The 2-terminal-set Path Cover Problem and its Polynomial Solution on Cographs*

Katerina Asdre and Stavros D. Nikolopoulos

Department of Computer Science, University of Ioannina
P.O.Box 1186, GR-45110 Ioannina, Greece
{katerina, stavros}@cs.uoi.gr

Abstract: In this paper, we study a generalization of the path cover problem, namely, the 2-terminal-set path cover problem, or 2TPC for short. Given a graph G and two disjoint subsets $\mathcal{T}^1$ and $\mathcal{T}^2$ of $V(G)$, a 2-terminal-set path cover of G with respect to $\mathcal{T}^1$ and $\mathcal{T}^2$ is a set of vertex-disjoint paths $\mathcal{P}$ that covers the vertices of G such that the vertices of $\mathcal{T}^1$ and $\mathcal{T}^2$ are all endpoints of the paths in $\mathcal{P}$ and all the paths with both endpoints in $\mathcal{T}^1 \cup \mathcal{T}^2$ have one endpoint in $\mathcal{T}^1$ and the other in $\mathcal{T}^2$. The 2TPC problem is to find a 2-terminal-set path cover of G of minimum cardinality; note that, if $\mathcal{T}^1 \cup \mathcal{T}^2$ is empty, the stated problem coincides with the classical path cover problem. The 2TPC problem generalizes some path cover related problems, such as the 1HP and 2HP problems, which have been proved to be NP-complete even for small classes of graphs. We show that the 2TPC problem can be solved in linear time on the class of cographs. The proposed linear-time algorithm is simple, requires linear space, and also enables us to solve the 1HP and 2HP problems on cographs within the same time and space complexity.

Keywords: path cover, fixed-endpoint path cover, perfect graphs, complement reducible graphs, cographs, linear-time algorithms.

1 Introduction

Framework–Motivation. A well studied problem with numerous practical applications in graph theory is to find a minimum number of vertex-disjoint paths of a graph G that cover the vertices of G. This problem, also known as the path cover problem (PC), finds application in the fields of database design, networks, code optimization among many others (see [1, 2, 19, 24]); it is well known that the path cover problem and many of its variants are NP-complete in general graphs [11]. A graph that admits a path cover of size one is referred to as Hamiltonian. Thus, the path cover problem is at least as hard as the Hamiltonian path problem (HP), that is, the problem of deciding whether a graph is Hamiltonian.

Several variants of the HP problem are also of great interest, among which is the problem of deciding whether a graph admits a Hamiltonian path between two points (2HP). The 2HP problem is the same as the HP problem except that in 2HP two vertices of the input graph G are specified, say, u and

*The research Project is co-funded by the European Union - European Social Fund (ESF) & National Sources, in the framework of the program "Pythagoras II" of the "Operational Program for Education and Initial Vocational Training" of the 3rd Community Support Framework of the Hellenic Ministry of Education.

v, and we are asked whether G contains a Hamiltonian path beginning with u and ending with v. Similarly, the 1HP problem is to determine whether a graph G admits a Hamiltonian path starting from a specific vertex u of G, and to find one if such a path does exist. Both 1HP and 2HP problems are also NP-complete in general graphs [11].

The path cover problem and several variants of it have numerous algorithmic applications in many fields. Some that have received both theoretical and practical attention are in the content of communication and/or transposition networks [25]. In such problems, we are given a graph (network) G and

(Problem A) a set $\mathcal{T}$ of $k = 2\lambda$ vertices of G, and the objective is to determine whether G admits a path cover of size λ that contains paths connecting pairs of vertices of $\mathcal{T}$, that is, G admits λ vertex-disjoint paths with both their endpoints in $\mathcal{T}$ (note that, the endpoints of a path P are the first vertex and the last vertex visited by P), or

(Problem B) a set $\mathcal{T}$ of $\lambda = k/2$ pairs of vertices of G (source-sink pairs), and the objective is to determine whether G admits for each pair (a_i, b_i), $1 \leq i \leq \lambda$, a path connecting a_i to b_i such that the set of λ paths forms a path cover.

Another path cover related problem that has received increased attention in recent years is in the context of communication networks. The only efficient way to transmit high volume communication, such as in multimedia applications, is through disjoint paths that are dedicated to pairs of processors. To efficiently utilize the network one needs a simple algorithm that, with minimum overhead, constructs a large number of edge-disjoint paths between pairs of two given sets of requests.

Both problems A and B coincide with the 2HP problem, in the case where $k = 2$. In [9], Damaschke provided a foundation for obtaining polynomial-time algorithms for several problems concerning paths in interval graphs, such as finding Hamiltonian paths and circuits, and partitions into paths. In the same paper, he stated that the complexity status of both 1HP and 2HP problems on interval graphs remains an open question; until now the complexities of 1HP and 2HP keep their difficulties even in the small subclass of split interval graphs – no polynomial algorithm is known.

Motivated by the above issues we state a variant of the path cover problem, namely, the 2-terminal-set path cover problem (2TPC), which generalizes both 1HP and 2HP problems, and also Problem B.

(Problem 2TPC) Let G be a graph and let $\mathcal{T}^1$ and $\mathcal{T}^2$ be two disjoint sets of vertices of $V(G)$. A *2-terminal-set path cover* of the graph G with respect to $\mathcal{T}^1$ and $\mathcal{T}^2$ is a path cover of G such that all vertices in $\mathcal{T}^1 \cup \mathcal{T}^2$ are endpoints of paths in the path cover and all the paths with both endpoints in $\mathcal{T}^1 \cup \mathcal{T}^2$ have one endpoint in $\mathcal{T}^1$ and the other in $\mathcal{T}^2$; a *minimum 2-terminal-set path cover* of G with respect to $\mathcal{T}^1$ and $\mathcal{T}^2$ is a 2-terminal-set path cover of G with minimum cardinality; the *2-terminal-set path cover problem* (2TPC) is to find a minimum 2-terminal-set path cover of the graph G.

Contribution. In this paper, we show that the 2-terminal-set path cover problem (2TPC) has a polynomial-time solution in the class of complement reducible graphs, or cographs [8]. More precisely, we establish a lower bound on the size of a minimum 2-terminal-set path cover of a cograph G on n vertices and m edges. We then define path operations, and prove structural properties for the paths of such a path cover, which enable us to describe a simple algorithm for the 2TPC problem. The proposed algorithm runs in time linear in the size of the input graph G, that is, in $O(n + m)$ time, and requires linear space. Figure 1 shows a diagram of class inclusions for a number of graph classes, subclasses of comparability and chordal graphs, and the current complexity status of the 2TPC problem on these classes; for definitions of the classes shown, see [6, 12]. Note that, if the problem is polynomially solvable on interval graphs, then it is also polynomially solvable on convex graphs [21].

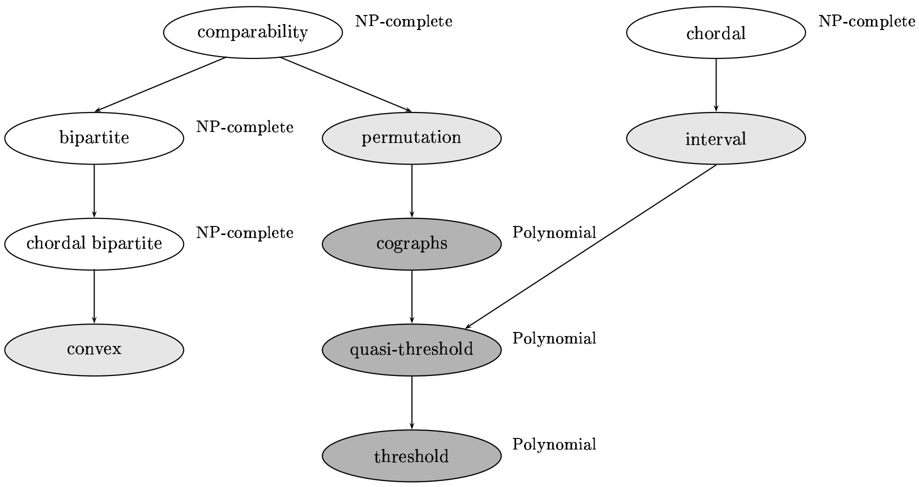

Figure 1: The complexity status (NP-complete, unknown, polynomial) of the 2TPC problem for some graph subclasses of comparability and chordal graphs. $A \rightarrow B$ indicates that class A contains class B.

The proposed algorithm for the 2TPC problem can also be used to solve the 1HP and 2HP problems on cographs within the same time and space complexity. Moreover, we have designed our algorithm so that it produces a minimum 2-terminal-set path cover of a cograph G that contains a large number of paths with one endpoint in $\mathcal{T}^1$ and the other in $\mathcal{T}^2$ (we can easily find a graph G and two sets $\mathcal{T}^1$ and $\mathcal{T}^2$ of vertices of $V(G)$ so that G admits two minimum 2-terminal-set path covers with different numbers of paths having one endpoint in $\mathcal{T}^1$ and the other in $\mathcal{T}^2$; for example, consider the graph G with vertex set $V(G) = \{a, b, c, d\}$, edge set $E(G) = \{ab, bc, ac, cd\}$, and $\mathcal{T}^1 = \{a\}$, $\mathcal{T}^2 = \{b\}$).

Related Work. The class of cographs has been extensively studied and several sequential and/or parallel algorithms for recognition and for classical combinatorial optimization problems have been proposed [8, 18, 19, 22]. Jung [18] studied the existence of a Hamiltonian path or cycle in a cograph, while Lin et al. [19] presented an optimal algorithm for the path cover problem on cographs. Nakano et al. [22] described an optimal parallel algorithm which finds and reports all the paths in a minimum path cover of a cograph in $O(\log n)$ time using $O(n/\log n)$ processors on a PRAM model. Recently, Asdre and Nikolopoulos proposed a linear-time algorithm for the k-fixed-endpoint path cover problem (kPC) on cographs and on proper interval graphs [3, 4]. Algorithms for optimization problems on other related classes of graphs have been also described [5, 13, 14, 15, 16, 23]. Moreover, algorithms for the path cover problem on other classes of graphs were proposed in [2, 10, 17, 24].

2 Theoretical Framework

The cographs admit a tree representation unique up to isomorphism. Specifically, we can associate with every cograph G a unique rooted tree $T_{co}(G)$ called the co-tree (or, modular decomposition tree [20]), which we can construct sequentially in linear time [7, 8]. The co-tree forms the basis for fast algorithms for problems such as isomorphism, coloring, clique detection, clusters, minimum weight dominating sets [6, 7], and also for the path cover problem [19, 22].

