UNIVERSITE D'ORLEANS
Faculté des Sciences
LIFO

Laboratoire d'Informatique Fondamentale d'Orléans
4, rue Léonard de Vinci, BP 6759
F-45067 Orléans Cedex 2
FRANCE

Rapport de Recherche
www : http://www.univ-orleans.fr/SCIENCES/LIFO/

Forbidden Subgraph Decomposition

I. Rusu, J. Spinrad
Université d'Orléans, LIFO

Rapport N^o **99-13**

Forbidden Subgraph Decomposition

Irena Rusu[1]
L.I.F.O., Université d'Orléans, bât. IIIA,
B.P. 6759, 45067 Orléans Cedex 2, France

Jeremy Spinrad[2]
Box 1679-B, CS Dept., Vanderbilt University,
Nashville, TN 37235, USA

Abstract

We define a new form of graph decomposition, based on forbidding a fixed bipartite graph from occurring as an induced subgraph of edges which cross a partition of the vertices. We show that some natural decompositions obtained in this way (including the generalized join proposed by Hsu [6]) are NP-complete.

Key Words. graph decomposition, totally decomposable graphs.

1 Introduction

This paper introduces a new notion of decomposing a graph. For any bipartite graph F, we define a F-free decomposition, which corresponds to partitioning the vertices into two subsets such that F does not occur as an induced subgraph of the edges which go from one subset to the other.

We study the complexity of finding a decomposition for small F, and a number of other questions which arise while studying these decompositions. Among other results, we show that it is NP-complete to determine whether a graph has a generalized join decomposition, thus resolving an open problem posed by Hsu [6].

The remainder of this section is devoted to definitions which allow new notions regarding these decompositions to be specified precisely. Later sections will be devoted to F-free decompositions with respect to certain specific graphs F.

For an arbitrary graph G, a partition P of its vertices into sets V_1, V_2 is denoted (V_1, V_2). Throughout the paper, a partition will be *ordered*, that is, (V_1, V_2) and (V_2, V_1) will be distinct partitions.

Let G_P be the bipartite graph with bipartition V_1, V_2, and an edge from $v_1 \in V_1$ to $v_2 \in V_2$ if and only if v_1 is adjacent to v_2 in G. Let F be a bipartite graph with (ordered) bipartition (X, Y).

[1]e-mail:irusu@titan.univ-orleans.fr
[2]e-mail:spin@vuse.vanderbilt.edu

We say that F is *tidily induced* in G_P if there exists an isomorphism f from F to some induced subgraph H of G_P, such that $f(x) \in V_1$ for all $x \in X$ and $f(y) \in V_2$ for all $y \in Y$. When no confusion is possible, we will skip the word "tidily".

The partition P is called an *F-free decomposition* of G if it meets the following criteria:

1) $|V_1| \geq |X|, |V_2| \geq |Y|$;

2) F is not tidily induced in G_P.

Moreover, P is called a *symmetric-F-free decomposition* if both (V_1, V_2) and (V_2, V_1) are F-free decompositions.

Of particular interest (because of their strong properties) are the graphs for which the decomposability is a hereditary property. A graph G is *totally decomposable* with respect to the F-free decomposition if every induced subgraph of G with at least $|X| + |Y|$ vertices has an F-free decomposition.

It seems natural to focus our first efforts on studying F-decompositions for small fixed graphs F. As we will show later, certain decompositions defined in this way correspond naturally to classical graph decomposition techniques from the literature.

Forbidden subgraphs defined by two vertices are too limited to consider; forbidding a single edge allows only division into disconnected sets, while forbidding the two vertex graph with no edges allows only division into disconnected sets of $\overline{G}$. Totally decomposable graphs with respect to these two decompositions are simply edgeless graphs, and cliques.

We consider now the forbidden subgraphs with three vertices.

2 Three Vertex Forbidden Subgraphs

Assuming without loss of generality that F (with the bipartition (X, Y) as before) has $|X| = 2, |Y| = 1$, there are only two forbidden subgraphs to consider separately; K_2 plus an isolated vertex in X, and $K_{1,2}$; the edgeless graph on three vertices is simply the bipartite complement of $K_{1,2}$, which corresponds to decomposing the complement graph.

