

COMPLEMENT REDUCIBLE GRAPHS

D.G. CORNEIL

Department of Computer Science, University of Toronto, Toronto, Canada

H. LERCHS

Versa Management Systems Ltd., Toronto, Canada

L. STEWART BURLINGHAM

Department of Computer Science, University of Toronto, Toronto, Canada

Received 1 October 1979

Revised 19 December 1980

In this paper we study the family of graphs which can be reduced to single vertices by recursively complementing all connected subgraphs. It is shown that these graphs can be uniquely represented by a tree where the leaves of the tree correspond to the vertices of the graph. From this tree representation we derive many new structural and algorithmic properties. Furthermore, it is shown that these graphs have arisen independently in various diverse areas of mathematics.

1. Introduction

A *complement reducible graph* (also called a *cograph*) is defined recursively as follows:

- (i) A graph on a single vertex is a complement reducible graph.
- (ii) If G_1, G_2, \dots, G_k are complement reducible graphs, then so is their union $G_1 \cup G_2 \cup \dots \cup G_k$.
- (iii) If G is a complement reducible graph, then so is its complement \bar{G} .

Cographs have arisen in many disparate areas of mathematics and have been independently rediscovered by various researchers. Names synonymous with cographs include D^* -graphs, P_4 restricted graphs, and HD or Hereditary Dacey graphs. Cographs themselves were introduced in the early 1970s by Lerchs [14,15] who studied their structural and algorithmic properties and enumerated the class. This work was extended by Stewart (Burlingham) [22], who also developed an $O(n^2)$ cograph recognition algorithm.

Previous work related to cographs can be found in the study of comparability graphs (see [7,8] for definitions and properties). Wolk [25,26] showed that D -graphs or diagonal graphs (a strict subset of cographs), are precisely the comparability graphs of rooted trees. This result was later quoted incorrectly as "A graph without

induced subgraph isomorphic to P_4 [i.e. a cograph] is the comparability graph of rooted trees" [2]. C_4 is a counter-example to this statement. Jung [11] has recently shown that G is a D^* -graph (i.e. a cograph) iff G is the comparability graph of a multitree (defined in Section 2). Since comparability graphs are perfect, this immediately implies that cographs are perfect. This result was proved earlier and independently by Lerchs and Seinsche [21]. Stewart (Burlingham) [22] has illustrated the relationship of cographs with other families of perfect graphs.

Cographs independently arose in the study of empirical logic where they were called HD (or Hereditary Dacey) graphs (see [6,23]). This work was advanced and presented in a graph theoretical formulation by Sumner [23].

In this paper we will assimilate the results which have previously appeared and present a fundamental theorem on cographs which will show the equivalence of eight different characterizations of this family of graphs. Furthermore, we will develop a unique tree representation of cographs and use this tree to develop polynomial time algorithms for such problems as isomorphism, Hamiltonicity, clique determination and colouring. We also examine the relationship of cographs with TSP digraphs recently introduced by Lawler [13]. First we present the terminology which will be used throughout the paper.

2. Terminology

The terminology used in this paper is compatible with [3]. We assume that all graphs are finite and that unless stated otherwise the term subgraph always refers to the notion of induced subgraph.