For convenience and ease of presentation, we binarize the co-tree $T_{co}(G)$ in such a way that each of its internal nodes has exactly two children [19, 22]. We shall refer to the binarized version of $T_{co}(G)$ as the modified co-tree of G and will denote it by $T(G)$. Thus, the left and right child of an internal node t of $T(G)$ will be denoted by t_ℓ and t_r, respectively. Let t be an internal node of $T(G)$. Then $G[t]$ is the subgraph of G induced by the subset V_t of the vertex set $V(G)$, which contains all the vertices of G that have as common ancestor in $T(G)$ the node t. For simplicity, we will denote by V_ℓ and V_r the vertex sets $V(G[t_\ell])$ and $V(G[t_r])$, respectively.

Let G be a cograph, $\mathcal{T}^1$ and $\mathcal{T}^2$ be two sets of vertices of $V(G)$ such that $\mathcal{T}^1 \cap \mathcal{T}^2 = \emptyset$, and let $\mathcal{P}_{2\mathcal{T}}(G)$ be a minimum 2-terminal-set path cover of G with respect to $\mathcal{T}^1$ and $\mathcal{T}^2$ of size $\lambda_{2\mathcal{T}}$; note that the size of $\mathcal{P}_{2\mathcal{T}}(G)$ is the number of paths it contains. The vertices of the sets $\mathcal{T}^1$ and $\mathcal{T}^2$ are called *terminal* vertices, and the sets $\mathcal{T}^1$ and $\mathcal{T}^2$ are called the *terminal sets* of G, while those of $V(G) - (\mathcal{T}^1 \cup \mathcal{T}^2)$ are called *non-terminal or free* vertices. Thus, the set $\mathcal{P}_{2\mathcal{T}}(G)$ contains three types of paths, which we call *terminal*, *semi-terminal*, and *non-terminal or free* paths:

(i) a *terminal path* P_t consists of at least two vertices and both its endpoints, say, u and v, are terminal vertices belonging to different sets, that is, $u \in \mathcal{T}^1$ and $v \in \mathcal{T}^2$;

(ii) a *semi-terminal path* P_s is a path having one endpoint in $\mathcal{T}^1$ or $\mathcal{T}^2$ and the other in $V(G) - (\mathcal{T}^1 \cup \mathcal{T}^2)$; if P_s consists of only one vertex (trivial path), say, u, then $u \in \mathcal{T}^1 \cup \mathcal{T}^2$;

(iii) a *non-terminal or free path* P_f is a path having both its endpoints in $V(G) - (\mathcal{T}^1 \cup \mathcal{T}^2)$; if P_f consists of only one vertex, say, u, then $u \in V(G) - (\mathcal{T}^1 \cup \mathcal{T}^2)$.

The set of the non-terminal paths in a minimum 2TPC of the graph G is denoted by N, while S and T denote the sets of the semi-terminal and terminal paths, respectively. Furthermore, let S^1 and S^2 denote the sets of the semi-terminal paths such that the terminal vertices belong to $\mathcal{T}^1$ and $\mathcal{T}^2$, respectively. Thus, $|S| = |S^1| + |S^2|$ and the following equation holds.

$$\lambda_{2\mathcal{T}} = |N| + |S| + |T| = |N| + |S^1| + |S^2| + |T| \tag{1}$$

From the definition of the 2-terminal-set path cover problem (2TPC), we can easily conclude that the number of paths in a minimum 2TPC can not be less than the number of the terminal vertices of the terminal set having maximum cardinality. Furthermore, since each semi-terminal path contains one terminal vertex and each terminal path contains two, the number of terminal vertices is equal to $|S| + 2|T| = |S^1| + |S^2| + 2|T|$. Thus, we have the following proposition, which also holds for general graphs:

Proposition 2.1. *Let G be a cograph and let $\mathcal{T}^1$ and $\mathcal{T}^2$ be two disjoint subsets of $V(G)$. Then $|\mathcal{T}^1| = |S^1| + |T|$, $|\mathcal{T}^2| = |S^2| + |T|$ and $\lambda_{2\mathcal{T}} \geq \max\{|\mathcal{T}^1|, |\mathcal{T}^2|\}$.*

Clearly, the size of a 2TPC of a cograph G, as well as the size of a minimum 2TPC of G, is less than or equal to the number of vertices of G, that is, $\lambda_{2\mathcal{T}} \leq |V(G)|$. Let $F(V(G))$ be the set of the free vertices of G; hereafter, $F(V) = F(V(G))$. Furthermore, let $\mathcal{P}$ be a set of paths and let $V_{\mathcal{P}}$ denote the set of vertices belonging to the paths of the set $\mathcal{P}$; hereafter, $F(\mathcal{P}) = F(V_{\mathcal{P}})$. Then, if $\mathcal{T}^1$ and $\mathcal{T}^2$ are two disjoint subsets of $V(G)$, we have $\lambda_{2\mathcal{T}} \leq |F(V)| + |\mathcal{T}^1| + |\mathcal{T}^2|$.

Let t be an internal node of the tree $T(G)$, that is, t is either an S-node or a P-node [20]. Then $\lambda_{2\mathcal{T}}(t)$ denotes the number of paths in a minimum 2TPC of the graph $G[t]$ with respect to $\mathcal{T}_t^1$ and $\mathcal{T}_t^2$, where $\mathcal{T}_t^1$ and $\mathcal{T}_t^2$ are the terminal vertices of $\mathcal{T}^1$ and $\mathcal{T}^2$ of the graph $G[t]$, respectively. Let t_ℓ and t_r be the left and the right child of node t, respectively. We denote by $\mathcal{T}_\ell^1$ and $\mathcal{T}_r^1$ (resp. $\mathcal{T}_\ell^2$ and $\mathcal{T}_r^2$) the terminal vertices of $\mathcal{T}^1$ (resp. $\mathcal{T}^2$) in V_ℓ and V_r, respectively, where $V_\ell = V(G[t_\ell])$ and $V_r = V(G[t_r])$. Let N_ℓ, S_ℓ and T_ℓ be the sets of the non-terminal, semi-terminal and terminal paths in a minimum 2TPC of $G[t_\ell]$, respectively. Similarly, let N_r, S_r and T_r be the sets of the non-terminal, semi-terminal and terminal paths in a minimum 2TPC of $G[t_r]$, respectively. Note that S_ℓ^1 and S_r^1 (resp. S_ℓ^2 and

S_r^2) denote the sets of the semi-terminal paths in a minimum 2TPC of $G[t_\ell]$ and $G[t_r]$, respectively, containing a terminal vertex of $\mathcal{T}^1$ (resp. $\mathcal{T}^2$). Obviously, Eq. (1) holds for $G[t]$ as well, with t being either an S-node or a P-node, that is,

$$\lambda_{2\mathcal{T}}(t) = |N_t| + |S_t| + |T_t| = |N_t| + |S_t^1| + |S_t^2| + |T_t| \tag{2}$$

where N_t, S_t and T_t are the sets of the non-terminal, the semi-terminal and the terminal paths, respectively, in a minimum 2TPC of $G[t]$, that is in $\mathcal{P}_{2\mathcal{T}}(t)$, and S_t^1 and S_t^2 denote the sets of the semi-terminal paths in $\mathcal{P}_{2\mathcal{T}}(t)$ containing a terminal vertex of $\mathcal{T}^1$ and $\mathcal{T}^2$, respectively. If t is a P-node, then $\mathcal{P}_{2\mathcal{T}}(t) = \mathcal{P}_{2\mathcal{T}}(t_\ell) \cup \mathcal{P}_{2\mathcal{T}}(t_r)$, where $\mathcal{P}_{2\mathcal{T}}(t_\ell)$ and $\mathcal{P}_{2\mathcal{T}}(t_r)$ are minimum 2TPCs corresponding to $G[t_\ell]$ and $G[t_r]$, respectively, and $\lambda_{2\mathcal{T}}(t) = \lambda_{2\mathcal{T}}(t_\ell) + \lambda_{2\mathcal{T}}(t_r)$. Furthermore, in the case where t is a P-node, we have

$$\begin{array}{lclcl} |N_t| & = & |N_\ell| + |N_r| & & \\ |S_t| & = & |S_\ell| + |S_r| & = & |S_\ell^1| + |S_\ell^2| + |S_r^1| + |S_r^2| \\ |T_t| & = & |T_\ell| + |T_r| & & \end{array}$$

Thus, we focus on computing a minimum 2TPC of the graph $G[t]$ for the case where t is an S-node. Before describing our algorithm, we establish a lower bound on the size $\lambda_{2\mathcal{T}}(t)$ of a minimum 2TPC $\mathcal{P}_{2\mathcal{T}}(t)$ of a graph $G[t]$. More precisely, we prove the following lemma.

Lemma 2.1. *Let t be an internal node of $T(G)$ and let $\mathcal{P}_{2\mathcal{T}}(t_\ell)$ and $\mathcal{P}_{2\mathcal{T}}(t_r)$ be a minimum 2TPC of $G[t_\ell]$ and $G[t_r]$, respectively. Then $\lambda_{2\mathcal{T}}(t) \geq \max\{\max\{|\mathcal{T}_t^1|, |\mathcal{T}_t^2|\},\ \lambda_{2\mathcal{T}}(t_\ell) - |F(V_r)|,\ \lambda_{2\mathcal{T}}(t_r) - |F(V_\ell)|\}$.*

Proof. Clearly, according to Proposition 2.1 and since $G[t]$ is a cograph, we have $\lambda_{2\mathcal{T}}(t) \geq \max\{|\mathcal{T}_t^1|, |\mathcal{T}_t^2|\}$. We will prove that $\lambda_{2\mathcal{T}}(t) \geq \lambda_{2\mathcal{T}}(t_\ell) - |F(V_r)|$. Assume that $\lambda_{2\mathcal{T}}(t) < \lambda_{2\mathcal{T}}(t_\ell) - |F(V_r)|$. Consider removing from this path cover all the vertices in V_r. What results is a set of paths which is clearly a 2TPC for $G[t_\ell]$. Since the removal of a free vertex in $F(V_r)$ will increase the number of paths by at most one, we obtain a 2TPC of $G[t_\ell]$ of size at most $\lambda_{2\mathcal{T}}(t) + |F(V_r)|$. The assumption $\lambda_{2\mathcal{T}}(t) < \lambda_{2\mathcal{T}}(t_\ell) - |F(V_r)|$ guarantees that $\lambda_{2\mathcal{T}}(t) + |F(V_r)| < \lambda_{2\mathcal{T}}(t_\ell)$, contradicting the minimality of $\mathcal{P}_{2\mathcal{T}}(t_\ell)$. Using similar arguments we can show that $\lambda_{2\mathcal{T}}(t) \geq \lambda_{2\mathcal{T}}(t_r) - |F(V_\ell)|$. Hence, the lemma follows. ∎

We next define four operations on paths of a minimum 2TPC of the graphs $G[t_\ell]$ and $G[t_r]$, namely *break*, *connect*, *bridge* and *insert* operations; these operations are illustrated in Fig. 2.

