(α, β)-Modules in Graphs*

Michel Habib[1], Lalla Mouatadid[2], Eric Sopena[3], and Mengchuan Zou[4]

[1] IRIF, UMR 8243 CNRS & Université Paris Cité, Paris, France
[2] Department of Computer Science, University of Toronto, Toronto, ON, Canada
[3] Univ. Bordeaux, CNRS, Bordeaux INP, LaBRI, UMR5800, F-33400 Talence, France
[4] State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing, China

Abstract. Modular Decomposition focuses on repeatedly identifying a module M (a collection of vertices that shares **exactly** the same neighbourhood outside of M) and collapsing it into a single vertex. This notion of *exactitude of neighbourhood* is very strict, especially when dealing with real world graphs.

We study new ways to relax this exactitude condition. However, generalizing modular decomposition is far from obvious. Most of the previous proposals lose algebraic properties of modules and thus most of the nice algorithmic consequences.

We introduce the notion of an **(α, β)-module**, a relaxation that maintains some of the algebraic structure. It leads to a new combinatorial decomposition with interesting properties. Among the main results in this work, we show that minimal (α, β)-modules can be computed in polynomial time, and we generalize series and parallel operation between graphs. This leads to (α, β)-cographs which have interesting properties. We study how can be generalized Gallai's Theorem corresponding to the case for $\alpha = \beta = 0$, but unfortunately we give evidence that computing such a decomposition tree can be difficult.

1 Introduction

First introduced for undirected graphs by Gallai in [20] to analyze the structure of comparability graphs, modular decomposition has been used and defined in many areas of discrete mathematics, including 2-structures, automata, partial orders, set systems, hypergraphs, clutters, matroids, boolean and submodular functions [14,15,18,22]. For a survey on modular decomposition, see [32] and for its algorithmic aspects [24]. Since they have been rediscovered in many fields, modules appear under various names in the literature, they have been called intervals, externally related sets, autonomous sets, partitive sets, homogeneous sets, and clans. In most of the above examples the family of modules of a given graph yields a kind of partitive family [6,8,9], and therefore leads to a unique modular decomposition tree that can be computed efficiently.

---

* This work is supported by the ANR-France Project Hosigra (ANR-17-CE40-0022). Preliminary results of this work were presented in [23].

Roughly speaking, elements of a module M behave exactly the same with respect to elements outside of M. Thus a module can be contracted to a single element without losing neighbourhood and connectivity information. This technique has been used to solve many optimization problems and has led to a number of elegant graph algorithms, see for instance [31]. Other direct applications of modular decomposition appear in areas such as computational protein-protein interaction networks and graph drawing [19,38]. Recently, new applications have appeared in the study of large networks [44,35], where a module is considered as a regularity or a community that has to be detected and understood.

Although it is well known that almost all graphs have no non-trivial modules [33], some graphs that arise from real data seem to have many non-trivial modules [37]. How can we explain such a phenomenon? It could be that the context in which this real data is generated has a clustering structure; but it could also be because we reach some known regularities as predicted by Szemerédi's Regularity Lemma [45]. In fact for every $\epsilon > 0$, Szemerédi's lemma asserts the existence of an n_0 such that all undirected graphs with more than n_0 vertices admit an ϵ-regular partition of their vertices. Such a partition is a kind of an *approximate* modular decomposition, and linear time algorithms for *exact* modular decomposition are known [24].

Our results. In this paper we introduce and study a new generalization of modular decomposition by relaxing the strict neighbourhood condition of modules with a tolerance of some errors (missing or extra edges). In particular, we define an (α, β)-module to be a set M whose elements behave exactly the same with respect to elements outside of M, except that each "outside" element can have either at most α missing edges or at most β extra edges connecting it to M. In other words, an (α, β)-module M can be turned into a module by adding at most α edges, or deleting at most β edges, at each element outside M. In particular, we recover the standard modular decomposition when $\alpha = \beta = 0$.

This new combinatorial decomposition is not only theoretically interesting but also can lead to practical applications. We first prove that every graph admits an (α, β)-modular decomposition tree which is a kind of generalization of Gallai's modular decomposition Theorem. But by no means such a tree is unique and we also give evidence that finding such a tree could be NP-hard. On the algorithmic side we propose a polynomial algorithm to compute a covering of the vertex set by minimal (α, β)-modules with a bounded overlap, in $O(m \cdot n^{\alpha+\beta+1})$ time. For the bipartite case, when we restrict (α, β)-modules on one side of the bipartition, we completely compute all these (α, β)-modules. In particular, we give an algorithm that computes a covering of the vertices of a bipartite graph in $O(n^{\alpha+\beta}(n + m))$ time, using maximal (α, β)-modules. This can be of great help for community detection in bipartite graphs.

Organization of the paper. Section 2 covers the necessary background on standard modular decomposition, introduces (α, β)-modules and illustrates various applications of (α, β)-modular decomposition. Sections 3 covers structural properties of (α, β)-modules and the NP-hardness results. Section 4 contains all the algorithmic results, in particular the computation of minimal (α, β)-modules as well as (α, β)-primality testing. Section 5 covers the complete determination of (α, β)-modules that lay one side of a bipartite graph. We conclude in Section 6 with an alternate relaxation of modular decomposition.

2 Modular Decomposition: A Primer

Let $G = (V(G), E(G))$ be a graph on $|V(G)| = n$ vertices and $|E(G)| = m$ edges. For two adjacent vertices $u, v \in V(G)$, uv denotes the edge in $E(G)$ with endpoints u and v. All the graphs considered here are simple (no loops, no multiple edges), finite and undirected. The complement of a graph $G = (V, E)$ is the graph $\overline{G} = (V(G), \overline{E(G)})$ where $uv \in \overline{E(G)}$ if and only $uv \notin E(G)$. We often refer to the sets of vertices and edges of G as V and E respectively, if G is clear from the context.

For a set of vertices $X \subseteq V$, we denote by $G(X)$ the induced subgraph of G generated by X. The set $N(v) = \{u : uv \in E\}$ is the *neighbourhood* of v and the set $\overline{N}(v) = \{u : u \neq v \text{ and } uv \notin E\}$ the *non-neighbourhood* of v. This notation can also be extended to sets of vertices: for a set $X \subseteq V$, we let

$$N(X) = \{y \in V \setminus X : \exists x \in X \text{ and } xy \in E(G)\},$$

and

$$\overline{N}(X) = \{y \in V \setminus X : \forall x \in X,\ xy \notin E(G)\}.$$

Note here that $N(X)$ is not the union of the sets $N(x)$ for all $x \in X$, but the set of vertices outside from X that have a neighbour in X.

Two vertices u and v are called *false twins* if $N(u) = N(v)$, and *true twins* if $N(u) \cup \{u\} = N(v) \cup \{v\}$.

A *Moore family* on a set X is a collection of subsets of X that contains X itself and is closed under intersection.

Definition 1. *A* ***module*** *of a graph $G = (V, E)$ is a set of vertices $M \subseteq V$ that satisfies*

$$\forall x, y \in M,\ N(x) \setminus M = N(y) \setminus M.$$

In other words, $V \setminus M$ is partitioned into two parts A, B such that there is a complete bipartite subgraph between M and A, and no edges between M and B. Observe that we have $A = N(M)$, and $B = \overline{N}(M)$.

A single vertex $\{v : v \in V\}$ is always a module, and so are the empty module and the set V. Such modules are called *trivial modules.* A graph with only trivial modules is called a *prime* graph. A module is *maximal* if it is not contained in any other non-trivial module.

A *modular decomposition tree* of a graph G is a tree $T(G)$ that captures the decomposition of G into modules. The leaves of $T(G)$ represent the vertices of G, the internal nodes of $T(G)$ capture operations on modules, and are labelled *parallel, series,* or *prime.* A parallel node captures the disjoint union of its children, whereas a series node captures the full connection of its children. A prime node is one whose children can only be decomposed into trivial modules.

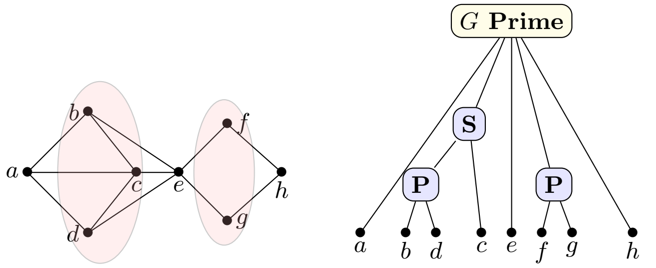

Fig. 1. A graph G (left) and its modular decomposition tree (right). Maximal modules are red, series and parallel nodes are labelled in the tree as S and P respectively.

Parallel and series nodes are often referred to as *complete* nodes. Fig. 1 illustrates a graph with its modular decomposition tree.

By the Modular Decomposition Theorem [9,20], every graph admits a *unique* modular decomposition tree. Other combinatorial objects also admit unique decomposition trees, partitive families in particular.

Two sets A and B *overlap* if $A \cap B \neq \emptyset$, $A \setminus B \neq \emptyset$, and $B \setminus A \neq \emptyset$. In a family of subsets $\mathcal{F}$ of a ground set V, a set $S \in \mathcal{F}$ is *strong* if S does not overlap with any other set in $\mathcal{F}$. We denote by Δ the *symmetric difference* of two sets:

$$A \Delta B = \{a : a \in A \setminus B\} \cup \{b : b \in B \setminus A\}.$$

Definition 2 ([9]). *A family of subsets $\mathcal{F}$ over a ground set V is* ***partitive*** *if*

(i) $\emptyset$, V, *and all singletons* $\{x : x \in V\}$ *belong to* $\mathcal{F}$, *and*
(ii) $\forall A, B \in \mathcal{F}$, *if* $A \cap B \neq \emptyset$ *then* $A \cup B \in \mathcal{F}$, $A \cap B \in \mathcal{F}$, $A \setminus B \in \mathcal{F}$, *and* $A \Delta B \in \mathcal{F}$.

Partitive families play a fundamental role in combinatorial decomposition [8,9]. Every partitive family admits a unique decomposition tree with only complete and prime nodes. The strong elements of $\mathcal{F}$ form a tree ordered by the inclusion relation [9].

A *complement reducible* graph is a graph whose decomposition tree has no prime nodes, that is, the graph is totally decomposable into parallel and series nodes only. Complement reducible graphs are also known as *cographs*, and are exactly the P_4-free graphs [43]. A modular decomposition tree of a cograph is often referred to as a *cotree*. Cographs have been widely studied, and many typical NP-hard problems (colouring, independent set, etc.) become tractable on cographs [11].

2.1 Generalizations of Modular Decomposition / Motivation

Finding a non-trivial tractable generalization of modules is not an easy task. Indeed, when trying to do so, we are faced with two main difficulties.

