515 and Discrete Applied Mathematics, LNCS 13179, pp. 48–61, Springer, Cham
516 (2022). https://doi.org/10.1007/978-3-030-95018-7_5.
- 517 [3] Anil, A., Changat, M.: Ptolemaic and Chordal Cover-Incomparability Graphs,
518 *Order* 39, 29–43 (2022). <https://doi.org/10.1007/s11083-021-09551-w>
- 519 [4] Anil, A., Changat, M., Goligranc, T., Sukumaran, B.: Ptolemaic
520 and planar cover-incomparability graphs, *Order* 38, 421–439(2021).
521 <https://doi.org/10.1007/s11083-021-09549-4>
- 522 [5] Blair, J.R.S., Peyton, B. (1993). An Introduction to Chordal Graphs and Clique
523 Trees. In: George, A., Gilbert, J.R., Liu, J.W.H. (eds) *Graph Theory and*
524 *Sparse Matrix Computation. The IMA Volumes in Mathematics and its Appli-*
525 *cations*, vol 56. Springer, New York, NY.
- 526 [6] Brešar, B., Changat, M., Klavžar, S., Kovše, M., Mathew, J., Mathews, A.:
527 Cover-incomparability graphs of posets, *Order* 25, 335–347 (2008).
- 528 [7] Brešar, B., Changat, M., Goligranc, T., Mathew, J., Mathews, A.: Cover-
529 incomparability graphs and chordal graphs, *Discrete Appl. Math.* 158, 1752
530 –1759(2010).
- 531 [8] Brešar, B., Goligranc, T., Changat, M., Sukumaran, B.: Cographs Which are
532 Cover-Incomparability Graphs of Posets, *Order* 32, 179–187(2015).
- 533 [9] Bretscher, A., Corneil, D.G., Habib, M., Paul, C.: A simple linear time LexBFS
534 cograph recognition algorithm, *SIAM J. Discrete Math.* 22 1277–1296(2008).
- 535 [10] Brightwell, G., West, W.B.: Partially ordered sets, Chapter 11 in *Handbook of*
536 *Discrete and Combinatorial Mathematics* (K.H. Rosen, ed.)CRC Press, Boca
537 Raton, 717–752(2000).
- 538 [11] Buneman, P.: A characterization of rigid circuit graphs, *Discrete Math.* 9, 205–
539 212(1974).
- 540 [12] Corneil, D.G., Lerchs, H., Stewart Burlingham, L.: Complement reducible
541 graphs, *Discrete Appl. Math.* 3, 163–174(1981).
- 542 [13] Corneil, D.G., Perl, Y., Stewart, L.K.: A linear recognition algorithm for
543 cographs, *SIAM J. Computing* 14, 926–934(1985).

- 544 [14] Cournier, A., Habib, M.: A new linear algorithm for modular decomposition
545 in S. Tison (Ed.), 19th International Colloquium Trees in Algebra and Pro-
546 gramming, CAAP'94, Vol. 787, Lecture Notes in Computer Science, Springer,
547 Berlin, 68–82(1994).
- 548 [15] Dahlhaus, E., Gustedt, J., McConnell, R.M.: Efficient and practical algorithms
549 for sequential modular decomposition, *J. Algorithms* 41(2), 360–387(2001).
- 550 [16] Dirac, G., A.: On rigid circuit graphs, *Abh. Math. Sem. Univ. HamDurg*, 25,
551 71-76(1961).
- 552 [17] Fulkerson, D.R., Gross, O.A.: Incidence matrices and interval graphs, *Pacific*
553 *J. Math.* 15, 835-855(1965).
- 554 [18] Gallai, T., Transitiv orientierbare Graphen, *Acta Math. Acad. Sci. Hung.* 18,
555 25–66(1967).
- 556 [19] Gavril, F.: The intersection graphs of a path in a tree are exactly the chordal
557 graphs, *J. Comb. Theory* 16, 47–56(1974).
- 558 [20] Habib, M., Paul, C.: A simple linear time algorithm for cograph recognition,
559 *Discrete Appl. Mathematics* 145, 183–197 (2005).
- 560 [21] Jung, H.A.: On a class of posets and the corresponding comparability graphs,
561 *Journal of Combin. Theory B* 24, 125-133 (1978)
- 562 [22] Maxová, J., Turzík, D.: Which distance-hereditary graphs are cover-
563 incomparability graphs?, *Discrete Appl. Math.*161, 2095–2100(2013).
- 564 [23] Maxová, J., Dubcová, M., Pavlíková, P., Turzík, D.: Which k-trees are cover-
565 incomparability graphs, *Discrete Appl. Math.* 167, 222–227(2014).
- 566 [24] Maxová, J., Pavlíková, P., Turzík, D.: On the complexity of cover-
567 incomparability graphs of posets, *Order* 26(3), 229–236(2009).
- 568 [25] McConnell, R.M., Spinrad, J.: Linear-time modular decomposition and effi-
569 cient transitive orientation of comparability graphs, in *Proceedings of the Fifth*
570 *Annual ACM-SIAM Symposium on Discrete Algorithms* Arlington, VA, ACM,
571 New York, 536–545(1994).
- 572 [26] Rose, D.J., Tarjan, R.E., Lueker, G.S.: Algorithmic Aspects of Vertex Elim-
573 ination on Graphs, *SIAM Journal on Computing* 5(2), 266-283 (1976).
- 574 [27] Seinsche, S.: On the property of the class of n-colorable graphs, *J. Comb.*
575 *Theory (B)*, 191–193(1974).
- 576 [28] Shibata, Y.: On the tree representation of chordal graphs, *Journal of Graph*
577 *Theory* 12, 421–428 (1988).
- 578 [29] Tarjan, R.E., Yannakakis, M.: Simple linear-time algorithms to test chordality
579 of graphs, test acyclicity of hypergraphs, and selectively reduce acyclic hyper-
580 graphs, *SIAM J. Comput.* 13, 566–579 (1984).