Classical graphs decomposition and their totally decomposable graphs

Ruzayn Quaddoura, Khalid Mansour,

Zarqa Private University, Zarqa, Jordan

Summary
In solving optimization problems on graphs, graph decomposition is considered to be a powerful tool for obtaining efficient solutions for these problems. Totally decomposable graphs with respect to some method of graphs decomposition are relevant to many different areas of applied mathematics and computer science. There is a considerable number of results in this area. The goal of this paper is to survey the state of art of the famous methods of graphs decomposition and their totally decomposable graphs.

Key words:
Graph Problems, Complexity

1. Introduction

In several domains of scientific activities, the efficient solution of a problem P is obtained by dividing the initial problem into sub-problems, then finding a solution for each one and finally constructing a solution for the initial problem from the obtained solutions of these sub-problems.

When a problem P is represented by a graph *G*, the above procedure can be used as follows: First, divide *G* into a set *H* of subgraphs Then apply a solution for each sub-graph in *H* and finally deduce the solution of P from the obtained solutions of each graph in *H*. Frequently the set *H* can be obtained by applying recursively to *G* one or more operators of decomposition.
Thus, the existence and the efficiency of a solution of problem P using the previous procedure depend on three issues:

1. Is it possible to decompose *G* efficiently into a set *H* of subgraphs? We recall that several problems of graphs decomposition are NP-complete (partition into forests, cliques, isomorphic subgraphs etc [GJ79]).
2. What is the complexity of a solution of P for a graph of *H*?
3. How and with which complexity can we deduce the solution of P for *G* from the given solutions of subgraph in *H*?

A popular approach to graph decomposition involves associating with a given graph *G* a rooted tree *T*(*G*) whose internal nodes correspond to the operators of decomposition that are applied and whose leaves correspond to certain subgraphs of *G* (e.g. vertices, edges, cliques, stable sets, cutsets). Of a particular interest are graphs for which the following conditions hold :

a) *T*(*G*) can be obtained efficiently, that is, in time polynomial in the size of *G*;
b) The tree *T*(*G*) is unique (up to isomorphism). This exclusively will permit to transfer the difficulty of recognition problem of *G* to the verification of a set of properties that concern *T*(*G*).

Tree representation satisfying the conditions mentioned above have been obtained for several classes of graphs including the cographs [CPS85], interval graphs [BL79], chordal graphs [Gol80], maximal outerplaner graphs [BJM78], P_4- reducible graphs [JO89], P_4-extendible graphs [JO91], P_4-sparse graphs [JO92], among many others. One of the best exponents of graphs are those that are totally decomposable. A graph *G* is totally decomposable with respect to the operators of decomposition that are used if it can be decomposed to its vertices by a finite sequence of some of these operators. In this case the set of leaves of *T*(*G*) is the set of vertices of *G*. In the majority of cases we consider that a problem P has a trivial solution for the singletons, consequently the difficulty of the solution of P is subject to points 1 and 3 above.

The purpose of this paper is to present the state of the art about the most known methods of graph decomposition and the classes of graphs that are totally decomposable with respect to these methods.
The paper is organized as follows : In the rest of this section, we are giving the basic concepts to be used throughout this paper. Section 2 presents the modular decomposition and cographs.. Section 3 presents the homogeneous decomposition, P_4-reducible graphs and P_4-sparse graphs. Section 4 presents split decomposition and completely separable graphs.

Manuscript received January 5, 2010
Manuscript revised January 20, 2010

The set of vertices of a graph G is denoted by $V(G)$ and the set of its edges by $E(G)$, (or simply V and E respectively) with cardinalities $|V(G)| = n$ and $|E(G)| = m$. For $X \subseteq V(G)$, $G[X]$ will denote the subgraph of G induced by X. The neighbor of a vertex v is $N(v) = \{w \mid vw \in E(G)\}$, while $N(X)$ ($X \subseteq V(G)$) is the set of vertices outside X adjacent to at least one vertex of X. A clique is a set of pairwise adjacent vertices and a stable set is a set of pairwise nonadjacent vertices. A graph whose vertices form a clique is called a complete graph. A graph $G = (V, E)$ is a star if there is a vertex u such that $E = \{uv : v \in V - \{u\}\}$. A chordless path on k vertices is denoted by P_k, a chordless cycle on k vertices is denoted by C_k, while a C_k with $k \geq 5$ is called long cycle or hole. Let Z be a set of graphs, we shall say that a graph G is Z-free if no induced subgraph of G is isomorphic to a graph of Z.

2. Modular decomposition

This form of decomposition is known by different names: Joint of family of graphs [Jol73], substitution decomposition [MR84], lexicographic sum [Sab61], X-joint [Hir51]. Modular decomposition can be seen as the research of a partition of the vertex set of a graph into modules.

Definition 1 Let $G = (V, E)$ be a graph. A set $A \subseteq V$ is called a module if for every pair of vertices x, y of A, x and y have the same neighbor in the exterior of A. (i.e. $N(x) - A = N(y) - A$)

This notion of module appears also in the bibliography under the terms: homogeneous part [Hab81], [Gal67], autonomous set [MR84], perfect set [Gol80], interval [Ill89], [Cou93], clan [ER90], [EGMS97]. The empty set, the singletons, and the vertex set of a graph are its trivial modules. A graph is called prime or indecomposable if its modules are only the trivial modules.

Theorem 1 [Hab81] Let $G = (V, E)$ be a graph such that $|V| \geq 2$, then one of the following cases is hold :

a. G is not connected.
b. $\overline{G}$ is not connected.
c. There exists $Y \subseteq V$, $|Y| \geq 4$, and a unique partition P of V such that Y is a maximal prime subgraph of G and all $S \in P$ are modules with $|S \cap Y| = 1$

Theorem 1 assures a decomposition scheme for any graph that is called modular decomposition. Obviously, a tree is associated with modular decomposition. The internal nodes of this tree represent some graph operations that is defined next, while its leaves correspond to prime graphs.

Definition 2 Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be nontrivial graphs. The graph $G = (V, E)$ is said to arise from G_1 and G_2 by a 0-operation (resp. 1-operation) if $V = V_1 \cup V_2$ and $E = E_1 \cup E_2$ (resp. $E = E_1 \cup E_2 \cup \{xy : x \in V_1, y \in V_2\}$).

