1999

On the p-Connectedness of Graphs – a Survey

Luitpold Babel

Stephan Olariu
Old Dominion University

Original Publication Citation

Babel, L., & Olariu, S. (1999). On the *p*-connectedness of graphs - a survey. *Discrete Applied Mathematics, 95*(1-3), 11-33. doi:10.1016/s0166-218x(99)00062-1

ELSEVIER
Discrete Applied Mathematics 95 (1999) 11–33
DISCRETE APPLIED MATHEMATICS

On the p-connectedness of graphs – a survey

Luitpold Babel[a,1], Stephan Olariu[b,*,2]

[a] Zentrum Mathematik, Technische Universität München, D-80290 München, Germany
[b] Department of Computer Science, Old Dominion University, Norfolk, VA 23529, USA

Received 31 October 1997; received in revised form 24 March 1998; accepted 23 December 1998

Abstract

A graph is said to be p-connected if for every partition of its vertices into two non-empty, disjoint, sets some chordless path with three edges contains vertices from both sets in the partition. As it turns out, p-connectedness generalizes the usual connectedness of graphs and leads, in a natural way, to a unique tree representation for arbitrary graphs.

This paper reviews old and new results, both structural and algorithmic, about p-connectedness along with applications to various graph decompositions.

1. Introduction and motivation

A number of concepts in graph theory find their roots in various areas of investigation. It is often the case that only after they have been defined one realizes how naturally these concepts bring and tie together seemingly unrelated topics, blending them in a more uniform body of knowledge. Such is the case – we strongly believe – with the concept of p-connectedness of graphs to which we devote this survey.

Quite often, real-life applications suggest the study of graphs that feature some *local density* properties, traditionally equated with the absence of chordless paths with four vertices and three edges, also known as P_4's. In particular, graphs that are unlikely to have more than a few P_4's appear in examination scheduling and semantic clustering of index terms [16,18,19]. In examination scheduling, for example, a *conflict graph* is readily constructed: the vertices represent different courses offered, while courses x and y are linked by an edge if, and only if, some student takes both of them. In

* Corresponding author.
E-mail address: olariu@cs.odu.edu (S. Olariu)

[1] Supported by the Deutsche Forschungsgemeinschaft (DFG).
[2] Supported in part by NSF grant CCR-95-22093 and by ONR grant N00014-97-1-0526

PII: S0166-218X(99)00062-1

the weighted version, the weight of edge xy stands for the number of students taking both x and y. Clearly, in any coloring of the conflict graph, vertices that are assigned the same color correspond to courses whose examinations can be held concurrently. It is usually anticipated that very few paths of length three will occur in the conflict graph. These applications have motivated both the theoretical and algorithmic study of the class of cographs [16,18,19] which contain no induced P_4's. Later, in a series of papers, Jamison and Olariu have studied the classes of P_4-reducible, P_4-sparse, P_4-extendible, and P_4-lite graphs obtained by relaxing in various ways the stringent requirement imposed by the absence of P_4's [34–37]. In all the classes mentioned above the P_4's interact with one another in very interesting, albeit straightforward, ways. In particular, either every vertex belongs to at most one P_4, as in the P_4-reducible graphs, or no set of five vertices induces more than one P_4, as is the case for the P_4-sparse graphs, or no set of six or more vertices form interacting P_4's as is the case for P_4-extendible graphs.

A powerful tool for obtaining efficient solutions to graph problems is the *divide-and-conquer* paradigm, one of whose manifestations is graph decomposition. A very desirable form of graph decomposition involves associating with a given graph G a rooted tree $T(G)$ whose leaves are subgraphs of G (e.g. vertices, edges, cliques, stable sets, cutsets) and whose internal nodes correspond to certain prescribed graph operations. Of a particular interest are classes of graphs G for which the following conditions hold:

- $T(G)$ can be obtained *efficiently*, that is, in time polynomial in the size of G;
- $T(G)$ is unique up to labeled tree isomorphism.

Tree representations satisfying the conditions mentioned above have been obtained for several classes including the cographs, P_4-reducible graphs, P_4-extendible graphs, and P_4-sparse graphs, among many others. A well-known form of graph decomposition is the *modular* decomposition (also called *substitution* decomposition). The modular decomposition has been discovered independently by researchers in many areas. The reader is referred to Möhring and Rademacher [50] where some applications are discussed.

A classic result of Lovász [47] asserts that a graph is perfect in the sense of Berge [24] whenever its complement is. This important result motivated Chvátal [12] to ask for a succinct certificate of perfection: Lovász's result suggested, in quite obvious terms, that a very desirable such certificate should apply both to the graph and to its complement. More generally, this suggests investigating graph properties that are invariant under complementation. It is a simple observation that the P_4 is self-complementary and, consequently, graph properties that are expressed in terms of P_4's only must also be invariant under complementation. Chvátal [12] proposed to call two graphs P_4-isomorphic if there exists a bijection between their vertices in such a way that a set of four vertices induces a P_4 in the first graph if and only if its image induces a P_4 in the second. Chvátal [12] conjectured and Reed [52] proved that a graph P_4-isomorphic to a perfect graph is also perfect. For various other related results the interested reader is referred to [13–15,25,27–29,32,51].

As it turns out, the concept of p-connectedness finds its original inspiration and motivation in all of the issues discussed above. In addition, it can be defined, at the elementary level, as an extension of the well-known connectedness in graphs. This is the approach we take in this survey although p-connectedness could have been, just as well, introduced in many other similar ways. The theory of p-connectedness was introduced by Jamison and Olariu in [40]. Since then it has developed into a rich body of knowledge with surprising ramifications both structural and algorithmic. Perhaps one of the most startling result that one derives from the concept of p-connectedness is a structure theorem for general graphs which, in turn, suggests a unique tree representation for arbitrary graphs: the leaves of this tree are the p-connected components along with weak vertices, that is, vertices of the graph that belong to no proper p-connected component. By refining this first result one obtains the *homogeneous decomposition* and the *separable-homogeneous decomposition* of arbitrary graphs. Like the modular decomposition, these decompositions produce a unique decomposition tree for arbitrary graphs; however, both the homogeneous and the separable-homogeneous decompositions go substantially beyond the modular decomposition in the sense that they decompose graphs that are prime with respect to the modular decomposition.

This survey is organized as follows: Section 2 presents basic definitions and establishes terminology used throughout this work; Section 3 discusses one of the main structural results pertaining to p-connectedness – a key ingredient in many of the subsequent results – along with a simple and natural decomposition for arbitrary graphs termed the primeval decomposition; Section 4 takes a new look at p-connectedness viewed from the perspective of p-chains, a natural analogue of paths; Section 5 introduces the concept of p-articulation-vertices, a natural analogue of articulation-vertices, and discusses the structure of those graphs all of whose vertices are p-articulation-vertices; Section 6 looks at two graph operations that allow us to construct new p-connected graphs out of old ones; Section 7 takes the opposite view, presenting two p-connectedness preserving graph operations along with a decomposition theorem for p-connected graphs that will be crucial in our subsequent decomposition schemes. Section 8 discusses the homogeneous decomposition and the separable-homogeneous decomposition of graphs; then several graph classes are analyzed whose p-connected components have a simple and intuitive structure; Section 9 investigates the concept of p-trees which are a natural analogue of trees; Section 10 deals with graphs which, in some local sense, contain only a restricted number of P_4's; finally, Section 11 offers a brief survey of known algorithmic results related to the concept of p-connectedness.

2. p-connected graphs

Before introducing the concept of p-connectedness, we present some basic definitions and establish notation that will be used throughout this survey. We consider finite graphs with no loops nor multiple edges. In addition to standard graph-theoretical terminology, compatible with [11], we need several new terms that we define next.