Forbidding K_2 plus an isolated vertex is equivalent to the well known notion of a module in the substitution decomposition [7]. We briefly present it in the next subsection. Then we show that $K_{1,2}$-free decompositions can be found in polynomial time, but it is NP-complete to determine whether G has a symmetric-$K_{1,2}$-free decomposition.

2.1 Edge Plus Isolated Vertex

It is interesting that the decomposition defined by forbidding $F = K_2 + K_1$ already gives us the notion of a module, which is the fundamental subgraph for the well known substitution decomposition.

A *module* in G is a set M of vertices which is "indistinguishable" from outside of M; i.e. for every vertex v of $V - M$, v is either adjacent to every vertex of M, or v is adjacent to no vertex in M. A module M is called *trivial* if $M = V$, or $|M| = 1$.

Any nontrivial module can be chosen to be V_1 to create an F-free decomposition of G, and for every F-free decomposition, V_1 must form a nontrivial module. Thus, it is possible to determine whether G has an F-free decomposition in linear time, though this is by no means trivial [10], [2], [3]. Using well known facts about the substitution decomposition [14], a graph is completely decomposable with respect to F-free decomposition if and only if it contains no P_4; in other words, if G is a cograph.

A graph is symmetrically decomposable with respect to F if and only if it can be partitioned into two modules. This is possible if and only if either G or $\overline{G}$ is disconnected, which can easily be tested in linear time; once again, completely decomposable graphs with respect to symmetric F-free decomposition are exactly the cographs.

2.2 $K_{1,2}$-free Decomposition and Star-Free Decomposition

We consider first the $K_{1,2}$-free decomposition, and then we notice that a similar approach is valid for $K_{1,i}$-free decompositions, for any value of i.

Note that the meaning of the $K_{1,2}$-free decomposition is that vertices are divided into sets V_1 and V_2 so that every vertex in V_1 has at most one neighbor in V_2.

Our first remark is that this decomposition can be found by a forcing algorithm in $O(n^2m)$ time. Indeed, for every pair of vertices x, y, we can find a decomposition with x and y in V_2 in $O(n + m)$ time, if such a decomposition exists. We mark all vertices not currently in V_2 with their number of neighbors in V_2; whenever a vertex gets a count of 2, we must bring it into V_2. If we stop before bringing every vertex into V_2, we have found a valid decomposition. This procedure looks at every edge once, and is thus $O(m)$ for a particular choice of x and y, giving an $O(n^2m)$ algorithm for determining whether G is decomposable.

It is also easy to see that G is totally decomposable if and only if G is triangle-free. Since a triangle cannot be decomposed, every totally decomposable graph must be triangle-free. If G is triangle-free, we can make any adjacent vertices x, y our set V_2; since G is triangle-free, no vertex is adjacent to both x and y, so this decomposition is valid.

Clearly, the same type of algorithm will work to find $K_{1,i}$-free decompositions for any value of i in $O(n^im)$ time; we start with each set of size i in V_2, and see whether all vertices are forced in V_2 in $O(m)$ time.

It is somewhat less obvious that the totally decomposable graphs for $K_{1,i}$-free decomposition are exactly those graphs without cliques of size $i+1$. One direction is clear, since a clique of size $i+1$ cannot be decomposed. To see that every graph without a clique of size $i+1$ can be decomposed, we consider two possibilities.

Let G_v be the graph obtained from G by removing the vertex v and all its incident edges. If G_v has a $K_{1,i-1}$-free decomposition, then v can be added to V_2 to obtain a $K_{1,i}$-free decomposition. Otherwise, we may assume that G_v has a clique of size i (using induction on the fact that totally $K_{1,i}$-free decomposable graphs are exactly those without cliques of size $i+1$). We place this i-clique in V_2, and all other vertices in V_1; since there is no clique of size $i+1$, this must give us a $K_{1,i}$-free decomposition.

2.3 Symmetric $K_{1,2}$-free Decomposition

We will show that this very natural form of decomposition, requiring that the edges connecting the two sets are a matching, cannot be found in polynomial time unless P=NP.

Theorem 1 *It is NP-complete to determine whether a graph G has a symmetric $K_{1,2}$-free decomposition.*

Proof. It is easy to verify whether a partition is a symmetric $K_{1,2}$-free decomposition, so this problem is clearly in NP.

