

Cograph editing: Merging modules is equivalent to editing P_4 s*

Adrian Fritz 

*Computational Biology of Infection Research, Helmholtz Centre for Infection Research,
Inhoffenstraße 7, D-38124 Braunschweig*

Marc Hellmuth 

*School of Computing, University of Leeds,
EC Stoner Building, Leeds LS2 9JT, England*

Peter F. Stadler † 

*Bioinformatics Group, Department of Computer Science, Universität Leipzig,
Härtelstrasse 16-18, D-04107 Leipzig, Germany*

Nicolas Wieseke 

*Swarm Intelligence and Complex Systems Group, Department of Computer Science,
Leipzig University, Augustusplatz 10, D-04109 Leipzig, Germany*

Received 23 May 2018, accepted 10 September 2019, published online 27 March 2020

Abstract

The modular decomposition of a graph $G = (V, E)$ does not contain prime modules if and only if G is a cograph, that is, if no quadruple of vertices induces a simple connected path P_4 . The cograph editing problem consists in inserting into and deleting from G a set F of edges so that $H = (V, E \triangle F)$ is a cograph and $|F|$ is minimum. This NP-hard combinatorial optimization problem has recently found applications, e.g., in the context of phylogenetics. Efficient heuristics are hence of practical importance. The simple characterization of cographs in terms of their modular decomposition suggests that instead of editing G one could operate directly on the modular decomposition. We show here that editing the

*We thank the anonymous referees for their helpful comments. This work was funded in part by the Deutsche Forschungsgemeinschaft (DFG), Proj. Nr. 432974470 (to PFS).

†PFS is also affiliated with the Interdisciplinary Center for Bioinformatics, the German Centre for Integrative Biodiversity Research (iDiv) Halle-Jena-Leipzig, the Competence Center for Scalable Data Services and Solutions Dresden-Leipzig, the Leipzig Research Center for Civilization Diseases, and the Centre for Biotechnology and Biomedicine at Leipzig University; the Max Planck Institute for Mathematics in the Sciences, Leipzig, Germany; the Institute for Theoretical Chemistry, University of Vienna, Vienna, Austria; the Center of noncoding RNA in Health and Technology (RTH) at the University of Copenhagen; Facultad de Ciencias of the National University of Colombia in Bogotá, Colombia; and the Santa Fe Institute, Santa Fe, NM.

induced P_4 s is equivalent to resolving prime modules by means of a suitable defined merge operation on the submodules. Moreover, we characterize so-called module-preserving edit sets and demonstrate that optimal pairwise sequences of module-preserving edit sets exist for every non-cograph. This eventually leads to an exact algorithm for the cograph editing problem as well as fixed-parameter tractable (FPT) results when cograph editing is parameterized by the so-called modular-width. In addition, we provide two heuristics with time complexity $O(|V|^3)$, resp., $O(|V|^2)$.

Keywords: Cograph editing, modular decomposition, module merge, prime modules, P_4 .

Math. Subj. Class. (2020): 05C75, 05C05, 92B10

1 Introduction

Cographs are of particular interest in computer science because many combinatorial optimization problems that are NP-complete for arbitrary graphs become polynomial-time solvable on cographs [4, 8, 20]. This makes them an attractive starting point for constructing heuristics that are exact on cographs and yield approximate solutions on other graphs. In this context it is of considerable practical interest to determine “how close” an input graph is to a cograph.

An independent motivation recently arose in biology, more precisely in molecular phylogenetics [14, 21, 35, 36, 37, 47]. In particular, *orthology*, a key concept in evolutionary biology in phylogenetics, is intimately tied to cographs [35]. Two genes in a pair of related species are said to be orthologous if their last common ancestor was a speciation event. The orthology relation on a set of genes forms a cograph [30], see [33] for a detailed discussion and [21, 22, 23, 31, 47] for generalizations of these concepts. This relation can be estimated directly from biological sequence data, albeit in a necessarily noisy form. Correcting such an initial estimate to the nearest cograph thus has recently become a computational problem of considerable practical interest in computational biology [35]. However, the (decision version of the) problem to edit a given graph with a minimum number of edits into a cograph is NP-complete [32, 34, 38, 39].