Obviously, operation 0 and 1 reflect conditions a and b of theorem 1.
A nontrivial module M of a graph G is called maximal if it does not properly contained in any other module of G.

Definition 3 The characteristic graph $C(G)$ of a graph G is the graph obtained by shrinking every maximal module to one representative vertex.

It is well known that if the graph G in theorem 1 (c) is not prime then every $S \in P$, $|S| > 1$, is a maximal module (see for example [Bau96]). This show that the maximal prime subgraph Y of G in theorem 1 (c) is isomorphic to the characteristic graph $C(G)$. Now we are able to define a 3-operation that reflect condition c in theorem 1.

Definition 4 Let $Y = \{(y_1,..., y_t\} \cup V_0, E_0)$, $G_i = (V_i, E_i)$, $i = 1, \ldots, t$ be nontrivial arbitrary graphs. The graph $G = (V, E)$ is said to arise from Y and $(G_1, \ldots, G_t)$ by a 3-operation if :

$$V = \cup_{i=0}^{t} V_i$$

$$E = (E_0 - \{xy_i : x \in V_0 \cup \{y_1,..., y_t\}, i = 1,...t\}) \cup_{i=1}^{t} E_i \cup E' \cup E''$$

With E' is obtained by making every vertex in V_i ($i = 1, \ldots, t$) adjacent to all vertices in V_0 adjacent to y_i, and E'' obtained by making every vertex in V_i ($i = 1, \ldots, t$) adjacent to all vertices in V_j if and only if $y_i y_j \in E_0$.

In other words, every vertex y_i in Y is replaced by the graph G_i. Note that G_i cannot represent a single vertex. Further in order to admit a unique inverse for the 3-operation we have to assume that every G_i remembers the identity of the vertex y_i it has replaced. With this notation every graph G can be obtained uniquely from its maximal prime subgraphs (which can also be single vertices) by a finite sequence of operations 0, 1, and 3.

We are in a position to describe the formal construction of the modular decomposition tree of an arbitrary graph G. The details are shown by the following recursive procedure. Figure 1 illustrates a graph G and its modular decomposition tree.

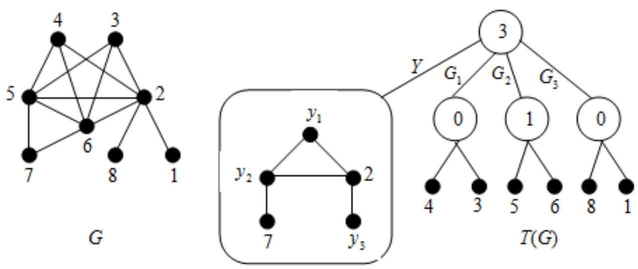

Fig 1. A graph G and its modular decomposition tree T(G)

Procedure Build Modular Decomposition Tree(G)
Input : an arbitrary graph $G = (V,E)$;
Output : the modular decomposition tree $T(G)$ corresponding to G.
begin
 if $|V| = 1$ then
  return the tree T having G as its unique vertex;
 else if G is disconnected then begin
  let $G_1, \ldots, G_p$ ($p \geq 2$) be the connected component of G;
  let $T_1, \ldots, T_p$ be the corresponding modular decomposition trees rooted at $r_1, \ldots, r_p$;
  return the tree $T(G)$ obtained by adding $r_1, \ldots, r_p$ as children of a 0-node
  end
 else if $\overline{G}$ is disconnected then begin
  let $\overline{G}_1, \ldots, \overline{G}_p$ ($p \geq 2$) be the connected components of $\overline{G}$;
  let $T_1, \ldots, T_p$ be the corresponding modular decomposition trees rooted at $r_1, \ldots, r_p$;
  return the tree $T(G)$ obtained by adding $r_1, \ldots, r_p$ as children of a 1-node
  end
 else {now G satisfies condition (c) in Theorem 1} begin
  let $C(G)$ be the characteristic graph of G;
  let α be the tree having the unique child $C(G)$;
  let $Y_1, \ldots, Y_p$ be the maximal modules of G;
  let $T_1, \ldots, T_p$ be the modular decomposition trees corresponding to $Y_1, \ldots, Y_p$,
  rooted at $r_1, \ldots, r_p$;
  return the tree $T(G)$ obtained by adding $\alpha, r_1, \ldots, r_p$ as children of a 3-node;
 end {if};
end; {Build Modular Decomposition Tree}

The efficient construction of modular decomposition tree was the subject of numerous works for a long time. D.D.Cown, L.O.James and R.G.Stanton in [CJS72] proposed an O(n^4) algorithm, M.Habib and M.C.Maurer in [HM79] and independently H.Buer and R.H.Mōring in [BM83] proposed an O(n^3) algorithm, then J.H.Mūller and R.H.Spinard in [MS89] proposed an incremental O(n^2) algorithm. Independently, R.McConnell and J.Spinard in [McS94], A.Cournier and M.Habib in [CH94], and E.Dahlhaus, J.Gustedt and R.McConnell in [DGM97] succeeded to obtain a linear algorithm in O($n + m$) time. Recently, Marc Tedder, Derek Corneil, Michel Habib, and Christophe Paul. Developed in [TCHP08] developed a simpler linear time algorithm. Finally E.Dahlhaus in [Dah95b] also proposed a parallel algorithm in O(($n + m$)$logn$) time.

The computation of modular decomposition is often the first step in many algorithms because of its succinct representation of a graph's structure. Indeed, since Gallai first noticed its importance to comparability graphs [Gal67], modular decomposition has been established as a fundamental tool in algorithmic graph theory. All efficient transitive orientation algorithms make essential use of modular decomposition (e.g., [McS00], [McS97]). It is frequently employed in recognizing different families of graphs, including interval graphs [Mo85], permutation graphs [PEL71], and cographs [CPS85]. Furthermore, restricted versions of many combinatorial optimization problems can be efficiently solved using modular decomposition ([FM05], [Gia96], [GRT97a], [GRT97b], and [GV97]). While the papers [[Mo85], [MR84]] provide older surveys of its numerous applications, new uses continue to be found, such as in the areas of graph drawing [PV06] and bioinformatics [GKB04].

