

Approximating Modular Decomposition is Hard

Michel Habib ^{*1,3}, Lalla Mouatadid², and Mengchuan Zou ^{**1,3}

¹ IRIF, UMR 8243 CNRS & Paris University, Paris, France

² Department of Computer Science, University of Toronto, Toronto, Ontario, Canada

³ Gang Project, INRIA Paris, France

Abstract. In order to understand underlying structural regularities in a graph, a basic and useful technique, known as modular decomposition, looks for subsets of vertices that have the **exact** same neighbourhood to the outside. These are known as modules and there exist linear-time algorithms to find them. This notion however is too strict, especially when dealing with graphs that arise from real world data. This is why it is important to relax this condition by allowing some noise in the data. However, generalizing modular decomposition is far from being obvious since most of the proposals lose the algebraic properties of modules and therefore most of the nice algorithmic consequences. In this paper we introduce the notion of ϵ -**module** which seems to be a good compromise that maintains some of the algebraic structure. Among the main results in the paper, we show that minimal ϵ -modules can be computed in polynomial time, on the other hand for maximal ϵ -modules it is already NP-hard to compute if a graph admits an 1-parallel decomposition, i.e. one step of decomposition of ϵ -module with $\epsilon = 1$.

1 Introduction

Introduced by Gallai in [13] to analyze the structure of comparability graphs, modular decomposition has been used and defined in many areas of discrete mathematics, including for graphs, 2-structures, automaton, partial orders, set systems, hypergraphs, clutters, matroids, boolean, and submodular functions [8,9,11,15], see [22] for a survey on modular decomposition. 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 yields a kind of partitive family [5,4], and therefore has a unique modular decomposition tree that can be computed efficiently.

Roughly speaking, elements of the module behave exactly the same with respect to the outside of the graph, and therefore a module can be contracted to a single element without losing information. This technique has been used to solve many optimization problems and has led to a number of elegant graph algorithms, see for instance [21]. On the other hand, direct applications of

* Corresponding author, habib@irif.fr

** This work is supported by the ANR-France Project Hosigra (ANR-17-CE40-0022)

modular decomposition in other areas include computational protein-protein interaction networks [12] and graph drawing [25], to name a few. More recently, new applications have appeared in the study of networks in social sciences [29], 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, in some recent experiments [24] in real data, many non-trivial modules were found in these graphs. How can we explain such a phenomena? It could be that the way the data is produced can generate modules, but it could also be because we reach some known regularities as predicted by Szemerédi’s regularity lemma [30]. In fact for every $\epsilon > 0$ Szemerédi’s lemma asserts that $\exists n_0$ such that all undirected graphs with more than n_0 vertices admits a ϵ -regular partition of the vertices. Such a partition is a kind of approximate modular decomposition. For graphs we now have linear-time algorithms to compute a modular decomposition tree, see [16]. In this paper we study a new generalization of modular decomposition, relaxing the strict neighbourhood condition of modules with a tolerance of some errors, i.e., some missing edges. The aims of this paper are twofold: first a theoretical study of an approximation of modular decomposition, and secondly a practical application for the computation of overlapping communities in bipartite graphs.

Organization of the Paper: We begin by giving necessary notations and a background on classical modular decomposition in Section 2, as well as illustrating some applications of ϵ -modular decomposition on various areas, on data compression and exact encodings for instance as well as in approximation algorithms. Section 3 introduces the notion of ϵ -modules and ϵ -modular decomposition, and their first basic properties. In Section 4, we give algorithmic results, in particular the computation of minimal ϵ -modules, as well as testing ϵ -primality. We then focus on two classes of graphs, bipartite graphs and 1-cographs (to be defined later) and conclude our discussion in the last section. In particular for bipartite graphs we can compute in $O(n^{2\cdot\epsilon}(n+m))$ a covering of the vertices using maximal ϵ -modules, in which two ϵ -modules can overlap on at most $2 \cdot \epsilon$ vertices. This can be of great help for community detection in bipartite graphs.

2 Approximations of modules

Let G be a simple, loop-free, undirected graph, with vertex set $V(G)$ and edge set $E(G)$, $n = |V(G)|$ and $m = |E(G)|$ are the number of vertices and edges of G respectively. For every $X \subseteq V(G)$, we denote by $G(X)$ the induced sub-graph generated by X . $N(v)$ denotes the neighbourhood of v and $\overline{N}(v)$ the non-neighbourhood, this notation could also be generalized to set of vertices, i.e. for $X \subseteq V(G)$, $N(X) = \{x \in V(G) \setminus X \text{ such that } \exists y \in X \text{ and } xy \in E(G)\}$ (resp. $\overline{N}(X) = \{x \in V(G) \setminus X \text{ such that } \forall y \in X \text{ and } xy \notin E(G)\}$). For $x, y \in V$, we call **false-twins** if $N(x) = N(y)$ and **true-twins** if $N(x) \cup \{x\} = N(y) \cup \{y\}$. A **Moore family** on a set X is a collection of subsets $S \subset X$ closed under intersection, and the set X itself.