- *Break* operation: Let $P = [p_1, p_2, \ldots, p_k]$ be a path of $\mathcal{P}_{2\mathcal{T}}(t_r)$ or $\mathcal{P}_{2\mathcal{T}}(t_\ell)$ of length k. We say that we *break* the path P in two paths, say, P_1 and P_2, if we delete an arbitrary edge of P, say the edge $p_i p_{i+1}$ $(1 \leq i < k)$, in order to obtain two paths which are $P_1 = [p_1, \ldots, p_i]$ and $P_2 = [p_{i+1}, \ldots, p_k]$. Note that we can break the path P in at most k trivial paths.

- *Connect* operation: Let P_1 be a non-terminal or a semi-terminal path of $\mathcal{P}_{2\mathcal{T}}(t_\ell)$ (resp. $\mathcal{P}_{2\mathcal{T}}(t_r)$) and let P_2 be a non-terminal or a semi-terminal path of $\mathcal{P}_{2\mathcal{T}}(t_r)$ (resp. $\mathcal{P}_{2\mathcal{T}}(t_\ell)$). We say that we *connect* the path P_1 with the path P_2, if we add an edge which joins two free endpoints of the two paths. Note that if $P_1 \in S_\ell^1$ (resp. $P_1 \in S_r^1$) then, if P_2 is also a semi-terminal path, $P_2 \in S_r^2$ (resp. $P_2 \in S_\ell^2$). Similarly, if $P_1 \in S_\ell^2$ (resp. $P_1 \in S_r^2$) then, if P_2 is also a semi-terminal path, $P_2 \in S_r^1$ (resp. $P_2 \in S_\ell^1$).

- *Bridge* operation: Let P_1 and P_2 be two paths of the set $N_\ell \cup S_\ell^1 \cup S_\ell^2$ (resp. $N_r \cup S_r^1 \cup S_r^2$) and let P_3 be a non-terminal path of the set N_r (resp. N_ℓ). We say that we *bridge* the two paths P_1 and P_2 using path P_3 if we connect a free endpoint of P_1 with one endpoint of P_3 and a free endpoint of P_2 with the other endpoint of P_3. The result is a path having both endpoints in $G[t_\ell]$ (resp. $G[t_r]$). Note that if $P_1 \in S_\ell^1$ (resp. $P_1 \in S_r^1$) then, if P_2 is also a semi-terminal path, $P_2 \in S_\ell^2$

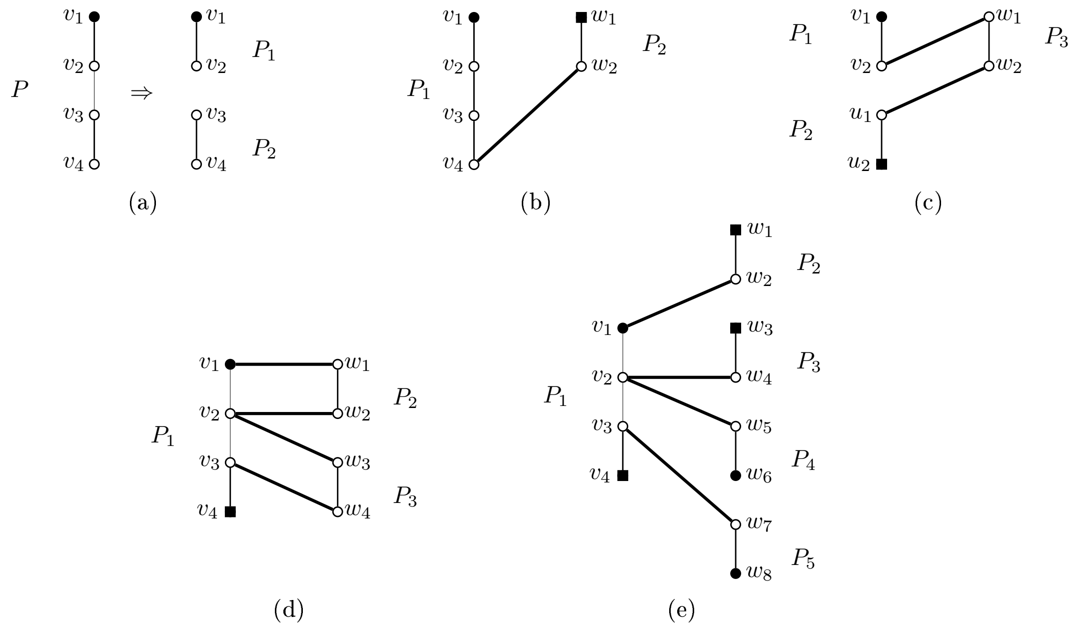

Figure 2: Illustrating (a) break, (b) connect, (c) bridge, (d) insert, and (e) connect-bridge operations; the vertices of $\mathcal{T}^1$ are denoted by black-circles, while the vertices of $\mathcal{T}^2$ are denoted by black-squares.

(resp. $P_2 \in S_r^2$). Similarly, if $P_1 \in S_\ell^2$ (resp. $P_1 \in S_r^2$) then, if P_2 is also a semi-terminal path, $P_2 \in S_\ell^1$ (resp. $P_2 \in S_r^1$).

- *Insert* operation: Let $P_1 = [t_1, p_1, \ldots, p_1', t_1']$ be a terminal path of the set T_ℓ (resp. T_r) and let $P_2 = [p_2, \ldots, p_2']$ be a non-terminal path of the set N_r (resp. N_ℓ). We say that we *insert* the path P_2 into P_1, if we replace the first edge of P_1, that is, the edge $t_1 p_1$, with the path $[t_1, p_2, \ldots, p_2', p_1]$. Thus, the resulting path is $P_1 = [t_1, p_2, \ldots, p_2', p_1, \ldots, p_1', t_1']$. Note that we can replace every edge of the terminal path so that we can insert at most $|F(\{P_1\})| + 1$ non-terminal paths, where $F(\{P_1\})$ is the set of the free vertices belonging to the path P_1. If the terminal path $P_1 = [t_1, p_1, \ldots, p_1^\ell, p_1^r, \ldots, p_1', t_1']$ is constructed by connecting a semi-terminal path of S_ℓ, say, $P_\ell = [t_1, p_1, \ldots, p_1^\ell]$ with a semi-terminal path of S_r, say, $P_r = [p_1^r, \ldots, p_1', t_1']$, then it obviously has one endpoint in $G[t_\ell]$ and the other in $G[t_r]$. In this case, if $P_2 \in N_\ell$ (resp. N_r) we can only replace the edges of P_1 that belong to $G[t_r]$ (resp. $G[t_\ell]$). On the other hand, if P_2 has one endpoint, say, p_2, in N_ℓ and the other, say, p_2', in N_r, we insert P_2 into P_1 as follows: $P_1 = [t_1, p_1, \ldots, p_1^\ell, p_2', \ldots, p_2, p_1^r, \ldots, p_1', t_1']$.

We can also combine the operations connect and bridge to perform a new operation which we call a *connect-bridge* operation; such an operation is depicted in Fig. 2(e) and is defined below.

- *Connect-Bridge* operation: Let $P_1 = [t_1, p_1, \ldots, p_k, t_1']$ be a terminal path of the set T_ℓ (resp. T_r), where $t_1 \in \mathcal{T}^2$ and $t_1' \in \mathcal{T}^1$, and let $P_2, P_3, \ldots, P_{\frac{s+1}{2}}$ be semi-terminal paths of the set S_r^1 (resp. S_ℓ^1) and $P_{\frac{s+1}{2}+1}, \ldots, P_s$ be semi-terminal paths of the set S_r^2 (resp. S_ℓ^2), where s is odd and $3 \le s \le 2k+3$. We say that we *connect-bridge* the paths $P_2, P_3, \ldots, P_s$ using vertices of P_1, if we perform the following operations: (i) connect the path P_2 with the path $[t_1]$; (ii) bridge $r = \frac{s-3}{2}$ pairs of different semi-terminal paths using vertices $p_1, p_2, \ldots, p_r$; and (iii) connect the path $[p_{r+1}, \ldots, p_k, t_1']$ with the last semi-terminal path P_s.

The Connect-Bridge operation produces two paths having one endpoint in $G[t_\ell]$ and the other endpoint in $G[t_r]$ and $\frac{s-3}{2}$ paths having both endpoints in $G[t_r]$ (resp. $G[t_\ell]$).

3 The Algorithm

We next present an optimal algorithm for the 2TPC problem on cographs. Our algorithm takes as input a cograph G and two subsets $\mathcal{T}^1$ and $\mathcal{T}^2$ of its vertices, where $\mathcal{T}^1 \cap \mathcal{T}^2 = \emptyset$, and finds the paths of a minimum 2TPC of G in linear time; it works as follows:

Algorithm Minimum_2TPC

1. Construct the co-tree $T_{co}(G)$ of G and make it binary; let $T(G)$ be the resulting tree;
2. Execute the subroutine *process*(*root*), where *root* is the root node of the tree $T(G)$; the minimum 2TPC $\mathcal{P}_{2\mathcal{T}}(root) = \mathcal{P}_{2\mathcal{T}}(G)$ is the set of paths returned by the subroutine;

where the description of the subroutine *process*( ) is as follows:

process (node t)

Input: node t of the modified co-tree $T(G)$ of the input graph G.

Output: a minimum 2TPC $\mathcal{P}_{2\mathcal{T}}(t)$ of the cograph $G[t]$.