The first one is to obtain a pseudo-generalization. Suppose for example that we change the definition of a module into: $\forall x, y \in M$, $N^*(x) \setminus M = N^*(y) \setminus M$, where $N^*(x)$ can mean something like "vertices at distance at most k" or "vertices joined by an odd path", etc. In many of these scenarios, it turns out that the problem transforms itself into the computation of precisely the modules of some auxiliary graph built from the original one. Some work in this direction avoiding this drawback can be found in [7].

The second difficulty is NP-hardness. Consider the notion of *roles* defined in sociology, where two vertices play the same role in a social network if they have the same set of colours in their neighbourhood. In this scenario, if a colouring of the vertices is given, then one can compute these *roles* in polynomial time. Otherwise, the problem is indeed a colouring problem which is NP-hard to compute [17].

In this work, we consider two variations of the notion of modules, both of which trying to avoid these two difficulties. Some of these new modules are polynomial to compute, and we believe they are worth studying further. We focus on the most promising relaxation, namely what we call (α, β)-modules.

Our initial idea was to allow some "errors" by saying that at most k edges (for some fixed integer k) could be missing in the complete bipartite subgraph between M and $N(M)$, denoted $(M, N(M))$, and, symmetrically, that at most k extra edges can exist between M and $\overline{N}(M)$. But by doing so, we lose most of the nice algebraic properties of modules which yield an underlying partitive family. Furthermore, most modular decomposition algorithms are based on these algebraic properties [9].

A second natural idea is to relax the condition on the complete bipartite subgraph $(M, N(M))$, for example by asking for a graph that does not contain any $2K_2$ (two disjoint edges). Unfortunately, as shown in [40], to test whether a given graph admits such a decomposition is NP-complete. In fact, in the same work, the authors studied a generalized join decomposition solving a question raised in [28] about perfection. A completely different type of generalization was proposed and studied by Ehrenfeucht and McConnell in [13] where they introduce the notion of a *k-structure* that unifies the prime decomposition on 2-structures as well as k-ary relations; now while this new notion of k-structures is a generalization of these two concepts, it is not itself a relaxation of the exactitude constraint of modular decomposition.

For all the above reasons and obstacles, we focus on (α, β)-modules which maintain some algebraic properties and thus allow to obtain nice algorithms.

Intuitively, we want the reader to think of an (α, β)-module as a subset of vertices that *almost* looks the same from the outside. So, if M is an (α, β)-module, then for all $x, y \in V \setminus M$, $N(x) \cap M$ and $N(y) \cap M$ are *almost* the same if both x, y have at least β neighbours in M each, or both x, y have at least α non-neighbours in M each. In other words, either x, y see nearly all of M or x, y do not see M with the exception of at most $\alpha + \beta$ "errors", where an error is either a missing edge or an extra edge. We use the integers α and β to bound the number of errors in the adjacency, according to their type.

Formally, we define an (α, β)-module as follows:

Definition 3. *An* $\boldsymbol{(\alpha, \beta)}$***-module*** *of a graph* $G = (V, E)$ *is a set of vertices* $M \subseteq V$ *that satisfies*

$$\forall x \in V \setminus M,\ |M \cap N(x)| \geq |M| - \alpha \text{ or } |M \cap N(x)| \leq \beta.$$

In other words, M can be turned into a (standard) module by adding at most α edges or deleting at most β edges at each vertex outside M.

This notion of missing or extra edges, that we call (α, β)-*errors*, finds application naturally in various fields, from data compression and exact encodings to approximation algorithms.

Indeed, modular decomposition is often presented as an efficient way to encode a graph. This encoding property is preserved under the (α, β)-modules. We want to be able to contract a non-trivial (α, β)-module (to be precisely defined later, see Definition 7) into a single vertex while keeping *almost* the entirety of the original graph, and then apply induction on the decomposition.

To this end, for a graph $G = (V, E)$, let M be a non-trivial (α, β)-module with X being the neighbourhood of M minus some α edges missing, and Y the non-neighbourhood of M with some extra β edges (we call these the α-neighbourhood and β-non-neighbourhood of M which we define formally in the following section). The point is, if we want an exact encoding of G, we can contract M into a unique vertex m adjacent to every vertex in X, and non-adjacent to any vertex in Y. We then keep track of the subgraph $G(M)$ and the errors that potentially arose from the missing edges in (M, X) (i.e., the missing α edges) and the extra edges in (M, Y) (i.e., the extra β edges). This new encoding has at least $|X| \cdot (|M| - \alpha - 1)$ edges less than the original encoding in the worst case and $|X| \cdot (|M| - 1)$ when M is a module.

3 Structural Properties of (α, β)-Modules

In order to maintain some of the algebraic properties of modules, and avoid running into the NP-complete scenarios previously mentioned, the (α, β) generalization of modules seems to be a good compromise.

We emphasize a few points concerning (α, β)-modules. Note first that we tolerate α or β "error-edges" per vertex outside the module, depending on how this vertex is connected to the (α, β)-module, and not $\alpha + \beta$ error-edges *per module*. Secondly, observe that when $\alpha = \beta = 0$, we recover the standard definition of modules (see Definition 1), which can be rephrased as follows.

Definition 4. *A module of a graph* $G = (V, E)$ *is a set of vertices* $M \subseteq V$ *that satisfies*

$$\forall x \in V \setminus M,\ M \cap N(x) = \emptyset \text{ or } M \cap N(x) = M.$$

Of course we only consider cases for which $\max(\alpha, \beta) < |V| - 1$. Fig. 2 illustrates an example of a graph with a $(1, 1)$-module.

Let us begin with some simple properties that directly follow from Definition 3.

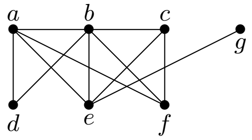

Fig. 2. The set $\{d, e, f\}$ is not a standard module, nor a $(1, 0)$ or a $(0, 1)$-module, only a $(1, 1)$-module.

Proposition 1. *If M is an (α, β)-module of G, then the following holds.*

1. *M is an (α', β')-module of G, for every $\alpha \leq \alpha'$ and $\beta \leq \beta'$.*
2. *M is a (β, α)-module of $\overline{G}$.*
3. *M is an (α, β)-module of every induced subgraph $G(N)$ of G with $M \subseteq N$.*
4. *Every (α, β)-module of $G(M)$ is an (α, β)-module of G.*

Proof.

1. Taking α' and β' such that $\alpha \leq \alpha'$ and $\beta \leq \beta'$ can only relax the module conditions.
2. Moving to the complement just interchanges the roles of α and β in the definition.
3. If the (α, β)-module conditions are satisfied for all vertices in $V(G) \setminus M$, then they are satisfied for all vertices in $V(G) \setminus N$ for $M \subseteq N$. Therefore M is an (α, β)-module of the induced subgraph $G(N)$.
4. Let N be an (α, β)-module of the subgraph $G(M)$. So every vertex in $M \setminus N$ satisfies the (α, β)-module conditions. Since M is supposed to be an (α, β)-module, every vertex in $V(G) \setminus M$ satisfies the (α, β)-module conditions for M and therefore also for $N \subseteq M$.

□

Definition 5. *Let $G = (V, E)$ be a graph and $A \subseteq V$ be a set of vertices. The **α-neighbourhood** and **β-non-neighbourhood** of A are, respectively,*

$$N_\alpha(A) = \{x \notin A : |N(x) \cap A| \geq |A| - \alpha\}, \text{ and}$$
$$\overline{N}_\beta(A) = \{x \notin A : |N(x) \cap A| \leq \beta\}.$$

*Moreover, if $x \in N_\alpha(A)$ (resp. $x \in \overline{N}_\beta(A)$), we say that x is an **α-neighbour** of A (resp. a **β-non-neighbour** of A) and that x is **α-adjacent** (resp. **β-non-adjacent**) to every vertex of A.*

Definition 6. *Let $G = (V, E)$ be a graph and $A \subseteq V$ be a set of vertices. A vertex $z \notin A$ is an **(α, β)-splitter** for A if*

$$\beta < |N(z) \cap A| < |A| - \alpha.$$

We denote by $S_{\alpha,\beta}(A)$ the set of (α, β)-splitters of A.

Hence, a set A is an (α, β)-module if and only if $S_{\alpha,\beta}(A) = \emptyset$. As an immediate consequence we have the following easy facts.

Lemma 1. *For every graph $G = (V, E)$ and every set of vertices $A \subseteq V$, the following holds.*

1. $N_\alpha(A) \cup \overline{N}_\beta(A) \cup S_{\alpha,\beta}(A) = V \setminus A$.
2. *If* $|A| \geq \alpha + \beta + 1$, *then* $N_\alpha(A) \cap \overline{N}_\beta(A) = \emptyset$.
3. *If* $|A| \leq \alpha + \beta + 1$, *then* $S_{\alpha,\beta}(A) = \emptyset$.
4. *If* $|A| = \alpha + \beta + 1$, *then* $N_\alpha(A)$ *and* $\overline{N}_\beta(A)$ *partition* $V \setminus A$.
5. *If A is an (α, β)-module of G and* $|A| \geq \alpha + \beta + 1$, *then* $N_\alpha(A)$ *and* $\overline{N}_\beta(A)$ *partition* $V \setminus A$.

Proof.

1. This directly follows from the definitions of these sets.
2. If $x \in N_\alpha(A)$, then $|N(x) \cap A| \geq |A| - \alpha \geq \beta + 1$ and thus $x \notin \overline{N}_\beta(A)$.
3. If $x \in S_{\alpha,\beta}(A)$, then $|N(z) \cap A| < |A| - \alpha \leq \beta + 1$, a contradiction.
4. We have $N_\alpha(A) \cap \overline{N}_\beta(A) = \emptyset$ by Item 2, and $S_{\alpha,\beta}(A) = \emptyset$ by Item 3. The result then follows from Item 1.
5. This follows from Items 1 and 2. ☐

Lemma 2. *For every graph $G = (V, E)$ and every set of vertices $A \subseteq V$, if $|A| \leq \alpha + \beta + 1$, then A is an (α, β)-module of G.*

Proof. Using Lemma 1.3, A admits no (α, β)-splitter and is thus an (α, β)-module of G. ☐

It thus seems that the subsets of size $\alpha + \beta + 1$ are crucial to the study of this new decomposition. In fact, if A is such a set, then for every vertex $z \notin A$, we have either $z \in N_\alpha(A)$ or $z \in \overline{N}_\beta(A)$, but not both (Lemma 1.3).