Let $G=(V,E)$ be a graph with vertex-set V and edge-set E. In the context of trees, vertices will be called *nodes*. We denote by n the cardinality of V. For a vertex v of G let $N(v)$ denote the set of all vertices adjacent to v, also called *neighbors* of v. If $U \subseteq V$ then $G(U)$ stands for the graph induced by U. Occasionally, to simplify the exposition, we shall blur the distinction between sets of vertices and the subgraphs they induce, using the same notation for both. The complement of G is denoted by $\overline{G}$. A *clique* is a set of pairwise adjacent vertices, a *stable set* is a set of pairwise nonadjacent vertices. G is termed a *split* graph if its vertices can be partitioned into a clique and a stable set.

We say that two sets X and Y of vertices of G are *nonadjacent* if no edge has one endpoint in X and the other in Y. X and Y are *totally adjacent* if every vertex in X is adjacent to all vertices in Y. Finally, sets X and Y that are neither nonadjacent nor totally adjacent are termed *partially adjacent*.

A vertex v is said to *distinguish* a set U of vertices if v is partially adjacent to U. A subset H of V with $1 < |H| < |V|$ is termed a *homogeneous set* if no vertex outside H distinguishes H, i.e. each vertex outside H is either nonadjacent or totally adjacent to H. A homogeneous set H is *maximal* if no other homogeneous set properly contains H. The graph obtained from a p-connected graph G by shrinking every maximal homogeneous set to a single vertex is called the *characteristic graph* of G.

As usual, we let P_k stand for the chordless path with k vertices and $k-1$ edges. The *length* of a path P_k is $k-1$. C_k is the chordless cycle with k vertices and k edges. The P_4 with vertices u,v,w,x and edges uv, vw, wx will be denoted by $uvwx$; v and w are the *midpoints* whereas u and x are the *endpoints* of the P_4.

The concept of p-connectedness was introduced by Jamison and Olariu in [40]. They define a graph $G=(V,E)$ to be *P_4-connected*, or *p-connected* for short, if for every partition of V into nonempty disjoint sets V_1 and V_2 there exists a *crossing* P_4, that is, a P_4 containing vertices from both V_1 and V_2. The *p-connected components* of a graph are the maximal induced subgraphs which are p-connected. Note that a p-connected component consists either of a single vertex or of at least four vertices. It is easy to see that

- each graph has a unique partition into p-connected components;
- the p-connected components are closed under complementation;
- every p-connected component is a connected subgraph of G and $\overline{G}$.

Vertices which do not belong to p-connected components of size at least four are referred to as *weak vertices*.

A p-connected graph is termed *separable* if its vertex-set can be partitioned into two nonempty disjoint sets V_1 and V_2 in such a way that every crossing P_4 has its midpoints in V_1 and its endpoints in V_2. We say that (V_1,V_2) is a *separation* of G. It is obvious that

- the complement of a separable p-connected graph is also separable.

The separation (V_1,V_2) of G becomes (V_2,V_1) in $\overline{G}$. Fig. 1 features a separable p-connected graph along with its characteristic graph. Separable p-connected

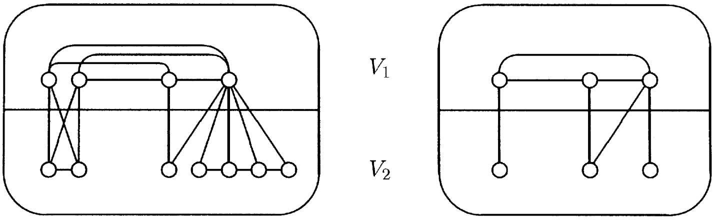

Fig. 1. A separable p-connected graph and its characteristic graph.

components play a crucial role in the theory of p-connectedness. The next three results summarize some of their most important properties.

Theorem 2.1 (Jamison and Olariu [40]). *Every separable p-connected graph has a unique separation. Furthermore, every vertex belongs to a crossing P_4 with respect to the separation.*

The next statement gives more detailed information about the structure of separable p-connected graphs.

Theorem 2.2 (Jamison and Olariu [40]). *Let G be separable p-connected with separation (V_1, V_2). The subgraph of G (respectively $\overline{G}$) induced by V_2 (respectively V_1) is disconnected. Furthermore, every connected component of the subgraph of G (respectively $\overline{G}$) induced by V_2 (respectively V_1) with at least two vertices is a homogeneous set in G.*

This immediately implies the following simple and useful result.

Corollary 2.3 (Jamison and Olariu [40]). *A p-connected graph is separable if and only if its characteristic graph is a split graph.*

3. Structure theorem and primeval decomposition

The introduction and the study of p-connected and separable p-connected graphs is justified by the following general result that provides the foundation of several decomposition schemes.

Theorem 3.1 (Structure Theorem Jamison and Olariu [40]). *For an arbitrary graph G exactly one of the following conditions is satisfied:*

1. *G is disconnected.*
2. *$\overline{G}$ is disconnected.*

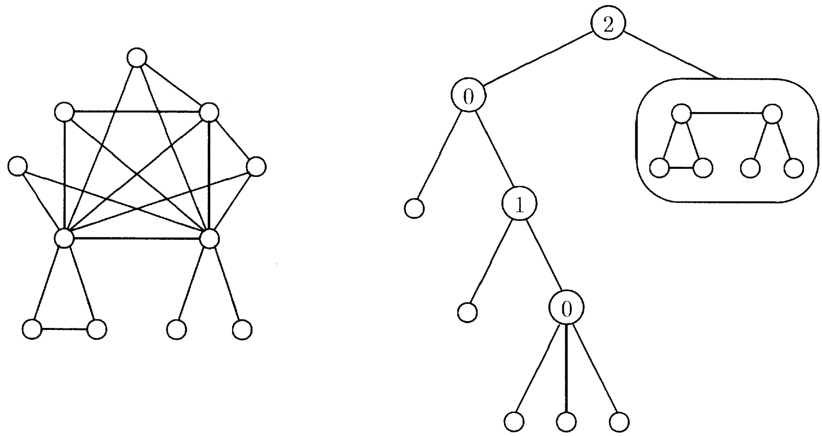

Fig. 2. A graph and the associated primeval tree.

3. *There is a unique proper separable p-connected component H of G with a partition (H_1, H_2) such that every vertex outside H is adjacent to all vertices in H_1 and to no vertex in H_2.*
4. *G is p-connected.*

As pointed out in [40], this theorem implies, in a natural way, a decomposition scheme for arbitrary graphs, called the *primeval decomposition*. In order to be more specific, we now define a number of graph operations.

Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be disjoint graphs. The *disjoint union* and the *disjoint sum* of G_1 and G_2 are the graphs which result, respectively, from the operations
- $G_1 \textcircled{0} G_2 = (V_1 \cup V_2, E_1 \cup E_2)$ and
- $G_1 \textcircled{1} G_2 = (V_1 \cup V_2, E_1 \cup E_2 \cup \{xy \mid x \in V_1, y \in V_2\})$.

Obviously, operations ⓪and ① reflect the first two cases of the Structure Theorem. Let $G_1 = (V_1, E_1)$ be separable p-connected with separation (V_1^1, V_1^2) and $G_2 = (V_2, E_2)$ be an arbitrary graph disjoint from G_1. The third case of the Structure Theorem is captured by the operation
- $G_1 \textcircled{2} G_2 = (V_1 \cup V_2, E_1 \cup E_2 \cup \{xy \mid x \in V_1^1, y \in V_2\})$.

As shown in [40], each graph can be obtained uniquely from its p-connected components and its weak vertices by a finite sequence of operations ⓪, ①and ②. Furthermore, the Structure Theorem suggests, in a natural way, a tree representation for arbitrary graphs which turns out to be unique up to isomorphism. The tree associated with a graph G is called the *primeval tree* of G. The internal nodes of the tree are labeled by integers $i \in \{0, 1, 2\}$, where an i-node indicates that the graph associated with the subtree rooted at this node is obtained from the graphs corresponding to its children by an ⓘoperation. The leaves of the tree are the p-connected components of G. Fig. 2 features a graph along with its associated primeval tree.