2.1 Totally decomposable graphs with respect to modular decomposition

Note that a graph involving the operation 3 in its modular decomposition tree can't be reduced to its vertex set by a finite sequence of operations 0, 1, and 3, since the operation 3 is not a combination of two separable graphs. Thus the modular decomposition tree of a totally decomposable graph contains only nodes of type 0 or 1. This sort of graphs called Cographs were independently discovered under various names and studied within various context, it can be found in the literature under the names : graphs D* [Jun 78] or graphs HD [Sum 74].

Definition 3 A cograph is defined recursively as follows :

1) A graph having one vertex is a cograph.
2) If G_1, $G_2, \ldots, G_k$ are cographs then $G_1 \cup G_2 \cup \ldots \cup G_k$ is a cograph.
3) If G is a cograph then $\overline{G}$ is a cograph.

Clearly, this definition implies that any cograph can be obtained from single vertices graphs by performing a finite number of graph operations involving union and complementation. It is also clear that this class of graphs is self-complemented, that is the complement of a connected cograph is necessarily not connected. Since the path P_4 is the smallest graph that is connected and its complement is also connected then every cograph is P_4-free. Lerchs in [Ler78] studied the structural and algorithmic properties of cographs and proved that the cographs are exactly the graphs P_4-free and that these graphs admit a unique up to isomorphism tree representation. This tree representation for a cograph G is obtained by associating with G a rooted tree $T(G)$ called a cotree, whose leaves are precisely the vertices of G while the internal nodes are labeled by the symbol $\overline{\cup}$ representing complemented union. In [Ler78] it is proved that from the definition of a cograph we can easily deduce that the complement of a connected cograph is disconnected. Thus, a cograph can be reduced to single vertices by recursively complementing connected subgraphs, and for this reason cographs are also called complement reducible graphs. A top-down traversal of $T(G)$ clearly describes this decomposition of a cograph G. The internal nodes of a cotree $T(G)$ can also be labeled by 0 and 1 in such a way that two vertices are adjacent in G, if and only if their least common ancestor in $T(G)$ is labeled by 1. In this form, the cotree of a cograph G is the tree $T(G)$ of the modular decomposition of G (theorem 1). This signifies that a graph is a cograph if and only if its modular decomposition tree contains only nodes of types 0 or 1 (see figure 2). In other words, the class of cographs is the smallest class containing the graph reduced to one vertex and closed under 0 and 1 operations.

Hence, this class of graphs provides an excellent paradigm of graphs possessing a unique tree representation, and, for such graphs, many results confirmed that a great number of intractable problems have efficient algorithmic solutions such as isomorphism, colouring, clique detection, clusters, minimum weight dominating sets, minimum fill-in (the smallest number of edges that have to be added to a graph to obtain a chordal graph), and hamiltonicity [CLB81], [CP84], [CPS84].

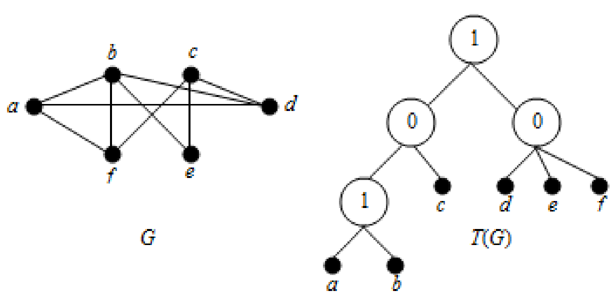

Fig. 2 A cograph G and its modular decomposition T(G)

The design of a recognition algorithm for cographs is also an interesting problem. The first linear time algorithm by Corneil et al. [CPS85] incrementally builds a cotree, starting from a single vertex and adding a new vertex at each step of the computation. It should be mentioned that Dahlhaus [Dah95a] proposed a nice parallel cograph recognition algorithm. Finally M. Habib and C. Paul in [HP05] proposed a simple linear time recognition algorithm for cographs.

3. Homogeneous Decomposition

B.Jamison and S.Olariu introduced in [JO95] another form of undirected graph decomposition called homogeneous decomposition in connection with the notion of p-connectedness. Homogeneous decomposition generalizes modular decomposition and produces a unique decomposition tree for an arbitrary graph.

Definition 4 Let $G = (V, E)$ be a graph**.** G is p-connected if for every partition of V into nonempty subsets V_1 and V_2 of V some P_4 in G contains vertices from both V_1 and V_2, (i.e. a crossing P_4). A maximal p-connected subgraph of G is called p-component.

Definition 5 A p-component H of a graph G is separable if $V(H)$ admits a partition H_1 and H_2 such that every crossing P_4 of H has its intermediate vertices in H_1 and its terminal vertices in H_2.

The structural theorem for p-connected graphs that lead to homogeneous decomposition is the following.

Theorem 2 [JO95] For an arbitrary graph $G = (V, E)$ exactly one of the following four cases holds :

a) G is not connected.
b) $\overline{G}$ is not connected.
c) G is p-connected.
d) There is a unique proper separable p-component H with a partition $\{H_1, H_2\}$ such that every vertex outside H is adjacent to all vertices in H_1 and independent of all vertices in H_2.

As in modular decomposition, theorem 2 assures a decomposition scheme for any graph called homogeneous decomposition. A rooted tree is associated with a given graph G, the leaves of this tree correspond to certain subgraphs whereas the internal nodes represent several graph operations. Obviously, operations 0 and 1 (definition 2) reflect the conditions a and b in theorem 2. Operation 2 that we are about to define will reflect condition d. More precisely :

Definition 6 Let $H = (U, F)$ be a p-connected graph which is separable with partition $\{U_1, U_2\}$ and let $K = (S, D)$ be an arbitrary graph disjoint from H. The graph $G = (V, E)$ is said to arise from H and K by a 2-operation if $V = U \cup S$ and $E = F \cup D \cup \{xy : x \in U_1, y \in S\}$.

Note that now every graph unless it is p-connected, can be obtained from its weak vertices (vertices that are contained in no p-component) and its p-components by a finite sequence of operations 0, 1 and 2 [JO95].