Formally for an undirected graph G , a module $M \subseteq V(G)$ satisfies $\forall x, y \in M$, $N(x) \setminus M = N(y) \setminus M$. In other words, $V(G) \setminus M$ is partitioned into X, Y such

that there is a complete bipartite between M and X , and no edge between M and Y . For convenience let us denote X (resp. Y) by $N(M)$ (resp. $\overline{N(M)}$). It is easy to see that all vertices within a module are at least false twins.

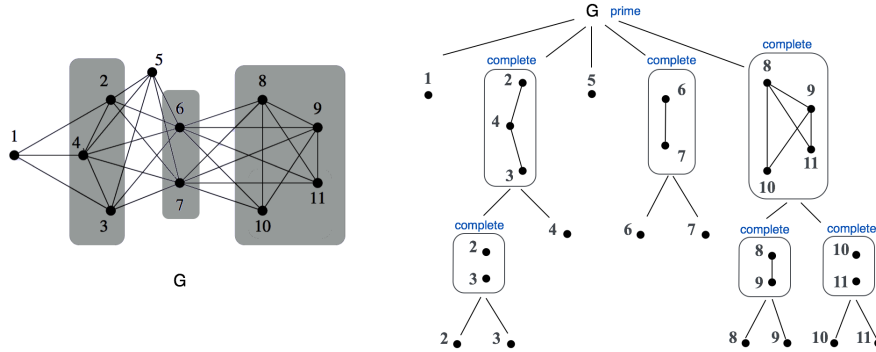


Fig. 1. Left, a graph with its maximal modules grouped. Right, its corresponding modular decomposition tree.

A single vertex $\{v\}$ and V are always modules, and called **trivial modules**. A graph that only has trivial modules is called a **prime graph**. By the Modular Decomposition Theorem [13,5], every graph admits a unique **modular decomposition tree**, in which a graph is decomposed via three types of internal nodes (operations): *parallel* (disjoint union) and *series* (connect every pair of nodes in disjoint sets X and Y), and *prime* nodes. The leaves represent the vertices of the graph, see Figure 1.

A graph is a **complement reducible** graph if there is no prime node in its decomposition tree [7]. Complement reducible graphs are also known as **cographs** in the literature, or P_4 -free graphs [28]. Cographs form a well studied graph class for which many classical *NP*-hard problems such as maximum clique, minimum coloring, maximum independent set, Hamiltonicity become tractable, see for instance [7].

Finding a non trivial tractable generalization of modules is not an easy task; indeed in trying to do so, we are faced with two main difficulties. The first one is to obtain a pseudo-generalization, for example if we change the definition of a module into: $\forall x, y \in M, Neighbour^*(x) \setminus M = Neighbour^*(y) \setminus M$, where $Neighbour^*(x)$ means something like “vertices at distance at most k ” or “joined by an odd path”, etc. As it turns out, in many of these cases, 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 [3]. The other one is NP-hardness. Consider the notion of *roles* defined in sociology, where two vertices play the same role in the social network if they have the same set of colours in the neighbourhood. If the colours of the vertices are given the problem is polynomially solvable, otherwise, this problem is a colouring one that is NP-hard to compute [10].

In this work, we will study two variations on the notion of modules both of which try to avoid these difficulties. Some aspects of them are polynomial to compute, and we believe they are worth studying further. We present the most promising one. Before we do so, we motivate this notion of ϵ -modules further, in two areas. The first one is in data compression and exact encodings, and the latter is its usefulness in approximation algorithms.

Before formally defining ϵ -modules, we want the reader to think of them as a subset of vertices that almost looks the same to the outside of the graph. Meaning, for all $x, y \in M$, an ϵ -module, $N(x) \setminus M$ and $N(y) \setminus M$ are the same with the exception of at most ϵ errors. Modular decomposition is often presented as an efficient way to encode a given graph. This property transmits to ϵ -modules. One can contract a non-trivial ϵ -module to a single vertex keeping almost the entirety of the original graph, and then recurse on the decomposition. To this end, let M be a non-trivial ϵ -module of G and X (resp. Y) be its neighbourhood (resp. non-neighbourhood). If we want an exact encoding of G , we can contract M to a unique vertex m connected to X , and not connected to Y , keep the subgraph $G(M)$ and keep track of the errors (i.e., the edges missing in the bipartite (M, X) and the edges that appear in the bipartite (M, Y)). Thus, this new exact encoding has at least $|M| \cdot (|X| - \epsilon) - 1$ edges fewer than the original encoding.

A second application of approximate modular decomposition is in approximation algorithms. Consider the classical colouring and independent set problems on cographs. Both algorithms use modular decomposition to give optimal linear time solutions to both problems. The way the algorithms work is by computing a modular decomposition tree – known as the cotree – and keeping track of the series or parallel internal nodes by scanning the tree from the leaves to the root. We later define extensions of a cograph and a cotree to an ϵ -cograph and ϵ -cotree, and show in particular for $\epsilon = 1$, we get a simple 2-approximation for 1-cographs for these two classical problems, just by summing over all ϵ errors. In particular, when $\epsilon = 1$, this means the neighbourhood of every pair $x, y \in M$ differs by at most one neighbour/non-neighbour with respect to the outside of M .