1. `if` t is a leaf
   `then` return($\{u\}$), where u is the vertex associated with the leaf t;
   `else` {t is an internal node that has a left and a right child denoted by t_ℓ and t_r, resp.}
   $\mathcal{P}_{2\mathcal{T}}(t_\ell) \leftarrow process(t_\ell)$;
   $\mathcal{P}_{2\mathcal{T}}(t_r) \leftarrow process(t_r)$;
2. `if` t is a P-node
   `then` return($\mathcal{P}_{2\mathcal{T}}(t_\ell) \cup \mathcal{P}_{2\mathcal{T}}(t_r)$);
3. `if` t is an S-node
   `then if` $|N_\ell| \le |N_r|$ `then` swap($\mathcal{P}_{2\mathcal{T}}(t_\ell), \mathcal{P}_{2\mathcal{T}}(t_r)$);
   $s^1 = |S^1_\ell| - |S^2_r|$;
   $s^2 = |S^2_\ell| - |S^1_r|$;
   `case 1:` $s^1 \ge 0$ and $s^2 \ge 0$
   call procedure $2TPC\_1$;
   `case 2:` $s^1 < 0$ and $s^2 < 0$
   `if` $|N_r| + \min\{|s^1|, |s^2|\} \le |F(S^1_\ell \cup S^2_\ell \cup N_\ell)|$
   `then` call procedure $2TPC\_2\_a$;
   `else` call procedure $2TPC\_2\_b$;
   `case 3:` ($s^1 \ge 0$ and $s^2 < 0$) or ($s^1 < 0$ and $s^2 \ge 0$)
   call procedure $2TPC\_3$;

We next describe the subroutine *process*( ) in the case where t is an S-node of $T(G)$. Note that, if $|N_\ell| \le |N_r|$, we swap $\mathcal{P}_{2\mathcal{T}}(t_\ell)$ and $\mathcal{P}_{2\mathcal{T}}(t_r)$. Thus, we distinguish the following three cases: (1) $s^1 \ge 0$ and $s^2 \ge 0$, (2) $s^1 < 0$ and $s^2 < 0$, and (3) ($s^1 \ge 0$ and $s^2 < 0$) or ($s^1 < 0$ and $s^2 \ge 0$). We next describe case 1; cases 2 and 3 are similar.

Case 1: $s^1 \geq 0$ and $s^2 \geq 0$

Let SN_r be the set of non-terminal paths obtained by breaking the set $S_r^1 \cup S_r^2 \cup N_r$ into $|N_\ell| - 1 + \min\{s^1, s^2\}$ non-terminal paths; thus, $|SN_r| \leq |F(S_r^1 \cup S_r^2 \cup N_r)|$. In the case where $|N_\ell| - 1 + \min\{s^1, s^2\} \geq F(S_r^1 \cup S_r^2 \cup N_r)$, the paths of SN_r are trivial (recall that $F(S_r^1 \cup S_r^2 \cup N_r)$ is the set of free vertices belonging to the set $S_r^1 \cup S_r^2 \cup N_r$). The paths of SN_r are used to bridge at most $2\min\{s^1, s^2\}$ semi-terminal paths of $S_\ell^1 \cup S_\ell^2$ and, if $|SN_r| - \min\{s^1, s^2\} > 0$, at most $|N_\ell|$ non-terminal paths of N_ℓ. We can construct the paths of a 2TPC using the following procedure:

Procedure 2TPC_1

1. connect the $|S_r^2|$ paths of S_r^2 with $|S_r^2|$ paths of S_ℓ^1, and the $|S_r^1|$ paths of S_r^1 with $|S_r^1|$ paths of S_ℓ^2;
2. bridge $2\min\{s^1, s^2\}$ semi-terminal paths of $S_\ell^1 \cup S_\ell^2$ using $\min\{s^1, s^2\}$ paths of SN_r;
3. bridge the non-terminal paths of N_ℓ using $|N_\ell| - 1$ non-terminal paths of SN_r; this produces non-terminal paths with both endpoints in $G[t_\ell]$, unless $|N_\ell| \leq |F(S_r^1 \cup S_r^2 \cup N_r)| - \min\{s^1, s^2\}$ where we obtain one non-terminal path with one endpoint in $G[t_\ell]$ and the other in $G[t_r]$;
4. if $|N_\ell| \leq |F(S_r^1 \cup S_r^2 \cup N_r)| - \min\{s^1, s^2\}$ insert the non-terminal path obtained in Step 3 into one terminal path which is obtained in Step 1;
5. if $|T_r| = |S_\ell^1| = |S_\ell^2| = 0$ and $|F(S_r^1 \cup S_r^2 \cup N_r)| \geq |N_\ell| + 1$ construct a non-terminal path having both of its endpoints in $G[t_r]$ and insert it into a terminal path of T_ℓ;
6. if $|T_r| = |S_r^1| = |S_r^2| = 0$ and $|F(N_r)| \geq |N_\ell| + \min\{s^1, s^2\}$ construct a non-terminal path having both of its endpoints in $G[t_r]$ and use it to connect two semi-terminal paths of $S_\ell^1 \cup S_\ell^2$;
7. if $s^1 - \min\{\min\{s^1, s^2\},\ |F(S_r^1 \cup S_r^2 \cup N_r)|\}$ (resp. $s^2 - \min\{\min\{s^1, s^2\},\ |F(S_r^1 \cup S_r^2 \cup N_r)|\}$) is odd and there is at least one free vertex in $S_r^1 \cup S_r^2 \cup N_r$ which is not used in Steps 1–6, or there is a non-terminal path having one endpoint in $G[t_\ell]$ and the other in $G[t_r]$, connect one non-terminal path with one semi-terminal path of S_ℓ^1 (resp. S_ℓ^2);
8. connect-bridge the rest of the semi-terminal paths of $S_\ell^1 \cup S_\ell^2$ (at most $2(|F(T_r)| + |T_r|)$) using vertices of T_r;
9. insert non-terminal paths obtained in Step 3 into the terminal paths of T_r;

Based on the procedure 2TPC_1, we can compute the cardinality of the sets N_t, S_t^1, S_t^2 and T_t, and thus, since $\lambda'_{2\mathcal{T}}(t) = |N_t| + |S_t| + |T_t|$ and $|S_t| = S_t^1 + S_t^2$, the number of paths in the 2TPC constructed by the procedure at node $t \in T(G)$. In this case, the values of $|N_t|$, $|S_t|$ and $|T_t|$ are the following:

$$
\begin{array}{rcl}
|N_t| & = & \max\{\mu - \alpha,\ 0\} \\
|S_t^1| & = & \min\{\sigma_\ell^1,\ \max\{\sigma_\ell^1 - |F(T_r)| - |T_r|,\ \max\{\sigma_\ell^1 - \sigma_\ell^2, 0\}\}\} \\
|S_t^2| & = & \min\{\sigma_\ell^2,\ \max\{\sigma_\ell^2 - |F(T_r)| - |T_r|,\ \max\{\sigma_\ell^2 - \sigma_\ell^1, 0\}\}\} \\
|S_t| & = & |S_t^1| + |S_t^2| \\
|T_t| & = & |S_r^1| + |S_r^2| + \min\{\min\{s^1,\ s^2\},\ |F(S_r^1 \cup S_r^2 \cup N_r)|\} + |T_\ell| + |T_r| + \frac{\sigma_\ell^1 + \sigma_\ell^2 - |S_t|}{2}
\end{array}
\tag{3}
$$

where

$$\begin{aligned}
\sigma_\ell^1 &= |S_\ell^1| - |S_r^2| - \min\{\min\{s^1,\ s^2\},\ |F(S_r^1 \cup S_r^2 \cup N_r)|\},\\
\sigma_\ell^2 &= |S_\ell^2| - |S_r^1| - \min\{\min\{s^1,\ s^2\},\ |F(S_r^1 \cup S_r^2 \cup N_r)|\},\\
\mu &= \max\{|N_\ell| - \pi_r, \max\{1 - \max\{|S_\ell^1|, |S_\ell^2|\}, 0\}\} - \max\{|F(T_r)| + |T_r| - \min\{\sigma_\ell^1, \sigma_\ell^2\}, 0\} -\\
&\quad \min\{\max\{\min\{|N_\ell| - \pi_r, \delta(\sigma_\ell^1), \delta(\sigma_\ell^2)\}, 0\}, \max\{\min\{F(S_r^1 \cup S_r^2 \cup N_r) - \min\{s^1, s^2\}, 1\}, 0\}\},\\
\alpha &= \min\{\max\{\min\{\pi_r - |N_\ell|,\ 1\},\ 0\},\ \max\{|T_\ell|,\ 0\}\},\ \text{and}\\
\pi_r &= \max\{|F(S_r^1 \cup S_r^2 \cup N_r)| - \min\{s^1,\ s^2\},\ 0\}.
\end{aligned}$$

In Eq. (3), σ_ℓ^1 (resp. σ_ℓ^2) is the number of semi-terminal paths of S_ℓ^1 (resp. S_ℓ^2) that are not connected or bridged at Steps 1–3. Furthermore, π_r is the number of free vertices in the set $S_r^1 \cup S_r^2 \cup N_r$ that are not used to bridge semi-terminal paths of $S_\ell^1 \cup S_\ell^2$ at Step 3 and δ is a function which is defined as follows: $\delta(x) = 1$, if x is odd, and $\delta(x) = 0$ otherwise. Note that at most $|F(T_r)| + |T_r|$ non-terminal paths can be inserted into the terminal paths of T_r or the terminal paths can connect-bridge at most $2(|F(T_r)| + |T_r|)$ semi-terminal paths.