To be able to define the homogeneous decomposition we notice that (according to theorem 1) the operation 3 (definition 3) reflects the condition c of theorem 2. Thus, the next step of homogeneous decomposition is using the 3-operation on all p-components appearing in theorem 2 and further decomposing the appearing non trivial modules with operation 0, 1, 2 and 3. Hence, every graph can be obtained from its characteristic p-components and weak vertices by a finite sequence of operations 0, 1, 2 and 3. Finally, the homogeneous decomposition is achieved by defining a 4-operation on all characteristic p-components which are split graphs. More precisely :

A graph $G = (V, E)$ is a split graph if there is a partition of the vertex set V into nonempty disjoint sets K and S such that K is a clique and S is a stable. This partition is not unique in general. The relationship between split graphs and separable p-components can be clarified the following theorem :

Theorem 3 [JO95] A p-component B is separable if and only if the characteristic p-component $C(B)$ induces a split graph.

It is easy to exhibit a unique tree representation of a split graph which is the characteristic p-component [JO95]. To clarify the last point, let the vertices of the split graph partition into nonempty disjoint sets K and S inducing a maximal clique and a stable set respectively. For every vertex s in S, $s \cup N(s)$ is a maximal clique as well. Now, the unique tree representation is obtained by placing K at the root and for all s in S, having the clique $s \cup N(s)$ as a child of the root.

Definition 7 Let $G = (V, E)$ be a split graph and let $\{K, S\}$ be a partition of V into a maximal clique K and a stable set S. The 4-operation defined on G is the association with G a unique tree representation obtained by placing K at the root and for all s in S having the clique $s \cup N(s)$ as a child of the root.

Now, we describe the formal construction of the homogeneous tree of an arbitrary graph G by the following recursive procedure.

Procedure Build Homogeneous Decomposition Tree (G)

Input : an arbitrary graph $G = (V, E)$;
Output : the homogeneous decomposition tree $T(G)$ corresponding to G.
begin
  if $|V| = 1$ then
    return the tree T having G as its unique vertex;
  else if G is p-connected then begin
    let $C(G)$ be the characteristic graph of G;
    decompose $C(G)$ by a 4-operation and let T' be the corresponding tree rooted at a 4-node;
    let $Y_1, \ldots, Y_p$ be the maximal modules of G;
    let $T_1, \ldots, T_p$ be the homogeneous decomposition trees corresponding to $Y_1, \ldots, Y_p$, rooted at $r_1, \ldots, r_p$;
    return the tree $T(G)$ obtained by adding $\alpha, r_1, \ldots, r_p$ as children of a 3-node;
    end
  else if G is disconnected then begin
    let $G_1, \ldots, G_p$ ($p \geq 2$) be the connected component of G;
    let $T_1, \ldots, T_p$ be the corresponding homogeneous decomposition trees rooted at $r_1, \ldots, r_p$;
    return the tree $T(G)$ obtained by adding $r_1, \ldots, r_p$ as children of a 0-node
    end
  else if $\overline{G}$ is disconnected then begin
    let $\overline{G}_1, \ldots, \overline{G}_p$ ($p \geq 2$) be the connected components of $\overline{G}$;
    let $T_1, \ldots, T_p$ be the corresponding homogeneous decomposition trees rooted at $r_1, \ldots, r_p$;
    return the tree $T(G)$ obtained by adding $r_1, \ldots, r_p$ as children of a 1-node
    end
  else {now G satisfies condition (c) in Theorem 2} begin
    let G_1 and G_2 be the decomposition of G by a 2-operation;
    let T_1, T_2 be the corresponding trees rooted at r_1 and r_2;
    return the tree $T(G)$ obtained by adding r_1, r_2 as children of a 2-node;
  end {if};
end; {Build Homogeneous Decomposition Tree}

Figure 3 illustrate the difference between modular decomposition and homogeneous decomposition of the graph of figure 1.

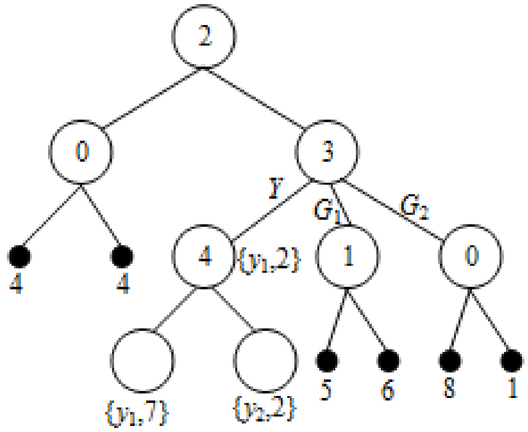

Fig. 3 The homogeneous decomposition of the graph G in figure 1.

A linear algorithm for the construction of homogeneous decomposition of a graph has been obtained by S. Baumann in [Bau96].

Note that for a number of classes of graphs the homogeneous decomposition can already be achieved in linear time including cographs [CPS85] (the modular decomposition tree for a cograph is exactly its homogeneous decomposition) as well as for a number of graphs with only few induced P_4s. P_4-reducible graphs [JO89], P_4-sparse graphs [JO92] and P_4-extendible graphs [HS95] are amongst them. Given such a tree representation for these graphs it is possible to find linear time algorithms for several optimization problems which are NP-hard in general. This is the case with the standard optimization problems (including maximum clique/stable set minimum vertex coloring/clique cover problems) [CLS81]. Other example is deciding hamiltonicity and computing the path cover number for P_4-sparse and P_4-extendible graphs [HT95] as well as standard optimization and minimum fill-in for P_4-sparse graphs [JO95b]. Another class of graphs with a simple P_4- structure are the (q, t) graphs [BO95] where for every induced subgraph with q vertices only t P_4s are allowed. In the case of the (q,q-4)-graphs linear algorithms for recognition, isomorphism and the standard optimization problems are known if the homogeneous decomposition tree is available.

3.1 Totally decomposable graphs with respect to homogeneous decomposition

As we noted in section 2.1, a graph that is totally decomposable with respect to homogeneous decomposition can't involve the operation 3 in its homogeneous decomposition tree. In the literature there are two classes of graphs that are totally decomposable with respect to homogeneous decomposition. This depends on the structure of the separable p-components. The smallest of them is the class of P_4-reducible graphs which was introduced in [JO89] as a natural generalization of the class of cographs. Another class which generalizes both cographs and P_4-reducible graphs is the class of P_4-sparse graphs.

Definition 8 A graph G is P_4-reducible graph if none of its vertices belongs to more than one induced P_4.