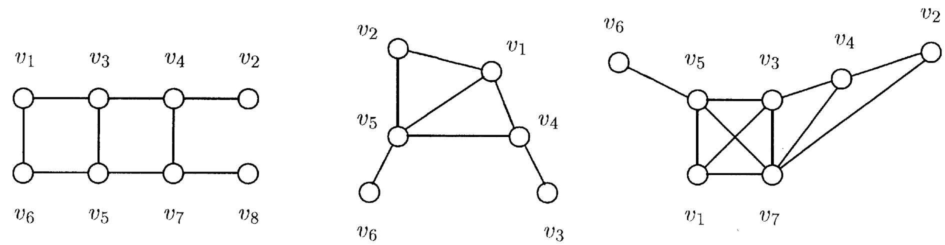

Fig. 3. Some examples of p-chains.

4. p-chains

Clearly, the p-connectedness generalizes the usual connectedness of graphs since a graph is connected if for every partition of the vertex-set into two nonempty, disjoint, sets some edge in the graph has endpoints in both sets of the partition. An equivalent and, perhaps, more common definition states that a graph is connected if and only if each pair of vertices is connected by a path, i.e. a sequence of vertices such that any two consecutive vertices induce an edge. Somewhat surprisingly, there is a very similar characterization of p-connected graphs in terms of p-chains, a natural analogue of paths in the context of p-connectedness.

Consider a graph $G=(V,E)$ and two vertices x and y in V. A *p-chain of length $k-1$ connecting x and y* is a sequence of distinct vertices $(v_1, v_2, \ldots, v_k)$ such that

- $x=v_1$, $y=v_k$, and
- for all i, $(1 \leqslant i \leqslant k-3)$, $X_i := \{v_i, v_{i+1}, v_{i+2}, v_{i+3}\}$ induces a P_4.

Two vertices are said to be *p-connected* if they coincide or else there exists a p-chain connecting them. Occasionally, a p-chain consisting of a single P_4 is termed a *trivial* p-chain. Furthermore, we say that vertices x and y are connected by a *unique* p-chain whenever the sequence of sets $X_1, X_2, \ldots, X_{k-3}$ is unique. Some of the simplest examples of p-chains include the chordless paths P_k and their complements $\overline{P_k}$ for $k \geqslant 4$. Several further examples are depicted in Fig. 3.

It is an important and useful property that p-chains are invariant under complementation, i.e. a p-chain in G is also a p-chain in $\overline{G}$. Thus, two vertices are p-connected in G if and only if they are p-connected in $\overline{G}$. We are now ready for the characterization of p-connected graphs by means of p-chains.

Theorem 4.1 (Babel and Olariu [7]). *A graph is p-connected if and only if every pair of vertices in the graph is p-connected.*

It is easy to show that p-connectedness of vertices in an arbitrary graph G is an *equivalence relation* on the vertex-set of G. The *equivalence classes* correspond precisely to the p-connected components of the graph.

Finally, it is worth mentioning that the task of checking whether or not a given graph is p-connected can be performed in time linear in the size of the graph [7]. At the same time, Babel and Olariu [7] have shown that given two arbitrary vertices in a graph, the task of finding a p-chain connecting them, if such a p-chain exists, can also be performed in time linear in the size of the graph.

Theorem 4.2 (Babel and Olariu [7]). *It can be tested in linear time whether a graph is p-connected. Moreover, for every pair of vertices in a p-connected graph, a connecting p-chain can be constructed in linear time.*

5. Minimally p-connected graphs

A vertex v in a p-connected graph G is called a *p-articulation-vertex* if $G-v$ is not p-connected. Obviously, a p-articulation-vertex in G is also a p-articulation-vertex in its complement $\overline{G}$. The following theorem presents equivalent conditions for a vertex to be a p-articulation-vertex.

Theorem 5.1 (Babel [4]). *Let $G=(V,E)$ be p-connected and $v \in V$. The following statements are equivalent*:
1. *v is a p-articulation-vertex.*
2. *There exist vertices x and y different from v such that every p-chain connecting x and y contains v.*
3. *There exists a partition V_1, V_2 of $V-\{v\}$ such that for every two vertices $x \in V_1$ and $y \in V_2$, every p-chain connecting x and y contains v.*

A p-connected graph $G=(V,E)$ is said to be *minimally p-connected* if for every choice of a vertex v, $G-v$ is not p-connected, i.e. every vertex in G is a p-articulation-vertex. It is obvious that a P_4 is minimally p-connected. Surprisingly, there are further graphs with this property.

A graph $G=(V,E)$ is termed a *spider* if its vertex-set V can be partitioned into disjoint sets S and K such that
- $|S|=|K|\geqslant 2$, S is a stable set, K is a clique;
- there exists a bijection $f: S \to K$ such that either

$$N(s)=\{f(s)\} \quad \text{for all vertices } s \in S$$

or else

$$N(s)=K-\{f(s)\} \quad \text{for all vertices } s \in S.$$

The smallest spider is the P_4, spiders with more than four vertices are referred to as *proper* spiders. If G has more than four vertices then, if the first of the two alternatives in the definition above holds, G is said to be a *thin spider*, otherwise G is a *thick spider*, as illustrated in Fig. 4. Obviously, the complement of a thin spider is a thick spider and vice versa. Furthermore, it is an easy observation that spiders are separable p-connected graphs with separation (K,S).

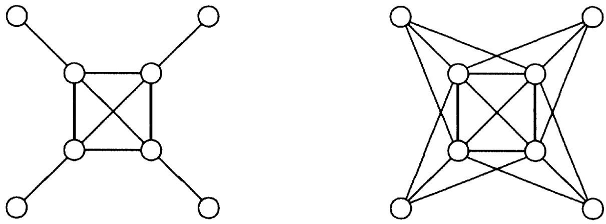

Fig. 4. The spiders with eight vertices.

It is easy to verify that spiders are minimally p-connected. Somewhat surprisingly, as the following theorem shows, they are the only graphs with this property.

Theorem 5.2 (Babel [4] and Babel and Olariu [6]). *Let $G=(V,E)$ be a p-connected graph. Then the following statements are equivalent*:
1. *G is minimally p-connected*;
2. *G contains only trivial p-chains*;
3. *G is a spider.*

It is folklore that every non-trivial connected graph contains two vertices such that the removal of either of them does not disconnect the graph. The following result extends the above result for p-connected graphs.

Theorem 5.3 (Babel [4]). *A p-connected graph which is not minimally p-connected contains at least two vertices which are not p-articulation-vertices.*

An immediate consequence of Theorem 5.2 is the following simple but important observation.

Corollary 5.4 (Babel and Olariu [6]). *If a graph G is p-connected then there is an ordering $(v_n, v_{n-1}, \dots, v_1)$ of its vertices and an integer $k \in \{4,5,\dots,n\}$ such that $G(\{v_i, v_{i-1} \dots, v_1\})$ is p-connected for $i=n, n-1, \dots, k+1$ and a spider for $i=k$.*

In other words, given a p-connected graph, we can repeatedly remove a vertex such that p-connectedness of the graph is preserved until we obtain a spider. In particular, this observation allows one to determine lower bounds on the number of P_4's which occur in a p-connected or an arbitrary graph.

Corollary 5.5 (Babel [4]). *A graph with s nontrivial p-connected components and t weak vertices contains at least $n-3s-t$ P_4's. In particular, a p-connected graph contains at least $n-3$ P_4's.*

The study of graphs which contain precisely this number of P_4's leads to the classes of p-forests and p-trees that will be discussed in Section 9.