Lemma 3. *If a vertex s is an (α, β)-splitter for a set A, then s is also an (α, β)-splitter for every set $B \supseteq A$ with $s \notin B$.*

Proof. Let s be an (α, β)-splitter of A. We thus have $\beta < |N(s) \cap A| < |A| - \alpha$. Now, if $A \subseteq B$ and $s \notin B$, then we have $\beta < |N(s) \cap A| \leq |N(s) \cap B|$. Similarly, $A \setminus N(s) \subseteq B \setminus N(s)$. Therefore $|B \setminus N(s)| > \alpha$ which implies $|N(s) \cap B| < |B| - \alpha$.

Therefore, we get $\beta < |N(s) \cap B| < |B| - \alpha$ and s is an (α, β)-splitter for B. ☐

Theorem 1. *For every graph $G = (V, E)$, the family of (α, β)-modules of G satisfies the following:*

(i) *The set V is an (α, β)-module of G, and every set $A \subseteq V$ with $|A| \leq \alpha + \beta + 1$ is an (α, β)-module of G.*

- **(ii)** *If A and B are two (α, β)-modules of G, then $A \cap B$ is an (α, β)-module of G. Moreover, the (α, β)-splitters of $A \setminus B$ and $B \setminus A$ can only belong to $A \cap B$.*

Proof.

- **(i)** This directly follows from Definition 3 and Lemma 2.
- **(ii)** Notice first that if both A and B are trivial (α, β)-modules with less than $\alpha+\beta+1$ vertices each then so are $A \cap B$, $A \setminus B$ and $B \setminus A$. It could be the case that $A = V$ and $B \neq V$ is also a trivial (α, β)-module. But then $A \cap B = B$ is still a trivial (α, β)-module, and (α, β)-splitters for $V \setminus B$ can only come from B. So property (ii) is satisfied for all (α, β)-modules satisfying condition (i).
  Suppose now that the cardinality of both A and B is at least $\alpha+\beta+2$ and at most $|V|-1$. If $A \cap B$ has an (α, β)-splitter outside of $A \cup B$ then, by Lemma 3, A and B would also have an (α, β)-splitter, a contradiction. If $A \cap B$ has an (α, β)-splitter in $B \setminus A$ (resp. in $A \setminus B$) then, by Lemma 3, again A (resp. B) would have an (α, β)-splitter. Therefore, $A \cap B$ is an (α, β)-module of G.
  Let us now consider $A \setminus B$. If $A \setminus B$ has an (α, β)-splitter in $B \setminus A$ then, by Lemma 3, A would have a (α, β)-splitter as well. The same conclusion arises for splitters outside of $A \cup B$. Hence, the only possible (α, β)-splitters for $A \setminus B$ and, similarly, for $B \setminus A$, are in $A \cap B$.
  Finally, we consider the case where one module is trivial and the other non-trivial. Suppose, without loss of generality, that A is a trivial (α, β)-module and B is a non-trivial one.
  If $A = V$, then $A \cap B = B$ is an (α, β)-module, and so is $B \setminus a = \emptyset$ – which has no (α, β)-splitters –. Furthermore, every (α, β)-splitter of $A \setminus B = V \setminus B$ must come from $B = A \cap B$.
  Otherwise, if $|A| \leq \alpha + \beta + 1$, then $|A \cap B| \leq \alpha + \beta + 1$ is also a trivial (α, β)-module, and so must $A \setminus B$ be. In this case, if $B \setminus A$ has an (α, β)-splitter in $V \setminus B$, then this vertex must also be an (α, β)-splitter for B, thus contradicting our assumption that B is an (α, β)-module. This implies that any (α, β)-splitter of $B \setminus A$ must be in $B \setminus A$, thereby completing the proof. □

Since the family of (α, β)-modules is closed under intersection, it yields a **discrete convexity**.

Given a set A, we can compute the minimal (under inclusion) (α, β)-module $M(A)$ that contains A, with strictly more than $\alpha+\beta+1$ elements, thus computing a **modular closure** via (α, β)-splitters. Furthermore, the dual cases of $(1, 0)$-modules and $(0, 1)$-modules seem very interesting.

Definition 7. *An (α, β)-module M of a graph $G = (V, E)$ is a* ***trivial (α, β)-module*** *if either $M = V$ or $|M| \leq \alpha + \beta + 1$.*

Definition 8. *A graph is an* ***(α, β)-prime graph*** *if it has only trivial (α, β)-modules.*

Observe here that when $\alpha = \beta = 0$, trivial (α, β)-modules are exactly trivial (standard) modules, and (α, β)-prime graphs are exactly prime graphs.

From Lemma 2, we directly get the following result.

Corollary 1. *A graph* $G = (V, E)$ *with* $|V| \leq \alpha + \beta + 1$ *has only trivial* (α, β)*-modules.*

However, we want to distinguish "truly" (α, β)-prime graphs and "degenerate" (α, β)-prime graphs.

Definition 9. *A graph* $G = (V, E)$ *is* ***(α, β)-degenerate*** *if* $|V| \leq \alpha + \beta + 2$.

Notice the difference between degenerate and trivial nodes. Just like the $\alpha = \beta = 0$ case where singletons are trivial modules but modules of two vertices are degenerate, the definition of (α, β)-degenerate graphs is different from that of trivial (α, β)-modules, and (α, β)-degenerate graphs will further be used to define the (α, β)-modular decomposition tree, instead of trivial modules in the classic modular decomposition.

We call a non-trivial (α, β)-module M a **minimal non-trivial (α, β)-module** if every (α, β)-module strictly contained in M is trivial. The following result directly follows from this definition.

Proposition 2. *If* A *and* B *are overlapping minimal non-trivial* (α, β)*-modules of a graph* G*, then* $A \cap B$ *is a trivial* (α, β)*-module of* G*.*

From Theorem 1, we get the following result.

Corollary 2. *For every graph* G*, the family of* (α, β)*-modules of* G *is a Moore family.*

In the standard setting for undirected graphs, if X and Y are overlapping modules, then $X \cup Y$, $X \setminus Y$, $Y \setminus X$ and $X \Delta Y$ are also modules [9,24] - i.e., modules form a partitive family. Unfortunately, this does not always hold in the (α, β) setting. But we can improve a little the algebraic setting of (α, β)-modules.

Theorem 2. *Let* A *and* B *be two overlapping* (α, β)*-modules of a graph* G*. If* $|A \cap B| \geq \alpha + \beta + 1$*, then* $A \cup B$ *is a* $(2\alpha, 2\beta)$*-module of* G*.*

Proof. Let $z \in V \setminus (A \cup B)$. We have $S_{\alpha,\beta}(B) = \emptyset$ since B is an (α, β)-module, and $|B| \geq \alpha + \beta + 1$ since $|A \cap B| \geq \alpha + \beta + 1$. Therefore, by Lemma 1.5, $N_\alpha(B)$ and $\overline{N}_\beta(B)$ partition $V \setminus B$. Suppose $z \in N_\alpha(B)$. Then, z has at most α non-neighbours in $A \cap B$ and thus at least $\beta + 1$ neighbours in $A \cap B$. This implies $z \in N_\alpha(A)$ since A is an (α, β)-module.

Consider first $A \cup B$. Since $z \in N_\alpha(B)$ and $z \in N_\alpha(A)$, it yields in the worst case, that z has at most α non-neighbours in $A \setminus B$ and at most α non-neighbours in $B \setminus A$. As $N_\alpha(B)$ and $\overline{N}_\beta(B)$ partition $V \setminus B$, to finish the proof we just use the same reasoning on $\overline{N}_\beta(B)$, we get that $A \cup B$ is a $(2\alpha, 2\beta)$-module. $\square$

Unfortunately similar results do not hold for differences and symmetric differences.

3.1 (α, β)-Modular Decomposition Trees

Definition 10. *Let* $G = (V, E)$ *be a graph. Two disjoint sets of vertices* $A, B \subseteq V$ *are said to be* ***α-connected*** *if* $A \subseteq N_\alpha(B)$ *and* $B \subseteq N_\alpha(A)$*. Similarly, they are said to be* ***β-non-connected*** *if* $A \subseteq \overline{N}_\beta(B)$ *and* $B \subseteq \overline{N}_\beta(A)$*.*

In other words, A and B are α-connected if every vertex in A is an α-neighbour of B and every vertex in B is an α-neighbour of A. They are β-non-connected if every vertex in A is a β-non-neighbour of B and every vertex in B is a β-non-neighbour of A.

Proposition 3. *If* A *and* B *are two disjoint* (α, β)*-modules of a graph* G *with* $|A|, |B| \geq \alpha + \beta + 1$*, then* $N_\alpha(A) \supseteq B$ *and* $\overline{N}_\beta(B) \supseteq A$ *are mutually exclusive.*

Proof. Suppose to the contrary that $N_\alpha(A) \supseteq B$ and $\overline{N}_\beta(B) \supseteq A$.

Let $m_{A,B}$ be the number of edges in G joining A and B.

We thus have

$$|B| \cdot (|A| - \alpha) \leq m_{A,B} \leq |A| \cdot \beta \implies |A| \cdot |B| \leq \beta \cdot |A| + \alpha \cdot |B|. \tag{1}$$

Now, let $|A| = \alpha + \beta + a$ and $|B| = \alpha + \beta + b$, with $a, b \in \mathbb{N}^*$. We then get

$$\alpha \cdot a + \beta \cdot b + ab \leq 0,$$

a contradiction.

□

Let us recall that $\mathcal{F}$ is called laminar if $\forall F, F' \in \mathcal{F}$ either $F \cap F' = \emptyset$ or $F \subseteq F'$ or $F' \subseteq F$, using notation introduced in [42]. Such a family is naturally tree-structured. A laminar family is called maximal if all its minimal elements are either trivial or prime.

Definition 11. *A maximal laminar family of* (α, β)*-modules is called an* (α, β)*-modular decomposition tree.*

For a graph G, if $\cup_{F \in \mathcal{F}} F \neq V$, one can always partition the remaining vertices using trivial (α, β)-modules and then the leaves of the associated tree partition V.

Theorem 3. *Every graph* $G = (V, E)$ *admits an* (α, β)*-modular decomposition tree, for every* (α, β) *with* $0 \leq \alpha, \beta \leq |V| - 1$*.*

Proof. Let $G = (V, E)$ be an arbitrary graph. If G is an (α, β)-prime graph or if $|V| \leq \alpha + \beta + 2$, then G admits only trivial (α, β)-modules. Let us partition V into sets with at most $\alpha + \beta + 1$ vertices. Adding V to these parts yields a maximal laminar family.

Suppose now that G is not an (α, β)-prime graph. Then G admits at least one non-trivial (α, β)-module. We then show how to obtain a partition $\mathcal{P} = \{M_1, \ldots, M_k\}$ of V with $k \geq 3$ and $|M_1| > \alpha + \beta + 1$.

Let M_1 be any maximal (under inclusion) non-trivial (α, β)-module, $|M_1| > \alpha + \beta + 1$ since it is non-trivial, and let R be the set of remaining vertices: $R = V \setminus M_1$.

