Adding an Edge in a Cograph

Stavros D. Nikolopoulos and Leonidas Palios

Department of Computer Science,
University of Ioannina, GR-45110 Ioannina, Greece
{stavros, palios}@cs.uoi.gr

Abstract. In this paper, we establish structural properties of cographs which enable us to present an algorithm which, for a cograph G and a non-edge xy (i.e., two non-adjacent vertices x and y) of G, finds the minimum number of edges that need to be added to the edge set of G such that the resulting graph is a cograph and contains the edge xy. The motivation for this problem comes from algorithms for the dynamic recognition and online maintenance of graphs; the proposed algorithm could be a suitable addition to the algorithm of Shamir and Sharan [13] for the online maintenance of cographs. The proposed algorithm runs in time linear in the size of the input graph and requires linear space.

Keywords: Perfect graphs, cographs, cotrees, connected components, co-connected components, optimization problems.

1 Introduction

In this paper, we study the following problem:

> *(Cograph,+1)-MinEdgeAddition*: Given a cograph G and a non-edge xy (i.e., a pair of non-adjacent vertices x and y) of G, find the minimum number of non-edges of G that need to be added to G so that the resulting graph is a cograph and contains xy as an edge.

This problem is an instance of a more general $(\Pi, +k)$-MinEdgeAddition problem in which we deal with a class Π of graphs and we want to have k given non-edges added. Similarly, we can define the $(\Pi, -k)$-MinEdgeAddition problem: we are given a graph G from a class Π and k edges of G which we want removed; since the removal of these edges yields a graph G' which may not necessarily belong to Π, we want to find the minimum number of non-edges of G which when added to G' give a graph in Π (note that the fact that we add non-edges of G prevents the addition of an edge of G which we want removed). Further extensions lead to the $(\Pi, \pm k)$-MinEdgeDeletion problem, in which we remove the minimum number of edges of G (instead of adding non-edges) in order to obtain a graph in Π.

The above problems are motivated by the dynamic recognition problem on (or on-line maintenance of) graphs: a series of requests for the addition or the deletion of an edge or a vertex (potentially incident on a number of edges) are submitted and each is executed only if the resulting graph remains in the same class of graphs. Several authors have studied this problem for different classes of graphs and have given algorithms supporting some or all the above operations; we mention the edges-only fully dynamic algorithm of Ibarra [8] for chordal graphs, the fully dynamic algorithm of Hell *et al.* [7] for proper interval graphs, and the fully dynamic algorithm of Shamir and Sharan [13] for cographs.

The *cographs*, short for *complement reducible graphs*, are defined as the class of graphs formed from a single vertex under the closure of the operations of union and complementation, namely: (i) a single-vertex graph is a cograph; (ii) the disjoint union of cographs is a cograph; (iii) the complement of a cograph is a cograph. Cographs were introduced in the early 1970s by Lerchs [11] who studied their structural and algorithmic properties. Along with other properties, Lerchs has shown that they admit a unique tree representation, up to isomorphism, called a *cotree*. Cographs have arisen in many disparate areas of applied mathematics and computer science and have been independently rediscovered by various researchers under various names such as D^*-graphs [10], P_4 restricted graphs [4,5], 2-parity graphs and Hereditary Dacey graphs or *HD*-graphs [15]. They are perfect and in fact form a proper subclass of permutation graphs and distance hereditary graphs; they contain the class of quasi-threshold graphs and, thus, the threshold graphs [1,9]. Furthermore, they are precisely the graphs which contain no induced subgraph isomorphic to a P_4 (i.e., a chordless path on four vertices).

The study of cographs led naturally to constructive characterizations that implied several linear-time recognition algorithms that also enabled the construction of the cotree in linear time [1,14]. Surprisingly, despite the structural simplicity of cographs, constructing linear-time recognition algorithms has been challenging. The first linear-time recognition and cotree-construction algorithm was proposed by Corneil, Perl, and Stewart [5] in 1985. Recently, Bretscher *et al.* [2] presented a simple linear-time recognition algorithm which uses a multi-sweep LexBFS approach; their algorithm either produces the cotree of the input graph or identifies an induced P_4. Additionally, since the cographs are perfect, many interesting optimization problems in graph theory, which are NP-complete in general graphs, have polynomial sequential solutions [1,9]; for example, for the problem of determining the minimum path cover for a cograph, Lin *et al.* [12] presented a linear-time algorithm, which can be used to produce a Hamiltonian cycle or path, if such a structure exists.

In this paper, we solve the (Cograph,+1)-MinEdgeAddition problem. We consider (what we call) the component-partition of a graph G with respect to any of its vertices v: this is related to the partition of the subgraph of G induced by the neighbors of v in G into co-components and to the partition of the subgraph induced by the non-neighbors of v into components. By taking advantage of the fact that a cograph contains no induced subgraph isomorphic to a P_4 [11], we establish structural properties for the component-partition of a cograph with respect to any of its vertices. These properties and the use of dynamic programming enable us to describe an algorithm for the above problem which runs in time linear in the size of the input graph.

2 Theoretical Framework

We consider finite undirected graphs with no loops or multiple edges. For a graph G, we denote by $V(G)$ and $E(G)$ the vertex set and edge set of G, respectively. Let S be a subset of the vertex set $V(G)$ of a graph G; the subgraph of G induced by S is denoted by $G[S]$.

The *neighborhood* $N(x)$ of a vertex x of the graph G is the set of all the vertices of G which are adjacent to x. The *closed neighborhood* of x is defined as $N[x] := N(x) \cup \{x\}$. The neighborhood of a subset S of vertices is defined as $N(S) := \big(\bigcup_{x \in S} N(x)\big) - S$ and its closed neighborhood as $N[S] := N(S) \cup S$. If two vertices x and y are adjacent in G, we say that *x sees y*; otherwise we say that *x misses y*. We extend this notion to vertex sets: $V_i \subseteq V(G)$ sees (misses) $V_j \subseteq V(G)$ if and only if every vertex $x \in V_i$ sees (misses) every vertex $y \in V_j$.

If the graph G contains a path from a vertex x to a vertex y, we say that *x is connected to y*. The *connected components* (or *components*) of G are the equivalence classes of the "is connected to" relation on the vertex set $V(G)$ of G. The *co-connected components* (or *co-components*) of G are the connected components of the complement $\overline{G}$ of G.