We reduce the problem one-in-three 3SAT with no negated literal [13] (see also [4]) to symmetric $K_{1,2}$-free decomposition. Consider a formula $\mathcal{C}$ with clauses $C_1, C_2, \ldots,$ C_k, each of them containing exactly three non-negated literals. We build a graph $G_{\mathcal{C}}$ such that $G_{\mathcal{C}}$ has a $K_{1,2}$-free decomposition if, and only if, $\mathcal{C}$ is satisfiable with exactly one true literal in each clause.

For each clause $C_i = (x_i^1 \vee x_i^2 \vee x_i^3)$ we create a graph G_i with 16 vertices as follows: for each literal x_i^j ($j = 1, 2, 3$), create vertices x_i^{j0}, x_i^{j1}, x_i^{j2}, x_i^{j3}, x_i^{j4} and join every pair of them by an edge. Then add a new vertex c_i which is adjacent to x_i^{10}, x_i^{20} and x_i^{30}.

Now, we put edges between the vertices in different graphs G_i, G_h (for every pair of i, h with $i \neq h$) as follows. If G_i and G_h have a common literal, then we add all possible edges between the 5-clique in G_i and the 5-clique in G_j corresponding to the common literal. We add an edge between c_i and c_h, no matter whether or not these clauses share a common literal.

We create the following connections between the clause C_i and the clause C_{i+1} (by convention, $C_{k+1} = C_1$). We create three new vertices z_i^1, z_i^2, z_i^3 and fix the following adjacencies (see Figure 1):

z_i^1 is adjacent to $x_i^{11}, x_{i+1}^{22}, x_{i+1}^{32}$
z_i^2 is adjacent to $x_i^{21}, x_{i+1}^{23}, x_{i+1}^{33}$
z_i^3 is adjacent to $x_i^{31}, x_{i+1}^{24}, x_{i+1}^{34}$
x_i^{11} is adjacent to x_{i+1}^{12}
x_i^{21} is adjacent to x_{i+1}^{13}
x_i^{31} is adjacent to x_{i+1}^{14}.

The graph in Figure 1 is called the *connection graph of* G_i, G_{i+1} and is denoted $Z_{i,i+1}$.

Finally, we add two supplementary vertices c_0, c_{-1} which are adjacent to every vertex c_i ($i = 1, \ldots, k$).

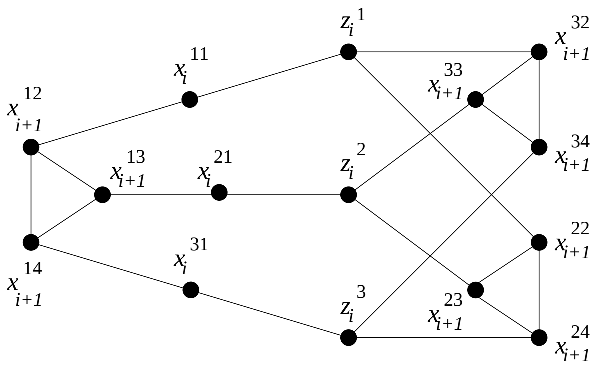

Figure 1: The connection graph $Z_{i,i+1}$

We now show that $\mathcal{C}$ is satisfiable if and only if G has a symmetric $K_{1,2}$-free decomposition.

$\Rightarrow$: Suppose that $\mathcal{C}$ is satisfiable with one true literal $t_i \in \{x_i^1, x_i^2, x_i^3\}$ per clause C_i. We partition the graph such that V_1 contains, for every i, the vertices obtained by the following rules:

Rule 1. the vertices t_i^0, t_i^1, t_i^2, t_i^3, t_i^{j4} created for t_i;

Rule 2. the vertex z_i^p such that z_i^p is adjacent in $Z_{i,i+1}$ both to a copy of t_i and a copy of t_{i+1} (if such a vertex exists; notice that if in $Z_{i,i+1}$ we have $t_{i+1} = x_{i+1}^1$, then no such vertex exists).

By definition, $V_2 = V - V_1$. We show that (V_1, V_2) is a valid partition of the set.