For a given vertex x in graph $G(V, E)$, $N(x)$ denotes $\{y \in V \mid (x, y) \in E\}$. Vertices x, y are called *siblings* if $N(x) - \{x, y\} = N(y) - \{x, y\}$. The siblings are said to be *strong* if they are adjacent and *weak* otherwise. A *kernel* of a graph is a maximal independent set and a *clique* is a maximal complete set. Note that $S \subseteq V$ is a kernel in $G(V, E)$ iff S is a clique in \bar{G} . \mathcal{C}_G and \mathcal{K}_G respectively denote the set of cliques and kernels of G . Furthermore, $\mathcal{C}_G(x)$ (respectively $\mathcal{C}_G(\bar{x})$) denotes the set of cliques of G containing (respectively not containing) the vertex x . A similar convention holds for kernels using $\mathcal{K}_G(x)$ and $\mathcal{K}_G(\bar{x})$. A graph is said to have the *clique-kernel intersection property* (or *CK-property*) iff every clique of G has one vertex in common with every kernel of G (i.e. $\forall C \in \mathcal{C}_G$ and $\forall K \in \mathcal{K}_G$, $|C \cap K| = 1$). A graph $G(V, E)$ is *equistable* iff $\exists t \in \mathbb{N}^+$ and a mapping $\phi: V \rightarrow \mathbb{N}^+$ such that $\forall S \subseteq V$, S is a kernel iff $\sum_{v \in S} \phi(v) = t$. For a digraph G , $\mathcal{U}(G)$ denotes the underlying undirected graph. The *scattering number* of a graph $G(V, E)$, denoted $s(G)$ equals $\max(c(G - S) - |S|)$ where $S \subseteq V$ and $c(G - S)$, the number of components in $G - S$ does not equal 1. A set S satisfying $c(G - S) \neq 1$ and $s(G) = c(G - S) - |S|$ is called a *scattering set* of G . A graph is *Hamilton-connected* if for any two distinct vertices x and y there is a Hamiltonian path joining them.

A graph is a *diagonal graph* or *D-graph* if for every path in G with edges (v_1, v_2) ,

(v_2, v_3) , (v_3, v_4) the graph also contains the edge (v_1, v_3) or (v_2, v_4) (see [25]). A graph is a *D*-graph* if for every path in G with edges (v_1, v_2) , (v_2, v_3) , (v_3, v_4) the graph also contains the edge (v_1, v_3) or (v_2, v_4) or (v_1, v_4) . (i.e. the graph does not contain an induced P_4 .) A graph $G(V, E)$ is a *Dacey graph* iff for every clique C of G and every pair of distinct vertices u and v we have

$$C \subseteq N(u) \cup N(v) \Rightarrow (u, v) \in E.$$

A graph is an *HD-graph* (or *Hereditary Dacey graph*) iff every induced subgraph is a Dacey graph (see [23]).

Given a partially ordered set (V, \leq) and $V_1, V_2 \subseteq V$, we write $V_1 \leq V_2$ if $v_1 \leq v_2$ for all $v_1 \in V_1, v_2 \in V_2$. Further

$$P(v_1) = \{v \in V \mid v \leq v_1\} \quad \text{and} \quad S(v_1) = \{v \in V \mid v_1 \leq v\}.$$

The poset (V, \leq) is called a *multitree* if $v \leq v'$ or $S(v) - S(v') \leq S(v) \cap S(v')$ for all $v, v' \in V$ (see [11]).

Given a rooted tree T , we denote the path from a leaf x to the root as $P_T(x)$. The *meet* of two such paths is the unique vertex which is on both paths at the furthest distance from the root.

3. Structural properties of cographs

From the definition of cographs we see that they are all such graphs that can be obtained from single node graphs by performing a finite number of operations involving union and complementation. As examples of cographs consider the family of graphs introduced by Moon & Moser [18]. These graphs are the complements of the disjoint union of complete graphs and, depending on the sizes of the complete graphs, can be shown to possess the maximum number of cliques of any graph on n vertices.

The definition of a cograph could lead to many different parsings of a given cograph. In order to represent a cograph by a unique parsing, we introduce the notion of a *normalized form of a cograph* as follows:

A connected cograph is in *normalized form* if it is expressed as a single vertex or the complemented union of k (≥ 2) connected cographs in normalized form. A disconnected cograph is in *normalized form* if it is represented as the complement of a connected cograph in normalized form.

The proof that the normalized form of a cograph is unique up to isomorphism is straightforward and left to the reader. Henceforth, we assume that all cographs are expressed in normalized form. A cograph and its normalized form are presented in Fig. 1, where the symbol $\bar{\cup}$ represents complemented union.

The rooted tree representing the parse structure of a cograph in normalized form is referred to as a *cotree*. The leaves of a cotree are the vertices of the corresponding cograph and each internal node represents the $\bar{\cup}$ operation. Every internal node,

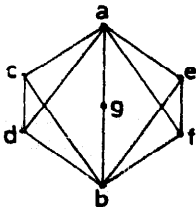


Fig. 1. $\bar{U}(a, b, \bar{U}(\bar{U}(c, d), \bar{U}(e, f), g))$

except possibly the root, will have two or more children; the root will have only one child exactly when the represented cograph is disconnected. Note that the cotree for a particular cograph is unique up to a permutation of the children of the internal nodes. Fig. 2 illustrates the cotree for the cograph presented in Fig. 1.