6. Reconstructing p-connected graphs

Two vertices u and v of a graph G are said to be *partners* if for some set S of three vertices in G, both $S \cup \{u\}$ and $S \cup \{v\}$ induce a P_4 in G. Occasionally, we shall say that v has a partner in a set U if v has a partner in a P_4 contained in U. Given a subset U of V, we denote by $T(U)$ and $I(U)$ the set of vertices in $V - U$ which are totally adjacent and nonadjacent to U, respectively. If U is separable p-connected with separation (U_1, U_2) then $P(U)$ denotes the set of vertices which are adjacent exactly to the vertices from U_1. Let X be a set inducing a P_4 in G. It is a simple observation that a vertex v has a partner in X if and only if $v \notin T(X) \cup I(X) \cup P(X)$ holds. More generally, we have the following result.

Theorem 6.1 (Babel and Olariu [8]). *Let $G=(V,E)$ be an arbitrary graph and let U be a proper subset of V such that $G(U)$ is p-connected. For every vertex v in $V-U$ the following statements are equivalent*:
1. *$G(U \cup \{v\})$ is p-connected.*
2. *v does not belong to $T(U) \cup I(U)$ and, if $G(U)$ is separable, also not to $P(U)$.*
3. *There is a set X of vertices in U such that X induces a P_4 and v has a partner in X.*

This result motivates to study the following extension procedure which starts with the vertex-set of a P_4 and adds a vertex whenever it has a partner in this set.

Procedure PARTNER ADDITION(G;X)
Input: A set X inducing a P_4 in an arbitrary graph G.
Output: A p-connected subgraph U of G.
begin
  Let $U := X$;
  while there exists a vertex $v \in V - U$
  with a partner in some P_4 in U
    do $U := U \cup \{v\}$;
  return(U);
end.

Theorem 6.1 implies that the set U returned by the above procedure, induces a p-connected graph. If $U \neq V$ then no vertex v outside U has a partner in some P_4 in U. This means that $G(U \cup \{v\})$ is not p-connected and every vertex $v \in V - U$ belongs either to one of the sets $T(U)$ and $I(U)$ or, if U is separable, to $P(U)$. If U is not separable or if $P(U)$ is empty, then U is homogeneous. Otherwise, we are in the situation where U is separable and $P(U)$ is nonempty. A set U with these properties is called a *separable-homogeneous set* (for an illustration of this concept see Fig. 5). In other words, a set U is separable-homogeneous whenever U is separable p-connected

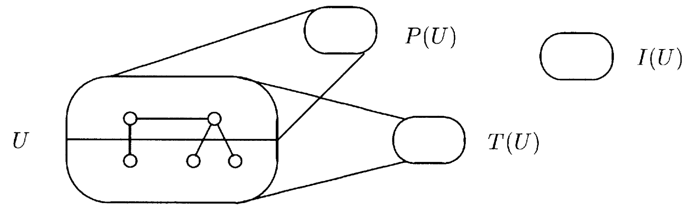

Fig. 5. A separable-homogeneous set U.

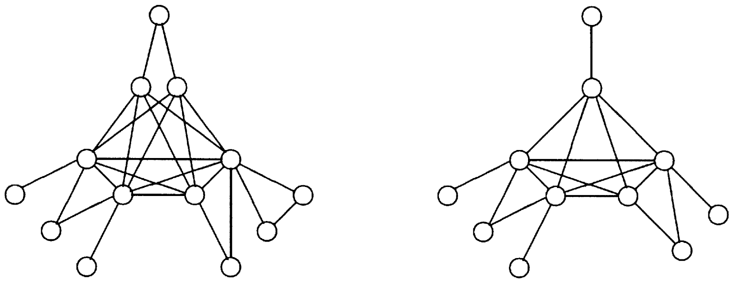

Fig. 6. A spider-like graph and its characteristic graph.

and $V = U \cup T(U) \cup I(U) \cup P(U)$ holds with $P(U) \neq \emptyset$. Using this notation we obtain the following result.

Lemma 6.2 (Babel and Olariu [8]). *Procedure PARTNER ADDITION returns a set U that induces a p-connected graph. If U is a proper subset of V then U is either homogeneous or separable-homogeneous.*

We shall say that a P_4 with vertex-set X *extends* to U *by partner addition* if for a suitable choice of vertices v, procedure PARTNER ADDITION, starting with X, returns U. Next, we characterize the graphs which contain a P_4 that extends to the whole vertex-set V.

Clearly, a nontrivial spider does not contain a P_4 that extends to V. Actually, this holds for a more general class of graphs. Note that, if G is a thin spider then the removal of all edges in the clique disconnects the graph leaving at least three connected components. Similarly, a graph G is called *thin-spider-like* if

- G is separable p-connected with separation (V_1, V_2);
- the removal of all edges in V_1 disconnects the graph leaving at least three connected components.

A graph is *thick-spider-like* if its complement is thin-spider-like. We shall refer to a graph that is thick-spider-like or thin-spider-like simply as *spider-like*. For an example of a spider-like graph see Fig. 6. The importance of spider-like graphs is exhibited in the following statement.

Theorem 6.3 (Babel and Olariu [8]). *For a graph $G=(V,E)$ the following statements are equivalent*:
1. *Some P_4 in G extends to V by partner addition.*
2. *G is p-connected and not spider-like.*
3. *Every vertex in G belongs to a P_4 that extends to V by partner addition.*

As seen before, partner addition is not a sufficient tool in order to reconstruct a p-connected graph (in the sense that p-connectedness is preserved after the addition of each vertex). Hence we have to extend the procedure by adding two vertices and proceeding with partner addition or by adding even three vertices and proceeding with partner addition. If partner addition gets stuck in a homogeneous set H, then three vertices must be added in order to extend H to a larger p-connected graph (since each P_4 which is crossing between H and $V-H$ has precisely one vertex in H). If it gets stuck in a separable-homogeneous set S then either two vertices (namely two adjacent vertices from $P(S)$ and $I(S)$ or two nonadjacent vertices from $P(S)$ and $T(S)$) or, in case $S \cup P(S)$ is a homogeneous set, again three vertices must be added. It is straightforward to verify that each p-connected graph can be reconstructed in this way.

7. Decomposing p-connected graphs

Let G be an arbitrary graph and let S be a separable-homogeneous set in G. We say that G^* results from G by *shrinking* S to a P_4 if S is replaced by a P_4 in the obvious way, i.e. a vertex v in G^* is either totally adjacent, nonadjacent or adjacent to the midpoints of the P_4, according to whether v belongs to $T(S)$, $I(S)$ or $P(S)$ in G. As the next result shows, the operations of shrinking a homogeneous set to a single vertex and that of shrinking a separable-homogeneous set to a P_4 preserve p-connectedness.

Theorem 7.1 (Babel and Olariu [8]). *Let G be a p-connected graph and let H and S be a homogeneous respectively a separable-homogeneous set in G. The following statements are satisfied*:

(a) *The graph obtained from G by shrinking H to a single vertex is p-connected.*
(b) *The graph obtained from G by shrinking S to a P_4 is p-connected.*

The next statement describes properties of separable-homogeneous sets in arbitrary, not necessarily p-connected graphs.

Theorem 7.2 (Babel and Olariu [8]). *Let G be an arbitrary graph and S, S' be separable-homogeneous sets in G with nonempty intersection such that no set contains the other. The following statements hold*:

(a) *$S \cup S'$ induces a spider-like graph.*
(b) If *$S \cup S' \neq V$ then $S \cup S'$ is homogeneous or separable-homogeneous.*

The previous theorem elucidates the way in which the separable-homogeneous sets relate to each other in an arbitrary graph. These results have been extended to reveal the interaction of maximal separable-homogeneous sets in a p-connected graph. Namely, if a p-connected graph G contains no homogeneous sets and is not spider-like, then any two maximal separable-homogeneous sets in G either coincide or are disjoint.

The previous results imply a decomposition theorem for p-connected graphs which will be stated next. For this purpose, call a graph *prime* if it contains no homogeneous set and no proper separable-homogeneous set, i.e. no separable-homogeneous set with more than four vertices.