We start by considering an arbitrary vertex $a \in V_1$ and showing that it has at most one neighbor in V_2. In the case where a is one of the vertices z_i^p put in V_1 by Rule 2, two of the three neighbors of a are already in V_1 and we are done. So we can consider that a is one of the vertices put in V_1 by Rule 1; then a is a vertex created for a true literal t_i. Each such vertex except t_i^1 is adjacent to at most one vertex which is not a copy of t_i, and thus has at most one neighbor not in V_1. The vertex t_i^1 may have two neighbors which are not copies of t_i (its two neighbors in $Z_{i,i+1}$), but one of these is also placed in V_1 (either $t_{i+1} = x_{i+1}^1$, so the neighbor x_{i+1}^{1j} of t_i^1 is in V_1, or $t_{i+1} = x_{i+1}^j$ with $j \in \{2, 3\}$ and then by Rule 2 the neighbor z_i^p of t_i^1 is in V_1).

Now, we consider every vertex b in V_2 and show that it has at most one neighbor in V_1. If b is some clause vertex c_i $(i = 1, \ldots, k)$, then it has exactly one neighbor not in V_2, corresponding to the true literal in the clause; furthermore, c_0, c_{-1} have only neighbors in V_2. If b is some vertex created for a false literal f_i, then, except when $b = f_i^1$, the vertex b has at most one neighbor which is not a copy of f_i; moreover, one of b_i^1's two other neighbors (the neighbor connecting vertices from different clauses) is also placed in V_2. Finally, if b is some vertex z_i^p then it will have at most one neighbor in V_1, else it would have been put in V_1 by Rule 2.

Therefore, if $\mathcal{C}$ is satisfiable with one true literal per clause, G is $K_{1,2}$-free decomposable.

$\Leftarrow$: Suppose that G has a symmetric $K_{1,2}$-free decomposition. Let V_2 be the set containing the clause vertices (all these vertices are on the same side of the partition since with c_0, c_{-1} they form a clique of size at least 3 which is indecomposable). For the same reason, the vertices corresponding to the same literal are either all in V_1 or all in V_2. We claim that the set of literals whose corresponding vertices are in V_1 contains exactly one true literal from each clause. For short, we will say that such a literal *is* in V_1.

First, we note that no clause C_i can have two true literals; otherwise, two of c_i's neighbors would be in V_1.

Next, we note that there is at least one true literal. This is equivalent to saying that V_1 contains at least one vertex created for some literal. If this is not true, then V_1 (which is non-empty) possesses at least one vertex z_i^r. Since the decomposition is $K_{1,2}$-free, one of its neighbors in $Z_{i,i+1}$ must also be in V_1.

By the two preceding remarks, there exists a clause C_i which has exactly one true literal (i.e. a literal in V_1). We will show that V_1 contains exactly one true literal from clause C_{i+1} (where, again, $C_{k+1} = C_1$). This will complete our proof since, by induction, V_1 will have exactly one true literal in each clause. Denote by x_i^j ($j \in \{1, 2, 3\}$) the true literal in C_i. We know that x_i^{j1} is adjacent in $Z_{i,i+1}$ to both a copy of x_{i+1}^1 and an intermediate vertex z_i^j. If the copy of x_{i+1}^1 is in V_1, then the clause C_{i+1} has a true literal, and we are done. Otherwise, since x_i^{j1} has at most one neighbor not in V_1, the intermediate vertex z_i^j is also in V_1. This intermediate vertex is adjacent to copies of two different literals from clause C_{i+1}, one of which must be in V_1 for this to be a valid symmetric $K_{1,2}$-free decomposition. Thus, one of these literals is true, and we have a satisfying assignment for $\mathcal{C}$ with exactly one true literal per clause. ∎

Remark. Notice that in the proof above the graph G_C always contains a clique of size 5. That means that finding a symmetric $K_{1,2}$-free decomposition is NP-complete even when reduced to graphs G with $\omega(G) \geq 5$ (where $\omega(G)$ is the size of the largest clique in G). We will use this observation in a subsequent proof.

It is easy to see that totally decomposable graphs with respect to symmetric $K_{1,2}$-free decomposition can contain no K_3 and no $K_{1,3}$. Conversely, every $(K_3, K_{1,3})$-free graph with four vertices or more either is a collection of stars (and then a decomposition is easy to find), or has at least two adjacent vertices of degree 2 which can be put on one side of the partition, while all the other vertices are on the other side.

3 Four vertex forbidden subgraphs

When F has four vertices, there are (up to symmetries and complementation) six cases to consider. One of them, $K_{1,3}$, was treaten in Section 2. The decompositions involving two edges sharing a common vertex seem less natural and will not be discussed in this paper. A more interesting case is the one of the P_4, which we discuss below and we propose as an open problem.