In order to establish various properties about cographs we label each internal node of a cotree as follows: the root is labelled 1, the children of a node with label 1 are labelled 0, and children of a node labelled 0 are labelled 1. Henceforth, we assume all cotrees to be labelled as such and we will refer to the internal nodes of cotrees as 0-nodes and 1-nodes.

An immediate consequence of the labelling is that two vertices x, y in a cograph G , with cotree T , are adjacent iff $P_T(x)$ and $P_T(y)$ meet at a 1-node. Other properties of G which may be derived from T are discussed in Section 4.

We now examine various structural properties of cographs and establish the equivalence of eight different characterizations of cographs. First we establish the property of heredity; recall that the term subgraph refers to induced subgraphs.

Lemma 1. *Every subgraph of a cograph is a cograph.*

Proof. For $n < 3$, the lemma follows immediately; thus assume $n \geq 3$. Since any subgraph of a graph can be obtained by removing vertices one by one, it is sufficient to show that the removal of a single vertex from a cograph yields a cograph. Let $G(V, E)$ be a cograph and T the associated cotree. The subgraph G' induced on $V - y$ is a cograph if and only if a cotree T' can be associated with it.

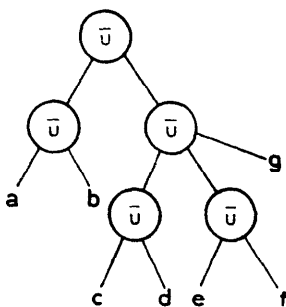


Fig. 2.

For the construction of T' we examine x , the parent of y in T , and consider the following cases:

- (i) x has more than two descendants: construct T' by removing the leaf y from T .
- (ii) x has exactly two descendants, y and y' . If y' is a leaf, construct T' by removing x and y from T and by connecting y' to the parent of x . If y' is a non-leaf, remove x , y and y' from T and connect all descendants of y' to the parent of x .

In both cases T' is a cotree uniquely representing the subgraph G' , thereby establishing that G' is a cograph. \square

We now present the fundamental theorem on cographs.

Theorem 2. *Given a graph G , the following statements are equivalent:*

- (1) G is a cograph.
- (2) Any nontrivial subgraph of G has at least one pair of siblings.
- (3) Any subgraph of G has the CK-property.
- (4) G does not contain P_4 as a subgraph.
- (5) The complement of any nontrivial connected subgraph of G is disconnected.
- (6) G is an HD-graph.
- (7) Every connected subgraph of G has diameter ≤ 2 .
- (8) G is the comparability graph of a multitree.

Proof. Throughout the proof we assume that $n \geq 3$ and that G is connected. The other cases follow immediately. As stated in the introductory section, some of the equivalences have been previously established. In particular, we have:

- (i) $(4) \Leftrightarrow (6) \Leftrightarrow (7)$ in [23].
- (ii) $(4) \Leftrightarrow (5)$ in [15,21].
- (iii) $(4) \Leftrightarrow (8)$ in [11].

To complete the proof we will show $(1) \Rightarrow (2) \Rightarrow (3) \Rightarrow (4)$ and $(5) \Rightarrow (1)$.

$(1) \Rightarrow (2)$. Because of Lemma 1, it is sufficient to show that any cograph has at least one pair of siblings. This follows by examining the leaves of the cotree T corresponding to an arbitrary cograph G . Any two leaves sharing the same parent are strong siblings if the parent is labelled 1 and weak siblings otherwise. Since every cotree must have at least one non-leaf vertex adjacent to at least two leaves, the property follows.