Theorem 7.3 (Decomposition Theorem, Babel and Olariu [8]). *Let G be a p-connected graph. Exactly one of the following statements is satisfied*:
1. *G is thin-spider-like.*
2. *G is thick-spider-like.*
3. *There is a maximal prime p-connected subgraph Y of G and a unique partition P of V such that for each $U \in P$ either*
- $|U| = 1$ *and* $|U \cap Y| = 1$ *holds or*
- *U is homogeneous and* $|U \cap Y| = 1$ *holds or*
- *U is separable-homogeneous and $U \cap Y$ induces a P_4.*

This theorem will be exploited in the next section to obtain a new graph decomposition that extends the modular decomposition in the sense that it goes further in decomposing graphs which are prime with respect to the modular decomposition.

8. Homogeneous and separable-homogeneous decomposition

The primeval decomposition described in Section 3 lays the foundation of the *homogeneous decomposition* [40], which additionally involves the homogeneous sets of the graph. Given the primeval tree, it constructs a new tree representation by introducing a graph operation which, loosely speaking, replaces homogeneous sets by single vertices (this operation will also occur in our new decomposition). The homogeneous decomposition properly extends the *modular* or *substitution decomposition* [49,50], a well-investigated and extremely useful technique to decompose a graph G into certain subgraphs, called *modules*. A module M is a set of vertices in G which cannot be distinguished from vertices in $V - M$, i.e. each vertex outside M is either totally adjacent or nonadjacent to M. In particular, the graph itself and each single vertex is considered to be a module. In this sense, homogeneous sets are precisely the nontrivial modules of a graph. The result of the modular decomposition is a tree that describes the submodules of G.

By virtue of Theorem 7.3 we are able to go substantially further and decompose graphs which are prime with respect to the modular and to the homogeneous decomposition. For this purpose, we now introduce several graph operations which are meant to capture the decomposition of spider-like graphs and which reflect the substitution of homogeneous and separable-homogeneous sets by single vertices and P_4's, respectively.

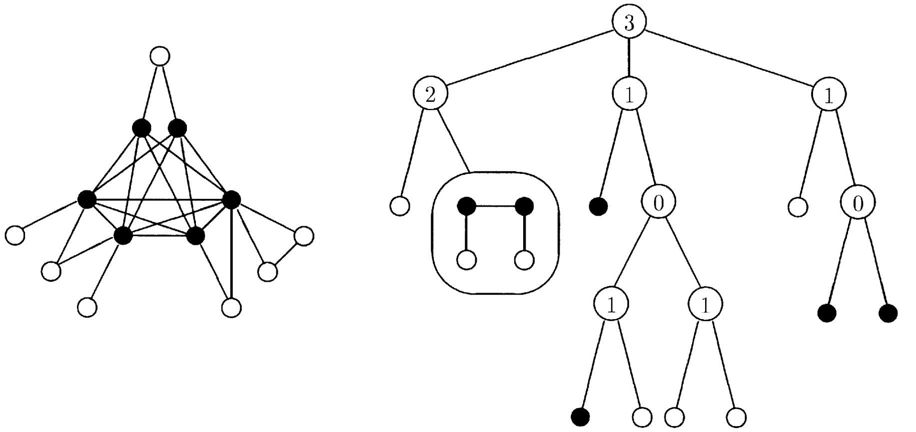

Fig. 7. Decomposing a thin-spider-like graph.

In a thin-spider-like graph G, the removal of the edges in the first set of the associated separation leaves at least three connected components. Let G_i, $i=1,2,\dots,t$, denote the subgraphs of G which are induced by the vertex-sets of these components. Note that, since the characteristic graph of G is a split graph, the characteristic graph of each subgraph G_i is a split graph, too. The reverse of the decomposition of G into the subgraphs $G_1,\dots,G_t$ is reflected by the following operation.

Let $G_i=(V_i,E_i)$, $i=1,2,\dots,t$, denote disjoint graphs with $t\geqslant 3$ and let $V_i=V_i^1\cup V_i^2$. The graph $G=(V,E)$ is said to arise from $G_1,\dots,G_t$ by a ③ operation if

- $V=\bigcup_{i=1}^t V_i$ and
- $E=\bigcup_{i=1}^t E_i\cup\{xy\,|\,x\in V_i^1, y\in V_j^1, 1\leqslant i<j\leqslant t\}$.

Similarly, $G=(V,E)$ arises from $G_1,\dots,G_t$ by a ④ operation if

- $V=\bigcup_{i=1}^t V_i$ and
- $E=\bigcup_{i=1}^t E_i\cup\{xy\,|\,x\in V_i^2, y\in V_j, 1\leqslant i<j\leqslant t\}$.

Clearly, a thin-spider-like graph results from a ③ operation applied to certain induced subgraphs, a thick-spider-like graph results from a ④ operation. We refer the reader to Fig. 7 illustrating the decomposition tree associated with a thin spider-like graph.

The reverse of shrinking homogeneous sets to single vertices and separable-homogeneous sets to P_4's is established by the following operation. Let $G_0=(V_0\cup\{y_1,y_2,\dots,y_s\}\cup X_{s+1}\cup\dots\cup X_t,E_0)$ be a graph such that each of the sets X_j induces a P_4. Let further $G_i=(V_i,E_i)$, $i=1,\dots,s$, be arbitrary and $G_j=(V_j,E_j)$ be separable p-connected graphs with separation (V_j^1,V_j^2), $j=s+1,\dots,t$. The graph $G=(V,E)$ arises from $G_0,G_1,\dots,G_t$ by means of a ⑤ operation if G is obtained by replacing every vertex y_i in G_0 by the graph G_i and each set X_j by the separable p-connected graph G_j in the obvious way, i.e.

- $V=\bigcup_{i=0}^t V_i$ and
- $E=\bigcup_{i=0}^t E_i-E'\cup E''\cup E'''$,

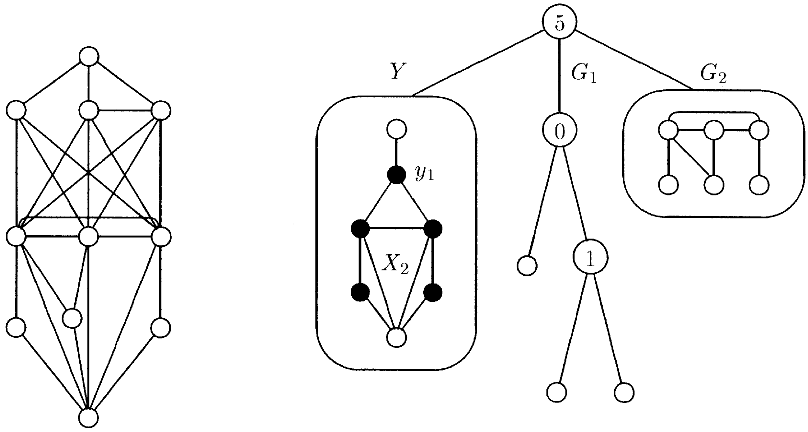

Fig. 8. Decomposing a p-connected graph.

where E' denotes the edges in G_0 which are incident to a vertex y_i or to a vertex from a set X_j, the set E'' arises by joining each vertex in V_i to every neighbor of y_i, and E''' arises by joining each vertex from V_j to every vertex which is totally adjacent to X_j and every vertex from V_j^1 to every vertex which is adjacent precisely to the midpoints of the P_4 induced by X_j (see also the example in Fig. 8).

As the following result shows, all graphs are constructible from certain atomic subgraphs by means of the operations defined above. More precisely, we have the following result.

Theorem 8.1 (Babel and Olariu [8]). *Every graph can be obtained uniquely from prime p-connected subgraphs by a finite sequence of operations* ⓪, ①, …, ⑤.