We consider three cases:

1. If $|R| > \alpha + \beta + 1$ and R contains B a non-trivial (α, β)-module of G. We set $M_2 = B$ and apply the same reasoning on the remaining vertices, until one of the 2 following case is reached.
2. If $|R| > \alpha + \beta + 1$ and R does not contain any non trivial (α, β)-module of G, then R is not an (α, β)-module of G, and we compute a partition P_1 of R into trivial modules, $\mathcal{P} = \mathcal{P} \cup P_1$. For sake of simplicity we can take as P_1 the partition of R into singletons, which is compatible with the classical modular decomposition since here R is an (α, β)-prime "part" of G. Note that it could be the case that $G(R)$ admits some non trivial (α, β)-modules which are not (α, β)-modules of G.
3. If $|R| \leq \alpha + \beta + 1$. Then R is a trivial (α, β)-module and let $\mathcal{P} = \mathcal{P} \cup \{R\}$.

Now for every non trivial and non prime M_i we recurse on $G(M_i)$ and using Proposition 1-4 we finally obtain a maximal laminar family and therefore a (α, β)-modular decomposition tree. □

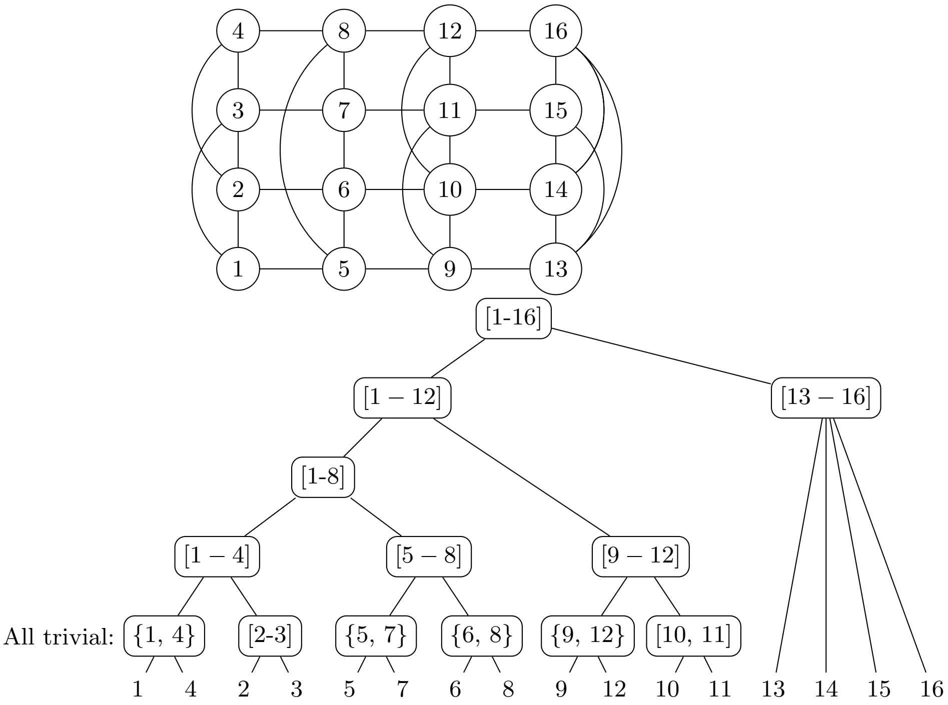

Fig. 3. A graph G and its (0,1)-decomposition where $\alpha = 0, \beta = 1$

Consider the graph in Fig. 3, and let $\alpha = 0, \beta = 1$. The set $[1-12] = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12\}$ is a maximal (0,1)-module, which results in the partition of the vertices $\{[1-12],[13-16]\}$ into (0,1)-modules. The set R as defined in Theorem 3 above is $R=[13-16]$, which is not a trivial module since $|R| > \alpha+\beta+1 = 2$.

We continue the decomposition by recursing on both sides to obtain $\{[1-8],[9-12]\}$ and $\{\{13,16\},\{14,15\}\}$. Similarly the sets $[1-8]$ and $[9-12]$ can be further decomposed.

As can be seen in the above example $[5-16]$ is also a maximal (0,1)-module overlapping $[1-12]$. Therefore maximal (α,β)-modules may overlap, which unfortunately means, that such an (α,β)-modular decomposition tree is not unique. Although such a decomposition tree can provide an exact encoding of the graph (if the (α,β)-errors are traced), it does not provide an encoding of all existing (α,β)-modules, see Fig. 4 for instance. Furthermore, although the above proof is constructive, Theorem 3 does not lead to an efficient algorithm for computing an (α,β)-modular decomposition tree. In fact, we do not know of any polynomial algorithm to compute M_1, a maximal non-trivial (α,β)-module, to start with.

Definition 12. *k-parallel (similarly for k-series) operations :*

For any integer k and two disjoint graphs G_1, G_2 with $V(G_1)\cap V(G_2)=\emptyset$, we define k-parallel$(G_1,G_2)$ to be the set of graphs defined on the vertex set $V(G_1)\cup V(G_2)$ by adding a k-factor cutset between $V(G_1)$ and $V(G_2)$, i.e. a bipartite graph with degree bounded by k.

Furthermore, $G \in$ k-series(G_1,G_2) if $\overline{G} \in$ k-parallel$(\overline{G_1},\overline{G_2})$.

As a consequence of this definition, these operations are symmetric, i.e., *k-parallel*$(G_1,G_2) =$ *k-parallel*(G_2,G_1).

Taking the decomposition viewpoint, when $G \in$ *k-series*(G_1,G_2) (resp. $G \in$ *k-parallel*(G_1,G_2)) we say that G admits a k-series (resp. k-parallel) decomposition into G_1, G_2.

A graph G admits a k-series decomposition if and only if $\overline{G}$ admits a k-parallel decomposition.

For a k-parallel composition with $k=1$ we simply add a matching between the two graphs G_1 and G_2 and we immediately notice the following property.

Proposition 4. *If $G \in$ k-parallel(G_1,G_2), then $V(G_1), V(G_2)$ are both $(0,k)$-modules of G.*

Proof. $\forall z \in V(G_2)$, z admits at most k neighbours in V_1 which is exactly the definition 3 of $V(G_1)$ being a $(0,k)$-module. The same reasoning holds for $V(G_2)$. □

As we shall see next, the problem of recognizing if a graph admits a 1-parallel decomposition is related to a nice combinatorial problem first studied in [21]. The problem of finding such a decomposition is equivalent to finding a *matching cutset* in a graph, i.e., an edge cut which is a matching, a well-known problem studied in [4,10,34].

Unfortunately however, it turns out –as one might expect– that finding such a matching cutset in an arbitrary graph is an NP-complete problem, as shown by Chvátal in [10].

Theorem 4 ([10]). *Deciding if a graph has a matching cutset is NP-complete.*

Corollary 3. *Deciding if a graph has a 1-parallel (resp. 1-series) decomposition is NP-complete.*

Proof. By definition, if a graph admits a 1-parallel decomposition it admits a matching cutset. The converse is also trivially true. For 1-series decomposition it suffices to consider the complement graph. □

But we can have a result a little more involved.

Let us consider a graph which is a $C_4 = [a, b, c, d]$. Clearly it can be decomposed either by a 1-parallel operation on the sets $\{a, b\}$ and $\{c, d\}$ or by a 1-series operation on the same sets. But we shall prove that it is a particular case.

Theorem 5. *For a graph G with strictly more that $4(\alpha+\beta+1)$ vertices, α-series and β-parallel operations are mutually exclusive.*

Proof. Suppose we have 2 different decompositions of $V(G)$: One α-series in $\mathcal{A} = \{A_1, A_2\}$ and β-parallel in $\mathcal{B} = \{B_1, B_2\}$. Let us define the following partition of the vertices of G: $\mathcal{P} = \{A_1 \cap B_1, A_1 \cap B_2, A_2 \cap B_1, A_2 \cap B_2\}$. We have to consider several cases depending on the size of $|\mathcal{P}|$.

1. $|\mathcal{P}| = 2$.
   Necessarily the partitions are the same, let $A_1 = B_1$ such that $|A_1| \geq n/2 \geq 2(\alpha + \beta + 1)$ then it contradicts Lemma 1-2 which forces $A_2 \cap B_2 = \emptyset$.
2. $|\mathcal{P}| = 3$.
   By symmetry we may assume that $B_1 \subsetneq A_1$, then $A_2 \subsetneq B_2$ and $B_1 \cap A_2 = \emptyset$. If $|B_1| > n/2$ then using Lemma 1-2 forces on $A_2 \cap B_2 = A_2 = \emptyset$, a contradiction. Else we consider B_2 with a similar reasoning.
3. $|\mathcal{P}| = 4$.
   In this case, all the 4 parts of $\mathcal{P}$ are non empty. the 2 partitions $\mathcal{A}, \mathcal{B}$ cross and one, say $A_1 \cap B_1$, has strictly more than $(\alpha + \beta + 1)$ vertices.
   So we can find a vertex $z \in A_2 \cap B_2$. Since $z \in A_2$, z has at most α non-neighbours in A_1. As $z \in B_2$, it has at most β neighbours in B_1. We have:
   $|A_1 \cap B_1| - \alpha \leq |N(z) \cap (A_1 \cap B_1)| \leq \beta$, which yields $|A_1 \cap B_1| \leq \alpha + \beta$, a contradiction.

□

Let us call (α, β)-series-parallel decomposable a graph G that admits an α-series or a β-parallel decomposition.

Corollary 4. *For any fixed integer k and a graph $G = (V, E)$ computing an (k, k)-series-parallel decomposition and a k-factor cutset are polynomially equivalent.*

Proof. Suppose we have a polynomial algorithm $\mathcal{A}$ to compute a k-factor cutset. By applying $\mathcal{A}$ on G and $\overline{G}$ we can obtain an (k, k)-series-parallel decomposition of G if exists one.

Conversely, if $|V| < 8k + 4$, we can check all 2^{8k+3} bipartitions of $V(G)$ to find if there exists k-series or k-parallel decomposition. This can be done in $O(1)$ since k is fixed.

Else if there was a polynomial algorithm that computes such a decomposition when it exists, i.e. a partition of the vertices $\mathcal{P} = \{A_1, A_2\}$. Then in linear time we can decide if it is a k-series or a k-parallel decomposition.

If we find a k-parallel decomposition, we can produce a k-factor cutset. And if we find a k-series decomposition using Theorem 5, we can answer: There is no k-factor cutset in G.

□

So if we can compute in polynomial time an $(1, 1)$-series-parallel decompositon of a graph with strictly more than 12 vertices, then we can answer polynomially if it has a matching cutset. A contradiction with the NP-hardness of finding a matching cutset in a graph of Theorem 4.