$(2) \Rightarrow (3)$. We proceed by induction on p the order of the subgraph in G . For $p = 1$ the subgraph trivially has the CK-property. We now show that if all subgraphs of order p in G have the CK-property, then so does any subgraph H of order $p + 1$. Indeed, let x, x' be siblings in H ; by (2) such siblings always exist. Consider then the subgraph $H' = H - \{x'\}$ of order p . The cliques and kernels of H can be expressed in terms of the cliques and kernels of H' as follows:

Case (i): x, x' are strong siblings. Any clique of H' not containing x remains a clique in H ; any clique of H' containing x becomes a clique in H when augmented with x' . Therefore

$$\mathcal{C}_H(x) = \mathcal{C}_{H'}(x), \quad \mathcal{K}_H(x) = \mathcal{K}_{H'}(x) + x'.$$

Similarly, any kernel of H' remains a kernel in H ; for any kernel of H' containing x , a new kernel is obtained in H by replacing x with x' . Therefore

$$\mathcal{K}_H(x') = \mathcal{K}_{H'}, \quad \mathcal{K}_H(x) = \mathcal{K}_{H'}(x) - x + x'.$$

Examination of the intersection of an arbitrary clique $C \in \mathcal{C}_H = \mathcal{C}_H(x) \cup \mathcal{C}_H(x')$ and kernel $K \in \mathcal{K}_H = \mathcal{K}_H(x') \cup \mathcal{K}_H(x)$ leads to $|C \cap K| = 1$ as required.

Case (ii): x, x' are weak siblings. This proof is identical to case (i) where the notions of clique and kernel are interchanged.

(3) \Rightarrow (4). The chain $P_4(w, x, y, z)$ cannot be a subgraph of G because P_4 does not have the CK-property; the clique (x, y) does not intersect the kernel (w, z) .

(5) \Rightarrow (1). We first show that if G has property (5) then so does \bar{G} . Assume the contrary and let H be a connected subgraph of \bar{G} such that \bar{H} is also connected. But \bar{H} is a connected subgraph of G and G satisfies property (5), hence, $\bar{H} = H$ is disconnected, contradicting our earlier assumption.

We prove the statement (5) \Rightarrow (1) by induction on n , the order of the graph. The statement holds trivially for $n \leq 3$. Assume it holds for $n < p$ and let G be of order p . Since property (5) and being a cograph are preserved under complementation we may examine either G or \bar{G} . In particular, we examine the one which is disconnected. By the inductive assumption, each of its connected components is a cograph thereby establishing that the graph itself is also a cograph. \square

Before presenting algorithms for various problems of cographs we state some relevant properties of cographs.

Lemma 3 [11]. *If $G(V, E)$ is a cograph, then*

- (i) *G has a Hamiltonian path iff $s(G) \leq 1$.*
- (ii) *G is Hamiltonian iff $s(G) \leq 0$ and $|V| \geq 3$.*
- (iii) *G is Hamiltonian connected iff $s(G) < 0$.*

Lemma 4 [19]. *Cographs are equistable.*

Lemma 5 [15]. *Cographs satisfy Ulam's conjecture.*

4. Algorithmic properties of cographs

As one might expect, various algorithmic problems which are thought to be difficult for graphs in general often may be solved in polynomial time for cographs. One obvious such problem is that of cograph isomorphism. Because of the unique cotree representation of a cograph, the linear time tree isomorphism algorithm [9] when applied to the cotrees yields a polynomial cograph isomorphism algorithm. Other families of graphs which are known to have polynomial isomorphism algorithms include planar graphs [10], interval graphs [16] and k -trees for fixed k [12]. It is

interesting to note that these last two families also have a unique tree representation.

From condition (4) of Theorem 2, we see that G is a cograph iff there is no induced subgraph isomorphic to P_4 . It is of interest to determine the isomorphism situation for classes of graphs which have a forbidden induced connected subgraph H other than P_4 . The following theorem shows that in some sense P_4 is the "largest" such forbidden subgraph for which the isomorphism problem is known to be polynomial.

Theorem 6 [5]. *Given H a connected graph, let \mathcal{G}_H denote the set of graphs where H is a forbidden subgraph. Then the isomorphism problem on \mathcal{G}_H is isomorphism complete¹ unless $H = P_1, P_2, P_3$ or P_4 in which case the problem is polynomial.*

The classes $\mathcal{G}_{P_1}, \mathcal{G}_{P_2}, \mathcal{G}_{P_3}, \mathcal{G}_{P_4}$ are respectively null graphs, void graphs, union of disjoint complete graphs and cographs.