Theorems 3.1 and 7.3 suggest, in a natural way, a tree representation for arbitrary graphs which is unique up to isomorphism. The tree $T(G)$ belonging to a graph G will be called the *separable-homogeneous tree* of G. The internal nodes of $T(G)$ are labeled with integers $i \in \{0, 1, \ldots, 5\}$, where an i-node means that the subgraph associated with this node as a root is constructed from the subgraphs associated with its children by an ⓘ operation. The leaves of the tree are the prime p-connected subgraphs of G (for recursive procedures which describe the formal construction of the homogeneous and the separable-homogeneous tree of an arbitrary graph G we refer to [40,8]).

9. p-trees

A different line of research tries to find new graph classes which constitute the analogue of known graph classes in the context of p-connectedness. For example, a *p-cycle* denotes a graph where each vertex belongs to at least two P_4's and which is minimal with this property, i.e. every proper induced subgraph has a vertex which

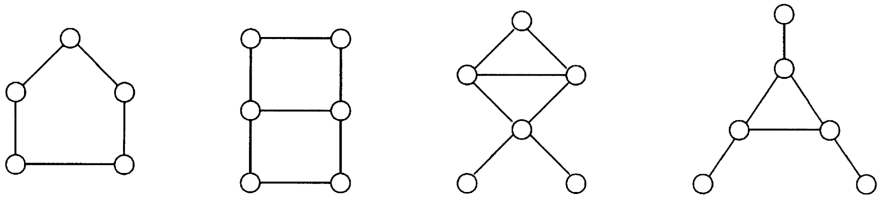

Fig. 9. Some examples of p-cycles.

belongs to at most one P_4. Obviously, p-cycles are p-connected graphs and do not contain any p-end-vertices (a vertex is a *p-end-vertex* if it belongs to exactly one P_4). Important examples of p-cycles are the chordless cycles C_k of length $k \geqslant 5$ and their complements, and spiders with six vertices. Some further examples are depicted in Fig. 9.

A *p-forest* is a graph which does not contain an induced p-cycle. The p-connected components of a p-forest are called *p-trees*. Thus, a p-tree is a p-connected graph without induced p-cycles. The smallest p-tree is the P_4 which occasionally is called the *trivial* p-tree.

p-forests and p-trees have been introduced and investigated by Babel in [2,4,32]. Among others, it has been shown that p-forests properly contain the classes of cographs and P_4-reducible graphs. On the other hand, p-forests are weakly triangulated and even brittle graphs.

As it turns out, p-trees are provided with structural properties which can be expressed in a quite analogous way to the numerous characterizations of ordinary trees. Here is the main result, the beauty of which gives an additional motivation to thoroughly explore this class of graphs.

Theorem 9.1 (Babel [4]). For a graph $G=(V,E)$ the following statements are equivalent:
1. *G is a p-tree.*
2. *G is p-connected and every p-connected induced subgraph of G contains at least one p-end-vertex.*
3. *G is p-connected, contains no proper induced spider and has exactly* $n-3$ P_4*'s.*
4. *G contains no induced p-*cycle *and has exactly* $n-3$ P_4's.
5. *G is p-connected, contains no proper induced spider and each vertex of a p-connected induced subgraph H of G is either a p-end-vertex or a p-articulation-vertex in H.*
6. *G contains no proper induced spider and each pair of vertices is connected either by a unique nontrivial p-chain or by trivial p-chains only.*

The next result points out a further interesting property of p-trees which also corresponds to a well-known property of trees.

Lemma 9.2 (Babel [4]). *A p-tree G contains at least two p-end-vertices u and v. Furthermore, if G is a nontrivial p-tree, then u and v do not belong to a common* P_4.

The previous structural results lay the foundation for the study of algorithmic properties of p-trees and p-forests. At the same time, homogeneous sets play an important role in the design of efficient algorithms. Fortunately, as asserted by the first part of the next statement, these sets are of a very simple nature.

Lemma 9.3 (Babel [2,32]). *Let G be a p-tree. Then the following holds*:
(a) *Each homogeneous set of G induces a cograph*,
(b) *G contains no separable-homogeneous sets.*

Based on the extension procedure presented in Section 6, Babel developed in [3] an efficient method for traversing a p-tree such that p-connectedness is preserved in each step and all P_4's of the graph are detected. The method depends on the following result which is an immediate consequence of the previous statement and of Lemma 6.2.

Theorem 9.4 (Babel [2,3]). *In a p-tree* $G=(V,E)$ *every* P_4 *extends to* V *by partner addition.*

The traversing technique can be extended to find efficiently the p-connected components of a p-forest. As a direct application, this allows to construct in linear time a perfect order for a p-forest which, in turn, allows to solve the classical optimization problems maximum clique, minimum coloring, maximum stable set and minimum clique cover [3].

The key for the construction of efficient recognition and isomorphism algorithms is the detailed study of the structure of p-chains in p-trees. For that purpose, call a p-chain $(v_1, v_2, \ldots, v_k)$ *simple* if there is no P_4 in $G(\{v_1, v_2, \ldots, v_k\})$ different from $(v_i, v_{i+1}, v_{i+2}, v_{i+3})$ with $1 \leqslant i \leqslant k-3$. It is an easy observation that every p-chain in a p-tree must be simple. In particular, every pair of vertices in a p-tree is connected by a simple p-chain (surprisingly, this is not true for arbitrary p-connected graphs).

Obviously, each path P_k consisting of $k \geqslant 4$ vertices and the complement of such a path are simple p-chains. Further examples are the graphs Q_k, $k \geqslant 4$, and R_k, $4 \leqslant k \leqslant 7$, as illustrated in Fig. 10, and the complements of these graphs. We omit a formal definition of the graphs Q_k and R_k, since their construction should be evident from the examples given in Fig. 10. As it turns out, there are no further simple p-chains.

Theorem 9.5 (Babel [2,32]). *A simple p-chain is isomorphic to one of the graphs* P_k, Q_k $(k \geqslant 4)$, R_k $(4 \leqslant k \leqslant 7)$, *or to the complement of one of these graphs.*

Using this result a further characterization of p-trees has been found in [2]. Roughly speaking, a p-tree consists of a simple p-chain which is extended – in a certain simple

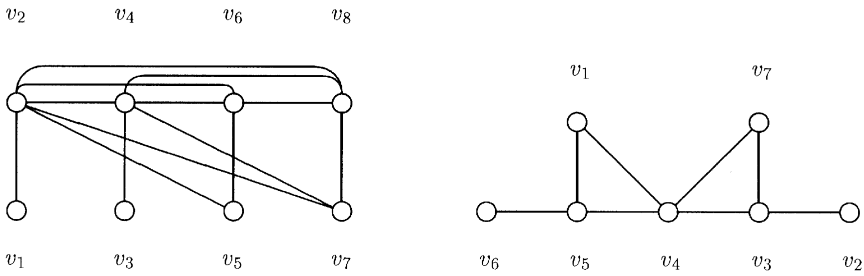

Fig. 10. The simple p-chains Q_8 and R_7.

pattern – by a number of p-end-vertices which, in turn, are replaced by cographs. This characterization allows to construct linear-time recognition as well as linear-time isomorphism algorithms for p-trees and p-forests.

10. Graphs with few P_4's

In recent years the study of graphs which – in some local sense – contain only a restricted number of P_4's, turned out to be of steadily increasing importance. The starting point and the original motivation for many investigations was the class of graphs where no P_4 is allowed to exist, commonly termed *cographs*. For these graphs, which have been investigated independently by many authors, a large number of interesting structural results have been obtained that culminate in a tree representation which is unique up to labeled tree isomorphism (see e.g. [16] for a discussion).