4 Correctness and Time Complexity

Let G be a cograph, $T(G)$ be the modified co-tree of G, and let $\mathcal{T}^1$ and $\mathcal{T}^2$ be the two terminal sets of G. Since our algorithm computes a 2TPC $\mathcal{P}'_{2\mathcal{T}}(t)$ of $G[t]$ of size $\lambda'_{2\mathcal{T}}(t)$ for each internal node $t \in T(G)$, and thus for the root $t = t_{root}$ of the tree $T(G)$, we need to prove that the constructed 2TPC $\mathcal{P}'_{2\mathcal{T}}(t)$ is minimum. Obviously, the size $\lambda_{2\mathcal{T}}(t)$ of a minimum 2TPC of the graph $G[t]$ is less than or equal to the size $\lambda'_{2\mathcal{T}}(t)$ of the 2TPC constructed by our algorithm. According to Proposition 2.1, if the size of the 2TPC constructed by our algorithm is $\lambda'_{2\mathcal{T}}(t) = \max\{|\mathcal{T}_t^1|, |\mathcal{T}_t^2|\}$, then it is a minimum 2TPC. After performing simple computations we get four specific values for the size $\lambda'_{2\mathcal{T}}(t)$ of the 2TPC constructed by our algorithm, that is, by the 2TPC procedures 1, 2_a, 2_b and 3. More precisely, if t is an internal S-node of $T(G)$, our algorithm returns a 2TPC of size $\lambda'_{2\mathcal{T}}(t)$ equal to either $\max\{|\mathcal{T}_t^1|, |\mathcal{T}_t^2|\} + 1$, $\max\{|\mathcal{T}_t^1|, |\mathcal{T}_t^2|\}$, $\lambda_{2\mathcal{T}}(t_\ell) - |F(V_r)|$, or $\lambda_{2\mathcal{T}}(t_r) - |F(V_\ell)|$; see Table 1. Specifically, in the case where $|S_\ell^1| = |S_\ell^2| = |T_r| = |S_r^1| = |S_r^2| = 0$ and $|N_\ell| = |V_r|$ procedure 2TPC_1 returns a 2TPC of the graph $G[t]$ of size $\lambda'_{2\mathcal{T}}(t) = \max\{|\mathcal{T}_t^1|, |\mathcal{T}_t^2|\} + 1$. We prove the following lemma, which shows that if the size of the 2TPC returned by our subroutine process(t) for the graph $G[t]$ is $\lambda'_{2\mathcal{T}}(t) = \max\{|\mathcal{T}_t^1|, |\mathcal{T}_t^2|\} + 1$ (procedure 2TPC_1), then it is a minimum 2TPC.

Lemma 4.1. *Let t be an S-node of $T(G)$ and let $\mathcal{P}_{2\mathcal{T}}(t_\ell)$ and $\mathcal{P}_{2\mathcal{T}}(t_r)$ be a minimum 2TPC of $G[t_\ell]$ and $G[t_r]$, respectively. If $|S_\ell^1| = |S_\ell^2| = |T_r| = |S_r^1| = |S_r^2| = 0$ and $|N_\ell| = |V_r|$, then the procedure 2TPC_1 returns a minimum 2TPC of $G[t]$ of size $\lambda'_{2\mathcal{T}}(t) = \max\{|\mathcal{T}_t^1|, |\mathcal{T}_t^2|\} + 1$.*

Proof. Since we can construct a 2TPC of size $\lambda'_{2\mathcal{T}}(t) = \max\{|\mathcal{T}_t^1|, |\mathcal{T}_t^2|\} + 1$, then the size $\lambda_{2\mathcal{T}}(t)$ of a minimum 2TPC is at most $\max\{|\mathcal{T}_t^1|, |\mathcal{T}_t^2|\} + 1$. We will show that we can not construct a minimum 2TPC of size less than $\max\{|\mathcal{T}_t^1|, |\mathcal{T}_t^2|\} + 1$, that is, we will show that $\lambda_{2\mathcal{T}}(t) \geq \max\{|\mathcal{T}_t^1|, |\mathcal{T}_t^2|\} + 1 \Leftrightarrow \lambda_{2\mathcal{T}}(t) > \max\{|\mathcal{T}_t^1|, |\mathcal{T}_t^2|\}$. Thus, we only need to prove that $\lambda_{2\mathcal{T}}(t) \neq \max\{|\mathcal{T}_t^1|, |\mathcal{T}_t^2|\}$. Note that by the assumption we have $\max\{|\mathcal{T}_t^1|, |\mathcal{T}_t^2|\} = |T_\ell|$. We assume that $\lambda_{2\mathcal{T}}(t) = \max\{|\mathcal{T}_t^1|, |\mathcal{T}_t^2|\}$, and, thus, $\lambda_{2\mathcal{T}}(t) = |T_\ell|$. There exists at least one non-terminal path in $G[t_\ell]$; for otherwise $|N_\ell| = 0$, and thus $V_r = \emptyset$, a contradiction. We ignore the terminal paths from the minimum 2TPC of $G[t_\ell]$ and apply the algorithm described in [19] to $G[t]$. The resulting minimum 2TPC contains only one (non-terminal) path which either has both endpoints in $G[t_\ell]$ or it has one endpoint in $G[t_\ell]$ and the other in $G[t_r]$. This non-terminal path can not be inserted into a terminal path of $G[t_\ell]$ because it does not have both endpoints in $G[t_r]$. Thus, $\lambda_{2\mathcal{T}}(t) = |T_\ell| + 1$, a contradiction. ∎

Procedures	Size of 2-terminal-set PC
Procedure 2TPC_1	$\max\{\lvert\mathcal{T}_t^1\rvert, \lvert\mathcal{T}_t^2\rvert\}+1$
All the procedures	$\max\{\lvert\mathcal{T}_t^1\rvert, \lvert\mathcal{T}_t^2\rvert\}$
Procedures 2TPC_1, 2TPC_2_a and 2TPC_3	$\lambda_{2\mathcal{T}}(t_\ell) - \lvert F(V_r)\rvert$
Procedure 2TPC_2_b	$\lambda_{2\mathcal{T}}(t_r) - \lvert F(V_\ell)\rvert$

Table 1: The size of the 2TPC that our algorithm returns in each case.

Moreover, if the size of the 2TPC returned by the process(t) is $\max\{|\mathcal{T}_t^1|, |\mathcal{T}_t^2|\}$ (all the procedures), then it is obviously a minimum 2TPC of $G[t]$. We prove that the size $\lambda'_{2\mathcal{T}}(t)$ of the 2TPC $\mathcal{P}'_{2\mathcal{T}}(t)$ that our subroutine process(t) returns is minimum.

Lemma 4.2. *Let t be an S-node of $T(G)$ and let $\mathcal{P}_{2\mathcal{T}}(t_\ell)$ and $\mathcal{P}_{2\mathcal{T}}(t_r)$ be a minimum 2TPC of $G[t_\ell]$ and $G[t_r]$, respectively. If the subroutine process(t) returns a 2TPC of $G[t]$ of size $\lambda'_{2\mathcal{T}}(t) = \max\{|\mathcal{T}_t^1|, |\mathcal{T}_t^2|\}$, then $\lambda'_{2\mathcal{T}}(t) \geq \max\{\lambda_{2\mathcal{T}}(t_\ell) - |F(V_r)|,\ \lambda_{2\mathcal{T}}(t_r) - |F(V_\ell)|\}$.*

Proof. Since $\lambda'_{2\mathcal{T}}(t) = \max\{|\mathcal{T}_t^1|, |\mathcal{T}_t^2|\}$, we have $\lambda'_{2\mathcal{T}}(t) = \lambda_{2\mathcal{T}}(t)$, that is, the 2TPC that the subroutine process(t) returns is minimum. Thus, the proof follows from Lemma 2.1. ∎

Let t be an S-node of $T(G)$ and let $\mathcal{P}_{2\mathcal{T}}(t_\ell)$ and $\mathcal{P}_{2\mathcal{T}}(t_r)$ be a minimum 2TPC of $G[t_\ell]$ and $G[t_r]$, respectively. Furthermore, we assume that the conditions $|S_\ell^1| = |S_\ell^2| = |T_r| = |S_r^1| = |S_r^2| = 0$ and $|N_\ell| = |V_r|$ do not hold together. We consider the case where the subroutine process(t) returns a 2TPC $\mathcal{P}'_{2\mathcal{T}}(t)$ of the graph $G[t]$ of size $\lambda'_{2\mathcal{T}}(t) = \lambda_{2\mathcal{T}}(t_\ell) - |F(V_r)|$ (cases 1, 2.a and 3). We prove the following lemma.

Lemma 4.3. *Let t be an S-node of $T(G)$ and let $\mathcal{P}_{2\mathcal{T}}(t_\ell)$ and $\mathcal{P}_{2\mathcal{T}}(t_r)$ be a minimum 2TPC of $G[t_\ell]$ and $G[t_r]$, respectively. If the subroutine process(t) returns a 2TPC of $G[t]$ of size $\lambda'_{2\mathcal{T}}(t) = \lambda_{2\mathcal{T}}(t_\ell) - |F(V_r)|$, then $\lambda'_{2\mathcal{T}}(t) > \max\{\max\{|\mathcal{T}_t^1|, |\mathcal{T}_t^2|\},\ \lambda_{2\mathcal{T}}(t_r) - |F(V_\ell)|\}$.*

Similarly we can show that if the subroutine process(t) returns a 2TPC of $G[t]$ of size $\lambda'_{2\mathcal{T}}(t) = \lambda_{2\mathcal{T}}(t_r) - |F(V_\ell)|$ (case 2.b), then $\lambda'_{2\mathcal{T}}(t) > \max\{\max\{|\mathcal{T}_t^1|, |\mathcal{T}_t^2|\},\ \lambda_{2\mathcal{T}}(t_\ell) - |F(V_r)|\}$. Thus, we can prove the following result.

Lemma 4.4. *Let t be an S-node of $T(G)$ and let $\mathcal{P}_{2\mathcal{T}}(t_\ell)$ and $\mathcal{P}_{2\mathcal{T}}(t_r)$ be a minimum 2TPC of $G[t_\ell]$ and $G[t_r]$, respectively. The subroutine process(t) returns a 2TPC $\mathcal{P}_{2\mathcal{T}}(t)$ of $G[t]$ of size*

$$\lambda'_{2\mathcal{T}}(t) = \begin{cases} \max\{|\mathcal{T}_t^1|, |\mathcal{T}_t^2|\}+1 & \text{if } |N_\ell| = |V_r| \text{ and} \\ & |S_\ell^1| = |S_\ell^2| = |\mathcal{T}_r^1| = |\mathcal{T}_r^2| = 0, \\ \max\{\max\{|\mathcal{T}_t^1|, |\mathcal{T}_t^2|\}, \lambda_{2\mathcal{T}}(t_\ell) - |F(V_r)|, \lambda_{2\mathcal{T}}(t_r) - |F(V_\ell)|\} & \text{otherwise.} \end{cases}$$

Obviously, a minimum 2TPC of the graph $G[t]$ is of size $\lambda_{2\mathcal{T}}(t) \leq \lambda'_{2\mathcal{T}}(t)$. On the other hand, we have proved a lower bound for the size $\lambda_{2\mathcal{T}}(t)$ of a minimum 2TPC of the graph $G[t]$ (see Lemma 2.1), namely,

$\lambda_{2\mathcal{T}}(t) \geq \max\{\max\{|\mathcal{T}_t^1|, |\mathcal{T}_t^2|\},\ \lambda_{2\mathcal{T}}(t_\ell) - |F(V_r)|,\ \lambda_{2\mathcal{T}}(t_r) - |F(V_\ell)|\}$. It follows that $\lambda'_{2\mathcal{T}}(t) = \lambda_{2\mathcal{T}}(t)$, and, thus, we can state the following result.