Jamison and Olariu in [JO89] proved the following fundamental result which is a constructive characterization of P_4-reducible graphs.

Theorem 4 [JO89] A graph G is P_4-reducible if and only if for every induced subgraph H of G exactly one of the following conditions are satisfied :

a) H is disconnected;
b) $\overline{H}$ is disconnected;
c) There exists a unique P_4 *abcd* in H such that every vertex outside {a, b, c, d} is adjacent to both b and c and nonadjacent to both a and d.

The homogeneous decomposition tree of a P_4 *abcd* is a tree having a root of type 4, marked by the clique *bc* and having the children *ab* and *cd* that are of type 1.Thus by theorem 4, a graph G is P_4-reducible if and only if the homogenous decomposition tree of G contains only nodes of type 0, 1, 2 or 4 such that any node of type 4 is marked by a clique of size exactly 2 (see figure 4). In other words, the class of P_4-reducible graphs is the smallest class containing the graph reduced to one vertex and closed under 0, 1, 2 and 4 operations such that any node of type 4 is marked by a clique of size 2.

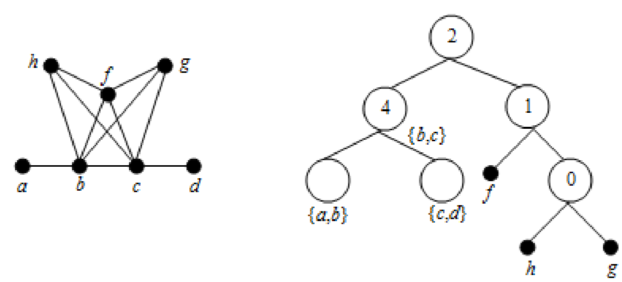

Fig. 4 A P_4-reducible graph and its homogeneous decomposition tree

Definition 9 A graph G is P_4-sparse graph if every set of five vertices in G induces at most one P_4.

The characterization of P_4-sparse graphs given by Jamison and Olariu in [JO92] is based on a special graph called spider whose definition is as follows :

A graph G is a spider if the vertex set $V(G)$ admits a partition into sets S, K and R such that :
a) S is a stable, K is a clique and $|S| = |K| \geq 2$.
b) Every vertex in R is adjacent to all vertices in K and misses all the vertices in S
c) There exists a bijection f between S and K such that either $N(x) = \{f(x)\}$ for every vertex x in S or else $N(x) = K\text{-}\{f(x)\}$ for every vertex x in S.

Observe that if a graph G is a spider then the characteristic graph of G is a split graph.

Theorem 5 [JO92] A graph G is a P_4-sparse graph if and only if for every induced subgraph of H of G exactly one of the following conditions are satisfied :

a) H is disconnected;
b) $\overline{H}$ is disconnected;
c) H is isomorphic to a spider.

Let G be a spider with partition $\{S, K, R\}$ as above. Then homogeneous decomposition tree of the characteristic graph of G is a tree having a root of type 4, marked by a clique of size at least 2 and having the children $x \cup N(x)$ where x is in S that are of type 1 .Thus by theorem 5, a graph G is P_4-sparse graph if and only if the homogenous decomposition tree of G contains only nodes of type 0, 1, 2 or 4 such that any node of type 4 is marked by a clique of size at least 2 (see figure 5). In other words, the class of P_4-sparse graphs is the smallest class containing the graph reduced to one vertex and closed under 0, 1, 2 and 4 operations such that any node of type 4 is marked by a clique of size at least 2.

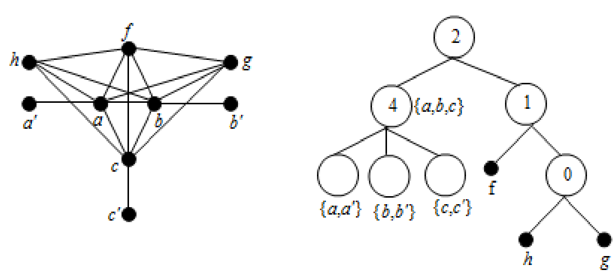

Fig. 5 A P_4-sparse graph and its homogeneous decomposition tree

4. Split decomposition

This form of decomposition was developed by W. Cunningham [Cun82] generalizes modular decomposition. Originally, the decomposition was seen primarily as a tool for dealing with directed graphs. Recently, however, the undirected version of this decomposition has been used as part of efficient algorithms for dealing with a number of different classes of graphs.

Definition 2.8 Let $G = (V, E)$ be a connected graph. A split of G is a partition of V into two sets V_1 and V_2 of such that :

1. $|V_1| \geq 2$ and $|V_2| \geq 2$,
2. There exist two sets $W_1 \subseteq V_1$ and $W_2 \subseteq V_2$ such that $\{uv : u \in V_1, v \in V_2\} = \{xy : x \in W_1, y \in W_2\}$

Note that a module (definition 1) is a particular case of the notion of split. Following this definition we define the simple split decomposition of a connected graph G as following.

Definition 4 Let $G = (V, E)$ be a connected graph and $\{V_1, V_2\}$ be a split of G. The simple split decomposition of G is the decomposition of G into G_1 and G_2 where G_i is the subgraph of G induced by V_i with an additional vertex v such that $N_{G_i}(v) = W_i$ (for $i \in \{1,2\}$).

The vertex v plays a special role and is called a marker. Note that G_1 and G_2 are connected. A graph is a prime with respect to split decomposition if it does not a split. It is clear that the smallest connected graph that is a prime consists of three vertices and every connected graph having three vertices is a prime. We say that G is decomposable into G_1 and G_2 if there is a split $\{V_1, V_2\}$ such that the simple decomposition of G by $\{V_1, V_2\}$ gives G_1 and G_2 (see figure 6).

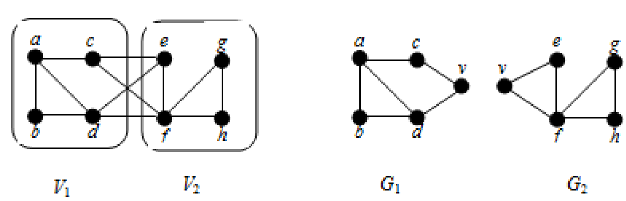

Fig. 6 A graph with a split $\{V_1, V_2\}$ and the simple decomposition by $\{V_1, V_2\}$

Additionally we define a related composition.