The study of cographs has been extended by Jamison and Olariu to graphs which contain a restricted number of induced paths of length three. The corresponding classes are called P_4-reducible, P_4-sparse, P_4-extendible and P_4-lite. In particular, P_4-*reducible graphs* [34] are defined as those graphs where no vertex belongs to more than one P_4. A graph is called P_4-*sparse* [37] if no set of five vertices induces more than one P_4 (this class was originally introduced by Hoang in [27]). Obviously, P_4-sparse graphs generalize both cographs and P_4-reducible graphs. P_4-*extendible graphs* [36] are graphs where each p-connected component consists of at most five vertices. Finally, a graph is P_4-lite [35] if every induced subgraph with at most six vertices either contains at most two P_4's or is isomorphic to a spider. It has been shown that these classes are provided with very nice structural properties. The most remarkable feature is the existence of a unique tree representation.

Historically, the previous classes have been presented and studied before the notion of p-connectedness has been introduced. With the knowledge of the results of Section 2, Babel and Olariu [6] proposed the generalizing concept of (q,t)-*graphs*. In such a graph no set of at most q vertices is allowed to induce more than t distinct P_4's. In this sense, the cographs are precisely the $(4,0)$-graphs and the P_4-sparse graphs coincide with the $(5,1)$-graphs. Furthermore, it turns out that the C_5-free P_4-extendible graphs are exactly the $(6,2)$-graphs. The following theorem states that the p-connected components of these graphs are of a rather simple structure.

Theorem 10.1 (Babel and Olariu [6]). *A p-connected component of a $(q,q-4)$-graph either contains less than q vertices or is isomorphic to a spider.*

As pointed out in Section 3, the Structure Theorem allows to give for any graph a tree representation which is unique up to isomorphism (the *primeval tree*). It is well known that each cograph arises from single vertices by a sequence of operations disjoint union and disjoint sum. Thus, in this special case, the leaves of the associated tree represent the vertices of the graph and the labels of the interior nodes are 0 and 1. For $(q,q-4)$-graphs the interior nodes are labeled 0, 1 and 2, the leaves represent graphs of restricted size or graphs which are isomorphic to graphs of a very simple nature, namely spiders.

In [23] Giakoumakis and Vanherpe studied structural and algorithmic properties of *extended P_4-reducible* and *extended P_4-sparse graphs*. These classes are obtained from P_4-reducible and P_4-sparse graphs, respectively, by also allowing C_5's as p-connected components. Hence, in the first case, the nontrivial leaves of the associated primeval tree are P_4's and C_5's, in the second case they are spiders and C_5's.

Another generalization of the previously mentioned graph classes are *P_4-tidy graphs*. They were introduced by I. Rusu and studied by Giakoumakis et al. in [22]. A graph is P_4-tidy if no P_4 has more than one partner (in other words, for every P_4 there exists at most one vertex outside which, together with three of its vertices, induces a P_4). As it turns out, P_4-tidy graphs strictly contain the classes of cographs, P_4-reducible, P_4-sparse, P_4-extendible and P_4-lite graphs. In our terminology, the structure of P_4-tidy graphs can be described as follows:

Theorem 10.2 (Giakoumakis et al. [22]). *A p-connected component of a P_4-tidy graph is either isomorphic to a spider (possibly with one vertex replaced by a homogeneous set of cardinality 2) or to one of the graphs P_5, $\overline{P_5}$, C_5.*

With the knowledge of the results of the previous section, recently the classes of $(q,q-3)$-graphs have been analyzed by Babel in [2]. Clearly, every $(q,q-4)$-graph is also a $(q,q-3)$-graph since here, in each set of q vertices, one more P_4 may be present. These graphs have also very nice structural properties which are described in the next theorem (a *disc* is a cycle or the complement of a cycle with at least five vertices).

Theorem 10.3 (Babel [2]). *A p-connected component of a $(q,q-3)$-graph, $q\geqslant 7$, either contains less than q vertices or is isomorphic to a spider, to a disc or to a p-tree.*

Hence, the leaves of the primeval tree associated to a $(q,q-3)$-graph represent spiders, discs, p-trees or graphs of restricted size. The importance of $(q,q-3)$-graphs becomes evident from the fact that they constitute rather comprehensive graph classes. In particular, the $(7,4)$-graphs properly contain all cographs, P_4-reducible graphs, P_4-sparse graphs, p-trees, and p-forests. The $(9,6)$-graphs additionally contain all extended P_4-reducible, extended P_4-sparse and P_4-extendible graphs.

11. Algorithmic features and applications

Graph decompositions are a very powerful tool to simplify difficult combinatorial optimization problems. In a divide-and-conquer manner, a problem on a graph is solved by independently studying the parts of the graph, and then combining the solutions for the parts into a solution for the whole graph. Often, we want to find certain graph parameters. In this context, the central questions read as follows.

- How fast can we construct the decomposition tree ?
- Given a graph together with its decomposition tree, how can we compute the parameters for the graph given the parameters for the leaves ?
- For which graph classes does this imply an efficient solution method ?

In the last years a large number of papers appeared giving (partial) answers to these questions. The following list of results is certainly far away from being complete.

The pioneering algorithm for the construction of tree representations is the algorithm for cographs described in [19]. Based on the techniques used there, linear-time algorithms have been developed for some graph classes with few P_4's, namely for P_4-reducible graphs [39], P_4-sparse graphs [38] and P_4-extendible graphs [30]. Later on, a linear-time algorithm has been presented in [9] which constructs the primeval decomposition tree of an arbitrary graph. This implies, among others, that the p-connected components of a graph can be found in linear time. A refinement of the former algorithm provides a linear-time method to obtain the homogeneous decomposition tree [9]. It is based, partly, on known methods which find the modular decomposition of a graph [20,48].

The algorithms [19,30,38,39] immediately imply linear-time recognition methods for the corresponding graph classes. Using the primeval decomposition algorithm of [9], the $(q,q-4)$-graphs and – combined with the recognition algorithm for p-trees [2] – even the $(q,q-3)$-graphs can be recognized in linear time if the value of q is fixed [2,6].

Assume now that an arbitrary graph is given together with its primeval (or homogeneous or separable-homogeneous) decomposition tree. General techniques for the computation of the clique number, the stability number, the chromatic number and the minimum clique cover number, even in the weighted case, are described in [2]. The problem of triangulating a graph and computing associated parameters such as minimum fill-in and treewidth is treated in [5], for results concerning the scattering and path covering number, see e.g. [31]. Other applications including dominating set, Steiner tree, vulnerability, vertex ranking, clustering problems, etc., are in prospect. One of the most important consequences of the unique tree representations is that the isomorphism problem can be solved in polynomial time whenever the isomorphism classes of the graphs associated to the leaves are known. This immediately follows from the fact that labeled tree isomorphism is solvable in linear time [1].

The latter observation implies linear-time (or at least polynomial-time) isomorphism tests for all the special graph classes mentioned in the previous section [2,6,16,19,30, 38,39]. Moreover, for a number of classes, the tree representations imply linear-time algorithms for problems which are NP-hard in general. This includes, among others, maximum clique, maximum stable set, minimum coloring, minimum clique cover [2,3,16,22,23,41], treewidth, pathwidth, minimum fill-in [5,10,18,41], clustering and domination [17], path covering number, scattering number and hamiltonicity [16,22,31, 42,45]. Finally, linear-time algorithms have also been presented to find maximum matchings in special graph classes [21,53].

Recently, increased attention has also been payed to the construction of parallel algorithms, see e.g. [26,33,43–46].

References

[1] A.V. Aho, J.E. Hopcroft, J.D. Ullman, The Design and Analysis of Computer Algorithms, Addison-Wesley, Reading, MA, 1974.

[2] L. Babel, On the P_4-structure of graphs, Habilitationsschrift, Zentrum Mathematik, Technische Universität München, 1997.

[3] L. Babel, Traversing and optimizing tree-like P_4-connected graphs, Report No. M-9711, Zentrum Mathematik, Technische Universität München, 1997.

[4] L. Babel, Tree-like P_4-connected graphs, Discrete Math. 191 (1998) 13–23.

[5] L. Babel, Triangulating graphs with few P_4s, Discrete Appl. Math. 89 (1998) 45–57.