Lemma 4.5. *Subroutine process(t) returns a minimum 2TPC $\mathcal{P}_{2\mathcal{T}}(t)$ of the graph $G[t]$, for every internal S-node $t \in T(G)$.*

Since the above result holds for every S-node t of the modified co-tree $T(G)$, it also holds when t is the root of $T(G)$ and $\mathcal{T}_t^1 = \mathcal{T}^1$ and $\mathcal{T}_t^2 = \mathcal{T}^2$. Thus, the following theorem holds:

Theorem 4.1. *Let G be a cograph and let $\mathcal{T}^1$ and $\mathcal{T}^2$ be two disjoint subsets of $V(G)$. Let t be the root of the modified co-tree $T(G)$, and let $\mathcal{P}_{2\mathcal{T}}(t_\ell)$ and $\mathcal{P}_{2\mathcal{T}}(t_r)$ be a minimum 2TPC of $G[t_\ell]$ and $G[t_r]$, respectively. Algorithm Minimum_2TPC correctly computes a minimum 2TPC of $G = G[t]$ with respect to $\mathcal{T}^1 = \mathcal{T}_t^1$ and $\mathcal{T}^2 = \mathcal{T}_t^2$ of size $\lambda_{2\mathcal{T}} = \lambda_{2\mathcal{T}}(t)$, where*

$$\lambda_{2\mathcal{T}}(t) = \begin{cases} \lambda_{2\mathcal{T}}(t_r) + \lambda_{2\mathcal{T}}(t_\ell) & \text{if } t \text{ is a P-node,} \\ \max\{|\mathcal{T}_t^1|, |\mathcal{T}_t^2|\} + 1 & \text{if } t \text{ is an S-node and} \\ & |N_\ell| = |V_r| \text{ and} \\ & |S_\ell^1| = |S_\ell^2| = |\mathcal{T}_r^1| = |\mathcal{T}_r^2| = 0, \\ \max\{\max\{|\mathcal{T}_t^1|, |\mathcal{T}_t^2|\}, \lambda_{2\mathcal{T}}(t_\ell) - |F(V_r)|, \lambda_{2\mathcal{T}}(t_r) - |F(V_\ell)|\} & \text{otherwise.} \end{cases}$$

Let G be a cograph on n vertices and m edges, $\mathcal{T}^1$ and $\mathcal{T}^2$ be two terminal sets, and let t be an S-node of the modified co-tree $T(G)$. From the description of the algorithm we can easily conclude that a minimum 2TPC $\mathcal{P}_{2\mathcal{T}}(t)$ of $G[t]$ can be constructed in $O(E(G[t]))$ time, since we use at most $|V(G[t_\ell])| \cdot |V(G[t_r])|$ edges to connect the paths of the minimum 2TPCs of the graphs $G[t_\ell]$ and $G[t_r]$; in the case where t is a P-node a minimum 2TPC is constructed in $O(1)$ time. Thus, the time needed by the subroutine process(t) to compute a minimum 2TPC in the case where t is the root of the tree $T(G)$ is $O(n+m)$; moreover, through the execution of the subroutine no additional space is needed. The construction of the co-tree $T_{co}(G)$ of G needs $O(n+m)$ time and it requires $O(n)$ space [7, 8]. Furthermore, the binarization process of the co-tree, that is, the construction of the modified co-tree $T(G)$, takes $O(n)$ time. Hence, we can state the following result.

Theorem 4.2. *Let G be a cograph on n vertices and m edges and let $\mathcal{T}^1$ and $\mathcal{T}^2$ be two disjoint subsets of $V(G)$. A minimum 2-terminal-set path cover $\mathcal{P}_{2\mathcal{T}}$ of G can be computed in $O(n+m)$ time and space.*

References

[1] G.S. Adhar and S. Peng, Parallel algorithm for path covering, Hamiltonian path, and Hamiltonian cycle in cographs, *Int'l Conference on Parallel Processing*, Vol. III: Algorithms and Architecture, Pennsylvania State University Press, 1990, pp. 364–365.

[2] S.R. Arikati and C.P. Rangan, Linear algorithm for optimal path cover problem on interval graphs, *Inform. Process. Lett.* **35** (1990) 149–153.

[3] K. Asdre and S.D. Nikolopoulos, A Linear-time Algorithm for the k-fixed-endpoint path cover problem on cographs, *Networks* **50** (2007) 231–240.

[4] K. Asdre and S.D. Nikolopoulos, A polynomial solution for the k-fixed-endpoint path cover problem on proper interval graphs, *Proc. 18th International Conference on Combinatorial Algorithms (IWOCA'07)*, Lake Macquarie, Newcastle, Australia, 2007.

[5] K. Asdre, S.D. Nikolopoulos, and C. Papadopoulos, An optimal parallel solution for the path cover problem on P_4-sparse graphs, *J. Parallel Distrib. Comput.* **67** (2007) 63–76.

[6] A. Brandstädt, V.B. Le, and J. Spinrad, *Graph classes – A survey*, SIAM Monographs in Discrete Mathematics and Applications, SIAM, Philadelphia, 1999.

[7] D.G. Corneil, H. Lerchs, and L. Stewart Burlingham, Complement reducible graphs, *Discrete. Appl. Math.* **3** (1981) 163–174.

[8] D.G. Corneil, Y. Perl, and L.K. Stewart, A linear recognition algorithm for cographs, *SIAM J. Comput.* **14** (1985) 926–984.

[9] P. Damaschke, Paths in interval graphs and circular arc graphs, *Discrete Math.* **112** (1993) 49–64.

[10] P. Damaschke, J.S. Deogun, D. Kratsch, and G. Steiner, Finding Hamiltonian paths in cocomparability graphs using the bump number algorithm, *Order* **8** (1992) 383–391.

[11] M.R. Garey and D.S. Johnson, *Computers and intractability: A guide to the theory of NP-completeness*, W.H. Freeman, San Francisco, 1979.

[12] M.C. Golumbic, *Algorithmic Graph Theory and Perfect Graphs*, Academic Press, New York, 1980. Second edition, Annals of Discrete Mathematics 57, Elsevier, 2004.

[13] W. Hochstättler and G. Tinhofer, Hamiltonicity in graphs with few P_4's, *Computing* **54** (1995) 213–225.

[14] S.Y. Hsieh, An efficient parallel strategy for the two-fixed-endpoint Hamiltonian path problem on distance-hereditary graphs, *J. Parallel Distrib. Comput.* **64** (2004) 662–685.

[15] S.Y. Hsieh, C.W. Ho, T.S. Hsu, and M.T. Ko, The Hamiltonian problem on distance-hereditary graphs, *Discrete. Appl. Math.* **154** (2006) 508–524.

[16] R.W. Hung and M.S. Chang, Linear-time algorithms for the Hamiltonian problems on distance-hereditary graphs, *Theoret. Comput. Sci.* **341** (2005) 411–440.

[17] R.W. Hung and M.S. Chang, Solving the path cover problem on circular-arc graphs by using an approximation algorithm, *Discrete. Appl. Math.* **154** (2006) 76–105.

[18] H.A. Jung, On a class of posets and the corresponding comparability graphs, *J. Combinatorial Theory (B)* **24** (1978) 125–133.

[19] R. Lin, S. Olariu, and G. Pruesse, An optimal path cover algorithm for cographs, *Comput. Math. Appl.* **30** (1995) 75–83.

[20] R.M. McConnell and J. Spinrad, Modular decomposition and transitive orientation, *Discrete Math.* **201** (1999) 189–241.

[21] H. Müller, Hamiltonian circuits in chordal bipartite graphs, *Discrete Math.* **156** (1996) 291–298.

[22] K. Nakano, S. Olariu, and A.Y. Zomaya, A time-optimal solution for the path cover problem on cographs, *Theoret. Comput. Sci.* **290** (2003) 1541–1556.

[23] S.D. Nikolopoulos, Parallel algorithms for Hamiltonian problems on quasi-threshold graphs, *J. Parallel Distrib. Comput.* **64** (2004) 48–67.

[24] R. Srikant, R. Sundaram, K.S. Singh, and C.P. Rangan, Optimal path cover problem on block graphs and bipartite permutation graphs, *Theoret. Comput. Sci.* **115** (1993) 351–357.

[25] Y. Suzuki, K. Kaneko, and M. Nakamori, Node-disjoint paths algorithm in a transposition graph, *IEICE Trans. Inf. & Syst.* **E89-D** (2006) 2600–2605.

APPENDIX

(To assist the reviewers)

A.1. The Cases 2 and 3 of the Subroutine process( )

Case 2: $s^1 < 0$ and $s^2 < 0$

In this case, we need $|N_r| + \min\{|s^1|, |s^2|\}$ paths of $G[t_\ell]$ in order to bridge $|N_r|$ non-terminal paths of N_r and $2\min\{|s^1|, |s^2|\}$ semi-terminal paths of $S_r^1 \cup S_r^2$. If $|N_\ell| < |N_r| + \min\{|s^1|, |s^2|\}$ we break the non-terminal paths of N_ℓ into at most $|F(N_\ell)|$ paths; in the case where $|F(N_\ell)| < |N_r| + \min\{|s^1|, |s^2|\}$ we also use (at most $|F(S_\ell^1 \cup S_\ell^2)|$) vertices of $S_\ell^1 \cup S_\ell^2$. Let $p = \min\{|N_r| + \min\{|s^1|, |s^2|\},\ |F(S_\ell^1 \cup S_\ell^2 \cup N_\ell)|\}$. We distinguish two cases:

2.a $|N_r| + \min\{|s^1|, |s^2|\} \leq |F(S_\ell^1 \cup S_\ell^2 \cup N_\ell)|$.