Implicit in the polynomial cograph isomorphism algorithm is the assumption that a polynomial algorithm exists for recognizing a cograph and for constructing its cotree. From Theorem 2, it is seen that several conditions of the theorem lead directly to polynomial recognition algorithms. An $O(n^2)$ recognition and cotree construction algorithm is described in [22]. This algorithm is incremental in the sense that one starts with the null graph and then adds the vertices of the given graph G one by one in any order to this graph. Since any subgraph of a cograph is a cograph the problem of constructing the cotree of cograph G is equivalent to the problem of constructing G 's cotree from the cotree representing $G - v$. An $O(n)$ algorithm for this problem immediately yields the $O(n^2)$ cograph recognition and cotree construction algorithms (see [22] for details).

As mentioned above, the unique tree representation of a cograph leads directly to a polynomial cograph isomorphism algorithm. We now show how this tree representation of the structure of a cograph leads to polynomial algorithms for many other traditionally very difficult problems; these algorithms are summarized in Table 1. Given a cotree one makes the appropriate substitutions for the leaves, 0-nodes and 1-nodes and progressing from the leaves to the root terminates with the appropriate answer. We assume that the k sons of an internal node have labels a_1, a_2, \dots, a_k . The δ_{a_i, a_j} is the standard Kronecker delta. By interchanging the operators on 1-nodes with the operators on 0-nodes for the first three algorithms one gets a corresponding result for kernels in place of cliques. The proofs of the first three algorithms are presented in [14]; Algorithm 4 is proved in [22].

To prove Algorithm 5, we proceed by induction on the size of the cograph. Assume that the operations work for all cographs with $< n$ nodes. If the cograph G is connected, the root node is a 1-node indicating that G is the complete interconnection of graphs G_1, G_2, \dots, G_k with corresponding scattering numbers $s(G_1), s(G_2), \dots, s(G_k)$. Because of this complete interconnection, a scattering set S

¹ A family of graphs is called *isomorphism complete* if the isomorphism problem on the class is polynomial time equivalent with the general graph isomorphism problem. See [4].

Table 1. Algorithms

Arguments of leaves	Operators on 1-nodes	Operators on 0-nodes	Results
1. $a = 1$	$a = \prod_1^k a_i$	$a = \sum_1^k a_i$	number of cliques
2. nodes of G	$a = \bigwedge_1^k a_i$	$a = \bigvee_1^k a_i$	generating formula for the set of cliques
3. $a = 1$	$a = \sum_1^k a_i$	$a = \max_1^k a_i$	a = size of largest clique and the chromatic #
$b = 1$	$b = \prod_1^k b_i$	$b = \sum_1^k \delta_{a, a_i} \cdot b_i$	b = number of maximum cliques
4. $a = 1$	$a = k! \cdot \prod_1^k a_i$	$\prod_1^k a_i$	# of transitive orientations of G
5. $a = -1$	$a = \max_1^k (a_i - \sum_{j \neq i} b_j)$	$a = \sum_1^k \max(a_i, 1)$	a = scattering number $s(G)$
$b = 1$	$b = \sum_1^k b_i$	$b = \sum_1^k b_i$	

of G must contain $k-1$ of the G_i 's. Thus for any scattering set S of G , $c(G-S) = c(G_i-S_i)$ where $S_i = S \cap G_i$. From this it follows that

$$s(G) = \max_{1 \leq i \leq k} \left(s(G_i) - \sum_{i \neq j} |G_{ij}| \right).$$

If the cograph G is not connected, i.e. the root node is a 0-node, then for any scattering set S , $c(G-S) = \sum_{1 \leq i \leq k} c(G-S_i)$ where $S_i = G_i \cap S$. Thus

$$\begin{aligned} s(G) &= \max_S \left[\sum_{\substack{i \in S, 1 \\ |S_i| > 0}} (c(G_i - S_i) - |S_i|) + \sum_{\substack{i \in S, 1 \\ |S_i| = 0}} c(G_i) \right] \\ &= \sum_{1 \leq i \leq k} \max(s(G_i), 1) \end{aligned}$$

since the G_i 's are rooted at 1 nodes.