To see this, let us recall the split-decomposition [1]. This is a partition of the graph G into two sets V_1, V_2 such that there exist sets $W_1 \subseteq V_1, W_2 \subseteq V_2$ with the property that every vertex in W_1 is adjacent to every vertex in W_2 and no other edge exists between a vertex in V_1 and a vertex in V_2.

It is easy to see that a partition is a split-decomposition if and only if it is both a P_4-free decomposition and a $2K_2$-free decomposition. Finding a split decomposition can be done in linear time [8], and the totally decomposable graphs are in this case the distance-hereditary graphs [5].

When we drop the condition on the $2K_2$-freeness of the decomposition, we obtain the P_4-free decomposition, which may be expressed in term of sets by asking that sets $W_1^i \subseteq V_1, W_2^i \subseteq V_2$ $(i = 1, 2, \ldots, k)$ exist such that every vertex in W_1^i is adjacent to every vertex in W_2^i and no other edge exists between a vertex in V_1 and a vertex in V_2. This is easy to obtain by noticing that any two vertices in V_1 (respectively in V_2) must have in V_2 (respectively in V_1) either the same neighborhood, or disjoint neighborhoods.

3.1 Forbidding $K_{2,2}$

This problem is NP-complete too, and very easy to prove so.

Theorem 2 *It is NP-complete to determine whether a graph G has a $K_{2,2}$-free decomposition.*

Proof. The problem is clearly in NP. To prove NP-completeness, we reduce the symmetric $K_{1,2}$-free decomposition to the $K_{2,2}$-free decomposition. To do this, just add to an instance G of the first problem two non-adjacent vertices x, y which are adjacent to every vertex of G. Call G' this new graph.

Now, if G has a symmetric $K_{1,2}$-free decomposition (V_1, V_2), then the decomposition $(V_1 \cup \{x\}, V_2 \cup \{y\})$ is a $K_{2,2}$-free decomposition of G'; otherwise, the $K_{2,2}$ induced by $a, b \in V_1 \cup \{x\}$ and $c, d \in V_2 \cup \{y\}$ would contain at most one vertex among x, y (since $xy \notin E$) and the three other vertices would induce a $K_{1,2}$ or $K_{2,1}$ in G.

Conversely, if G' has a $K_{2,2}$ decomposition (W_1, W_2), then x, y are on different sides of the partition (else they would form a $K_{2,2}$ with two arbitrary vertices on the other side of the partition). Say $x \in W_1, y \in W_2$. Then $(W_1 - \{x\}, W_2 - \{y\})$ is a symmetric $K_{1,2}$-free partition of G. ∎

3.2 Forbidding $2K_2$

This decomposition is of particular interest, having been studied under several different names. Hsu [6] called this the generalized join decomposition (this generalizes Cunningham's split decomposition, also known as join decomposition, which corresponds to symmetrically forbidding both $2K_2$ and P_4), and asked whether this could be found in polynomial time. The decomposition was also studied by Maire [9]; both authors were interested in this decomposition because it preserves perfection. We will show that finding a $2K_2$-free decomposition is NP-complete.

Theorem 3 *It is NP-complete to decide whether a graph has a $2K_2$-free decomposition.*

Proof. The problem of determining whether a graph has a $2K_2$-free decomposition is clearly in NP. We reduce the problem of finding a symmetric $K_{1,2}$-free decomposition for graphs G with $\omega(G) \geq 5$ to the problem of finding a $2K_2$-free decomposition.

Let G be the input graph to the symmetric $K_{1,2}$-free decomposition problem (case $\omega(G) \geq 5$). Then G is not a star. We create a graph G' as input to the $2K_2$-free decomposition problem as follows.

We first show that we can assume G has no vertex of degree 1. Indeed, let L be the set of vertices of G which have at least one neighbour of degree 1. Two cases can occur if $L \neq \emptyset$:

• If $|L| \geq 2$, or $L = \{z\}$ and z has exactly one neighbor of degree two or more, then we put $G' = G$. Now, G has always a symmetric $K_{1,2}$-free decomposition, while G' has always a $2K_2$-free decomposition. The proof is finished in this case.