2.1 Subset families

Subset families: Two sets A and B *overlap* if $A \cap B \neq \emptyset$, $A \setminus B \neq \emptyset$, and $B \setminus A \neq \emptyset$. Let \mathcal{F} be a family of subsets of a ground set V . A set $S \in \mathcal{F}$ is called *strong* if $\forall S' \neq S \in \mathcal{F} : S$ does not overlap S' . Let Δ be the symmetric difference operation.

Definition 1. [5] A family of subsets \mathcal{F} over a ground set V is **partitive** if it satisfies the following properties: (i) \emptyset, V and all singletons $\{x\}$ for $x \in V$ belong to \mathcal{F} . (ii) $\forall A, B \in \mathcal{F}$ that overlap, $A \cap B, A \cup B, A \setminus B$ and $A \Delta B \in \mathcal{F}$

Partitive families play fundamental roles in combinatorial decompositions [5,4]. Every partitive family admits a unique decomposition tree, with two types of nodes: *complete* and *prime*. It is well known that the strong elements of \mathcal{F} form a tree ordered by the inclusion relation [5]. In this decomposition tree, every node corresponds to a set of the elements of the ground set V of \mathcal{F} , and the leaves of the tree are single elements of V .

3 ϵ -Modules and basic properties

One first idea to accept some errors is to say that at most k edges, for some fixed integer k , could be missing in the complete bipartite between M and $N(M)$, and symmetrically that at most k edges can exist between M and $\overline{N(M)}$. But doing so we loose most of the nice algebraic properties of modules of graphs which yield partitive families. Furthermore most algorithms for modular decomposition are based on these algebraic properties [5].

Another natural idea is to relax the condition on the complete bipartite between M and $N(M)$, for example asking for a graph that does not contain any $2K_2$. Unfortunately as shown in [27] to test whether a given graph admits such a decomposition is NP-complete. In fact they studied a generalized join decomposition solving a question asked in [18] studying perfection. This is why the following generalization of module defined for any integer ϵ , seems to be a good compromise⁴.

Definition 2. *A subset $M \subseteq V(G)$ is an ϵ -module if $\forall x \in V(G) \setminus M$, either $|M \cap N(x)| \leq \epsilon$ or $|M \cap N(x)| \geq |M| - \epsilon$*

In other words, we tolerate ϵ edges of errors per node outside the ϵ -module, and not ϵ errors per module. It should be noticed that with $\epsilon = 0$, we recover the usual definition of modules [16], i.e., $\forall x \in V(G) \setminus M$, either $M \cap N(x) = \emptyset$ or $M \cap N(x) = M$. Necessarily we will only consider $\epsilon < |V(G)| - 1$.

Let us consider the first simple properties yielded by this definition.

Proposition 1. *If M is an ϵ -module for G , then*

- (a) *M is an σ -module for G , for every $\epsilon \leq \sigma$.*
- (b) *M is an ϵ -module for \overline{G} .*
- (c) *M is an ϵ -module for every induced subgraph H of G such that $M \subseteq V(H)$.*
- (d) *every ϵ -module of $G(M)$ is an ϵ -module of G .*

Definition 3. ϵ -neighbourhoods. *For $A \subseteq V(G)$, let us denote by $N_\epsilon(A)$ (resp. $\overline{N}_\epsilon(A)$) the vertices of $V(G) \setminus A$, that are connected (resp. not connected) to M except for at most ϵ vertices. Similarly $S_\epsilon(A) = \{x \in V(G) \setminus A \text{ such that } \epsilon < |N(x) \cap A| < |A| - \epsilon\}$. $S_\epsilon(A)$ is called the set of ϵ -splitters of A .*

Equivalently a module can therefore be defined as a subset of vertices having no ϵ -splitter.

Lemma 1. *Some easy facts, for $A \subseteq V(G)$.*

- (i) *If $2 \cdot \epsilon + 1 \leq |A|$ then $N_\epsilon(A) \cap \overline{N}_\epsilon(A) = \emptyset$.*
- (ii) *If $|A| \leq 2 \cdot \epsilon + 1$ then $S_\epsilon(A) = \emptyset$.*
- (iii) *If $|A| = 2 \cdot \epsilon + 1$ then $N_\epsilon(A)$ and $\overline{N}_\epsilon(A)$ partition $V(G) \setminus A$.*
- (iv) *If $|A| < 2 \cdot \epsilon$ then $N_\epsilon(A) = \overline{N}_\epsilon(A)$.*

⁴ We use ϵ to denote small error, despite being greater than 1.

So the subsets of vertices having size $2 \cdot \epsilon + 1$ seem to be crucial to study this new decomposition. If A is such a set, for every $z \notin A$ either $z \in N_\epsilon(A)$ or $z \in \overline{N_\epsilon(A)}$, but not both.

Lemma 2. *If s is a ϵ -splitter for a set A , then s is also a ϵ -splitter for every set $B \supseteq A$ such that $s \notin B$.*

Proof. Since $\epsilon < |A \cap N(x)|$ and $N(x) \cap A \subseteq N(x) \cap B$ we have $\epsilon < |B \cap N(x)|$. $|A \cap N(x)| < |A| - \epsilon$ is equivalent to $|A \setminus N(x)| > \epsilon$. But $A \setminus N(x) \subseteq B \setminus N(x)$ implies $|B \setminus N(x)| > \epsilon$. So $\epsilon < |B \cap N(x)| < |B| - \epsilon$. \square

Theorem 1. *The family of ϵ -modules of a graph satisfies:*

- (i) $V(G)$ is an ϵ -module and $\forall A \subseteq V(G)$ such that $|A| \leq 2 \cdot \epsilon + 1$ are ϵ -modules.
- (ii) $\forall A, B \subseteq V(G)$ ϵ -modules then, $A \cap B$ is an ϵ -module and for the subsets $A \setminus B$ and $B \setminus A$ their ϵ -splitters can only belong to $A \cap B$.