3 The Component-Partition

Let us consider for a vertex v of a graph G the partition of the subgraphs $G[N(v)]$ and $G[V(G)-N[v]]$ into co-components and connected components, respectively; then, we define:

Definition 1. *Let G be a graph and v a vertex of G. We define the* component-partition of G with respect to v*, denoted by $(v;\widehat{\mathcal{C}}_{1..\ell};\mathcal{C}_{1..k})$, as the partition of the vertex set $V(G)$*

$$V(G) \;\; = \;\; \{v\} \; + \; \widehat{\mathcal{C}}_1 + \widehat{\mathcal{C}}_2 + \ldots + \widehat{\mathcal{C}}_\ell \; + \; \mathcal{C}_1 + \mathcal{C}_2 + \ldots + \mathcal{C}_k,$$

where $\widehat{\mathcal{C}}_1, \widehat{\mathcal{C}}_2, \ldots, \widehat{\mathcal{C}}_\ell$ are the co-connected components of $G[N(v)]$ and $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_k$ are the connected components of $G[V(G) - N[v]]$.

In particular, we restrict our attention to component-partitions such that there are no P_4s with vertices in both $N(v)$ and $V(G) - N[v]$; thus, we define:

Definition 2. *Let G be a graph, v a vertex of G, and $(v;\widehat{\mathcal{C}}_{1..\ell};\mathcal{C}_{1..k})$ the component-partition of G with respect to v. We say that this component-partition is* good *if and only if G contains no P_4 with a vertex in some $\widehat{\mathcal{C}}_i$ $(1 \le i \le \ell)$ and a vertex in some $\mathcal{C}_j$ $(1 \le j \le k)$.*

Our interest in good component-partitions comes from the property described in the following observation:

Observation 1. *Suppose that the component-partition $(v;\widehat{\mathcal{C}}_{1..\ell};\mathcal{C}_{1..k})$ of a graph G with respect to a vertex v is good. If G contains a P_4, then all the* *vertices of the P_4 belong to the same co-component $\widehat{\mathcal{C}}_i$ or to the same component $\mathcal{C}_j$.*

Observation 1 follows from the fact that the vertices of any P_4 in the subgraph $G[N[v]]$ all belong to the same co-component of $G[N(v)]$, and the vertices of any P_4 in the subgraph $G[V(G) - N[v]]$ all belong to the same component of $G[V(G) - N[v]]$. Additionally, the definition of a good component-partition and the fact that the cographs do not contain P_4s clearly imply:

Observation 2. *If G is a cograph, then the component-partition of any induced subgraph of G with respect to any of its vertices is good.*

In Lemma 1 we establish necessary and sufficient conditions for a component-partition to be good.

Lemma 1. *Let G be a graph, v a vertex of G, and $(v;\widehat{\mathcal{C}}_{1..\ell};\mathcal{C}_{1..k})$ the component-partition of G with respect to v. Then, the component-partition of G with respect to v is good if and only if the following two conditions hold:*

(i) for each co-component $\widehat{\mathcal{C}}_i$ and each component $\mathcal{C}_j$, $\widehat{\mathcal{C}}_i$ either sees or misses $\mathcal{C}_j$;
(ii) if, for each co-component $\widehat{\mathcal{C}}_i$, $1 \leq i \leq \ell$, we define the set $\widehat{I}_i = \{\, j \mid \widehat{\mathcal{C}}_i \text{ sees } \mathcal{C}_j \,\}$, then the co-components of $G[N(v)]$ have the following monotonicity property: $|\widehat{I}_i| \leq |\widehat{I}_j|$ implies that $\widehat{I}_i \subseteq \widehat{I}_j$.

Condition (ii) of Lemma 1 can be phrased in another equivalent way, as given in the following corollary.

Corollary 1. *Let G be a graph, v a vertex of G, and $(v;\widehat{\mathcal{C}}_{1..\ell};\mathcal{C}_{1..k})$ the component-partition of G with respect to v. Then, the component-partition of G with respect to v is good if and only if the following two conditions hold:*

(i) for each co-component $\widehat{\mathcal{C}}_i$ and each component $\mathcal{C}_j$, $\widehat{\mathcal{C}}_i$ either sees or misses $\mathcal{C}_j$;
(ii) Suppose that the ordering of the co-components $\widehat{\mathcal{C}}_1, \widehat{\mathcal{C}}_2, \ldots, \widehat{\mathcal{C}}_\ell$ corresponds to their ordering by non-decreasing $|\widehat{I}_i|$, where $\widehat{I}_i = \{\, j \mid \widehat{\mathcal{C}}_i \text{ sees } \mathcal{C}_j \,\}$. If we associate each component $\mathcal{C}_i$, $1 \leq i \leq k$, with the set $I_i = \{\, j \mid \mathcal{C}_i \text{ sees } \widehat{\mathcal{C}}_j \,\}$, then the components of $G[V(G) - N[v]]$ have the following property: if $I_i \neq \emptyset$ and h is the minimum element of I_i, then $I_i = \{h, h+1, \ldots, \ell\}$.

We also note that because the co-components of the neighbors of a vertex and the components of its non-neighbors trade places in the complement of the graph, then properties similar to those described in conditions (ii) of Lemma 1 and Corollary 1 hold for the sets I_i and $\widehat{I}_i$, respectively.

Let us assume that the component-partition $(v;\widehat{\mathcal{C}}_{1..\ell};\mathcal{C}_{1..k})$ of the graph G is good. We partition the set of co-components $\{\widehat{\mathcal{C}}_1, \widehat{\mathcal{C}}_2, \ldots, \widehat{\mathcal{C}}_\ell\}$ of the subgraph $G[N(v)]$ into a collection of sets $\widehat{S}_1, \widehat{S}_2, \ldots, \widehat{S}_{\ell'}$ defined as follows:

Definition 3. *Consider the equivalence relation R on the set of co-components $\{\widehat{\mathcal{C}}_1, \widehat{\mathcal{C}}_2, \ldots, \widehat{\mathcal{C}}_\ell\}$ such that $(\widehat{\mathcal{C}}_i, \widehat{\mathcal{C}}_j) \in R$ if and only if $\widehat{I}_i = \widehat{I}_j$, i.e., $\widehat{\mathcal{C}}_i$ and $\widehat{\mathcal{C}}_j$ see the same components of the subgraph $G[V(G) - N[v]]$. We define the sets $\widehat{S}_1, \widehat{S}_2, \ldots, \widehat{S}_{\ell'}$ as the equivalence classes of the relation R where, for every i, j such that $1 \leq i < j \leq \ell'$, and every $\widehat{\mathcal{C}}_r \in \widehat{S}_i$ and $\widehat{\mathcal{C}}_s \in \widehat{S}_j$, it holds that $\widehat{I}_r \subset \widehat{I}_s$.*