• If $L = \{z\}$ and z has at least two neighbors of degree two or more, then G has symmetric $K_{1,2}$-free decomposition if and only if the graph obtained from G by removing all its vertices of degree one has such a decomposition. Then we remove these vertices from G, and recall G the new graph (which has no more vertices of degree one). For the new graph G, the set L is now empty.

Thus we can assume G has no vertex of degree 1. Notice that the case where $L = \{z\}$ and z has no neighbour of degree two or more cannot occur, since G is not a star. The construction of G' is described below. Let $d(v)$ denote the degree of vertex v in G.

For each vertex v of G, create $2 * d(v)$ vertices $v_1, v_2, \ldots, v_{2d(v)}$. Connect these vertices to form a chordless cycle $v_1, ..., v_{2d(v)}$. Then, choose any order $u_1, u_2, ..., u_{d(v)}$ for the neighbors of v.

For each edge (w, x) of G, let i be the position of x among the neighbors of w, and let j be the position of w in the order of neighbors of x. Add edges between w_i and x_j, and between $w_{i+d(w)}$ and $x_{j+d(x)}$. We will say that these two edges of G' are *copies* of the edge (w, x) of G. Furthermore, add *crossedges* from w_i to $x_{j+d(x)}$, and from $w_{i+d(w)}$ to x_j in G'.

Finally, let (u, v), (w, x) be a pair of edges in G. If the two edges do not share any endpoints in G, take the four endpoints of the two edges corresponding to (u, v) in G', and connect each to all four endpoints of the two edges which represent (w, x) in G'. These edges will be called *filling edges*.

We now show that G has a symmetric $K_{1,2}$-free decomposition if and only if G' has a $2K_2$-free decomposition.

$\Rightarrow$: First, suppose that G has a symmetric $K_{1,2}$-free decomposition (V_1, V_2). The decomposition of G' into (V_1', V_2') is defined as follows:

$V_1' = \{u \mid u \text{ is a copy of some vertex } x \in V_1\}$
$V_2' = \{v \mid v \text{ is a copy of some vertex } y \in V_2\}.$

We will show that this decomposition is $2K_2$-free.

By contradiction, consider a $2K_2$ given by the edges (u_i, v_j), (w_k, x_l) which goes between V_1' and V_2' in G' (that is, $u_i, w_k \in V_1$, $v_j, x_l \in V_2$). We consider three possibilities, depending on the number of edges in the $2K_2$ which are copies of edges in G (two, one or zero edges).

Case 1. Both (u_i, v_j) and (w_k, x_l) are copies of edges of G. Suppose first that u_i and w_k are copies of the same vertex. Since every vertex in V_1 has at most one neighbor in V_2, v_j and x_l are also copies of the same vertex, i.e. these are two copies of the same edge. This cannot occur, since our construction adds crossedges between copies of the same edge. Thus, all four endpoints correspond to distinct vertices. But in this case, the construction adds all filling edges between the two edges. We have a contradiction.

Case 2. The edge (u_i, v_j) is a copy of an edge of G, but (w_k, x_l) is not. So (w_k, x_l) is a filling edge or a crossedge.

If (w_k, x_l) is a filling edge, there exist edges (b, x), (w, c) of G with all four endpoints distinct. Now, at least one edge (between u_i and x_l) is missing in G' between the endpoints corresponding to (b, x) and (u, v). So, these edges must share a common point (else all filling edges would have been added): either $b = u$ or $x = v$. Since the decomposition of G leaves only a matching between V_1 and V_2, in either case we must have both $b = u$ and $x = v$. Similarly, we would require $w = u$ and $c = v$; since there are only two copies of any edge, this case cannot occur.

If (w_k, x_l) is a crossedge, then it was added between two copies of the same edge (w, x). Since edges between V_1 and V_2 form a matching, neither u nor v can be equal to w or x; thus, we would add all edges between the corresponding endpoints, violating the assumption that these form a $2K_2$ in G'.

Case 3. Neither (u_i, v_j) nor (w_k, x_l) are copies of edges of G.

First, suppose that both edges are filling edges. Say that (u_i, v_j) was added because of edges (u, d) and (e, v) in G, and that (w_k, x_l) was added because of edges (w, f) and (g, x) in G. Since we did not add all edges between (u, d) and (g, x), some endpoint must be shared; we may assume that $u = g$. Since (u, d) and (u, x) both go from V_1 to V_2 in G, we must have $d = x$. Since (u, d) and (g, x) are copies of the same edge, by construction we should have added a crossedge from u_i to x_l.