Proof. (i) By definition $V(G)$ has no ϵ -splitter. Let $A \subseteq V(G)$, such that $|A| \leq 2 \cdot \epsilon + 1$ and let $x \in V(G) \setminus A$.

Suppose $|N(x) \cap A| = k > \epsilon$ but since $|A| \leq 2 \cdot \epsilon + 1$, $\epsilon \geq |A| - \epsilon - 1$

Therefore: $|N(x) \cap A| = k \geq |A| - \epsilon$ and A has no ϵ -splitter.

- (ii) First we notice that if $A, B \subseteq V(G)$ are 2 trivial modules, obviously $A \cap B$, $A \setminus B$ and $B \setminus A$ are trivial ϵ -modules.

Let $A, B \subseteq V(G)$ be two non trivial ϵ -modules. If $A \cap B$ has an ϵ -splitter outside of $A \cup B$ then using Lemma 2 also A, B would have an ϵ -splitter, a contradiction. Suppose now that $A \cap B$ admits an ϵ -splitter in $B \setminus A$ but then with the same Lemma we know that A would have an ϵ -splitter. Therefore $A \cap B$ is an ϵ -module. Let us now consider $A \setminus B$, if admits an ϵ -splitter in $B \setminus A$, using again Lemma 2, A would have a ϵ -splitter too. Similarly if the ϵ -splitter is outside $A \cup B$. Then the only potential ϵ -splitters for $A \setminus B$ and $B \setminus A$ are in $A \cap B$. \square

Corollary 1. *A graph G with $|V(G)| \leq 2 \cdot \epsilon + 2$ admits only trivial modules.*

By convention we will call such a graph ϵ -**degenerate** in order to distinguish with really ϵ -prime graphs.

Corollary 2. *If A, B are overlapping minimal ϵ -modules then $A \cap B$ is a trivial ϵ -module.*

We know then the ϵ -modules generate a Moore family of subsets worth studying. For usual modules as can be seen in [5,16], $A \cup B$, $B \setminus A$ and $A \setminus B$ are also modules. Unfortunately this does not always hold for ϵ -modules. Moreover we cannot bound the error as can be seen the next proposition.

Proposition 2. *Let $A, B \subseteq V(G)$ be two non trivial ϵ -modules, then*

1. there could be $c = \Omega(\min(|A|, |B|))$, s.t. $A \cup B$ is not an ϵ -module, $\forall \epsilon \leq c$.
2. there could be $c = \Omega(n)$, s.t. $A \setminus B$ is not an ϵ -module, for all $\epsilon \leq c$.

In fact we can prove a weaker result.

Theorem 2. *Let $A, B \subseteq V(G)$ be two non trivial overlapping ϵ -modules, if $|A \cap B| \geq 2\epsilon + 1$ then $A \cup B, A\Delta B$ (i.e., symmetric difference) are 2ϵ -modules.*

Proof. Let $z \in V(G) \setminus B$. Since B is an ϵ -module then $S_\epsilon(B) = \emptyset$, since B is non trivial, $|B| \geq 2 \cdot \epsilon + 2$, therefore $N_\epsilon(B)$ and $\overline{\epsilon N_\epsilon(B)}$ partition $V(G) \setminus B$, using Lemma 1. Suppose $z \in N_\epsilon(B)$, z has at most ϵ non neighbors in $A \cap B$. Therefore it has at least $\epsilon + 1$ neighbors in $A \cap B$, therefore $z \in N_\epsilon(A)$. For $A \cup B$ in the worst case z has at most ϵ non-neighbors in $A \setminus B$ and at most ϵ non-neighbors in $B \setminus A$. Therefore $A \cup B$ is a 2ϵ -module. For $A\Delta B$, the worst case is obtained when a given vertex $z \in A \cap B$ has ϵ errors in $A \setminus B$ and ϵ errors in $B \setminus A$. Therefore $A\Delta B$ is a 2ϵ -module. \square

Theorem 1 allows us to define a graph convexity. Since the family of ϵ -modules is closed under intersection, it yields a graph convexity and we can compute the minimal under inclusion ϵ -module $M(A)$ that contains a given set A , with strictly more that $2 \cdot \epsilon + 1$ elements, computing a **modular closure** via ϵ -splitters.

3.1 A Symmetric Variation of ϵ -Modules

One could want to restrict the definition of the ϵ -modules in a symmetric way. Here symmetric means that the condition is applied symmetrically on the vertices of the ϵ -module M and on the vertices outside, i.e., $V(G) \setminus M$.