The value ℓ' is equal to the number of distinct sets $\widehat{I}_i$, and thus each set $\widehat{S}_j$ is nonempty. It is not difficult to see that the partition sets $\widehat{S}_1, \widehat{S}_2, \ldots, \widehat{S}_{\ell'}$ have the following properties:

- If a connected component $\mathcal{C}$ of the subgraph $G[V(G) - N[v]]$ sees a co-component $\widehat{\mathcal{C}}_i \in \widehat{S}_j$, then $\mathcal{C}$ sees all the co-components in $\widehat{S}_j$.
- Let us consider the ordering of the co-components $\{\widehat{\mathcal{C}}_1, \widehat{\mathcal{C}}_2, \ldots, \widehat{\mathcal{C}}_\ell\}$ consisting of an arbitrary ordering of the elements of the set $\widehat{S}_1$ followed by an arbitrary ordering of the elements of $\widehat{S}_2$ and so on up to the set $\widehat{S}_{\ell'}$. In this ordering, the co-components $\widehat{\mathcal{C}}_i$, $1 \le i \le \ell$, are ordered by non-decreasing value of $|\widehat{I}_i|$.

In light of these properties and the fact that the component-partition $(v; \widehat{\mathcal{C}}_{1..\ell}; \mathcal{C}_{1..k})$ is good (thus condition (ii) of Corollary 1 holds), we have:

Definition 4. *We define the following partition of the set of components* $\{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_k\}$ *of the subgraph* $G[V(G) - N[v]]$*:*

$$\begin{aligned} S_1 &= \{\, \mathcal{C}_j \mid \forall \widehat{\mathcal{C}} \in \widehat{S}_1,\ \mathcal{C}_j \textit{ sees } \widehat{\mathcal{C}} \,\} \\ S_i &= \{\, \mathcal{C}_j \mid \forall \widehat{\mathcal{C}} \in \widehat{S}_i \textit{ and } \widehat{\mathcal{C}}' \in \widehat{S}_{i-1},\ \mathcal{C}_j \textit{ sees } \widehat{\mathcal{C}} \textit{ but misses } \widehat{\mathcal{C}}' \,\} \qquad (2 \le i \le \ell') \\ S_{\ell'+1} &= \{\, \mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_k \,\} - \bigcup_{i=1,\ldots,\ell'} S_i \end{aligned}$$

The definition of the sets $\widehat{S}_j$, $j = 1, 2, \ldots, \ell'$, implies that $S_i \neq \emptyset$ for all $i = 2, 3, \ldots, \ell'$. However, $S_{\ell'+1}$ and S_1 may be empty. In particular, $S_{\ell'+1}$ is empty if and only if the graph G is connected; in fact, $S_{\ell'+1}$ contains the connected components of G except for the component to which v belongs. Figure 1 illustrates the partitions of the set of co-components and of the set of components described above and their adjacencies in a good component-partition of the graph G with respect to vertex v; the dotted ovals indicate the partition sets, and the circles inside the ovals indicate the components or co-components belonging to the partition set.

In terms of the partitions into sets $\widehat{S}_1, \widehat{S}_2, \ldots, \widehat{S}_{\ell'}$ and $S_1, S_2, \ldots, S_{\ell'}, S_{\ell'+1}$, the cotree of a cograph G has a very special structure, which is described in

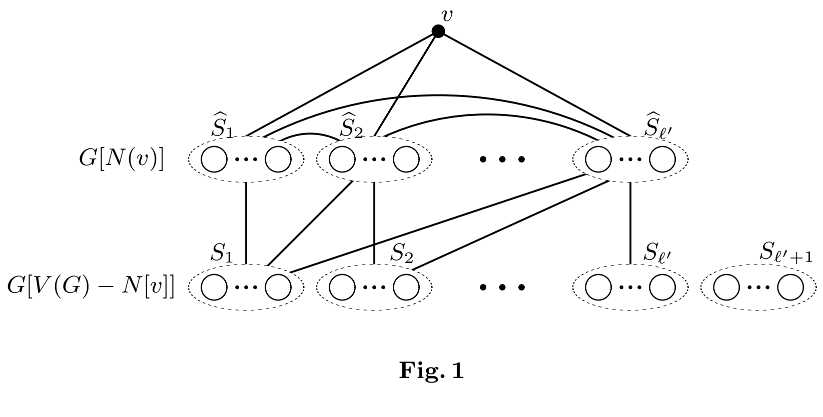

Fig. 1

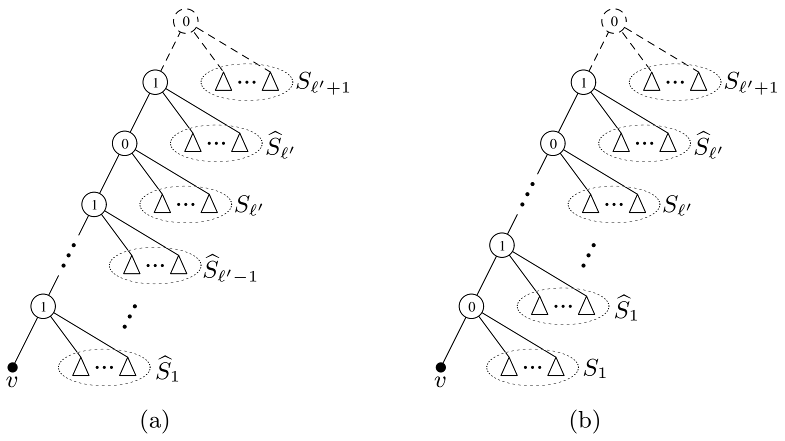

Fig. 2

the following observation (because of Observation 2, the sets $\widehat{S}_1, \widehat{S}_2, \ldots, \widehat{S}_{\ell'}$ and $S_1, S_2, \ldots, S_{\ell'}, S_{\ell'+1}$ are well defined).

Observation 3. *Let G be a cograph, v a vertex of G, and $\widehat{S}_1, \widehat{S}_2, \ldots, \widehat{S}_{\ell'}$ and $S_1, S_2, \ldots, S_{\ell'}, S_{\ell'+1}$, respectively, the partitions of the co-connected components of $G[N(v)]$ and of the connected components of $G[V(G) - N[v]]$ as defined above. Then,*

(i) if $S_1 = \emptyset$, the cotree of G has the general form depicted in Figure 2(a);

(ii) if $S_1 \neq \emptyset$, the cotree of G has the general form depicted in Figure 2(b).