[6] L. Babel, S. Olariu, On the isomorphism of graphs with few P_4s, Discrete Appl. Math. 84 (1998) 1–3.

[7] L. Babel, S. Olariu, A new characterization of P_4-connected graphs, in: F. d'Amore, P. Franciosa, A. Marchetti-Spaccamela (Eds.), Graph-Theoretic Concepts in Computer Science, 22nd International Workshop, WG'96, Lecture Notes in Computer Science, vol. 1197, Springer, Berlin, 1996, pp. 17–30.

[8] L. Babel, S. Olariu On the Separable-homogeneous Decomposition of Graphs, in: R.H. Möhring (Ed.), Graph-Theoretic Concepts in Computer Science, 23rd International Workshop, WG'97, Lecture Notes in Computer Science, vol. 1335, Springer, Berlin, 1997, pp. 25–37.

[9] S. Baumann, A linear algorithm for the homogeneous decomposition of graphs, Report No. M-9615, Zentrum Mathematik, Technische Universität München, 1996.

[10] H.L. Bodlaender, R.H. Möhring, The pathwidth and treewidth of cographs, SIAM J. Discrete Math. 6 (1993) 181–188.

[11] J.A. Bondy, U.S.R. Murty, Graph Theory with Applications, North-Holland, Amsterdam, 1976.

[12] V. Chvátal, A semi-strong perfect graph conjecture, Ann. Discrete Math. 21 (1984) 279–280.

[13] V. Chvátal, C. Hoang, On the P_4-structure of perfect graphs, I: Even decompositions, J. Combin. Theory Ser. B 39 (1985) 209–219.

[14] V. Chvátal, On the P_4-structure of perfect graphs, III: Partner decompositions, J. Combin. Theory Ser. B 43 (1987) 349–353.

[15] V. Chvátal, W.J. Lenhart, N. Sbihi, Two-colourings that decompose perfect graphs, J. Combin. Theory Ser. B 49 (1990) 1–9.

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

[17] D.G. Corneil, Y. Perl, Clustering and domination in perfect graphs, Discrete Appl. Math. 9 (1984) 27–39.

[18] D.G. Corneil, Y. Perl, L.K. Stewart, Cographs: recognition, applications and algorithms, Congr. Numer. 43 (1984) 249–258.

[19] D.G. Corneil, Y. Perl, L.K. Stewart, A linear recognition algorithm for cographs, SIAM J. Comput. 14 (1985) 926–934.
[20] A. Cournier, M. Habib, A new linear time algorithm for modular decomposition, Trees in Algebra and Programming, Lecture Notes in Computer Science, vol. 787, Springer, Berlin, 1994, pp. 68–84.
[21] J.L. Fouquet, I. Parfenoff, H. Thuillier, An $O(n)$ time algorithm for maximum matching in P_4-tidy graphs, Inform. Process. Lett. 62 (1997) 281–287.
[22] V. Giakoumakis, F. Roussel, H. Thuillier, On P_4-tidy graphs, Discrete Math. Theoret. Comput. Sci. 1 (1997) 17–41.
[23] V. Giakoumakis, J. Vanherpe, On extended P_4-reducible and extended P_4-sparse graphs, Theoret. Comput. Sci. 180 (1997) 269–286.
[24] M.C. Golumbic, Algorithmic Graph Theory and Perfect Graphs, Academic Press, New York, 1980.
[25] R.B. Hayward, L.B. Lenhart, On the P_4-structure of perfect graphs, IV: partner graphs, J. Combin. Theory Ser. B 48 (1990) 135–139.
[26] X. He, Parallel algorithms for cograph recognition with applications, J. Algorithms 15 (1993) 284–313.
[27] C.T. Hoáng, Doctoral Dissertation, McGill University, Montreal, 1985.
[28] C.T. Hoáng, On the P_4-structure of perfect graphs, II: odd decompositions, J. Combin. Theory Ser. B 39 (1985) 220–232.
[29] C.T. Hoáng, S. Hougardy, F. Maffray, On the P_4-structure of perfect graphs, V: overlap graphs, J. Combin. Theory Ser. B 67 (1996) 212–237.
[30] W. Hochstättler, H. Schindler, Recognizing P_4-extendible graphs in linear time, Technical Report No. 95.188, Universität zu Köln, 1995.
[31] W. Hochstättler, G. Tinhofer, Hamiltonicity in graphs with few P_4s, Computing 54 (1995) 213–225.
[32] L. Babel, Recognition and isomorphism of tree-like P_4-connected graphs, Discrete Appl. Math, to appear.
[33] B. Jamison, R. Lin, S. Olariu, A fast parallel recognition algorithm for P_4-reducible graphs, Proceedings of Fifth Jerusalem Conference on Information Technology, Jerusalem, Israel, 1990, pp. 402–413.
[34] B. Jamison, S. Olariu, P_4-reducible graphs, a class of uniquely tree representable graphs, Stud. Appl. Math. 81 (1989) 79–87.
[35] B. Jamison, S. Olariu, A new class of brittle graphs, Stud. Appl. Math. 81 (1989) 89–92.
[36] B. Jamison, S. Olariu, On a unique tree representation for P_4-extendible graphs, Discrete Appl. Math. 34 (1991) 151–164.
[37] B. Jamison, S. Olariu, A unique tree representation for P_4-sparse graphs, Discrete Appl. Math. 35 (1992) 115–129.
[38] B. Jamison, S. Olariu, Recognizing P_4-sparse graphs in linear time, SIAM J. Comput. 21 (1992) 381–406.
[39] B. Jamison, S. Olariu, A linear-time algorithm to recognize P_4-reducible graphs, Theoret. Comput. Sci. 145 (1995) 329–344.
[40] B. Jamison, S. Olariu, p-components and the homogeneous decomposition of graphs, SIAM J. Discrete Math. 8 (1995) 448–463.
[41] B. Jamison, S. Olariu, Linear time optimization algorithms for P_4-sparse graphs, Discrete Appl. Math. 61 (1995) 155–175.
[42] H.A. Jung, On a class of posets and the corresponding comparability graphs, J. Combin. Theory Ser. B 24 (1978) 125–133.
[43] R. Lin, S. Olariu, An NC recognition algorithm for cographs, J. Parallel Distributed Comput. 13 (1991) 76–90.
[44] R. Lin, S. Olariu, An optimal parallel matching algorithm for cographs, J. Parallel Distributed Comput. 22 (1994) 26–37.
[45] R. Lin, S. Olariu, G. Pruesse, An optimal path cover for cographs, Comput. Math. Appl. 30 (1995) 75–83.
[46] R. Lin, S. Olariu, A fast parallel algorithm to recognize P_4-sparse graphs, Discrete Appl. Math. 81 (1997) 191–215.
[47] L. Lovász, A characterization of perfect graphs, J. Combin. Theory Ser. B 13 (1972) 95–98.
[48] R. McConnell, J. Spinrad, Linear-time modular decomposition and efficient transitive orientation of comparability graphs, Fifth Annual ACM-SIAM Symposium of Discrete Algorithms, 1994, pp. 536–545.

[49] R.H. Möhring, Algorithmic aspects of comparability graphs and interval graphs, in: I. Rival (Ed.), Graphs and Orders, Dordrecht, Holland, 1985.
[50] R.H. Möhring, F.J. Rademacher, Substitution decomposition and connections with combinatorial optimization, Ann. Discrete Math. 19 (1984) 257–356.
[51] S. Olariu, On the structure of unbreakable graphs, J. Graph Theory 15 (1991) 349–373.
[52] B.A. Reed, A semi-strong perfect graph theorem, J. Combin. Theory Ser. B 43 (1987) 223–240.
[53] C.H. Yang, M.S. Yu, An $O(n)$ time algorithm for maximum matchings in cographs, Inform. Process. Lett. 47 (1993) 89–93.