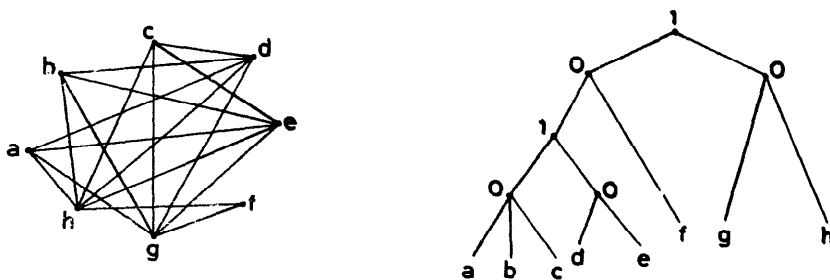


Fig. 3.

In a fairly straightforward manner, Algorithm 5 can be altered to produce all scattering sets or the number of scattering sets. For example, consider the cograph G and its cotree presented in Fig. 3. The number of cliques is 14 with the generating formula

$$(((a \vee b \vee c) \wedge (d \vee e)) \vee f) \wedge (g \vee h).$$

The size of the largest clique is 3 and the number of such cliques is 12. Similarly the number of kernels is 3 with the generating formula

$$(((a \wedge b \wedge c) \vee (d \wedge e)) \wedge f) \vee (g \wedge h)$$

yielding the kernels

$abcf$
 def
 gh

The number of transitive orientations is 4. The scattering number is 0 indicating that G is Hamiltonian but not Hamilton connected. (One Hamiltonian cycle is $\langle a, d, b, e, c, g, f, h, a \rangle$, however since $\deg(f) = 2$ there cannot be a $g-h$ Hamilton path.)

Although the clique and Hamiltonian problems are polynomial for cographs, one may not infer that the general induced and partial subgraph isomorphism problems are also polynomial. In fact, Agarwal [1] has shown that the partial subgraph isomorphism problem for cographs is NP-complete. The status of the induced subgraph isomorphism problem is presently unknown.

5. Relationship of cographs with TSP digraphs

We now examine the relationship of cographs with TSP digraphs, a family of digraphs recently presented by Lawler [13] (see also [24]). These digraphs are defined recursively as follows:

A digraph on a single node is *transitive series parallel*, or *TSP*. If $G_1 = (V_1, E_1)$, $G_2 = (V_2, E_2)$, ..., $G_k = (V_k, E_k)$, $k \geq 2$, $V_i \cap V_j = \emptyset$ for all $i, j = 1, 2, \dots, k$, $i \neq j$, are TSP digraphs, then

(i) $G_1 | G_2 | \dots | G_k = (V_1 \cup V_2 \cup \dots \cup V_k, E_1 \cup E_2 \cup \dots \cup E_k)$, the *parallel composition* of G_1, G_2, \dots, G_k , is TSP, and

(ii) $G_1 \rightarrow G_2 \rightarrow \dots \rightarrow G_k = (V_1 \cup V_2 \cup \dots \cup V_k, E_1 \cup E_2 \cup \dots \cup E_k \cup ((V_i \times V_j), i, j = 1, 2, \dots, k, i < j))$, the *series composition* of G_1, G_2, \dots, G_k , is TSP.

Only those graphs which can be obtained by a finite number of series and parallel operations on single node digraphs are TSP.

A second characterization, involving a forbidden subgraph, is the following. A digraph is TSP if and only if it is transitive and does not contain an induced subgraph isomorphic to the graph of Fig. 4.



Fig. 4.

Finally, a tree representation of TSP digraphs is presented. Any TSP digraph G can be uniquely represented by a tree structure in which the leaves correspond to the vertices of G and each internal node is either an S -node or a P -node, representing series or parallel composition, respectively. An additional restriction is that no child of an S -node is an S -node and no child of a P -node is a P -node. The order of the children of S -nodes completely defines the directions of the edges of G whereas the order of the children of P -nodes is irrelevant. The SP-tree representation of a TSP digraph is in fact unique up to a permutation of the children of the P -nodes. Fig. 5 illustrates a TSP digraph and one of its corresponding SP-tree representations.