In either case, the dashed part[1] *appears in the tree if and only if $S_{\ell'+1} \neq \emptyset$.*

The circular nodes labeled with a 0 or a 1 in Figure 2 are 0-nodes and 1-nodes, respectively, whereas the shaded node is a leaf node; the triangles denote the cotrees of the corresponding connected components or co-components.

4 Adding an Edge in a Cograph

Let G be a cograph and let x, y be two vertices of G which are not adjacent. We want to solve the (Cograph,+1)-MinEdgeAddition problem for G, x, y, i.e., we wish to make x and y adjacent, while adding the minimum number of non-edges of G so that the resulting graph is a cograph. Instrumental in the algorithm that we will be presenting is the component-partition of the graph G with respect to a vertex of G (see Definition 1) and in particular the partitions into sets $\widehat{S}_i$ and

[1] Lerchs' definition required that the root of a cotree be a 1-node [11]; here, we relax this condition and allow the root to be a 0-node as well, thus obtaining cotrees whose internal nodes all have at least two children, and whose root is a 1-node if and only if the corresponding cograph is connected.

S_j (see Definitions 3 and 4); since G is a cograph, Observation 2 holds and thus the adjacencies between the $\widehat{S}_i$s and S_js are as shown in Figure 1.

In particular, let $\widehat{S}_1(x)$, $\widehat{S}_2(x), \ldots$, $\widehat{S}_{\ell'_x}(x)$ and $S_1(x)$, $S_2(x), \ldots$, $S_{\ell'_x}(x)$, $S_{\ell'_x+1}(x)$ be the sets of the co-components of the subgraph $G[N(x)]$ and of the connected components of the subgraph $G[V(G) - N[x]]$, respectively. Since x and y are non-adjacent, then y belongs to a set, say, $S_{k_x}(x)$; in particular, let $\mathcal{C}_y$ be the component in $S_{k_x}(x)$ to which y belongs. Similarly, let $\widehat{S}_1(y)$, $\widehat{S}_2(y), \ldots$, $\widehat{S}_{\ell'_y}(y)$ and $S_1(y)$, $S_2(y), \ldots$, $S_{\ell'_y}(y), S_{\ell'_y+1}(y)$ be the sets of the co-components of $G[N(y)]$ and of the connected components of $G[V(G) - N[y]]$, respectively, and suppose that x belongs to the component $\mathcal{C}_x$ of the set $S_{k_y}(y)$. Because the elements of the sets $\widehat{S}_{k_x+i}(x)$ and $\widehat{S}_{k_y+i}(y)$, $i \geq 0$, and $S_{k_x+i}(x)$ and $S_{k_y+i}(y)$, $i \geq 1$, correspond to the subtrees of the cotree of G hanging from the nodes in the path from the parent of the least common ancestor of x and y to the root (see Figure 2), it holds that

$$\begin{array}{lrcll} & \widehat{S}_{k_x+i}(x) & = & \widehat{S}_{k_y+i}(y) & \text{for all } i \geq 0 \\ \text{and} & S_{k_x+i}(x) & = & S_{k_y+i}(y) & \text{for all } i \geq 1 \end{array}$$

which also implies that $\ell'_x - k_x = \ell'_y - k_y$. Moreover, from any subtrees, other than those containing x and y, hanging from the least common ancestor of x and y, we have:

$$S_{k_x}(x) - \{\mathcal{C}_y\} \;=\; S_{k_y}(y) - \{\mathcal{C}_x\}.$$

For the sake of simplicity of the notation, let us define

$$V_i(x) \;=\; \bigcup_{1 \leq t \leq i} \big(\widehat{S}_t(x) \cup S_t(x)\big) \quad \text{and} \quad V_i(y) \;=\; \bigcup_{1 \leq t \leq i} \big(\widehat{S}_t(y) \cup S_t(y)\big).$$

Note that $V_0(x) = \emptyset$ and $V_0(y) = \emptyset$. Then, the properties of a good component-partition (in light of Observation 2) imply (see also Figure 1):

P1: the common neighbors of x and y are precisely the vertices in $\widehat{S}_{k_x}(x) \cup \widehat{S}_{k_x+1}(x) \cup \ldots \cup \widehat{S}_{\ell'_x}(x) = \widehat{S}_{k_y}(y) \cup \widehat{S}_{k_y+1}(y) \cup \ldots \cup \widehat{S}_{\ell'_y}(y)$;

P2: $\mathcal{C}_y = \{y\} \cup V_{k_y-1}(y)$ and similarly, $\mathcal{C}_x = \{x\} \cup V_{k_x-1}(x)$.

In order to show Property P2, we note that $\mathcal{C}_y$ is the connected component to which y belongs after all the common neighbors of x and y have been removed; then, Property P2 follows from considering the removal of the vertices in $\widehat{S}_{k_y}(y) \cup \widehat{S}_{k_y+1}(y) \cup \ldots \cup \widehat{S}_{\ell'_y}(y)$ (see Property P1) in the component-partition of the graph G with respect to y.

Let G' be an optimal solution to the (Cograph,+1)-MinEdgeAddition problem, i.e., G' is a cograph for which $V(G') = V(G)$, $E(G) \cup \{xy\} \subseteq E(G')$, and $|E(G')|$ is minimum. Clearly, Observation 2 holds for G'; the properties of G' are described in the following two lemmata.