In this case, $p = |N_r| + \min\{|s^1|, |s^2|\}$ and the number of non-terminal paths (or free vertices) of $G[t_\ell]$ is sufficient to bridge non-terminal paths of N_r and semi-terminal paths of $S_r^1 \cup S_r^2$. In detail, let SN_ℓ be the set of non-terminal paths obtained by breaking the set $S_\ell^1 \cup S_\ell^2 \cup N_\ell$ into p non-terminal paths in order to bridge $2\min\{|s^1|, |s^2|\}$ semi-terminal paths of $S_r^1 \cup S_r^2$ and all the non-terminal paths of N_r. If $p < |N_\ell|$ then $SN_\ell = N_\ell$. Obviously, $|SN_\ell| \leq |F(S_\ell^1 \cup S_\ell^2 \cup N_\ell)|$. Note that, if $p < |N_\ell|$ then the non-terminal paths of N_r are used to bridge the paths of N_ℓ. More precisely, we use paths of the set SN_r (it is the set of non-terminal paths that we get by breaking the set $S_r^1 \cup S_r^2 \cup N_r$) in order to obtain $|N_\ell| - \min\{|s^1|, |s^2|\}$ non-terminal paths. If $p \geq |N_\ell|$ we set $SN_r = N_r$ and we use at most $|F(S_\ell^1 \cup S_\ell^2 \cup N_\ell)|$ paths obtained by $S_\ell^1 \cup S_\ell^2 \cup N_\ell$ in order to bridge non-terminal paths of N_r and semi-terminal paths of $S_r^1 \cup S_r^2$, that is, we use the set SN_ℓ. As a result, we construct $\min\{|s^1|, |s^2|\}$ terminal paths having both of their endpoints in $G[t_r]$ and we have at least one non-terminal path, if $p < |N_\ell|$, and exactly one non-terminal path, otherwise. Note that, in the second case, we can construct the non-terminal path in such a way that one endpoint is in SN_ℓ and the other is in N_r. We construct the paths of a 2TPC at node $t \in T(G)$ using the following procedure:

Procedure 2TPC_2_a

1. connect the $|S_\ell^1|$ paths of S_ℓ^1 with $|S_\ell^1|$ paths of S_r^2, and the $|S_\ell^2|$ paths of S_ℓ^2 with $|S_\ell^2|$ paths of S_r^1;
2. if $|T_\ell| = |T_r| = 0$ and $p \geq |N_\ell|$, use N_r to bridge $p - \min\{|s^1|, |s^2|\} + 1$ paths of SN_ℓ and use the constructed non-terminal path having both of its endpoints in $G[t_\ell]$ to bridge two semi-terminal paths of $S_r^1 \cup S_r^2$;
3. bridge semi-terminal paths of $S_r^1 \cup S_r^2$ using paths of SN_ℓ;
4. if $|T_r| = 0, |T_\ell| \neq 0$, $p \geq |N_\ell|$ and $|F(S_r^1 \cup S_r^2 \cup N_r)| \geq |SN_\ell| - \min\{|s^1|, |s^2|\}$ construct a non-terminal path having both of its endpoints in $G[t_r]$ and use a terminal path of T_ℓ to insert the constructed non-terminal path;
5. bridge the remaining paths of SN_ℓ using the paths of SN_r. This produces non-terminal paths one of which has one endpoint in $G[t_\ell]$ and the other in $G[t_r]$;
6. if $|s^2| - \min\{|s^1|, |s^2|\}$ (resp. $|s^1| - \min\{|s^1|, |s^2|\}$) is odd, we connect one non-terminal path with one semi-terminal path of S_r^1 (resp. S_r^2);
7. insert at most $|F(T_r)| + |T_r|$ non-terminal paths obtained in Step 5 into the terminal paths of T_r;

Based on the path operations performed by procedure 2TPC_2_a, we can compute the cardinalities of the sets N_t, S_t and T_t:

$$\begin{array}{rcl}
|N_t| & = & \max\{\mu - \alpha,\ 0\} \\
|S_t^1| & = & \max\{|s^2| - |s^1|,\ 0\} \\
|S_t^2| & = & \max\{|s^1| - |s^2|,\ 0\} \\
|S_t| & = & |S_t^1| + |S_t^2| \\
|T_t| & = & |S_\ell^1| + |S_\ell^2| + \min\{|s^1|,\ |s^2|\} + |T_\ell| + |T_r|
\end{array} \tag{4}$$

where

$$\begin{array}{rcl}
\mu & = & \max\{|N_\ell| - F(S_r^1 \cup S_r^2 \cup N_r),\ 0\} - \min\{|s^1|,\ |s^2|\} - |F(T_r)| - |T_r| - \\
 & & \max\{\delta(|s^1| - \min\{|s^1|,\ |s^2|\}), \delta(|s^2| - \min\{|s^1|,\ |s^2|\})\},\ \text{and} \\
\alpha & = & \min\{\max\{\min\{F(S_r^1 \cup S_r^2 \cup N_r) - |N_\ell|,\ 1\},\ 0\},\ \max\{|T_\ell|,\ 0\}\}.
\end{array}$$

2.b $|N_r| + \min\{|s^1|, |s^2|\} > |F(S_\ell^1 \cup S_\ell^2 \cup N_\ell)|$.

In this case, $p = |F(S_\ell^1 \cup S_\ell^2 \cup N_\ell)|$ and the number of free vertices of $G[t_\ell]$ (that is, in $S_\ell^1 \cup S_\ell^2 \cup N_\ell$) is not sufficient to bridge non-terminal paths of N_r and semi-terminal paths of $S_r^1 \cup S_r^2$. In detail, let SN_ℓ be the set of the trivial, non-terminal paths, obtained by breaking the set $S_\ell^1 \cup S_\ell^2 \cup N_\ell$ into $|F(S_\ell^1 \cup S_\ell^2 \cup N_\ell)|$ non-terminal paths. We can construct the paths of a 2TPC at node $t \in T(G)$ using the following procedure:

Procedure 2TPC_2_b

1. connect the $|S_\ell^1|$ paths of S_ℓ^1 with $|S_\ell^1|$ paths of S_r^2, and the $|S_\ell^2|$ paths of S_ℓ^2 with $|S_\ell^2|$ paths of S_r^1;
2. bridge $2\min\{|s^1|, |s^2|\}$ semi-terminal paths of $S_r^1 \cup S_r^2$ using $\min\{|s^1|, |s^2|\}$ paths of SN_ℓ;
3. bridge the non-terminal paths of N_r using the rest of the non-terminal paths of SN_ℓ. This produces non-terminal paths such that both endpoints belong to $G[t_r]$;
4. connect-bridge the rest of the semi-terminal paths of $S_r^1 \cup S_r^2$ (at most $2(|F(T_\ell)| + |T_\ell|)$) using vertices of T_ℓ;
5. insert non-terminal paths obtained in Step 3 into the terminal paths of T_ℓ;

Based on the procedure 2TPC_2_b, we can compute the cardinalities of non-terminal, semi-terminal and terminal sets:

$$\begin{array}{rcl}
|N_t| & = & \max\{\mu,\ 0\} \\
|S_t^1| & = & \min\{\sigma_r^1,\ \max\{\sigma_r^1 - |F(T_\ell)| - |T_\ell|,\ \max\{\sigma_r^1 - \sigma_r^2, 0\}\}\} \\
|S_t^2| & = & \min\{\sigma_r^2,\ \max\{\sigma_r^2 - |F(T_\ell)| - |T_\ell|,\ \max\{\sigma_r^2 - \sigma_r^1, 0\}\}\} \\
|S_t| & = & |S_t^1| + |S_t^2| \\
|T_t| & = & |S_\ell^1| + |S_\ell^2| + \min\{\min\{|s^1|,\ |s^2|\},\ |F(S_\ell^1 \cup S_\ell^2 \cup N_\ell)|\} + |T_\ell| + |T_r| + \frac{\sigma_r^1 + \sigma_r^2 - |S_t|}{2}
\end{array} \tag{5}$$

where

$$\begin{array}{rcl}
\sigma_r^1 & = & |S_r^1| - |S_\ell^2| - \min\{\min\{|s^1|,\ |s^2|\},\ |F(S_\ell^1 \cup S_\ell^2 \cup N_\ell)|\}, \\
\sigma_r^2 & = & |S_r^2| - |S_\ell^1| - \min\{\min\{|s^1|,\ |s^2|\},\ |F(S_\ell^1 \cup S_\ell^2 \cup N_\ell)|\}, \\
\mu & = & |N_r| - \pi_\ell - \min\{\max\{\min\{|F(S_\ell^1 \cup S_\ell^2 \cup N_\ell)| - \min\{|s^1|,\ |s^2|\},\ 1\},\ 0\}, \\
 & & \max\{\min\{|N_r| - \pi_\ell,\ \delta(|s^1|),\ \delta(|s^2|)\},\ 0\}\} - \max\{|F(T_\ell)| + |T_\ell| - \min\{\sigma_r^1, \sigma_r^2\},\ 0\},\ \text{and} \\
\pi_\ell & = & \max\{|F(S_\ell^1 \cup S_\ell^2 \cup N_\ell)| - \min\{|s^1|,\ |s^2|\},\ 0\}.
\end{array}$$

In Eq. (5), σ_r^1 (resp. σ_r^2) is the number of semi-terminal paths of S_r^1 (resp. S_r^2) that are not connected or bridged at Steps 1–3. Moreover, π_ℓ is the number of free vertices that belong to the set $S_\ell^1 \cup S_\ell^2 \cup N_\ell$ and are not used to bridge semi-terminal paths of $S_r^1 \cup S_r^2$ (at Step 3). Again, $\delta(x) = 1$, if x is odd, and $\delta(x) = 0$ otherwise. Note that at most $|F(T_\ell)| + |T_\ell|$ non-terminal paths can be inserted into the terminal paths of T_ℓ or the terminal paths can connect-bridge at most $2(|F(T_\ell)| + |T_\ell|)$ semi-terminal paths.