Our only remaining case is that at least one of the edges of the $2K_2$ (let us say (u_i, v_j)) is a crossedge, i.e. it was added between copies of a single edge (u, v). Since neither u nor v can have another neighbor on the other side of the partition, any edges of G which caused (w_k, x_l) to be added cannot have u or v as an endpoint; thus, crossedges should have been added, contradicting the assumption that these edges form a $2K_2$.

$\Leftarrow$: Suppose that G' has a partition (V_1', V_2') such that no $2K_2$ exists between V_1' and V_2'. We will show that G has a symmetric $K_{1,2}$-free decomposition.

We divide this proof into two cases. In the first case, we assume that for every vertex v of G, all copies of v are on the same side of the partition of G'. In the second case, we assume that there is a vertex of G such that some copy of v occurs on each side of the partition of G'.

Case 1. Suppose that all copies of every vertex lie on a single side of the partition of G'. Then let the partition of G be

$$V_1 = \{x \mid \text{all copies of}\, x\, \text{are in}\, V_1'\},$$
$$V_2 = \{y \mid \text{all copies of}\, y\, \text{are in}\, V_2'\}.$$

We will show that (V_1, V_2) is a symmetric $K_{1,2}$-free decomposition of G.

In the contrary case, we may assume that there is a vertex x in V_1 which has two neighbors y, z in V_2. In G', the copies (x_i, y_k) of (x, y) and (x_j, z_l) of (x, z) go from V_1' to V_2', and since $(x, y), (x, z)$ share an endpoint of G, the construction will not add any edges between the corresponding endpoints in G'; thus we contradict the assumption that V_1', V_2' is a $2K_2$-free decomposition.

Case 2. Assume that there is some vertex v of G which has copies in both V_1' and V_2' in the decomposition of G'. Let our partition have the minimum number of vertices which have copies in both V_1' and V_2'.

The copies of v form an induced cycle in G'. We claim that the only way to split an induced cycle $c_1, ..., c_k$ into two pieces of a $2K_2$-free decomposition is to put one vertex on one side, and all other vertices on the other side. Otherwise, call a *segment* of the cycle every maximal set of successive vertices along the cycle, such that all the vertices are on the same side. Now, the cycle cannot have two segments $c_i, \ldots, c_j$, $c_k, \ldots, c_l$ in V_1 since the edges (c_j, c_{j+1}) and (c_l, c_{l+1}) violate the $2K_2$-free condition. If it has only one segment $c_i, \ldots, c_j$ in V_1, then $j = i$ or else (c_{i-1}, c_i) and (c_j, c_{j+1}) give a $2K_2$.

Thus, we may assume that v_i is in V_1', and all other copies of v are in V_2'. Placing v_i in V_2', and changing the position of no other vertex, gives a partition of G' which cannot be a $2K_2$-free decomposition, otherwise the decomposition (V_1', V_2') would not have the minimum number of vertices with copies on both sides of the partition. We have two possibilities:

- either $V_1' - \{v_i\}$ contains only one vertex w_j; then, since the decomposition (V_1', V_2') is $2K_2$-free, we must have either $N_{V_2'}(v_i) \subseteq N_{V_2'}(w_j)$ or $N_{V_2'}(w_j) \subseteq N_{V_2'}(v_i)$. Assume that the first holds and call t_p the neighbour of v_i corresponding to an edge vt of G. Now, $w_j \neq t_p$ since (t_p, v_{i-1}) is not an edge. So (w_j, t_p) is an edge of G'. This cannot be a copy of an edge of G (t_p is in only one such copy), neither a crossing edge (w_j would be a copy of v_i), so (w_j, t_p) is a filling edge corresponding to edges (v, t), (w, y) of G. But then by construction v_i is adjacent to $w_{j+d(w)}$ (the endpoint of the other copy of (w, y) in G', while w_j is non-adjacent to $w_{j+d(w)}$. Then $N_{V_2'}(v_i) \subseteq N_{V_2'}(w_j)$ is contradicted.