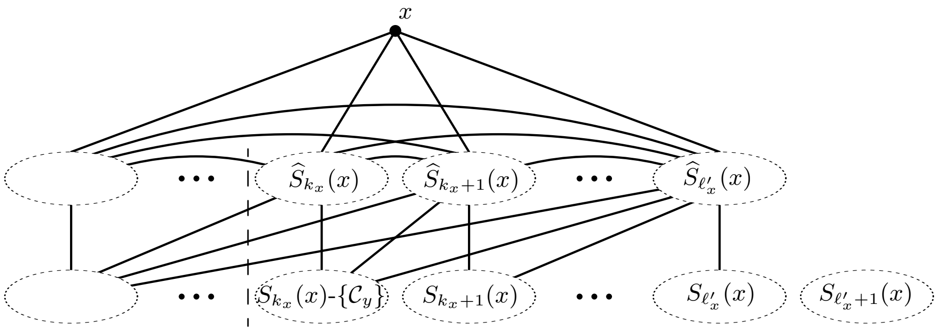

Fig. 3 $S_{k_x}(x) - \{C_y\} \neq \emptyset$: the rightmost sets $\widehat{S}'_i(x)$ and $S'_i(x)$, $i = r_x - (\ell'_x - k_x), \ldots, r_x$, in the partitions of the subgraphs $G'[N_{G'}(x)]$ and $G'[V(G') - N_{G'}[x]]$

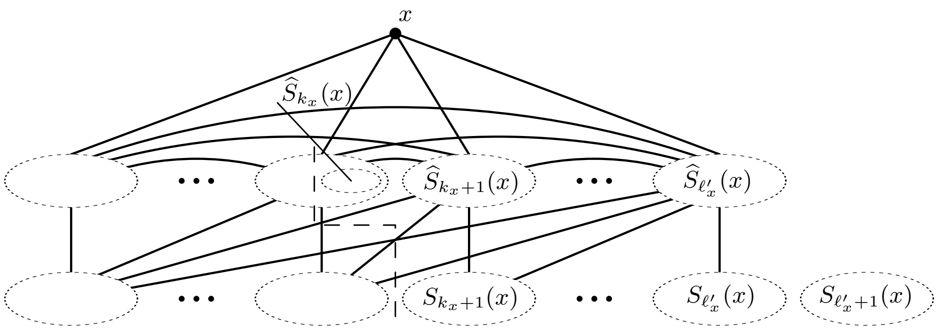

Fig. 4 $S_{k_x}(x) = \{C_y\}$: the rightmost sets $\widehat{S}'_i(x)$ and $S'_i(x)$, $i = r_x - (\ell'_x - k_x) + 1, \ldots, r_x$, and $\widehat{S}'_{r_x-(\ell'_x-k_x)}(x)$ in the partitions of $G'[N_{G'}(x)]$ and $G'[V(G') - N_{G'}[x]]$

Lemma 2. *Let G be a cograph, x, y be two non-adjacent vertices of G, and let*

- $\widehat{S}_1(x), \widehat{S}_2(x), \ldots, \widehat{S}_{\ell'_x}(x)$ *and* $S_1(x), S_2(x), \ldots, S_{\ell'_x}(x), S_{\ell'_x+1}(x)$,
- $\widehat{S}_1(y), \widehat{S}_2(y), \ldots, \widehat{S}_{\ell'_y}(y)$ *and* $S_1(y), S_2(y), \ldots, S_{\ell'_y}(y), S_{\ell'_y+1}(y)$,
- k_x, k_y, $\mathcal{C}_y$, $V_i(x)$, *and* $V_i(y)$

be as described above. Then, for the partition of the subgraphs $G'[N_{G'}(x)]$ and $G'[V(G') - N_{G'}[x]]$ of an optimal graph G' into sets of co-components $\widehat{S}'_1(x)$, $\widehat{S}'_2(x), \ldots, \widehat{S}'_{r_x}(x)$, and connected components $S'_1(x), S'_2(x), \ldots, S'_{r_x}(x), S'_{r_x+1}(x)$ respectively, the following properties hold:

(i) $\widehat{S}'_{r_x-i}(x) = \widehat{S}_{\ell'_x-i}(x) = \widehat{S}_{\ell'_y-i}(y)$ *for all* $i = 0, 1, 2, \ldots, \ell'_x - k_x - 1$, *and* $S'_{r_x+1-i}(x) = S_{\ell'_x+1-i}(x) = S_{\ell'_y+1-i}(y)$ *for all* $i = 0, 1, 2, \ldots, \ell'_x - k_x$ *(see Figures 3 and 4);*

(ii) if $S_{k_x}(x)$ *contains at least one connected component in addition to* $\mathcal{C}_y$, *then* $\widehat{S}'_{r_x-(\ell'_x-k_x)}(x) = \widehat{S}_{k_x}(x) = \widehat{S}_{k_y}(y)$ *and* $S'_{r_x-(\ell'_x-k_x)}(x) = S_{k_x}(x) - \{\mathcal{C}_y\}$ *(see Figure 3);*

(iii) if $S_{k_x}(x)$ *contains just the connected component* $\mathcal{C}_y$, *then all the co-components in* $\widehat{S}_{k_x}(x)$ *form co-components in* $\widehat{S}'_{r_x-(\ell'_x-k_x)}(x)$ *(see Figure 4).*

In terms of the cotree of the graph G, Lemma 2 implies that all changes that need to be done in order to obtain the cotree of the graph G' are restricted in the subtree rooted at the least common ancestor of x and y.

The remaining sets of co-components and components in the component-partition of the optimal graph G' with respect to x are obtained from an optimal "grouping" of the vertices in $V_{k_x-1}(x) \cup \mathcal{C}_y = V_{k_x-1}(x) \cup \{y\} \cup V_{k_y-1}(y)$ (see Property P2). The following lemma gives the possible cases of such a "grouping." It does not take into account case (iii) of Lemma 2; if this case applies, then the set $\widehat{S}'_{r_x-(\ell'_x-k_x)}(x)$, in addition to the co-components that result by the "grouping," contains the co-components in $\widehat{S}_{k_x}(x)$ as well (see Figure 4).

Lemma 3. *Let G be a cograph, x, y be two non-adjacent vertices of G, and let*

- $\widehat{S}_1(x), \widehat{S}_2(x), \ldots, \widehat{S}_{\ell'_x}(x)$ *and* $S_1(x), S_2(x), \ldots, S_{\ell'_x}(x), S_{\ell'_x+1}(x)$,
- $\widehat{S}_1(y), \widehat{S}_2(y), \ldots, \widehat{S}_{\ell'_y}(y)$ *and* $S_1(y), S_2(y), \ldots, S_{\ell'_y}(y), S_{\ell'_y+1}(y)$,
- $V_i(x)$ *and* $V_i(y)$

be as described above. Then, an optimal "grouping" of the vertices in $V_i(x) \cup \{y\} \cup V_j(y)$ in order to form sets of co-components $\widehat{S}'_1(x)$, $\widehat{S}'_2(x), \ldots, \widehat{S}'_k(x)$ and sets of connected components $S'_1(x)$, $S'_2(x), \ldots, S'_k(x)$ in the component-partition of an optimal graph G' with respect to vertex x is of one of the following forms:

(a) $k = 1$, *$\widehat{S}'_k(x)$ contains a single co-component induced by all the vertices in $V_i(x) \cup \{y\} \cup V_j(y)$, and $S'_k(x) = \emptyset$ (see Figure 5(a));*

(b) provided that $V_j(y) \neq \emptyset$, $k = i+1$, $\widehat{S}'_k(x)$ consists of a single co-component involving just vertex y, $S'_k(x)$ consists of a single connected component induced by the vertices in $V_j(y)$, whereas the remaining sets $\widehat{S}'_1(x), \widehat{S}'_2(x), \ldots, \widehat{S}'_{k-1}(x)$ and $S'_1(x), S'_2(x), \ldots, S'_{k-1}(x)$ are identical to $\widehat{S}_1(x), \widehat{S}_2(x), \ldots, \widehat{S}_i(x)$ and $S_1(x), S_2(x), \ldots, S_i(x)$, respectively (see Figure 5(b));

(c) provided that $j \geq 2$ or $j = 1$ and $S_1(y) \neq \emptyset$, $\widehat{S}'_k(x) = \widehat{S}_j(y)$ and $S'_k(x) = S_j(y)$ (see Figure 5(c));

(d) provided that $i \geq 2$ or $i = 1$ and $S_1(x) \neq \emptyset$, $\widehat{S}'_k(x) = \widehat{S}_i(x)$ and $S'_k(x) = S_i(x)$ (see Figure 5(d)).

Lemma 2 and the fact that the vertices in $\widehat{S}_{k_x}(x) \cup \widehat{S}_{k_x+1}(x) \cup \ldots \cup \widehat{S}_{\ell'_x}(x)$ see all the vertices in $\{x, y\} \cup V_{k_x-1}(x) \cup V_{k_y-1}(y)$ (see Property P1) imply that new edges are added only as a result of the "grouping;" it is important to note that we do not need to add new edges connecting vertices in the same set $\widehat{S}'_t(x)$ or $S'_t(x)$ as the vertices in each such set induce subgraphs not containing any P_4s. Then, in light of Lemma 3, we get a recursive expression for the number of additional edges that such an optimal "grouping" requires; this is given in Lemma 4.

Lemma 4. *Suppose that the conditions of Lemma 2 hold and let $cost(i, j)$ denote the number of edges with both endpoints in $\{x\} \cup V_i(x) \cup \{y\} \cup V_j(y)$ which need to be added to G in an optimal "grouping" of the vertices in $V_i(x) \cup \{y\} \cup V_j(y)$ to form sets $\widehat{S}'_t(x)$ and $S'_t(x)$. Then,*

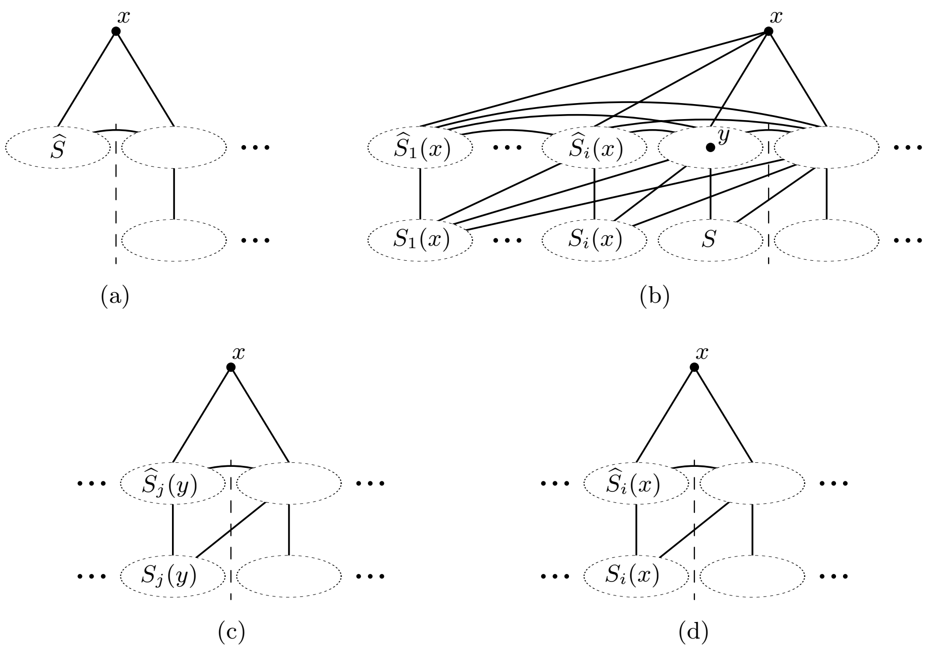

Fig. 5 Cases (a)-(d) of Lemma 3 where $\widehat{S}$ contains a single co-component induced by $V_i(x) \cup \{y\} \cup V_j(y)$ and S contains a single component induced by $V_j(y)$

(i) the number of additional edges in the graph G' (optimal solution for the problem (Cograph,+1)-MinEdgeAddition for G, x, y) is cost$(k_x - 1, k_y - 1)$;

(ii) the value cost(i, j) is the minimum among the costs of the cases below provided that they apply:

(a) $1 + \sum_{t=1}^{i} |S_t(x)| + \sum_{t=1}^{j} \big(|\widehat{S}_t(y)| + |S_t(y)|\big)$;

(b) $1 + \sum_{t=1}^{i} \big(|\widehat{S}_t(x)| + |S_t(x)|\big) + \sum_{t=1}^{j} |S_t(y)|$, *provided that* $j \geq 1$;

(c) $cost(i, j-1) + |\widehat{S}_j(y)| \cdot \Big(1 + \sum_{t=1}^{i} \big(|\widehat{S}_t(x)| + |S_t(x)|\big)\Big)$, *provided that* $j \geq 2$ *or* $j = 1$ *and* $S_1(y) \neq \emptyset$;

(d) $cost(i-1, j) + |\widehat{S}_i(x)| \cdot \Big(1 + \sum_{t=1}^{j} \big(|\widehat{S}_t(y)| + |S_t(y)|\big)\Big)$, *provided that* $i \geq 2$ *or* $i = 1$ *and* $S_1(x) \neq \emptyset$.

It is important to note the symmetry between cases (a) and (b) and between cases (c) and (d) with respect to x and y, as it is expected. Let us now consider some special cases.