Definition 5 Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be two connected graphs, such that $|V_1 \cap V_2| = 1$. Let $\{v\} = V_1 \cap V_2$. Then $G_1 * G_2$ is the graph $G = (V, E)$ where :

$V = (V_1 \cup V_2) \setminus \{v\}$

$E = E(G_1) \cup E(G_2) \cup \{xy : x \in N_{G_1}(v), y \in N_{G_2}(v)\}$
$\setminus \{xv : x \in V_1 \cup V_2\}$

Obviously if G is decomposable into G_1 and G_2 then $G = G_1 * G_2$.

The split decomposition of a connected graph is the recursive decomposition of the graph using the simple decomposition until none of the obtained graphs can be decomposed further. Therefore the split decomposition of a graph gives a collection D of prime graphs. The split decomposition tree of the graph G is the tree T in which each vertex h corresponds to a prime graph G_h^* in D. Furthermore two vertices h and h' in T are adjacent if the corresponding graphs G_h^* and $G_{h'}^*$ share a common marker (see Fig. 4). Cunningham [cun82] showed that every connected graph has a unique decomposition by splits into prime graphs, stars and complete graphs with minimum number of components. Thus every connected graph can be obtained uniquely from its minimal components which are prime graphs, stars and complete graphs, by a finite sequence of the operation *. It is not hard to see that if G is not connected then the parameters we want to compute on G can be easily computed from the parameters of the connected components of G. The following recursive procedure describes the formal construction of split decomposition tree of an arbitrary connected graph.

Procedure Build Split Decomposition Tree (G)
Input : an arbitrary connected graph $G = (V, E)$;
Output : the split decomposition tree $T(G)$ corresponding to G.
Begin
 if $V(G) \leq 3$ or G is prime then
  return the tree T having G as its unique vertex;
 else begin
  find a split $\{V_1, V_2\}$ of G;
  write $G = G_1 * G_2$ as in definition 5;
  let T_1, T_2 be the corresponding split decomposition trees of G_1 and G_2 respectively;
  return the tree $T(G)$ having the two adjacent vertices T_1 and T_2;
 end {if};
end {Build Split Decomposition Tree}

The first algorithm to compute a split decomposition was given by Cunningham with a complexity of $O(n^3)$ [Cun82]. The complexity has been improved to $O(nm)$ in [GHS89] and to $O(n^2)$ in [MS94]. Finally Dahlhaus gave a linear time algorithm in [Dah94].

There are several papers presenting algorithms for optimization problems using split decomposition. Cunningham gave an algorithm for the independent set problem [Cun82]. Cicerone and Di Stefano showed how to apply this algorithm to parity graphs [CD99]. Split decomposition was also used for circle graph recognition ([GHS89], [Spi94]) and parity graph recognition [Dah00]. M. Rao in [Rao08] showed how to use the split decomposition to solve some NP-hard optimization problems on graphs. The author gave algorithms for clique problem and domination type problems. In addition, he presented an algorithm to compute a coloration of a graph using its split decomposition. Finally he showed that the clique width of a graph is bounded if and only if the clique width of each representative graph in its split decomposition is bounded.

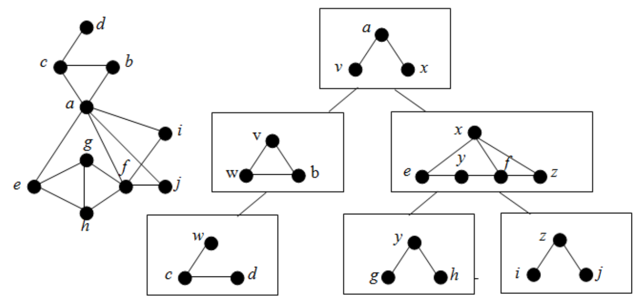

Fig. 7 A graph and its split decomposition tree. The markers are *v*, *w*, *x*, *y* and *z*.

4.1 Totally decomposable graphs with respect to split decomposition

P.L. Hammer and F. Maffray in [HM90] provided a characterization for the undirected graphs that are totally decomposable by split decomposition (i.e. those graphs that can be decomposed until obtaining only graphs having less than four vertices). This family of graphs is called in [HM90] completely separable graphs and characterized by theorem 6. In this theorem, a long cycle means a cycle with at least five vertices. A chord of a cycle is any edge joining two nonconsecutive vertices of the cycle, Two chords *ac* and *bd* are crossing if the four vertices *a*, *b*, *c*, *d* lie in this order on the cycle.

Theorem 6 [HM90] The following properties are equivalent :

a) G is completely separable;
b) G contains none of the configurations shown in figure 8;
c) Every long cycle has two crossing chords;
d) Every induced subgraph of G has a module consists of two vertices or a vertex having exactly one neighbor;

e) Given any two vertices *u* and *v*, all paths from *u* to *v* have the same length.

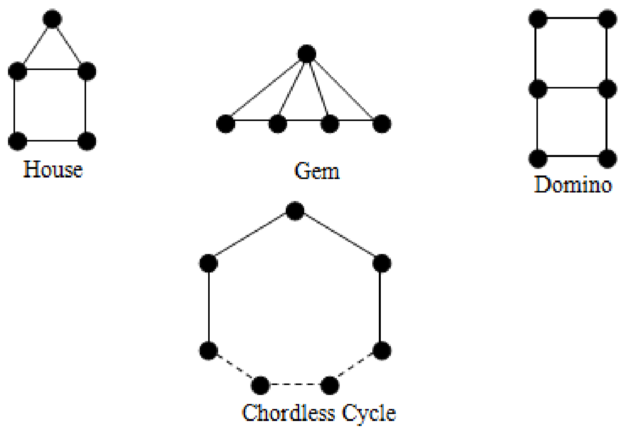

Fig. 8 The forbidden configuration of completely separable graph

Linear time algorithms for the recognition, maximum weighted stable set and maximum weighted clique for completely separable graphs are proposed in [HM90].