Open Problem 1 *Does the NP-hardness hold for any $k > 1$? We conjecture yes.*

In the standard modular decomposition setting, the notion of strong modules, i.e., modules that do not overlap any other module, is quite central. In the (α, β)-modular decomposition setting, observe that there are no strong (α, β)-modules other than $\{V\}$ and the singletons $\{v : v \in V\}$. This comes from the fact that when $\max\{\alpha, \beta\} \geq 1$, every subset of vertices of size 2 is a trivial (α, β)-module. Now, assume there is a standard strong module $A \neq V$ with $|A| > 1$. By taking any vertex $v \in A$ and any vertex $u \in V \setminus A$, we get an (α, β)-module of size 2 which overlaps A.

In the next section we will see not only that (α, β)-modular decomposition trees are not unique but also that it could be NP-complete to find if a particular one exists.

3.2 (α, β)-Cographs

Let us focus on the (α, β)-series and (α, β)-parallel decompositions.

Definition 13. *The class of (α, β)-cographs is the smallest class of graphs containing all graphs having less than $\alpha + \beta + 1$ vertices and closed under α-series and β-parallel operations.*

Another way to see this new class of graphs.

An $\boldsymbol{(\alpha, \beta)}$**-cograph** is a graph that can be totally decomposed with α-series and β-parallel decompositions until we reach graphs having less than $\alpha + \beta + 1$ vertices.

Using Definition 13 above, it is clear that standard cographs are precisely the $(0, 0)$-cographs. Moreover if G is an (α, β)-cograph, then G is also an (α', β')-cograph, for all $\alpha' \geq \alpha$ and $\beta' \geq \beta$.

Furthermore one can associate to any (α, β)-cograph a decomposition tree called an (α, β)-cotree, whose nodes are labelled with α-series, or β)-parallel and the leaves correspond to graphs with at most $(\alpha + \beta + 1)$ vertices.

Every node of the cotree corresponds to an (α, β)-module of G, and therefore it yields a maximal laminar family. Therefore this cotree is a particular case of (α, β)-modular decomposition tree.

Consider the two examples illustrated in Figures 4 and 5. Fig. 4 shows a $(1, 1)$-cograph H that admits a unique $(1, 1)$-cotree. Fig. 5 shows a $(0, 1)$-cograph G that admits *two* different $(0, 1)$-cotrees. Moreover, if we replace each vertex of G in Fig. 5 by an isomorphic copy of G, and repeat this process, we can build a $(0, 1)$-cograph which has *exponentially many* different $(0, 1)$-cotrees.

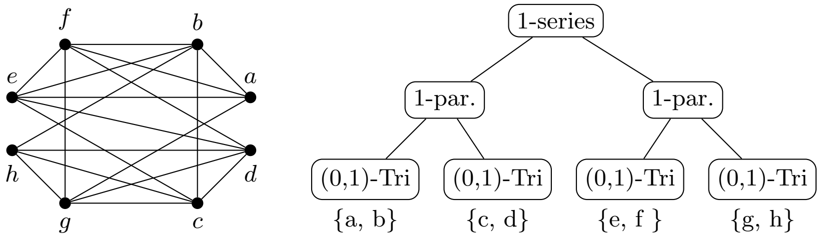

Fig. 4. A $(1, 1)$-cotree of the $(1, 1)$-cograph H on the left. Tri means trivial module. Notice that H is not a usual cograph since it contains two induced P_4's: $\{a, b, c, d\}$ and $\{e, f, g, h\}$.

It should be noticed on this example that we may be forced to have 2 consecutive k-parallel nodes on a path from the root to a leaf in these news cotrees. For usual cographs if we have 2 consecutive parallel nodes we can merge them in a unique bigger parallel node having more children. But for the graph of Fig. 5 we cannot build a bigger 1-parallel node since $\{a, b\}$ is not 1-parallel with the remaining vertices. But al least we know that for an (α, β)-cograph with strictly more that $4(\alpha + \beta + 1)$ vertices all its cotrees have a the same label on their root, using Theorem 5.

Definition 14. *Using the terminology of [12] for combinatorial decompositions, we will say that a graph* $G = (V, E)$ *is* $\boldsymbol{(\alpha, \beta)}$***-brittle*** *if every subset of* V *is an* (α, β)*-module.*

Of course, $(\alpha, 0)$-complete graphs (i.e., complete graphs missing at most α edges), and $(0, \beta)$-independent graphs (i.e., independent sets with at most β edges) are (α, β)-brittle, but they are not the only obvious ones; any path P_k is also $(1, 1)$-brittle.

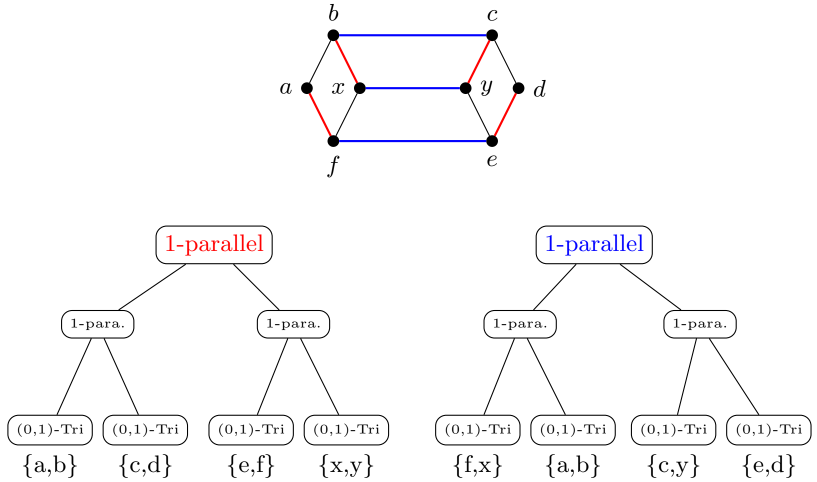

Fig. 5. The graph G on the top is a $(0,1)$-cograph, with two different $(0,1)$-cotrees. The internal nodes have the same labels but the partitions of V induced by the leaves of the $(0,1)$-cotrees are not the same.

As already seen previously, all graphs G with $|V| \leq \alpha+\beta+2$ are (α,β)-brittle, and we called them (α,β)-degenerate to distinguish them from the "truly" (α,β)-prime graphs. All these remarks raise the question of the characterization of (α,β)-brittle graphs. Clearly, any graph G with minimum degree at least $|V|-\alpha$ or maximum degree at most β is (α,β)-brittle.

Definition 15. *Let $G=(V,E)$ be a graph and $\mathcal{P}=\{V_1,\ldots,V_k\}$ be the partition of V associated with an (α,β)-modular decomposition of G. If every union of parts from $\{V_1,\ldots,V_k\}$ is an (α,β)-module, we say that such a decomposition is an* ***(α,β)-modular brittle decomposition*** *of G.*

From Theorem 4, we get the following result.

Proposition 5. *For a graph $G=(V,E)$ with $|V| \geq 2(\alpha+\beta+1)$, if G admits an (α,β)-modular brittle decomposition then it admits an α-parallel or β-series decomposition.*

Proof. Suppose G admits a (α,β)-modular brittle decomposition with partition $\mathcal{P}=\{V_1,\ldots,V_k\}$. Since it is a brittle decomposition, $\bigcup\limits_{1<i\leq k} V_i$ is a (α,β)-module, and therefore the bipartition $\{V_1, \bigcup\limits_{1<i\leq k} V_i\}$ is (α,β)-modular. One of the 2 part, say V_1 has more than $(\alpha+\beta+1)$ vertices. Therefore using Lemma 1-2, either we have an α-parallel decomposition or a β-series one. $\square$

Open Problem 2 *Can we completely characterize the (α, β)-brittle graphs?*

Recall that the Modular Decomposition Theorem [20] (Gallai's Theorem) says that every graph admits a unique modular decomposition tree, using series, parallel and prime nodes. In this tree series and parallel nodes can be brittle nodes.

In the (α, β) setting, k-parallel (resp. k-series) nodes are not always brittle. Furthermore using Proposition 5 an (α, β)-brittle node can be decomposed using α-parallel or β-series decompositions. Therefore a Gallai type Theorem would have 3 kinds of nodes mutually exclusive: k-series, k-parallel or (α, β)-prime. But we do not know yet under which condition such a decomposition tree always exists.

In the above-mentioned work [10], Chvátal showed that the matching cutset problem is NP-complete on graphs with maximum degree four, and polynomial on graphs with maximum degree three. In fact the problem of finding a perfect matching cutset is also NP-hard [27]. On the other hand, computing a matching cutset in the following graph classes is polynomial:

- graphs with maximum degree three [10],
- weakly chordal graphs and line-graphs [34],
- Series-Parallel graphs [39],
- claw-free graphs and graphs with bounded clique-width, as well as graphs with bounded treewidth [4],
- graphs with diameter 2 [5]
- $(K_{1,4}, K_{1,4}+e)$-free graphs [29], and median graphs [36].

.....

From the practical side, a particular subclass of $(0, 1)$-cographs has been introduced and studied in network theory [30], namely the class of networks obtained by starting from the one vertex graph and, at step t, taking two graphs obtained in $t-1$ steps and joining them by a perfect matching. This class, called **Matching Composition Networks** in [46], contains all hypercubes, as well as all crossed, twisted and Möbius hypercubes. In general, $PMG(k)$ is a family of graphs recursively defined, that starts with all connected graphs on k vertices and, at every step, add any graph that can be obtained by selecting two graphs within the family having the same order and joining them with a perfect matching.

More formally, the family $\mathbf{PMG}(4)$, for instance, is defined as follows.

Definition 16 ($\mathbf{PMG}(4)$). *We start with the following seven connected graphs on four vertices:*

- $P_4, C_4, K_4, K_{1,3}$,
- *a triangle with a pending edge,*
- *two triangles having an edge in common.*

At every step, a new graph is obtained from two graphs in the family having the same order by joining them with a perfect matching.

Knowing that $\mathbf{PMG}(1) = \mathbf{PMG}(2)$ and that they contain hypercubes, crossed, twisted and Möbius hypercubes, we end this section with the following recognition problem.

Open Problem 3 *Given $G = (V, E)$ with $|V| = 2^n$, what is the complexity of recognizing whether $G \in \boldsymbol{PMG}(k)$ or to one of its non-trivial subclasses?*

Although computing 1-parallel and 1-series decompositions are NP-hard, the status of the recognition of (0,1)-cographs (resp. (1,0)-cographs, (1,1)-cographs) are not clear yet, since these graphs are supposed to be totally decomposable with these two operations this could make a difference. As for example the particular case of $(0, 1)$-cographs made up with median graphs for which an $O(nlogn)$ recognition algorithm exists [26]. But we conjecture that (0,1)-cographs (resp. (1,0)-cographs, (1,1)-cographs) are NP-complete to recognize.