- If $i = 0$ and $j = 0$, then only case (a) applies and $cost(0, 0) = 1$.
- If $i = 0$ and $j = 1$, then case (d) does not apply while the cost in case (a) is no smaller that the cost in case (b), thus, $cost(0, 1)$ is the minimum of $1 + \sum_{t=1}^{1} |S_t(y)|$ and of $cost(0, j-1) + |\widehat{S}_j(y)|$ assuming that case (c) applies: if $S_1(y) \neq \emptyset$ then case (c) applies and since $cost(0, 0) = 1$, we have that $cost(0, 1) = \min\{1 + |S_1(y)|, 1 + |\widehat{S}_1(y)|\}$; if $S_1(y) = \emptyset$ then case (c) does

not apply and thus $cost(0,1)$ is $1 + |S_1(y)| = 1$. In either case, $cost(0,1) = \min\{1 + |S_1(y)|, 1 + |\widehat{S}_1(y)|\}$.

- If $i = 1$ and $j = 0$, then cases (b) and (c) do not apply, so that $cost(1,0)$ is the minimum of $1 + \sum_{t=1}^{1} |S_t(x)|$ and of $cost(0,0) + |\widehat{S}_i(x)|$ assuming that case (d) applies. The case is symmetric to the previous case so that $cost(1,0) = \min\{1 + |S_1(x)|, 1 + |\widehat{S}_1(x)|\}$.
- If $i = 0$ and $j \geq 2$, then case (d) does not apply, while the cost in case (a) is no smaller than the cost in case (b) since $1 + \sum_{t=1}^{j} \big(|\widehat{S}_t(y)| + |S_t(y)|\big) < 1 + \sum_{t=1}^{j} |S_t(y)|$; thus, $cost(0,j)$ is the minimum of $1 + \sum_{t=1}^{j} |S_t(y)|$ and of $cost(0,j-1) + |\widehat{S}_j(y)|$.
- If $i \geq 2$ and $j = 0$, then case (c) does not apply, while the cost in case (b) is no smaller than the cost in case (a) since $1 + \sum_{t=1}^{i} \big(|\widehat{S}_t(x)| + |S_t(x)|\big) < 1 + \sum_{t=1}^{i} |S_t(x)|$; thus, $cost(i,0)$ is the minimum of $1 + \sum_{t=1}^{i} |S_t(x)|$ and of $cost(i-1,0) + |\widehat{S}_i(x)|$.

Based on Lemma 4 and the above discussion, we give below our algorithm. The algorithm uses four matrices $A_x[\,]$, $B_x[\,]$, $A_y[\,]$, and $B_y[\,]$, such that $A_v[i] = \sum_{t=1}^{i} |\widehat{S}_t(v)|$ and $B_v[i] = \sum_{t=1}^{i} |S_t(v)|$. It also uses a 2-dimensional array $cost[\,,\,]$, where it saves the values of $cost(\,,\,)$. The algorithm receives as input a cograph G on n vertices and two non-adjacent vertices x, y of G, and outputs the minimum number of edges that need to be added to G so that x, y become adjacent and the resulting graph is a cograph (we note that the algorithm can be easily modified to produce the set of edges that need to be added, instead of their number only, within the same time and space complexity). In detail, it works as follows:

Algorithm Add-Edge-in-Cograph

```
1. Compute the sets
       $\widehat{S}_1(x), \widehat{S}_2(x), \ldots, \widehat{S}_{\ell'_x}(x)$ of co-components of $G[N(x)]$   and
       $S_1(x), S_2(x), \ldots, S_{\ell'_x}(x), S_{\ell'_x+1}(x)$ of conn. components of $G[V(G) - N[x]]$;
   find the set $S_{k_x}(x)$ to which $y$ belongs;
   compute the sets
       $\widehat{S}_1(y), \widehat{S}_2(y), \ldots, \widehat{S}_{\ell'_y}(y)$ of co-components of $G[N(y)]$   and
       $S_1(y), S_2(y), \ldots, S_{\ell'_y}(y), S_{\ell'_y+1}(y)$ of conn. components of $G[V(G) - N[y]]$;
   find the set $S_{k_y}(y)$ to which $x$ belongs;
2. $A_x[0] \leftarrow 0$;      $B_x[0] \leftarrow 0$;
   for $i = 1, 2, \ldots, k_x - 1$ do
       $A_x[i] \leftarrow A_x[i-1] + |\widehat{S}_i(x)|$;
       $B_x[i] \leftarrow B_x[i-1] + |S_i(x)|$;
   $A_y[0] \leftarrow 0$;      $B_y[0] \leftarrow 0$;
   for $i = 1, 2, \ldots, k_y - 1$ do
       $A_y[i] \leftarrow A_y[i-1] + |\widehat{S}_i(y)|$;
       $B_y[i] \leftarrow B_y[i-1] + |S_i(y)|$;
3. $cost[0,0] \leftarrow 1$;
   for $j = 1, 2, \ldots, k_y - 1$ do
       $cost[0,j] \leftarrow \min\{1 + B_y[j],\ cost[0,j-1] + A_y[j] - A_y[j-1]\}$;
```

```
for $i = 1, 2, \ldots, k_x - 1$ do
    $cost[i, 0] \leftarrow \min\{1 + B_x[i],\ cost[i-1, 0] + A_x[i] - A_x[i-1]\}$;
    for $j = 1, 2, \ldots, k_y - 1$ do
        $val1 \leftarrow 1 + B_x[i] + A_y[j] + B_y[j]$;                    $\{case\ (a)\}$
        $val2 \leftarrow 1 + A_x[i] + B_x[i] + B_y[j]$;                    $\{case\ (b)\}$
        if $j \geq 2$ or ($j = 1$ and $B_y[1] \neq 0$)                     $\{case\ (c)\}$
        then $val3 \leftarrow cost[i, j-1] + (A_y[j] - A_y[j-1]) \cdot (1 + A_x[i] + B_x[i])$
        else $val3 \leftarrow n^2$;
        if $i \geq 2$ or ($i = 1$ and $B_x[1] \neq 0$)                     $\{case\ (d)\}$
        then $val4 \leftarrow cost[i-1, j] + (A_x[i] - A_x[i-1]) \cdot (1 + A_y[j] + B_y[j])$
        else $val4 \leftarrow n^2$;
        $cost[i, j] \leftarrow \min\{val1,\ val2,\ val3,\ val4\}$;
return($cost[k_x - 1, k_y - 1]$).
```

Note that whenever a value $cost[\,,\,]$ is needed for another cost-computation, it has already been computed. The correctness of Algorithm Add-Edge-in-Cograph follows from Lemma 4, the discussion of the special cases, and the definitions of the arrays $A_x[\,]$, $B_x[\,]$, $A_y[\,]$, and $B_y[\,]$, which also imply that $A_x[i] - A_x[i-1] = |\widehat{S}_i(x)|$, $B_x[j] - B_x[j-1] = |S_j(x)|$, and similarly for $A_y[\,]$ and $B_y[\,]$.