Definition 4. *An ϵ -module M is **symmetric** if every $x \in M$ is adjacent (resp. non-adjacent) to all vertices in $N(M)$ (resp. $\overline{N(M)}$) except for at most ϵ vertices.*

In other words for $\epsilon = 1$, in the bipartite $M, N(M)$ only a matching is missing. It is a restriction of the ϵ -modules and all the previous results could be generated similarly for symmetric ϵ -modules.

Proposition 3. *If $\mathcal{P} = \{V_1, \dots, V_k\}$ is a partition of $V(G)$ into ϵ -modules, then the V_i 's are necessarily symmetric ϵ -modules.*

With this definition in mind, we present extensions of the series and parallel nodes in the classical setting, as well as introduce a new graph class we call *1-cographs*, the definition of which we present below.

Using Proposition 1(d) and mimicking the case of modular decomposition we may define an ϵ -tree decomposition as follows.

Definition 5. *An ϵ -tree decomposition is a tree whose nodes are labelled with ϵ -modules ordered by inclusion with 4 types of nodes ϵ -series, ϵ -parallel, ϵ -prime and ϵ -degenerate. Each level of the tree corresponds to a partition of $V(G)$, starting with $\{V(G)\}$ at the root and the leaves correspond to a partition of $V(G)$ into ϵ -degenerate nodes.*

For standard modular decomposition the notion of strong modules as modules that do not overlap with any other is central. For ϵ -modular decomposition we can observe that there are no strong modules other than V and $\{v\}, v \in V$ that are strong ϵ -modules. The reason is that, for $\epsilon \geq 1$, any subset of vertices of size 2 is a trivial ϵ -module, then assume there is a classical strong module $V_1 \neq V$, $|V_1| > 1$, then take any vertex $v \in V_1$ and any vertex $u \in V \setminus V_1$, then $\{u, v\}$ is a ϵ -module and overlapping with V_1 .

3.2 ϵ -Series and ϵ -Parallel Operations

Definition 6. For a graph G with $|V(G)| \geq 2\epsilon + 3$, we say that G admits an ϵ -series (resp. ϵ -parallel) decomposition if there exists a partition of $V(G)$, $\mathcal{P} = \{V_1, \dots, V_k\}$ such that: $\forall i, 1 \leq i \leq k, |V_i| \geq 2\epsilon + 1$ and $\forall x \in V_i$ and for every $j \neq i, x$ is adjacent (resp. non-adjacent) to all vertices of V_j with perhaps ϵ errors.

Using Proposition 3, all the V_i 's are necessarily symmetric ϵ -modules. Furthermore in such cases every union of V_i 's are also symmetric ϵ -modules. Fortunately with $\epsilon = 1$ the problem of recognizing if a graph admits an 1-parallel decomposition corresponds to a nice combinatorial problem first studied in [14]. The complexity of this problem known as finding a matching cut-set is now well-known [6,23,1] and therefore we have:

Theorem 3. Finding if a graph admits an 1-parallel decomposition is NP-hard.

Proof. Let G be a graph with minimum degree 3, and suppose that it admits an 1-parallel decomposition into V_1, \dots, V_k . Necessarily $\forall i, |V_i| > 1$, since there is no pending vertex. Therefore $\{V_1, \cup_{1 < i \leq k} V_i\}$ is a matching cut set of G . So using [6], deciding if a graph admits 1-parallel decomposition is NP-complete. \square

Definition 7. An ϵ -cograph is a graph that is decomposable with respect to ϵ -series, ϵ -parallel decompositions until we reach only degenerate subgraphs.

Using this definition above, it is clear that cographs are precisely the 0-cographs and let us call ϵ -cotree and ϵ -modular decomposition the corresponding tree and decomposition of an ϵ -cograph.

Proposition 4. A graph is an ϵ -cograph iff it admits a ϵ -cotree using only ϵ -series and ϵ -parallel internal nodes.

Proof. Suppose that G admits a ϵ -series composition with a partition $\mathcal{P} = \{V_1, \dots, V_k\}$. First we must notice that these two operations are exclusive. It is the case since every part has at least $2 \cdot \epsilon + 1$ vertices, we cannot have 2 parts V_i, V_j both ϵ -connected and ϵ -disconnected. Therefore we start a ϵ -cotree starting with a node labelled ϵ -series and recurse on all the subgraphs $G(V_i)$ using proposition d. \square

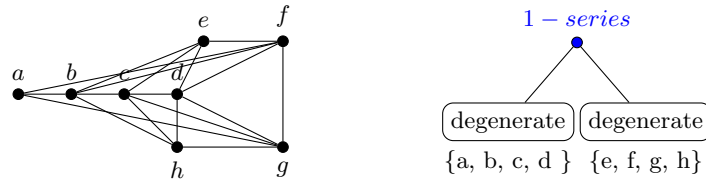


Fig. 2. 1-MD(H): A 1-cotree of a 1-cograph graph H . Notice that H is not a cograph since it contains 2 induced P_4 's, namely $H(\{a, b, c, d\})$ and $H(\{e, f, g, h\})$.