3.3 The Structure of (α, β)-Prime Graphs

It is well-known that usual prime graphs are connected and their complement also. we can generalize this result using another connectivity

Proposition 6. *For an (α, β)-prime graph with strictly more than $2(\alpha + \beta + 1)$ vertices, G does not admit an α-factor cutset and its complement $\overline{G}$ does not admit a β-factor cutset.*

Proof. If G admits an α-parallel decomposition into G_1, G_2, then at least one of $|V(G_1)|$ or $|V(G_2)|$ is greater than $(\alpha+\beta+1)$ and therefore G admits a non-trivial (α, β)-module. Same argument for a β-series decomposition of G which leads to a β-factor cutset on the complement graph. □

Proposition 7. *The only $(1, 1)$-prime graph of order 5 is C_5.*

Proof. Let $C_5 = [a, b, c, d, e]$. To prove that C_5 is a $(1, 1)$-prime graph, we just have to prove that every subset of four vertices is not a $(1, 1)$-module, which is obvious since for every such subset A, the remaining vertex not in A is connected by exactly two edges to A.

For the other direction, let $G = (V, E)$ be a $(1, 1)$-prime graph of order 5, as there is no subset as non-trivial $(1, 1)$-module. We consider any subset of V of size four, denoted as M. Let $x \in V \setminus M$, since M is not a $(1, 1)$-module, $|N(x) \cap M| > 1$ and $|N(x) \cap M| < 3$, thus $|N(x) \cap M| = 2$ and $deg(x) = 2$. For every subset of four vertices, we have the same argument, thus G is a 2-regular graph, and the only one is C_5. □

Algorithm 1: Computing minimal (α, β)-modules.

```
   Input: A graph $G = (V, E)$ and a set $A \subseteq V$ with $|A| \geq \alpha + \beta + 2$.
   Output: $M(A)$, the unique minimal $(\alpha, \beta)$-module that contains $A$
1  $M_0 \leftarrow A$, $i \leftarrow 0$ ;
2  $S \leftarrow \{x \in V \setminus M_0 : \beta < |N(x) \cap M_0| < |M_0| - \alpha\}$;
3  while $S \neq \emptyset$ do
4      $i \leftarrow i + 1$;
5      $M_i \leftarrow M_{i-1} \cup S$;
6      $S \leftarrow \{x \in V \setminus M_i : \beta < |N(x) \cap M_i| < |M_i| - \alpha\}$;
7  $M(A) \leftarrow M_i$;
```

Notice here that the Petersen graph can be obtained by a $(0, 1)$-parallel operation made on two copies of a C_5.

Obviously, we have the following inclusion: For all $\alpha \leq \alpha'$ and all $\beta \leq \beta'$, the family of (α', β')-prime graphs is included in the family of (α, β)-prime graphs. But can we improve this result? In the standard setting, the prime graphs are nested. In particular, P_4 is the smallest prime graph, and all primes on n vertices contain a prime subgraph on either $n - 1$ or $n - 2$ vertices, as shown in [41].

We pose the following problem.

Open Problem 4 *Are the* (α, β)*-prime graphs nested?*

4 Computing the minimal (α, β)-modules

Despite the negative hardness results in the previous sections, we shall now examine how to compute all minimal (α, β)-modules of a given graph in polynomial time. As mentioned earlier, non-trivial (α, β)-modules have strictly more than $\alpha + \beta + 2$ elements; and since they are closed under intersection, (α, β)-modules have an underlying graph convexity, and thus (see Algorithm 1), we can compute the minimal (α, β)-module $M(A)$ that contains a given set A with $|A| > \alpha + \beta + 2$, by computing a **modular closure** via (α, β)-splitters. In fact, we build a series of subsets M_i that starts with $M_0 = A$ and satisfies $M_i \subseteq M_{i+1}$ for every $i \geq 0$.

Proposition 8. *Algorithm 1 computes the unique minimal* (α, β)*-module that contains* A *in* $O(m \cdot n)$ *time.*

Proof. If A is an (α, β)-module, then in line 2, $S = \emptyset$; otherwise, all the elements of S have to be added into $M(A)$. In other words, using Lemma 3, there is no (α, β)-module M such that : $A \subsetneq M \subsetneq A \cup S$. At the end of the while loop, either $M_i = V$ or we have found a non-trivial (α, β)-module that contains A.

This algorithm obviously runs in $O(m \cdot n)$ time naivily or in $O((\alpha + \beta) \cdot m)$ time using standard algorithmic techniques. □

Algorithm 2: Computing minimal (α, β)-modules.

```
Input: A graph $G$ and $A \subseteq V(G)$ with $|A| \geq \alpha + \beta + 2$.
Output: $M(A)$, the minimal $(\alpha, \beta)$-module that contains $A$
$OPEN \leftarrow A$;
$M(A) \leftarrow \emptyset$;
foreach $u \in V$ do
    $CLOSED(u) \leftarrow FALSE$; $edge(u) \leftarrow 0$; $non\text{-}edge(u) \leftarrow 0$;
while $OPEN \neq \emptyset$ do
    Select a vertex $z$ from $OPEN$ and delete $z$ from $OPEN$;
    Add $z$ to $M(A)$;
    $CLOSED(z) \leftarrow TRUE$;
    foreach $u$ neighbour of $z$ do
        if $CLOSED(u) = FALSE$ and $u \notin M(A)$ then
            $edge(u) \leftarrow edge(u) + 1$;
            if $\beta < edge(u)$ and $\alpha < non\text{-}edge(u)$ then
                Add $u$ to OPEN
    foreach $v$ non-neighbour of $z$ do
        if $CLOSED(v) = FALSE$ and $v \notin M(A)$ then
            $non\text{-}edge(v) \leftarrow non\text{-}edge(v) + 1$;
            if $\beta < edge(v)$ and $\alpha < non\text{-}edge(v)$ then
                Add $v$ to OPEN
```

Algorithm 2 proposes a different implementation that uses a graph search approach to compute the minimal (α, β)-module containing A. This will allow us to achieve a linear running time.

Theorem 6. *Algorithm 2 can be implemented in $O(m + n)$ time.*

Proof. We can implement Algorithm 2 as a kind of a graph search, using an algorithm less naive than Algorithm 1. Algorithm 2 also computes the minimal (α, β)-module that contains A in a graph search manner.

At the end of Algorithm 2, the set $M(A)$ contains a minimal (α, β)-module that contains A. At first glance, this algorithm requires $O(n^2)$ operations, since for each vertex we must consider all its neighbours and all its non-neighbours.

However, if we use a partition refinement technique as defined in [25], starting with a partition of the vertices as $\mathcal{P} = \{A, V \setminus A\}$. We then keep in the same part, $B(i, j)$, vertices x, y with *edge*$(x) =$ *edge*$(y) = i$ and *non-edge*$(x) =$ *non-edge*$(y) = j$. This way, when visiting a vertex z, it suffices to compute

$$B'(i + 1, j) = B(i, j) \cap N(z), \text{ and}$$
$$B''(i, j + 1) = B(i, j) - N(z),$$

for each part $B(i, j)$ in the current partition. This can be done in $O(|N(z)|)$ time.

It should be noted that the parts need not to be sorted in the current partition, and we may have different parts with the same (edge, non-edge) values.

Algorithm 2 can thus be implemented in $O(m + n)$ time. □

Theorem 7. *Using Algorithm 2, one can compute all the minimal non-trivial* (α, β)*-modules of a given graph in* $O(m \cdot n^{\alpha+\beta+2})$ *time.*

Proof. To do so, it sufices to use Algorithm 2 starting from every subset of $\alpha + \beta + 2$ vertices. Since there exists $O(n^{\alpha+\beta+2})$ such subsets, and Algorithm 2 runs in $O(m + n)$ time for each one, we get a total complexity of $O(m \cdot n^{\alpha+\beta+2})$ time.

□

Corollary 5. *Using theorem 7, one can compute a covering of* V *with an overlapping family of minimal* (α, β)*-modules in* $O(m \cdot n^{\alpha+\beta+2})$ *time. Moreover, the overlapping of any two members of the obtained covering is bounded by* $\alpha + \beta + 1$*.*

Proof. Using theorem 7, we can compute an overlapping family of minimal (α, β)-modules in $O(m \cdot n^{\alpha+\beta+2})$ time. But this family can possibly not be a full covering of V since some vertices may not belong to any minimal non-trivial (α, β)-module. To obtain a full covering, we then simply add the remaining vertices as singletons. □

Corollary 5 can be very interesting if we are looking for overlapping communities in social networks, where the overlapping is bounded by $\alpha + \beta + 1$.

Going a step further, we can use Theorem 2 and merge every pair A, B of (α, β)-modules with $|A \cap B| \geq \alpha + \beta + 1$, either by keeping $A \cup B$ as a $(2\alpha, 2\beta)$-module, or by computing $M(A \cup B)$, the minimal (α, β)-module that contains $A \cup B$. This depends however on the structure of the maximal (α, β)-modules, and unfortunately we do not know yet under which conditions there exists a unique partition into maximal (α, β)-modules.

Corollary 6. *Checking if a graph is* (α, β)*-prime can be done in* $O(m \cdot n^{\alpha+\beta+2})$ *time.*

Proof. Easy using Theorem 7. □

5 An Application on Bipartite Graphs

In this section, let $G = (X, Y, E)$ be a bipartite graph with parts X and Y. By allowing $\alpha + \beta$ errors in the decomposition, (α, β)-modules can be made up with vertices from both X and Y. However, in some applications, we are forced to consider X and Y separately. Consider for instance the setting where X and Y represent the sets of customers and products, or the sets of DNA sequences and organisms, in which case one would want to find regularities on each side of the bipartition.

Definition 17. *For a given bipartite graph* $G = (X, Y, E)$*, we let*

$$\mathcal{F}_{\alpha,\beta}(X) = \{M : M \text{ is an } (\alpha, \beta)\text{-module of } G \text{ and } M \subseteq X\}.$$

Note that X *is not always an* (α, β)*-module of* G*.*

Proposition 9. *For every two sets* $A, B \in \mathcal{F}_{\alpha,\beta}(X)$*,* $A \cap B$*,* $A \setminus B$ *and* $B \setminus A$ *are all in* $\mathcal{F}_{\alpha,\beta}(X)$*.*

Proof. Using Theorem 1, the only (α, β)-splitters of the sets $A \setminus B$ and $B \setminus A$ must belong to $A \cap B$; but since $A, B \subseteq X$, and X is an independent set, this is not possible. $\square$

It should be noticed here that for $A \subseteq X$, the minimal (α, β)-module that contains A does not always belong to $\mathcal{F}_{\alpha,\beta}(X)$, since we may have to add (α, β)-splitters from Y. Therefore we have to use an algorithmic approach different from those developed in the previous section in order to compute $\mathcal{F}_{\alpha,\beta}(X)$.