Case 3: ($s^1 \geq 0$ and $s^2 < 0$) or ($s^1 < 0$ and $s^2 \geq 0$)

Let SN_r be the set of non-terminal paths which are used to bridge at most $|N_\ell|$ non-terminal paths of N_ℓ; it is obtained by breaking the set $S_r^1 \cup S_r^2 \cup N_r$ into $|N_\ell| - 1$ non-terminal paths. In the case where $|N_\ell| - 1 \geq F(S_r^1 \cup S_r^2 \cup N_r)$, the paths of SN_r are trivial. We can construct the paths of a 2TPC using the following procedure:

Procedure 2TPC_3

1. connect $\min\{|S_\ell^1|, |S_r^2|\}$ paths of S_ℓ^1 with $\min\{|S_\ell^1|, |S_r^2|\}$ paths of S_r^2, and $\min\{|S_\ell^2|, |S_r^1|\}$ paths of S_ℓ^2 with $\min\{|S_\ell^2|, |S_r^1|\}$ paths of S_r^1;
2. bridge the non-terminal paths of N_ℓ using $|N_\ell| - 1$ non-terminal paths of SN_r; this produces non-terminal paths with both endpoints in $G[t_\ell]$, unless $|N_\ell| \leq |F(S_r^1 \cup S_r^2 \cup N_r)|$ where we obtain one non-terminal path with one endpoint in $G[t_\ell]$ and the other in $G[t_r]$;
3. if $|N_\ell| \leq |F(S_r^1 \cup S_r^2 \cup N_r)|$ insert the non-terminal path obtained in Step 2 into one terminal path which is obtained in Step 1;
4. if there is at least one free vertex in $S_r^1 \cup S_r^2 \cup N_r$ which is not used in Steps 1–3, or there is a non-terminal path having one endpoint in $G[t_\ell]$ and the other in $G[t_r]$, connect one non-terminal path with one semi-terminal path of $S_\ell^1 \cup S_\ell^2$;
5. if there is a non-terminal path having at least one endpoint in $G[t_\ell]$, connect it with one semi-terminal path of $S_r^1 \cup S_r^2$;
6. insert non-terminal paths obtained in Step 2 into the terminal paths of T_r;

Based on the procedure 2TPC_3, we can compute the values of $|N_t|$, $|S_t|$ and $|T_t|$:

$$\begin{array}{rcl} |N_t| & = & \max\{|N_\ell| - |F(V_r)| - |T_r| - \max\{\sigma_r^1,\ \sigma_r^2\},\ 0\} \\ |S_t^1| & = & \sigma_\ell^1 + \sigma_r^1 \\ |S_t^2| & = & \sigma_\ell^2 + \sigma_r^2 \\ |S_t| & = & |S_t^1| + |S_t^2| \\ |T_t| & = & \min\{|S_\ell^1|,\ |S_r^2|\} + \min\{|S_\ell^2|,\ |S_r^1|\} + |T_\ell| + |T_r| \end{array} \qquad (6)$$

where

$$\begin{array}{rcl} \sigma_\ell^1 & = & \max\{|S_\ell^1| - |S_r^2|,\ 0\}, \\ \sigma_\ell^2 & = & \max\{|S_\ell^2| - |S_r^1|,\ 0\}, \\ \sigma_r^1 & = & \max\{|S_r^1| - |S_\ell^2|,\ 0\},\ \text{and} \\ \sigma_r^2 & = & \max\{|S_r^2| - |S_\ell^1|,\ 0\}. \end{array}$$

In Eq. (6), σ_ℓ^1 (resp. σ_ℓ^2) is the number of semi-terminal paths of S_ℓ^1 (resp. S_ℓ^2) that are not connected at Step 1 (resp. Step 2) and σ_r^1 (resp. σ_r^2) is the number of semi-terminal paths of S_r^1 (resp. S_r^2) that are not connected at Step 2 (resp. Step 1).

A.2. Proof of Lemma 4.3

Lemma 4.3. *Let t be an S-node of $T(G)$ and let $\mathcal{P}_{2\mathcal{T}}(t_\ell)$ and $\mathcal{P}_{2\mathcal{T}}(t_r)$ be a minimum 2TPC of $G[t_\ell]$ and $G[t_r]$, respectively. If the subroutine process(t) returns a 2TPC of $G[t]$ of size $\lambda'_{2\mathcal{T}}(t) = \lambda_{2\mathcal{T}}(t_\ell) - |F(V_r)|$, then $\lambda'_{2\mathcal{T}}(t) > \max\{\max\{|\mathcal{T}_t^1|, |\mathcal{T}_t^2|\},\ \lambda_{2\mathcal{T}}(t_r) - |F(V_\ell)|\}$.*

Proof. We consider the cases 1, 2.a and 3. In these cases, the size $\lambda'_{2\mathcal{T}}(t)$ of the constructed 2TPC is computed using Eqs. (3), (4) and (6) and the fact that $\lambda'_{2\mathcal{T}}(t) = |N_t| + |S_t| + |T_t|$. After performing simple computations, we conclude that in these cases the subroutine process(t) (cases 1, 2.a) returns $\lambda'_{2\mathcal{T}}(t) = \lambda_{2\mathcal{T}}(t_\ell) - |F(V_r)|$ if the following condition holds:

$$|N_\ell| - \max\{|s^1|, |s^2|\} > |F(V_r)| + |T_r|. \tag{7}$$

In case 3 subroutine process(t) returns $\lambda'_{2\mathcal{T}}(t) = \lambda_{2\mathcal{T}}(t_\ell) - |F(V_r)|$ if the following condition holds:

$$|N_\ell| + \min\{s^1, s^2\} > |F(V_r)| + |T_r|. \tag{8}$$

We will show that (i) $\lambda'_{2\mathcal{T}}(t) > \max\{|\mathcal{T}_t^1|, |\mathcal{T}_t^2|\}$ and, (ii) $\lambda'_{2\mathcal{T}}(t) > \lambda_{2\mathcal{T}}(t_r) - |F(V_\ell)|$. According to Proposition 2.1 and since $G[t]$ is a cograph, we have:

$$|\mathcal{T}_t^1| = |S_\ell^1| + |S_r^1| + |T_\ell| + |T_r| \text{ and } |\mathcal{T}_t^2| = |S_\ell^2| + |S_r^2| + |T_\ell| + |T_r|.$$

(i) We first show that $\lambda'_{2\mathcal{T}}(t) = \lambda_{2\mathcal{T}}(t_\ell) - |F(V_r)| > \max\{|\mathcal{T}_t^1|, |\mathcal{T}_t^2|\}$. Consider the case where $s^1 \geq 0$ and $s^2 \geq 0$, and let $s^1 < s^2$; equivalently,

$$|S_\ell^1| - |S_r^2| < |S_\ell^2| - |S_r^1| \;\Leftrightarrow |S_\ell^1| + |S_r^1| + |T_\ell| + |T_r| < |S_\ell^2| + |S_r^2| + |T_\ell| + |T_r| \;\Leftrightarrow |\mathcal{T}_t^1| < |\mathcal{T}_t^2|.$$

By Proposition 2.1 and Eq. (7) we obtain $|N_\ell| + |S_\ell^1| + |S_\ell^2| + |T_\ell| - |F(V_r)| > |T_r| + \max\{s^1, s^2\} + |S_\ell^1| + |S_\ell^2| + |T_\ell| = |\mathcal{T}_t^2| - |S_r^2| + |S_\ell^1| + \max\{s^1, s^2\} = |\mathcal{T}_t^2| + s^1 + s^2 > |\mathcal{T}_t^2| = \max\{|\mathcal{T}_t^1|, |\mathcal{T}_t^2|\}$.

Since $\lambda_{2\mathcal{T}}(t_\ell) - |F(V_r)| = |N_\ell| + |S_\ell^1| + |S_\ell^2| + |T_\ell| - |F(V_r)|$, it follows that $\lambda_{2\mathcal{T}}(t_\ell) - |F(V_r)| > \max\{|\mathcal{T}_t^1|, |\mathcal{T}_t^2|\}$.

Now let $s^1 \geq s^2$; equivalently,

$$|S_\ell^1| - |S_r^2| \geq |S_\ell^2| - |S_r^1| \Leftrightarrow |\mathcal{T}_t^1| \geq |\mathcal{T}_t^2|.$$

By Proposition 2.1 and Eq. (7) we obtain $|N_\ell| + |S_\ell^1| + |S_\ell^2| + |T_\ell| - |F(V_r)| > |T_r| + \max\{s^1, s^2\} + |S_\ell^1| + |S_\ell^2| + |T_\ell| = |\mathcal{T}_t^1| - |S_r^1| + |S_\ell^2| + \max\{s^1, s^2\} = |\mathcal{T}_t^1| + s^2 + s^1 > |\mathcal{T}_t^1| = \max\{|\mathcal{T}_t^1|, |\mathcal{T}_t^2|\}$.

Similarly, we can show that $\lambda'_{2\mathcal{T}}(t) = \lambda_{2\mathcal{T}}(t_\ell) - |F(V_r)| > \max\{|\mathcal{T}_t^1|, |\mathcal{T}_t^2|\}$ for the cases where $s^1 < 0$ and $s^2 < 0$, $s^1 \geq 0$ and $s^2 < 0$, and $s^1 < 0$ and $s^2 \geq 0$.

(ii) We next show that $\lambda'_{2\mathcal{T}}(t) = \lambda_{\mathcal{T}}(t_\ell) - |F(V_r)| > \lambda_{2\mathcal{T}}(t_r) - |F(V_\ell)|$. Consider the case where $s^1 \geq 0$ and $s^2 \geq 0$. From Eq. (7) and since

$$|N_r| \leq |N_\ell| \leq |F(N_\ell)| \Leftrightarrow |N_r| \leq |F(V_\ell)| \Leftrightarrow |N_r| - |F(V_\ell)| \leq 0,$$

we obtain $|N_r| + |S_r^1| + |S_r^2| + |T_r| - |F(V_\ell)| < |N_r| + |S_r^1| + |S_r^2| - |F(V_\ell)| + |N_\ell| - \max\{s^1, s^2\} - |F(V_r)| < |S_r^1| + |S_r^2| + |N_\ell| - \max\{s^1, s^2\} - |F(V_r)| < |S_\ell^2| + |S_\ell^1| + |N_\ell| - |F(V_r)| \leq \lambda_{\mathcal{T}}(t_\ell) - |F(V_r)|$.

Similarly, we can show that $\lambda'_{2\mathcal{T}}(t) = \lambda_{\mathcal{T}}(t_\ell) - |F(V_r)| > \lambda_{2\mathcal{T}}(t_r) - |F(V_\ell)|$ for the cases where $s^1 < 0$ and $s^2 < 0$, $s^1 \geq 0$ and $s^2 < 0$, and $s^1 < 0$ and $s^2 \geq 0$.