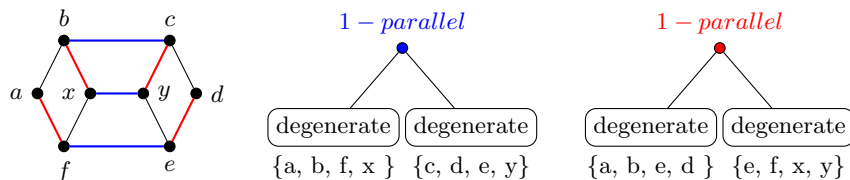


Fig. 3. This 1-cograph G admits 2 different 1-cotrees, the internal nodes have the same label but the partitions of $V(G)$ induced by the leaves are not the same.

Let us consider now the 2 examples described in Figures 2 and 3. The first one shows a 1-cograph H that admits a unique 1-cotree. The second one shows a 1-cograph G that admits 2 different legitimate 1-cotrees. Moreover by substituting in each vertex of G a graph isomorphic to G and if we repeat this process we can build a 1-cograph which admit **exponentially many** different legitimate 1-cotrees. At this particular time regarding for a graph the existence of ϵ -tree decomposition is not clear and as shown with ϵ -cographs we cannot ask for a unique one if it exists.

Unfortunately, it turns out, as one might expect, that finding this matching cutset is an NP-complete problem, as was shown by Chvatal in [6]. In the same work, Chvatal showed in particular that the problem is NP-hard on graphs with maximum degree four, and polynomial on graphs with maximum degree three.

Furthermore, it was shown that computing a matching cutsets in the following graph classes is polynomial: for graphs with max degree three [6], for weakly chordal graphs and line-graphs [23], for Series Parallel graphs [26], claw-free graphs and graphs with bounded clique width, as well as graphs with bounded treewidth [1], graphs with diameter 2 [2]. and for $(K_{1,4}, K_{1,4} + e)$ -free graphs [20].

Therefore, to check if any of these graphs are 1-cographs, it suffices to run the corresponding matching cutset algorithms on either the graph or its complement.

But we conjecture that even 1-cographs that can be decomposed into *exactly* two cographs are hard to recognize in the general case.

4 Computing the minimal ϵ -modules

Despite the negative results of the previous sections, we shall now examine how to compute all minimal ϵ -modules in polynomial time. As seen previously non trivial ϵ -module have strictly more than $2 \cdot \epsilon + 1$ elements. Since ϵ -module family is closed under intersection, it yields a graph convexity and we can compute the minimal under inclusion ϵ -module $M(A)$ that contains a given set A , with strictly

more than $2 \cdot \epsilon + 1$ elements, computing a **modular closure** via ϵ -splitters. In fact we built a series of subsets M_i starting with $M_0 = A$, and satisfying $M_i \subseteq M_{i+1}$.

Algorithm 1: Computing minimal ϵ -modules.

Input: a graph G and $A \subseteq V(G)$ with $|A| \geq 2 \cdot \epsilon + 2$.
Output: $M(A)$ the minimal ϵ -module that contains A

- 1 $M_0 \leftarrow A, i \leftarrow 0$;
- 2 $S \leftarrow \{x \in V(G) \setminus M_0 \text{ such that } \epsilon < |N(x) \cap M_0| < |M_0| - \epsilon\}$;
- 3 **while** $S \neq \emptyset$ **do**
- 4 $i \leftarrow i + 1$;
- 5 $M_i \leftarrow M_{i-1} \cup S$;
- 6 $S \leftarrow \{x \in V(G) \setminus M_i \text{ such that } \epsilon < |N(x) \cap M_i| < |M_i| - \epsilon\}$;
- 7 $M(A) \leftarrow M_i$;

Proposition 5. *Algorithm 1 computes the minimal ϵ -module that contains A .*

Proof. If A is an ϵ -module, then at line 2, $S = \emptyset$, else all the elements of S have to be added to A . In other words, using Lemma 2 there is no ϵ -module M such that $A \subsetneq M \subsetneq A \cup S$. At the end of the While loop either $M_i = V(G)$ or we have found a non trivial ϵ -module. \square

Theorem 4. *Algorithm 1 can be implemented in $O(m + n)$.*

Proof. In fact we can implement it as a kind of graph search as follows.

Algorithm 2: Computing minimal ϵ -modules.

Input: a graph G and $A \subseteq V(G)$ with $|A| \geq 2 \cdot \epsilon + 2$.
Output: $M(A)$ the minimal ϵ -module that contains A

- 1 $OPEN \leftarrow A$;
- 2 $M(A) \leftarrow \emptyset$;
- 3 $\forall u \in V(G), CLOSED(u) \leftarrow FALSE; edge(u) \leftarrow 0; nonedge(u) \leftarrow 0$;
- 4 **while** $OPEN \neq \emptyset$ **do**
- 5 $z \leftarrow Choice(OPEN)$, Delete z from $OPEN$;
- 6 Add z to $M(A)$;
- 7 $CLOSED(z) \leftarrow TRUE$;
- 8 \forall neighbours u of z
- 9 **if** $CLOSED(u) = FALSE$ and $u \notin M(A)$ **then**
- 10 $edge(u) \leftarrow edge(u) + 1$;
- 11 **if** $\epsilon < edge(u)$ and $\epsilon < nonedge(u)$ **then**
- 12 Add u to $OPEN$
- 13 \forall non neighbours v of z
- 14 **if** $CLOSED(v) = FALSE$ and $v \notin M(A)$ **then**
- 15 $nonedge(v) \leftarrow nonedge(v) + 1$;
- 16 **if** $\epsilon < edge(u)$ and $\epsilon < nonedge(u)$ **then**
- 17 Add v to $OPEN$