Definition 18. *Two sets* $A, B \subseteq V$*,* $A \neq B$*, with* $|A| = |B| = \alpha + \beta + 1$*, are said to be* ***true*** $\boldsymbol{(\alpha, \beta)}$***-twin*** *(resp.* ***false*** $\boldsymbol{(\alpha, \beta)}$***-twin****) in* G *if they satisfy the following three conditions:*

1. $A \cup B$ *is an* (α, β)*-module,*
2. $\forall x \in A$*,* $x \in N_\alpha(B)$ *(resp.* $x \in \overline{N}_\beta(B)$*),*
3. $\forall y \in B$*,* $y \in N_\alpha(A)$ *(resp.* $y \in \overline{N}_\beta(A)$*).*

Observe that A and B are false (α, β)-twin sets in G if and only if A and B are true (β, α)-twin sets in $\overline{G}$.

Furthermore when applying Definition 18 to bipartite graphs, we obviously only have false (α, β)-twin sets.

Proposition 10. *A set* $M \subseteq X$ *is an* (α, β)*-module if and only if* M *is a union of false* (α, β)*-twin sets.*

Proof. Let $A, B \subseteq M$, with $|A| = |B| = \alpha + \beta + 1$. Pick any vertex $z \in Y$. If $z \in N_\alpha(A)$, then $z \in N_\alpha(M)$ and therefore $z \in N_\alpha(B)$. Therefore, A and B are false (α, β)-twin sets since they are both included in X.

The converse directly follows from Definition 18. $\square$

Consequently, in terms of $(\alpha + \beta + 1)$-tuples, the sets of false (α, β)-twin sets partition the $(\alpha + \beta + 1)$-tuples. Furthermore, using the notion of false (α, β)-twin sets, we obtain the following theorem (recall that for a graph G, $\mathcal{F}_{\alpha,\beta}$ is the set of its (α, β)-modules whose elements are in X).

Theorem 8. *For a given bipartite graph* $G = (X, Y, E)$*, the maximal elements of* $\mathcal{F}_{\alpha,\beta}(X)$ *can be computed in* $O(n^{\alpha+\beta}(n + m))$ *time.*

Proof. To do so, we first build an auxiliary bipartite graph, $G' = (\mathcal{A}, Y, E(G'))$, which represents the labelled incidence graph of the $(\alpha + \beta + 1)$-tuples of vertices of X. The set of vertices of G' is thus the set $\mathcal{A}$ of these $(\alpha + \beta + 1)$-tuples,

By Lemma 1.4, we know that every such tuple T yields a partition of Y into $N_\alpha(T)$ and $\overline{N}_\beta(T)$. The set of edges of G' is then defined by setting, for every $T \in \mathcal{A}$ and $y \in Y$,

$$Ty \in E(G') \text{ if and only if } y \in N_\alpha(T),$$

which implies

$$Ty \notin E(G') \text{ if and only if } y \in \overline{N}_\beta(T).$$

Since every vertex in X belongs to at most $O(n^{\alpha+\beta})$ tuples from $\mathcal{A}$, the number of edges in $E(G')$ is in $O(m \cdot n^{\alpha+\beta})$.

Given the auxiliary graph G', we now partition $\mathcal{A}$ into false twins. To this aim, we use every vertex in Y to refine $\mathcal{A}$ with respect to (α, β)-neighbourhood. This can be done in $O(n^{\alpha+\beta+1} + n^{\alpha+\beta} \cdot m)$ time, using standard partition refinement techniques [25].

Let $Q = \{\mathcal{A}_1, \dots \mathcal{A}_k\}$ be such a partition. We prove the following claim.

Claim: *No element of $\mathcal{F}_{\alpha,\beta}$ can contain two $(\alpha + \beta + 1)$-tuples from different parts of Q.*

Proof. Let $A_i \in \mathcal{A}_i$, $A_j \in \mathcal{A}_j$, with $i \neq j$, and S be a subset of X such that $A_i \cup A_j \subseteq S$. Since $i \neq j$, there is a vertex $y \in Y$ such that (w.l.o.g.) $A_i \in N_\alpha(y)$ and $A_j \in \overline{N}_\beta(y)$. Hence we have

$$|A_i| - \alpha \leq |S \cap N(y)| \leq |S| - |A_j| + \beta,$$

which gives

$$\beta + 1 \leq |S \cap N(y)| \leq |S| - \alpha - 1,$$

and thus y is an (α, β)-splitter for S. □

Therefore, to find the maximal elements of $\mathcal{F}_{\alpha,\beta}$, we can restrict the search to the $\mathcal{A}_i$'s. Let us now examine how to generate them. To this aim, we define a labelling λ that assigns to each ordered pair (y, A), with $y \in Y$ and $A \in \mathcal{A}$, a subset of A as follows.

- If $yA \in E(G')$ and $a_1, \dots, a_k$, $k \leq \alpha$, are the vertices from A non adjacent to y, then we set $\lambda(y, A) = \{a_1, \dots, a_k\}$.
- Symmetrically, if $yA \notin E(G')$ and $a_1, \dots, a_h$, $h \leq \beta$, are the vertices from A adjacent to y, then we set $\lambda(y, A) = \{a_1, \dots, a_h\}$.

This labelling can be done while constructing the graph G'.

Then, a maximal element F of $\mathcal{F}_{\alpha,\beta}$ is just a maximal union of elements of some $\mathcal{A}_i$, $1 \leq i \leq k$, satisfying the following:

- For every vertex $y \in Y$,
  - if every element of $\mathcal{A}_i$ is adjacent to y, then $|\cup_{A \in F} \lambda(y, A)| \leq \alpha$,
  - otherwise, $|\cup_{A \in F} \lambda(y, A)| \leq \beta$.

Note that all vertices in $\mathcal{A}_i$ are false twins, since the graph G' is bipartite, and therefore connected the same way to Y.

To produce these maximal sets, we start with $\alpha = \beta = 0$, in which case the only maximal module has an empty label. Let M_0 denote this module and $\mathcal{M}_{0,0} = \{M_0\}$ denote the set of maximal elements at this step. We then increase either α or β by one, and recursively compute the new set, $\mathcal{M}_{\alpha+1,\beta}$ or $\mathcal{M}_{\alpha,\beta+1}$, of maximal elements from the previously computed set $\mathcal{M}_{\alpha,\beta}$ (note that every maximal (α,β)-module is contained in a maximal $(\alpha+1,\beta)$-module and in a maximal $(\alpha,\beta+1)$-module as well).

For $\alpha=\beta=0$, M_0 is unique. For $\alpha=\beta=1$, there are at most $|Y|^2$ maximal $(1,1)$-modules in $\mathcal{F}_{1,1}$. Hence, there are at most $|Y|^{\alpha+\beta}$ maximal (α,β)-modules in $\mathcal{F}_{\alpha,\beta}$. This computation is therefore bounded in the whole by

$$(\alpha+\beta)(\Sigma_{i=0}^{i=\alpha+\beta}|Y|^i)\cdot(|X|^{\alpha+\beta+1}),$$

which is in the order of $O((|Y|^{\alpha+\beta+1})\cdot(|X|^{\alpha+\beta+1}))$. □

Note that these maximal elements of $\mathcal{F}_{\alpha,\beta}(X)$ may overlap. It remains to test the quality of the covering obtained on some real data graphs. We leave this as something to explore for data analysts.

6 Conclusion

Before we conclude, we want first to expose the reader to a different way to approach the approximation of modules.

6.1 *k*-splitter Modules: An Alternate Approximation

Another natural way to approach the problem of approximating modules is by restricting the number of splitters a module can have. Recall that in the standard modular decomposition setting, a splitter of a module M in a graph $G=(V,E)$ is a vertex $v\in V\setminus M$ such that there exists at least two vertices $a,b\in M$ with $av\in E$ and $bv\notin E$. By restricting the number of splitters outside a module, we get the following definition – which intuitively just allows at most k "errors" in terms of connectivity.

Definition 19. *For a given graph $G=(V,E)$, a subset M of V is a **k-splitter module** if M has at most k splitters.*

Notice then that by setting $k=0$ in the above definition, we recover the standard modular decomposition setting [24], i.e., for every $x\in V\setminus M$, either $M\cap N(x)=\emptyset$ or $M\cap N(x)=M$. So, for this approximate setting, we will necessarily only consider the case $k<|V(G)|-1$.

We begin with some obvious remarks.

Proposition 11. *If M is a k-splitter for G, then the following holds.*

1. *M is a k'-splitter module for G, for every $k'\geq k$.*
2. *M is a k-splitter module for $\overline{G}$.*
3. *If s is a splitter for M, then s is also a splitter for every set $M'\supseteq M$ with $s\notin M'$.*

Proposition 12. *The family of k-splitter modules of a graph $G = (V, E)$ satisfies the following.*

1. *Every set $A \subseteq V$ with $|A| \leq 1$ or $|A| \geq |V| - k$ is a k-splitter module of G. (We call such a set A a* ***trivial k-splitter module****.)*
2. *For every two k-splitter modules $A, B \subseteq V$ of G with $A \cap B \neq \emptyset$, $A \cup B$ is a $2k$-splitter module of G.*
3. *For every two k-splitter modules $A, B \subseteq V$ of G with $A \cap B \neq \emptyset$, $A \cap B$ is a $2k$-splitter module of G.*

Proof.

1. This follows from the definition.
2. There cannot be a splitter of $A \cup B$ that is not a splitter of either A or B since $A \cap B \neq \emptyset$. We get the $2k$ in the worst case, when both A and B have two disjoint sets of k splitters outside of $A \cup B$.
3. A splitter of $A \cap B$ in $V \setminus (A \cup B)$ is a splitter of both A and B. A splitter of $A \cap B$ in A is a splitter of B and a splitter of $A \cap B$ in B is a splitter of A. Therefore, the number of splitters of $A \cap B$ is at most the sum of the numbers of splitters of A and B, i.e., $2k$.

□

Proposition 13. *If A and B are two non-trivial k-splitter modules of a graph $G = (V, E)$, then $A \setminus B$ is a $(k + |A \cap B|)$-module of G.*

Proof. There are at most k splitters of $A \setminus B$ in $V \setminus A$, and at most $|A \cap B|$ splitters of $A \setminus B$ in $A \cap B$. □

Proposition 14. *For a graph $G = (V, E)$ and a subset $A \subseteq V$, there may exist different minimal (under inclusion) k-splitter modules containing A.*

Proof. Suppose A admits $k + 1$ splitters. Every one of these $k + 1$ splitters can be added to A in order to obtain a k-splitter module. □

In conclusion, this approximation variation is not closed under intersection, unfortunately. There was no way to define some sort of convexity, and thus no easy way to define a closure operator with this notion, which is why we have focused our study on (α, β)-modules instead.