5. Conclusion and Future work

In solving optimization problems on graphs, graph decomposition is considered to be a powerful tool for obtaining efficient solutions of these problems. Classes of graphs that have been extensively studied in this sense are totally decomposable graphs. In this paper we have given an overview about the most known methods of graph decomposition and the classes of graphs that are totally decomposable with respect to these methods. From the viewpoint of modular decomposition, every graph can be obtained uniquely from its maximal prime subgraphs (which can be single vertices) by a finite sequence of operations 0, 1, and 3. The smallest decomposable graph with respect to modular decomposition is the cograph. From the viewpoint of homogeneous decomposition, every graph can be obtained from its characteristic *p*-components and weak vertices by a finite sequence of operations 0, 1, 2, 3 and 4. The smallest decomposable graph with respect to homogeneous decomposition is the P_4-reducible graph. From the viewpoint of split decomposition, every connected component of a graph can be obtained uniquely from its minimal components which are prime graphs, stars and complete graphs, by a finite sequence of the operation *. The smallest decomposable graph with respect to split decomposition is the completely separable graph.

The only linear algorithm for the homogeneous decomposition was proposed by Baumann in 1996 [Bau96]. This algorithm is an extended of the modular decomposition algorithm developed independently by McConnell and Spinard [McS94] and Cournier and Habib [CH94]. These algorithms are unfortunately so complex and viewed as theoretical contributions. Recently, a simple linear modular decomposition algorithm has been developed by Tedder et al. in [TCHP08]. A promising future direction is the development of a simple linear homogeneous decomposition using the strategy in [TCHP08]. Also all linear algorithms proposed for P_4-reducible graphs and P_4-sparse graphs in [JO89] and [JO92] respectively are complex and a considerable effort is necessary to understand them. It is interesting to design a simple recognition algorithms for these classes by studying the algorithmic framework used in [HP05] for a simple cograph recognition. Finally designing a simple linear time split decomposition still an open problem.

References

[Bau96] S. Baumann. A linear algorithm for the homogeneous decomposition of graph. *Tech. Report M-9615, Institut fūr Mathematik, TU Mūnchen, 1996*.

[BJM78] T. Beyer, W. Jones, and S. Mitchell, A linear algorithm for isomorphism of maximal outerplanar graphs. *CS-TR-78-1, Dep. Of computer science, Univ. of Oregon, Eugene, Oregon, 1978.*

[BL79] K. S Booth and G. S. Lueker, A linear time algorithm for deciding interval graph isomorphism. *Journal of the ACM 26, 1979, 183-195.*

[BO95] L. Babel, S. Olariu, On the isomorphism of graphs with few P_4's. in M. Nagl. editor, *Graph Theoretic Concepts in Computer Science, 21th International Workshop, WG'95, Lecture Notes in Computer Science 1995, 24-36.*

[BM83] H. Buer and R.H. Mōring. A fast algorithm for the decomposition of graphs and posets. *Math. Oper .Res, 1983, 8 :170–184.*

[CD99] S. Cicerone, D. Di Stefano, On the extension of bipartite graphs to parity graphs. *Discrete Appl. Math. 1999, 95: 1–3, 181–195*.

[CH94] A. Cournier and M. Habib. A new linear algorithm for modular decomposition. *Lecture Notes in Computer Science, CAAP' 1994 :68–8.*

[CLB81] D.G. Corneil, H. Lerchs, and L. Stewart Burlingham. Complement reducible graphs. *Discrete Appl. Math. 1981 3 :163–174.*

[CJS72] D.D. Cowan, I.O. James, and R.G. Stanton. Graph decomposition for undirected graphs. *3rd S-E Conf. on Combinatorics, Graph Theory and Computing, 1972, 281–290.*

[CP84] D.G. Corneil and Y. Perl. Clustering and domination in perfect graphs. *Discrete Appl. Math. 1984, 9 : 27-39.*

[CPS84] D.G. Corneil, Y. Perl, and L.K. Stewart. Cographs : recognition, Application and Algorithms. *Proc. 15th Southeastern Conf. on Combinatorics, Graph Theory and Computing, LSU, 1984.*

[CPS85] D.G. Corneil, Y. Perl, and L.K. Stewart. A linear recognition algorithm for cographs. *SIAM Journal of computing, 1985, 14 :926–934.*

[Cou93] A. Cournier. Sur quelques Algorithmes de D´ecomposition des graphes. *PhD thesis, Univeristé des Sciences et Techniques du Languedoc, 1993.*

[Cun82] W.H. Cunningham. Decomposition of directed graphs. *SIAM Journal Alg. Discrete Math. 1982, 3 :214–228.*

[Dah94] E. Dahlhaus. Efficient parallel and linear time sequential split decomposition *(extended abstract). FSTTCS, 1994, 171–180.*

[Dah95a] E. Dahlhaus, Efficient parallel algorithms of cographs and distance hereditary graphs, *Discrete Appl. Math. 1995, 57: 29–54.*

[Dah95b] E. Dahlhaus. Efficient parallel modular decomposition. *Lecture Notes in Computer Science, 1995, 290–302, 1995.*

[Dah00] E. Dahlhaus, Parallel algorithms for hierarchical clustering and applications to split decomposition and parity graph recognition, *Journal of Algorithms 2000, 36- 2: 205–240.*

[DGM97] E. Dahlhaus, J. Gustedt, and R. McConnell. Efficient and practical modular decomposition. *SODA 1997, 26–35.*

[EGMS97] A. Ehrenfeucht, H.N. Gabow, R.M. McConnell, and S.J. Sullivan. An $O(n^2)$ divide and conquer algorithm for the prime tree decomposition of two structures and modular decomposition of graphs. *Journal of Algorithms, 1997, 16 :283–294.*

[ER90] A. Ehrenfeucht and G. Rozenberg. Theory of 2-structures, part i : Clans, basic, subclasses and morphemes. *Theoretical Computer Science, 1990, 70: 305–342.*

[FM05] Celina M. H. de Figueiredo and Frèdéric Maffray. Optimizing bull-free perfect graphs. *SIAM Journal of Discrete. Math., 2005, 18-2: 226–240, 2005.*

[Gal67] T. Gallai. Transitiv orienterbare graphen. *Acta Math.Acad.Sci. Hungar, 1967,18 :25–66.*

[GHS89] C.P. Gabor, W.L. Hsu, K.J. Supowit, Recognizing circle graphs in polynomial time, *J. ACM 1989, 36-3: 435–473.*

[Gia96] V. Giakoumakis. P_4-laden graphs : A new class of brittle graphs. *Information Processing Letters, pages1996, 60 : 29–36.*

[GJ79] M.R. Garey and D.S Johnson. Computers and Intractability : A gide to the Theory of NP-Completeness. *Freeman, San Francisco, 1979*.