At the end of this algorithm 2 the set $M(A)$ contains a minimal ϵ -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. But if we use a partition refinement technique as defined in [17], starting with a partition of the vertices in $\{A, V(G) - A\}$, then we keep in the a same part $B(i, j)$ vertices x, y , such that $edge(x) = edge(y) = i$ and $nonedge(x) = nonedge(y) = j$. Then when visiting a vertex it suffices for each part $B(i, j)$ of the current partition to compute $B'(i + 1, j) = B(i, j) \cap N(z)$ and $B''(i, j + 1) = B(i, j) - N(z)$, which can be done in $O(|N(z)|)$. It should be noticed that the parts need not to be sorted in the current partition and we may have different parts with the same (edge, nonedge) values. Therefore can be implemented in $O(m + n)$. \square

Theorem 5. *Using Algorithm 1, one can compute all minimal non-trivial ϵ -modules in $O(m \cdot n^{2 \cdot \epsilon + 1})$.*

Proof. It suffices to use Algorithm 1 starting from every subset with $2 \cdot \epsilon + 2$ vertices. There exist $O(n^{2 \cdot \epsilon + 2})$ such subsets. And therefore this yields an algorithm in $O(m \cdot n^{2 \cdot \epsilon + 2})$. But we can do all the partition refinements in the whole, using the neighbourhood of one vertex only once. Since a vertex may belong to at most $n^{2 \cdot \epsilon + 1}$ parts, it yields an algorithm working in $O(m \cdot n^{2 \cdot \epsilon + 1})$. \square

If we consider the $\epsilon = 0$ case, this gives an implementation of the algorithm in [19] which also computes all minimal modules in $O(m \cdot n)$, to be compared to the original one in $O(n^4)$.

Corollary 3. *Using theorem 5, one can compute a covering of $V(G)$ with an overlapping family of minimal ϵ -modules in $O(m \cdot n^{2 \cdot \epsilon + 1})$ and for any two members of the covering their overlapping is bounded by $2 \cdot \epsilon + 1$.*

Proof. Using theorem 5, we can compute an overlapping family of minimal ϵ -modules in $O(m \cdot n^{2 \cdot \epsilon + 1})$. Perhaps it is not a covering of $V(G)$, since some vertices may not belong to any minimal non-trivial ϵ -module. To obtain a covering we simply add as singletons the remaining vertices. \square

This could be very interesting if we are looking for overlapping communities in social networks, the overlapping being bounded by $2 \cdot \epsilon + 1$.

To go a step further we can use Theorem 2 and merge every pair A, B of ϵ -modules such that $|A \cap B| \geq 2 \cdot \epsilon + 1$, either keeping $A \cup B$ as a $2 \cdot \epsilon$ -module or compute $M(A \cup B)$ the minimal ϵ -module that contains $A \cup B$. But this depends on the structure of the maximal ϵ -modules, and unfortunately we do not know yet under what conditions there exists a unique partition into maximal ϵ -modules.

Corollary 4. *Checking if a graph is ϵ -prime can be done in $O(m \cdot n^{2 \cdot \epsilon + 1})$.*

Proof. It suffices to test whether for every set with $2 \cdot \epsilon + 2$ vertices if its closure is equal to $V(G)$. So either we find a non-trivial ϵ -module or the graph is ϵ -prime. Since every non-trivial ϵ -module necessarily contains one of the sets with $2 \cdot \epsilon + 2$ vertices. \square

Corollary 5. *Finding for a graph G the smallest ϵ such that G has an ϵ -module can be done in $O(\log n \cdot m \cdot n^{2 \cdot \epsilon + 1})$*

Proof. To find such an ϵ we can use the above primality test in a dichotomic way, just adding a $\log n$ factor to the complexity. \square

5 The Bipartite Case

Let us consider now a bipartite graph $G = (X, Y, E(G))$. Unfortunately the ϵ -modules can be made up with vertices of both X, Y . But in some applications we are forced to consider X and Y separately. As for example in the case where X is a set of customers (resp. DNA sequences) and Y a set of products (resp. organisms), usually one wants to find regularities on each side of the bipartite graph. Let $\mathcal{F}_\epsilon(X) = \{M \mid \epsilon\text{-module of } G \text{ such that } M \subseteq X\}$. It should be noticed that X is not always an ϵ -module of G .

Proposition 6. $\forall A, B \in \mathcal{F}_\epsilon(X), A \cap B, A \setminus B, B \setminus A \in \mathcal{F}_\epsilon(X)$.

Proof. Using Theorem 1, the only ϵ -splitters of the sets $A \setminus B$ and $B \setminus A$ must belong to $A \cap B$. But since $A, B \subseteq X$, which is an independent set, it is impossible. \square

As a consequence, using a notion of false ϵ -twins, we obtain.

Theorem 6. *For a bipartite graph $G = (X, Y, E(G))$, the maximal elements of $\mathcal{F}_\epsilon(X)$ can be computed in $O(n^{2 \cdot \epsilon}(n + m))$.*

It should be noticed that these maximal elements of $\mathcal{F}_\epsilon(X)$ may overlap, but the overlap is bounded by $2 \cdot \epsilon$. Furthermore the experimentation on real data has still to be done to evaluate the quality of the covering obtained.