- or the partition $(V_1' - \{v_i\}, V_2' \cup \{v_i\})$ contains a $2K_2$. Thus, there must be some w_j in V_1' such that (v_i, w_j) is an edge of G'. By our construction, $(v_{i+d(v)}, w_j)$ is also an edge of G'. The edges (v_{i-1}, v_i) and $(v_{i+d(v)}, w_j)$ form a $2K_2$ in G', violating the assumption that we have a $2K_2$-free decomposition of G'. ∎

Considering graphs which are totally decomposable with respect to $2K_2$-free decomposition yields a subclass of weakly triangulated graphs (i.e. graphs with no chordless cycle of length at least five and with no complement of such cycle) and a superclass of domination graphs (i.e. graphs in which every induced subgraph has two vertices with comparable neighborhoods). We were not able to characterize this class, but we discuss some related aspects in a paper devoted to domination graphs [12].

4 Conclusions and Open Problems

We have defined a new form of graph decomposition, and studied some important special cases. We have shown that it is NP-complete to determine whether we can decompose a graph so that there is only a "matching" going between the two sets, and also to determine whether a graph has a generalized join.

A supplementary interest of these two decompositions comes from their perfection-preserving properties. The generalized join preserves perfection (see [6], [9]) , i.e. V_1 and V_2 induce perfect graphs, then G is a perfect graph too. For the "matching" decomposition, the result is a little bit weaker, in the sense that we can only claim that no minimal imperfect graph different from a C_{2k+1} ($k \geq 2$) has a symmetric $K_{1,2}$-free decomposition. For this, it is sufficient to notice that the intersection graph of the maximum cliques of G is disconnected (no maximum clique can have vertices in both V_1 and V_2), and this is not possible by Preissmann's result [11].

On the other hand, it would be interesting to characterize completely when a decomposition with a forbidden subgraph is polynomial or NP-complete. A particular open case is the P_4-decomposition.

Furthermore, it seems likely that the time complexity for $K_{1,i}$-free decomposition, i.e. allowing each vertex in V_1 to have at most i-1 neighbors in V_2, can be improved considerably.

Applications of these decompositions would be interesting to find, especially for those which can be found in polynomial time.

References

[1] W. H. Cunningham – Decomposition of directed graphs, *SIAM J. Alg. Discrete Methods* 3 (1982), 214–228.

[2] A. Cournier, M. Habib - A new linear algorithm for modular decomposition, *in* Lecture Notes in Computer Science 787 (1994), 68–84.

[3] E. Dahlhaus, J. Gustedt, R. M. McConnell - Efficient and practical modular decomposition, *in* 8th ACM-SIAM Symposium on Discrete Algorithms (1997), 26–35.

[4] M. R. Garey, D. S. Johnson - Computers and intractability, A Guide to the Theory of NP-completness, Freeman, San-Francisco (1979).

[5] P. Hammer, F. Maffray - Completely separable graphs, *Discrete Appl. Math.* 27 (1990), 85–99.

[6] W.-L. Hsu - Decomposition of perfect graphs, *Journal of Combinatorial Theory* Ser. B 43 (1987), 70–94.

[7] L. Lovász - Normal hypergraphs and the perfect graph conjecture, *Discrete Mathematics* 2 (1972) 253–267.

[8] E. Dahlhaus – Efficient parallel and linear time sequential split decomposition, *dans* FSTTCS: Foundations of Software Technology and Theoretical Computer Science, Lecture Notes in Computer Science 880 (1994), 171–180.

[9] F. Maire - Des polyominos aux graphes parfaits ; contribution à l'étude des propriétés combinatoires de ces structures, Thèse de doctorat, Université Paris 6 (1993).

[10] R. M. McConnell, J. Spinrad - Linear-time modular decomposition and efficient transitive orientation of comparability graphs, 5th ACM-SIAM Symposium on Discrete Algorithms (1994), 536–545.

[11] M. Preissmann - Locally perfect graphs, *J. Combin. Theory* Ser. B 50 (1990), 22–40.

[12] I. Rusu, J. Spinrad - A note on domination graphs, preprint.

[13] T. J. Schaefer - The complexity of satisfiability problems, *Proceedings 10th. Annual ACM Symposium on Theory of Computing*, Association for Computing Machinery, New York (1978), 216–226.

[14] D. Seinsche - On a property of the class of n-colorable graphs, *J. Combin. Theory* ser. B 16 (1974), 191–193.