[GKB04] J. Gagneur, R. Krause, T. Bouwmeester, and G. Casari. Modular decomposition of protein-protein interaction networks. *Genome Biology, 2004, 5-8: R57.*

[Gol80] M.C. Golumbic. Algorithmic graph theory and perfect graphs. *Academic press, New-York, 1980.*

[GRT97a] V. Giakoumakis, F. Rossel, and H. Thuillier. On P_4-tidy graphs. *Discrete Math. and Theoretical Computer Science, 1997, 1 :17–41.*

[GRT97b] V. Giakoumakis, F. Rossel, and H. Thuillier. Scatterring number and modular decomposition. *Discrete Math. 1997, 165-166 :321–342.*

[GV97] V. Giakoumakis and J.M. Vanherpe. On extended P_4-reducible and extended P_4-sparse graphs. *Theoretical Computer Science, 1997, 18 :389–402.*

[Hab81] M. Habib. Substitution des structures combinatoires, Théorie et Algorithmes. *University Pierre et Marie Curie (Paris VI), 1981. Thèse d'habilitation.*

[Hir51] T. Hiragushi. On the dimension of partially ordered sets. *Sci.Rep.Kanazawa University, 1951, 77–94.*

[HM79] M. Habib and M.C. Maurer. On the X-joint decomposition for undirected graphs. *Discrete Applied Mathematics, 1979, 3: 198–207.*

[HM90] P.L. Hammer and F. Maffray. Completely separable graphs. *Discrete Applied Mathematics, 1990, 27 :85–99.*

[HP05] M. Habib and C. Paul. A simple linear time algorithm for cograph recognition, *Discrete Applied Mathematics 2005, 145: 183 – 197*

[HS95] W.Hochstätter and H. schindler, Recognizing P_4-extendible graphs in linear time, *University of Cologne, ZPR, Report No. 95.188*

[HT95] W. Hochstätter and G. Tinhofer, Hamiltonicity in graphs with few P_4's. *Computing 1995, 54: 213-225.*

[Ill89] P. Ill. L'intervalle en théorie des relations. *1989. Thèse de Doctorat*.

[JO89] B. Jamison and S. Olariu, P_4-reducible graphs, a class of uniquely tree representable graphs, *Studies in Applied Mathematics, 1989, 81: 79-87*.

[JO91] B. Jamison and S. Olariu, On a unique tree representation for P_4-extendible graphs, *Discrete Applied Mathematics, 1991, 34: 151-164.*

[JO92] B. Jamison and S. Olariu, A unique tree representation for P_4-sparse graphs, *Discrete Applied Mathematics, 1992, 35: 115-129*.

[JO95] B. Jamison and S. Olariu. p-components and the homogeneous decomposition of graphs. SIAM Journal on Discrete Mathematics, 1995, 8 :448–463.

[JO95b] B. Jamison and S. Olariu. Linear time optimization algorithms for P_4-sparse graphs. *Discrete Applied Mathematics, 1995, 61: 155-175.*

[Jol73] J.L. Jolivet. Sur le joint d'une famille des graphes. *Discrete Mathematics, 1973, 5: 145–158.*

[Jun78] H.A. Jung. On a class of posets and the corresponding comparability graphs. Journal of Computation Theory (B), 1978, 14 :125–133.

[Ler78] H. Lerchs. On the clique-kernel structure of graphs. *Technical report, Dept. Computer Science, University of Toronto, 1978.*

[Mo85] R.H. Möhring. Algorithmic aspects of comparability graphs and interval graphs. *In I. Rival, editor, Graphs and Orders, pages 41–101. D. Reidel, Boston, 1985.*

[MR84] R.H. Möhring and F.J. Radermacher. Substitution decomposition and connection with combinatorial optimization. *Ann. Discrete Mathematics, 1984, 19: 257–356.*

[MS89] J.H. Mūller and J. Spinard. Incremental modular decomposition. *Journal of Algorithms, 1989, 16: 149–160.*

[MS94] T.H. Ma and J. Spinard. An O(n^2) algorithm for undirected split decomposition. *Journal of Algorithms, 1994, 16 :145–160.*

[McS94] R.M. McConnelle and J. Spinard. Linear time modular decomposition and efficient transitive orientation of comparability graphs. *In Proc. 5th ACM-SIAM Symp, volume Proc.5th ACM-SIAM Symp, pages 536–545, 1994.*

[McS97] R.M. McConnell and J. Spinard. Linear time transitive orientation. *In 8th ACMSIAM Symp, Discrete Algorithms, pages 19–25, 1997.*

[McS00] R.M. McConnell and J. Spinrad. Ordered vertex partitioning. *Discrete Mathematics and Theoretical Computer Science, 2000, 4: 45–60.*

[PEL71] A. Pnueli, S. Even, and A. Lempel. Transitive orientation of graphs and identification of permutation graphs. *Canad. J. Math., 1971, 23:160–175.*

[PV06] C. Papadopoulos and C. Voglis. Drawing graphs using modular decomposition. *In Patrick Healy and Nikola S. Nikolov, editors, Graph Drawing, Limerick, Ireland, September 12-14, 2005, pages pp. 343–354. Springer, 2006.*

[Rao08] M. Rao. Solving some NP-complete problems using split decomposition. *Discrete Applied Mathematics. 2008, 156: 2768–2780.*

[Sab61] G. Sabidussi. Graph derivates. *Math. 1961, Z,76 :385–401.*

[Spi94] J. Spinrad, Recognition of circle graphs, *Journal of Algorithms. 1994, 16-2 264–282.*

[Sum74] D.P. Sumner. Dacey graphs. *J.Australian Math.Soc. 1974, 18: 492–502*.

[TCHP08] Marc Tedder, Derek Corneil, Michel Habib, and Christophe Paul. Simpler Linear-Time Modular Decomposition via Recursive Factorizing Permutations. Lecture Notes In Computer Science, 2008, 5125 : 634 - 645