Time and Space Complexity: Suppose that the input cograph G has n vertices and m edges. Then, the sets $\widehat{S}_1(x), \widehat{S}_2(x), \ldots, \widehat{S}_{\ell'_x}(x)$ and $S_1(x), S_2(x), \ldots, S_{\ell'_x+1}(x)$ can be computed in $O(n+m)$ time and space either by computing the cotree of G [5], or by computing the co-components of $G[N(x)]$ [3,6] and the connected components of $G[V(G) - N[x]]$ and then by placing them in the appropriate $\widehat{S}_i(x)$ or $S_i(x)$ based on their number of incident edges to vertices in $V(G) - N[x]$ and in $N(x)$ respectively. Finding the set $S_{k_x}(x)$ can be done in constant time. Similarly, the computation of the corresponding sets $\widehat{S}_i(y)$ and $S_i(y)$, and finding $S_{k_y}(y)$ takes $O(n+m)$ time and space. For the complexity of Steps 2 and 3, we observe that ℓ'_x and ℓ'_y are $O(\sqrt{m})$: since every vertex in any co-component of $\widehat{S}_i$ ($1 \leq i \leq \ell'_x$) sees every vertex in the co-components of $\widehat{S}_j$ for $j \neq i$, there exist at least $\ell'_x(\ell'_x - 1)/2$ edges connecting vertices in different co-components of $G[N(x)]$; since G contains a total of m edges and there are at least ℓ'_x edges connecting x to its neighbors, we conclude that $m \geq \ell'_x + \ell'_x(\ell'_x - 1)/2 > {\ell'_x}^2/2$, from which the result for ℓ'_x follows; a similar argument holds for ℓ'_y. Step 2 takes $O(\sqrt{m}) = O(n)$ time, since $k_x \leq \ell'_x$ and $k_y \leq \ell'_y$. Step 3 takes $O(k_x \cdot k_y) = O(\ell'_x \cdot \ell'_y) = O(m)$ time. The space needed by Algorithm Add-Edge-in-Cograph is equal to the space needed for the representation of the input graph G and the space taken by the arrays $A_x[\,]$, $B_x[\,]$, $A_y[\,]$, $B_y[\,]$, and $cost[\,,\,]$; hence, it is $O(n + m + k_x \cdot k_y) = O(n+m)$. Therefore, Algorithm Add-Edge-in-Cograph takes $O(n+m)$ time and space.

5 Concluding Remarks

In this paper, we described a linear-time algorithm for the (Cograph,+1)-Min-EdgeAddition problem; instrumental in our construction are the properties of the component-partition of a cograph that we establish. Since the cographs are complement-invariant, the approach we used when applied on the complement of the given graph gives a solution to the (Cograph,−1)-MinEdgeDeletion problem.

It would be interesting to obtain efficient algorithms for the (Cograph,−1)-MinEdgeAddition and the (Cograph,+1)-MinEdgeDeletion problems as well as for the extensions of all these problems in which k edges or non-edges are involved. Finally, it would also be interesting to study the problems for other classes of graphs; an obvious immediate next step would be to consider the class of P_4-sparse graphs, a superclass of the class of cographs.

Acknowledgment. The authors would like to thank A. Brandstädt for proposing the four variants $(\Pi, \pm 1)$-MinEdgeAddition/Deletion problems, D. Corneil for suggesting a solution to the (Cograph,−1)-MinEdgeDeletion problem, and D. Kratsch for constructive suggestions and comments.

References

1. A. Brandstädt, V.B. Le, and J.P. Spinrad, *Graph Classes: A Survey*, SIAM Monographs on Discrete Mathematics and Applications, 1999.
2. A. Bretscher, D. Corneil, M. Habib, and C. Paul, A simple linear time LexBFS cograph recognition algorithm, *Proc. 29th Int'l Workshop on Graph Theoretic Concepts in Comput. Sci. (WG'03)*, LNCS 2880 (2003) 119–130.
3. K.W. Chong, S.D. Nikolopoulos, and L. Palios, An optimal parallel co-connectivity algorithm, *Theory Comput. Systems* **37** (2004) 527–546.
4. D.G. Corneil, H. Lerchs, and L. Stewart-Burlingham, Complement reducible graphs, *Discrete Appl. Math.* **3** (1981) 163–174.
5. D.G. Corneil, Y. Perl, and L.K. Stewart, A linear recognition algorithm for cographs, *SIAM J. Comput.* **14** (1985) 926–934.
6. E. Dahlhaus, J. Gustedt, and R.M. McConnell, Partially Complemented Representations of Digraphs, *Discrete Math. & Theoret. Comput. Sci.* **5** (2002) 147–168.
7. P. Hell, R. Shamir, and R. Sharan, A fully dynamic algorithm for recognizing and representing proper interval graphs, *SIAM J. Comput.* **31** (2002) 289–305.
8. L. Ibarra, Fully dynamic algorithms for chordal graphs, *Proc. 10th Annual ACM-SIAM Symp. on Discrete Algorithms (SODA'99)*, (1999) 923–924.
9. M.C. Golumbic, *Algorithmic Graph Theory and Perfect Graphs*, Academic Press, Inc., 1980.
10. H.A. Jung, On a class of posets and the corresponding comparability graphs, *J. Combin. Theory Ser. B* **24** (1978) 125–133.
11. H. Lerchs, On cliques and kernels, *Technical Report*, Department of Computer Science, University of Toronto, March 1971.
12. R. Lin, S. Olariu, and G. Pruesse, An optimal path cover algorithm for cographs, *Computers Math. Applic.* **30** (1995) 75–83.
13. R. Shamir and R. Sharan, A fully dynamic algorithm for modular decomposition and recognition of cographs, *Discrete Appl. Math.* **136** (2004) 329–340.
14. J.P. Spinrad, *Efficient Graph Representations*, American Mathematical Society, 2003.
15. D.P. Sumner, Dacey graphs, *J. Austral. Math. Soc.* **18** (1974) 492–502.