The preceding characterization of TSP digraphs obviously parallels three of the properties of cographs. The precise relationship between the two families can be stated as follows.

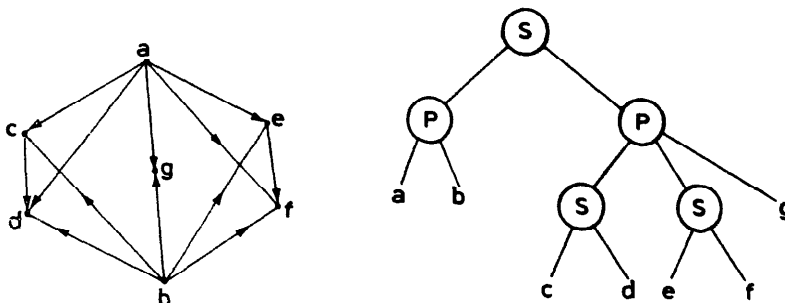


Fig. 5.

Theorem 7. *A graph G is a cograph if and only if there exists an orientation G' of G such that G' is TSP.*

Proof. (\Rightarrow) Let G be a cograph; then G is a comparability graph, i.e., G is transitively orientable. Let G' be a transitive orientation of G . G' is transitive and does not contain the graph of Fig. 4 as a subgraph since G contains no P_4 . Hence, G' is TSP.

(\Leftarrow) Let G' be a TSP digraph and G its underlying graph. G' does not contain the graph of Fig. 4 as an induced subgraph and, in fact, cannot contain any other orientation of P_4 since none of the others are transitive. Hence, G does not contain a subgraph isomorphic to P_4 and therefore is a cograph. \square

In other words, the class of TSP digraphs is exactly the set of all transitive orientations of cographs and cographs are simply the underlying graphs of the TSP digraphs. The correspondence between SP-trees and cotrees is established in the

next two theorems. The proofs, omitted here, are based on the fact that two vertices x, y of a TSP digraph have an edge between them iff $P_T(x)$ and $P_T(y)$ meet at an S -node. For details, see [22].

Theorem 8. *Consider the following alterations of a cotree:*

- (i) *substitute S -nodes for 1-nodes and P -nodes for 0-nodes throughout, and*
- (ii) *delete the root if it has only one child.*

If T is the cotree representing a cograph G , then the tree T' obtained by performing the above modifications to T is the SP-tree representation of a TSP digraph G' and $\psi(G') = G$.

Theorem 9. *Consider the following modifications to an SP-tree:*

- (i) *if the root of the SP-tree is a P -node, then add an S -node as its parent, and*
- (ii) *replace each S -node with a 1-node and each P -node with a 0-node.*

If T' is an SP-tree representing the TSP digraph G' , then the tree T obtained by performing the above operations on T' is the cotree representing the cograph $G = \psi(G')$.

6. Concluding remarks

In this paper we have greatly exploited the fact that a cograph can be uniquely represented by a tree. It is natural to ask if other combinatorial structures are also tree representable. As mentioned earlier, interval graphs and k -trees for fixed k fall into this class. A similar result holds for a class of electrical structures. In 1892 P.A. MacMahon [17] (see also [20]) enumerated what he called "combinations of resistances" (now called "two-terminal series parallel networks"). His enumeration formulas are identical with Lerchs' enumeration formulas of cographs [14]. This leads immediately to a tree representation of two-terminal series parallel networks and a one-to-one relationship between these trees and cotrees. Thus many results for cographs immediately apply to two-terminal series parallel networks.

Very recently, the tree representation of graphs has been extended to hookup classes of graphs. The hookup class $[A, B]$ denotes the set of graphs formed by starting with graph A and recursively adding a vertex adjacent to all vertices in an induced subgraph isomorphic to B . For example, k -trees are the hookup class $[K_k, K_k]$. It is shown in [12] that the isomorphism problem on hookup classes is either isomorphism complete or polynomial. The polynomial algorithms result from a unique tree representation for each graph in the given hookup classes.