6.2 Conclusions and Perspectives

In this work, we introduce a new notion of modular decomposition relaxation. This notion of (α, β)-module yields many interesting questions, both from a theoretical or practical point of view. Standard modular decomposition is too restrictive for graphs that arise from real data; do (α, β)-modules indeed often arise in this setting? We believe this relaxation of modular decomposition can definitely find applications in practice.

On the theory side, this new combinatorial decomposition may help to better understand graph structuration that can be obtained when grouping vertices that have similar neighbourhood. Such an idea has been successfully used with the notion of twin-width [3,2]. $Twinwidth$ 0 graphs are exactly the cographs. $(0, 1)$-cographs seems to be a wide class of graphs and it could be interesting to characterize them for example by a list of forbidden subgraphs.

Very easily we have that if $G \in k\text{-}parallel(G_1, G_2)$ then $Twinwidth(G) \leq k + Max\{Twinwidth(G_1), Twinwidth(G_2)\}$. It could be interesting to study the relationships between these notions and in particular, we raise the following question:

Open Problem 5 *Can we compute polynomially the twinwidth of a given (α, β)-cograph when one of its cotree is given?*

Furthermore it is related to fundamental combinatorial objects as for example matching cutsets and their generalization. In particular the (α, β)-cographs that contain many known graph classes seem to be very promissing.

Another natural and useful application of (α, β)-modules concerns approximation algorithms. Similarly to how cographs are the totally decomposable graphs with respect to standard modular decomposition, we can define (α, β)-cographs as the totally decomposable graphs with no (α, β)-prime graphs. Now consider the classical colouring and independent set programs on cographs. The linear time algorithms for these problems both use modular decomposition. Roughly speaking, the algorithms compute a modular decomposition tree, and keep track of the series and parallel internal nodes of the cotree by scanning the tree from the leaves to the root. Now for $\alpha + \beta \leq 1$, we can get a simple 2-approximation algorithm for (α, β)-cographs for both colouring and independent set, just by summing over all the (α, β)- errors.

Our work leaves many interesting questions open (five open questions and one conjecture), as the study of (α, β)-prime graphs for instance. We have also exhibited new classes of graphs, such as the $(1, 1)$-cographs, that contain many interesting subclasses and on which it would be interesting to consider the Erdös-Hajnal conjecture [16], which holds for cographs and is closed under substitution [1]. Our work could also be related to the very interesting new generalization of cographs introduced in [3].

We have also presented polynomial time algorithms that we believe could all be improved. It is important however to keep in mind that since the number of unions of overlapping minimal modules can be exponential, it is thus hard to compute from the minimal (α, β)-modules some hierarchy of modules. However, perhaps a better way to decompose a graph is to first compute the families of minimal modules with small values of α and β, and then consider a hierarchy of overlapping families.

Acknowledgements

The author wish to thank the reviewers for many valuable corrections and suggestions which greatly improved the paper.

References

1. Berger, E., Choromanski, K., Chudnovsky, M.: On the Erdős-Hajnal conjecture for six-vertex tournaments. European Journal of Combinatorics **75**, 113–122 (2019)
2. Bonnet, É., Geniet, C., Kim, E.J., Thomassé, S., Watrigant, R.: Twin-width II: small classes. In: Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA). pp. 1977–1996. SIAM (2021)
3. Bonnet, E., Kim, E.J., Thomassé, S., Watrigant, R.: Twin-width I: tractable FO model checking (2020)
4. Bonsma, P.: The complexity of the matching-cut problem for planar graphs and other graph classes. Journal of Graph Theory **62**(2), 109–126 (2009)
5. Borowiecki, M., Jesse-Józefczyk, K.: Matching cutsets in graphs of diameter 2. Theoretical Computer Science **407**(1-3), 574–582 (2008)
6. Bui-Xuan, B., Habib, M., Limouzy, V., de Montgolfier, F.: Algorithmic aspects of a general modular decomposition theory. Discrete Applied Mathematics **157**(9), 1993–2009 (2009)
7. Bui-Xuan, B., Habib, M., Limouzy, V., de Montgolfier, F.: Algorithmic aspects of a general modular decomposition theory. Discrete Applied Mathematics **157**(9), 1993–2009 (2009)
8. Bui-Xuan, B., Habib, M., Rao, M.: Tree-representation of set families and applications to combinatorial decompositions. European Journal of Combinatorics **33**(5), 688–711 (2012)
9. Chein, M., Habib, M., Maurer, M.C.: Partitive hypergraphs. Discrete Mathematics **37**(1), 35–50 (1981)
10. Chvátal, V.: Recognizing decomposable graphs. Journal of Graph Theory **8**(1), 51–53 (1984)
11. Corneil, D.G., Lerchs, H., Burlingham, L.S.: Complement reducible graphs. Discrete Applied Mathematics **3**(3), 163–174 (1981)
12. Cunningham, W., Edmonds, J.: A combinatorial decomposition theory. Canadian Journal of Mathematics **32**, 734–765 (1980)
13. Ehrenfeucht, A., McConnell, R.: A k-structure generalization of the theory of 2-structures. Theor. Comput. Sci. **132**, 209–227 (sep 1994)
14. Ehrenfeucht, A., Harju, T., Rozenberg, G.: Theory of 2-structures. In: Automata, Languages and Programming, 22nd International Colloquium, ICALP95, Szeged, Hungary, July 10-14, 1995, Proceedings. pp. 1–14 (1995)
15. Ehrenfeucht, A., Rozenberg, G.: Theory of 2-structures, part II: representation through labeled tree families. Theoretical Compututer Science **70**(3), 305–342 (1990)
16. Erdős, P., Hajnal, A.: Ramsey-type theorems. Discrete Applied Mathematics **25**(1), 37 – 52 (1989)
17. Fiala, J., Paulusma, D.: A complete complexity classification of the role assignment problem. Theoretical Compututer Science **349**(1), 67–81 (2005)
18. Fujishige, S.: Submodular Functions and Optimization. North-Holland (1991)
19. Gagneur, J., Krause, R., Bouwmeester, T., Casari, G.: Modular decomposition of protein-protein interaction networks. Genome Biology **5**(8) (2004)
20. Gallai, T.: Transitiv orientierbare Graphen. Acta Mathematica Academiae Scientiarum Hungaricae **18**, 25–66 (1967)
21. Graham, R.: On primitive graphs and optimal vertex assignments. Annals of the New York Academy of Sciences **175**, 170–186 (1970)

22. Habib, M., de Montgolfier, F., Mouatadid, L., Zou, M.: A general algorithmic scheme for combinatorial decompositions with application to modular decompositions of hypergraphs. Theor. Comput. Sci. **923**, 56–73 (2022)
23. Habib, M., Mouatadid, L., Zou, M.: Approximating modular decomposition is hard. In: Conference on Algorithms and Discrete Applied Mathematics. pp. 53–66. Springer (2020)
24. Habib, M., Paul, C.: A survey of the algorithmic aspects of modular decomposition. Computer Science Review **4**(1), 41–59 (2010)
25. Habib, M., Paul, C., Viennot, L.: Partition refinement techniques: An interesting algorithmic tool kit. International Journal of Foundations of Computer Science **10**(2), 147–170 (1999)
26. Hagauer, J., Imrich, W., Klavzar, S.: Recognizing median graphs in subquadratic time. Theor. Comput. Sci. **215**(1-2), 123–136 (1999)
27. Heggernes, P., Telle, J.A.: Partitioning graphs into generalized dominating sets. Nordic Journal of Computing **5**(2), 128–142 (1998)
28. Hsu, W.: Decomposition of perfect graphs. Journal of Combinatorial Theory, Series B **43**(1), 70–94 (1987)
29. Kratsch, D., Le, V.B.: Algorithms solving the matching cut problem. Theoretical Compututer Science **609**, 328–335 (2016)
30. Lai, P.L., Tan, J.J., Tsai, C.H., Hsu, L.H.: The diagnosability of the matching composition network under comparison diagnosis model. IEEE Transactions on Computers **53-8**, 1064–1069 (2004)
31. Möhring, R.H.: Algorithmic aspects of the substitution decomposition in optimization over relations, set systems and boolean functions. Annals of Operations Research **6**, 195–225 (1985)
32. Möhring, R., Radermacher, F.: Substitution decomposition for discrete structures and connections with combinatorial optimization. Annals of Discrete Mathematics **19**, 257–356 (1984)
33. Möhring, R.H.: Almost all comparability graphs are upo. Discrete Mathematics **50**, 63–70 (1984)
34. Moshi, A.M.: Matching cutsets in graphs. Journal of Graph Theory **13**(5), 527–536 (1989)
35. Mouatadid, L.: Mining frequent patterns on knowledge graphs. In: Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining. p. 1647. WSDM '22 (2022)
36. Mulder, M.: The structure of median graphs. Discrete Mathematics **24(2)**, 197–204 (1978)
37. Nabti, C., Seba, H.: Querying massive graph data: A compress and search approach. Future Generation Computer Systems **74**, 63–75 (2017)
38. Papadopoulos, C., Voglis, C.: Drawing graphs using modular decomposition. Journal of Graph Algorithms and Applications **11**(2), 481–511 (2007)
39. Patrignani, M., Pizzonia, M.: The complexity of the matching-cut problem. In: Graph-Theoretic Concepts in Computer Science, 27th International Workshop, WG 2001, Boltenhagen, Germany, June 14-16, 2001, Proceedings. pp. 284–295 (2001)
40. Rusu, I., Spinrad, J.P.: Forbidden subgraph decomposition. Discrete Mathematics **247**(1-3), 159–168 (2002)
41. Schmerl, J., Trotter, W.: Critically indecomposable partially ordered sets, graphs, tournaments and other binary relational structures. Discrete Mathematics **117**, 191–205 (1991)
42. Schrijver, A.: Combinatoriall Optimization, vol. 24. Springer Verlag (2002)

43. Seinsche, D.: On a property of the class of n-colorable graphs. Journal of Combinatorial Theory, Series B **16**, 191–193 (1974)
44. Serafino, P.: Speeding up graph clustering via modular decomposition based compression. In: Proceedings of the 28th Annual ACM Symposium on Applied Computing, SAC '13, Coimbra, Portugal, March 18-22, 2013. pp. 156–163 (2013)
45. Szemerédi, E.: On sets of integers containing no k elements in arithmetic progression. Acta Arithmetica **27**, 199–245 (1975)
46. Wang, G., Lin, C., Cheng, B., Fan, J., Fan, W.: Structure fault-tolerance of the generalized hypercube. The Computer Journal **62**(10), 1463–1476 (2019)