5.1 Conclusions and perspectives

The polynomial algorithms presented here have to be improved. Since it is hard to compute from the minimal ϵ -modules some hierarchy of modules – because we may have to consider an exponential number of unions of overlapping minimal ones – perhaps a good way to analyze a graph is to compute the families of minimal ones with $\epsilon = 1, 2, 3 \dots$ and consider a hierarchy of overlapping families.

This notion of ϵ -modules yields many interesting questions both theoretical and practical. As for example for $\epsilon = 1$ to characterize 1-cographs or graphs that admits a 1-modular decomposition tree. The study of 1-primers is also worth to be done. On the other hand are there many ϵ -modules in real data? A natural consequence of this work is to extend the Courcelle’s cliquewidth parameter into an ϵ -cliquewidth and similarly to define an ϵ -split operation in graphs.

Acknowledgments. The authors wish to thank anonymous reviewers for their helpful comments.

References

1. Bonsma, P.: The complexity of the matching-cut problem for planar graphs and other graph classes. *JGT* **62**(2), 109–126 (2009)
2. Borowiecki, M., Jesse-Józefczyk, K.: Matching cutsets in graphs of diameter 2. *Theoretical Computer Science* **407**(1-3), 574–582 (2008)
3. 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)
4. Bui-Xuan, B., Habib, M., Rao, M.: Tree-representation of set families and applications to combinatorial decompositions. *Eur. J. Comb.* **33**(5), 688–711 (2012)
5. Chein, M., Habib, M., Maurer, M.C.: Partitive hypergraphs. *Discrete mathematics* **37**(1), 35–50 (1981)
6. Chvátal, V.: Recognizing decomposable graphs. *JGT* **8**(1), 51–53 (1984)
7. Corneil, D.G., Lerchs, H., Burlingham, L.S.: Complement reducible graphs. *Discrete Applied Mathematics* **3**(3), 163–174 (1981)
8. 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)
9. Ehrenfeucht, A., Rozenberg, G.: Theory of 2-structures, part II: representation through labeled tree families. *Theor. Comput. Sci.* **70**(3), 305–342 (1990)
10. Fiala, J., Paulusma, D.: A complete complexity classification of the role assignment problem. *Theor. Comput. Sci.* **349**(1), 67–81 (2005)
11. Fujishige, S.: *Submodular Functions and Optimization.* North-Holland (1991)
12. Gagneur, J., Krause, R., Bouwmeester, T., Casari, G.: Modular decomposition of protein-protein interaction networks. *Genome Biology* **5**(8) (2004)
13. Gallai, T.: Transitiv orientierbare Graphen. *Acta Mathematica Academiae Scientiarum Hungaricae* **18**, 25–66 (1967)
14. Graham, R.: On primitive graphs and optimal vertex assignments. *Ann. N.Y. Acad. Sci.* **175**, 170–186 (1970)
15. Habib, M., de Montgolfier, F., Mouatadid, L., Zou, M.: A general algorithmic scheme for modular decompositions of hypergraphs and applications pp. 251–264 (2019)
16. Habib, M., Paul, C.: A survey of the algorithmic aspects of modular decomposition. *Computer Science Review* **4**(1), 41–59 (2010)
17. Habib, M., Paul, C., Viennot, L.: Partition refinement techniques: An interesting algorithmic tool kit. *Int. J. Found. Comput. Sci.* **10**(2), 147–170 (1999)
18. Hsu, W.: Decomposition of perfect graphs. *JCTB* **43**(1), 70–94 (1987)
19. James, L.O., Stanton, R.G., Cowan, D.D.: Graph decomposition for undirected graphs. In: *Proc. 3rd Southeastern International Conference on Combinatorics, Graph Theory, and Computing* (Florida Atlantic Univ., Boca Raton, Flo. pp. 281–290 (1972)
20. Kratsch, D., Le, V.B.: Algorithms solving the matching cut problem. *Theor. Comput. Sci.* **609**, 328–335 (2016)
21. 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)
22. Möhring, R., Radermacher, F.: Substitution decomposition for discrete structures and connections with combinatorial optimization. *Annals of Discrete Mathematics* **19**, 257–356 (1984)

23. Moshi, A.M.: Matching cutsets in graphs. *JGT* **13**(5), 527–536 (1989)
24. Nabti, C., Seba, H.: Querying massive graph data: A compress and search approach. *Future Generation Comp. Syst.* **74**, 63–75 (2017)
25. Papadopoulos, C., Voglis, C.: Drawing graphs using modular decomposition. *J. Graph Algorithms Appl.* **11**(2), 481–511 (2007)
26. 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)
27. Rusu, I., Spinrad, J.P.: Forbidden subgraph decomposition. *Discrete Mathematics* **247**(1-3), 159–168 (2002)
28. Seinsche, D.: On a property of the class of n -colorable graphs. *JCTB* **16**, 191–193 (1974)
29. 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)
30. Szemerédi, E.: On sets of integers containing no k elements in arithmetic progression. *Acta Arithmetica* **27**, 199–245 (1975)