Acknowledgement

The authors wish to thank the Natural Sciences and Engineering Research Council of Canada for financial assistance.

References

- [1] R.K. Agarwal, An investigation of the subgraph isomorphism problem, M.Sc. Thesis, Dept. of Comp. Sci., University of Toronto, TR 138/80 (1980).
- [2] J.C. Arditti and D. DeWerra, A note on a paper by D. Seinsche, *J. Combinatorial Theory (B)* 21 (1976) 90.
- [3] J.A. Bondy and U.S.R. Murty, *Graph Theory with Applications*, (MacMillan, New York, 1976).
- [4] K.S. Booth and C.J. Colbourn, Problems polynomially equivalent to graph isomorphism, Dept. of Comp. Sci., University of Waterloo, TR 77-04 (1979).
- [5] C. Colbourn and M. Colbourn, private communication (1978).
- [6] D.J. Foulis, Empirical logic, Xeroxed course notes, Univ. of Massachusetts (1969–1970).
- [7] P.C. Gilmore and A.J. Hoffman, A characterization of comparability and of interval graphs, *Canad. J. Math.* 16 (1964) 539–548.
- [8] M.C. Golumbic, *Algorithmic Graph Theory and Perfect Graphs*, (Academic Press, New York, 1980).
- [9] J.E. Hopcroft and R.E. Tarjan, Isomorphism of planar graphs, in: R. Miller, ed., *Complexity of Computer Computations*, (Plenum, New York, 1972).
- [10] J.E. Hopcroft and J.K. Wong, Linear time algorithm for isomorphism of planar graphs, *Proc. Sixth Annual ACM Symp. on the Theory of Computing* (1974) 172–184.
- [11] H.A. Jung, On a class of posets and the corresponding comparability graphs, *J. Combinatorial Theory (B)* 24 (1978) 125–133.
- [12] M.M. Klawe, D.G. Corneil and A. Proskurowski, Isomorphism testing in hookup classes, IBM Research Report RJ3091 (38307) (1981).
- [13] E.L. Lawler, Graphical algorithms and their complexity, *Math. Centre Tracts* 81 (1976) 3–32.
- [14] H. Lerchs, On cliques and kernels, Dept. of Comp. Sci., University of Toronto (March 1971).
- [15] H. Lerchs, On the clique-kernel structure of graphs, Dept. of Comp. Sci., University of Toronto (October 1972).
- [16] G.S. Leuker and K.S. Booth, A linear time algorithm for deciding interval graph isomorphism, *J. Assoc. Comput. Mach.* 26 (2) (1979) 183–195.
- [17] P.A. MacMahon, The combination of resistances, *The Electrician* 28 (1892) 601–602.
- [18] J.W. Moon and L. Moser, On cliques in graphs, *Israel J. Math.* 3 (1965) 23–28.
- [19] C. Payan, A class of threshold and domishold graphs: equistable and equidominating graphs, *Discrete Math.* 29 (1980) 47–52.
- [20] J. Riordan, *An Introduction to Combinatorial Analysis*, (Wiley, New York, 1958) 139–143.
- [21] D. Seinsche, On a property of the class of n -colorable graphs, *J. Combinatorial Theory (B)* 16 (1974) 191–193.
- [22] L. Stewart, Cographs, a class of tree representable graphs, M.Sc. Thesis, Dept. of Comp. Sci., University of Toronto, TR 126/78 (1978).
- [23] D.P. Sumner, Dacey graphs, *J. Australian Math. Soc.* 18 (4) (1974) 492–502.
- [24] J. Valdes, R.E. Tarjan and E.L. Lawler, The recognition of series parallel digraphs, *Proc. Eleventh Annual ACM Symp. on the Theory of Computing* (1979) 1–12.
- [25] E.S. Wolk, The comparability graph of a tree, *Proc. AMS* 13 (1962) 789–795.
- [26] E.S. Wolk, A note on 'The comparability graph of a tree', *Proc. AMS* 16 (1965) 17–20.