As noted already in [7], the input for several combinatorial optimization problems, such as exam scheduling or several variants of clustering problems, is naturally expected to have few induced paths on four vertices (P_4 s). Since graphs without an induced P_4 are exactly the cographs, available cograph editing algorithms focus on efficiently removing P_4 s, see e.g. [16, 24, 25, 38, 39, 53]. The FPT-algorithm introduced in [38, 39] takes as input a graph that is first edited to a so-called P_4 -sparse graph and then to a cograph. The basic strategy is to destroy the P_4 s in the subgraphs by branching into six cases that eventually leads to an $O(4.612^k |V|^{9/2})$ -time algorithm, where k is the number of required edits. Algorithms that compute the kernel of the (parameterized) cograph editing problem [24, 25] as well as the exact $O(3^{|V|} |V|)$ -time algorithm [53] use the modular-decomposition tree as a guide to locate the forbidden P_4 s using the fact that these are associated with prime modules. Nevertheless, the basic operation in all of these algorithms is still the direct destruction of the P_4 s.

Cographs are recursively defined as follows: K_1 is a cograph, the disjoint union of

E-mail addresses: adrian.fritz@helmholtz-hzi.de (Adrian Fritz), mhellmuth@mailbox.org (Marc Hellmuth), studla@bioinf.uni-leipzig.de (Peter F. Stadler), wieseke@informatik.uni-leipzig.de (Nicolas Wieseke)

cographs is a cograph, and the join of cographs is a cograph. This recursive definition associates a vertex labeled tree, the cotree, with each cograph, where a vertex label “0” denotes a disjoint union, while “1” indicates the join of the children is formed. It has been shown in [7] that each cograph has a unique cotree and conversely, every tree whose interior vertices are labeled alternatingly defined a unique cograph. A simple recognition algorithm starts with an input graph G . If G is connected, then a node labeled “1” is inserted into the tree, the complement graph \overline{G} is formed and the algorithm proceeds recursively on the connected components of \overline{G} . If G is not connected, the tree-node is labeled “0”, and the algorithm recurses on the components of G . If both G and \overline{G} are connected, then G is not a cograph, and the algorithm terminates with a negative answer. A natural heuristic for cograph editing proceeds by finding a minimal cut in G or \overline{G} , removes the cut-edges and proceeds with the modified graph. This idea is pursued in [14, 15].

A very different heuristic for cograph modification was recently proposed by Crespelle [11]. It corrects the neighborhood of each vertex separately. More precisely, an inclusion-minimal cograph editing H_k of the induced subgraph $G_k := G[\{x_1, \dots, x_k\}]$ is computed from the correction H_{i-1} of G_{i-1} in such a way that only edges involving x_i are inserted or deleted. It has the useful property that in each step the number of inserted or deleted edges is minimum, and it inserts or deletes no more than $|E(G)|$ edges in total. It is based on a general property of single-vertex augmentations in hereditary graph classes that are stable under the addition of universal vertices and isolated vertices, see e.g. [48]. A key advantage is that it has linear time complexity, i.e., $O(|V| + |E|)$.

Cotrees are a special case of the much more general modular decomposition tree, which is well-defined for every graph and conveys detailed information about its structure in a hierarchical manner [19]. A subset $M \subseteq V$ is called a module of a graph $G = (V, E)$, if all members of M share the same neighbors in $V \setminus M$. A prime module is a module that is characterized by the property that both, the induced subgraph $G[M]$ and its complement $\overline{G[M]}$, are connected subgraphs of G . Cographs play a particular role in this context as their modular decompositions are of a special form: they are characterized by the absence of prime modules. In particular, the cotree of a cograph coincides with its modular decomposition tree [19]. It is natural to ask, therefore, whether the modular decomposition tree can be manipulated in a such a way that all prime modules of a given graph are converted into “series” or “parallel” modules for which either $G[M]$ and or $\overline{G[M]}$ is disconnected. This is equivalent to converting G into a cograph G^* . Every minimum edit set clearly is inclusion-minimal. However, not every minimum edit set – and in particular not every inclusion-minimal edit set – respects the module structure of G . Figure 1 below shows a pertinent example. In contrast to the editing approach of [11], we pursue an approach that is modul-preserving in the sense that each module of G is also a module of the edited graph G^* . We argue that this property is desirable in the context of orthology detection, because the corrected modular decomposition tree, i.e., the cotree of G^* has a direct interpretation as event-labeled gene tree [30, 35].

An alternative way of looking at the connection between cographs and their modular decomposition trees is to interpret the destruction of all P_4 s in a cograph editing algorithm as the resolution of all prime modules in the edited graph G^* . This simple observation suggests to edit the modules of G . The min-cut approach of [14] is one possibility to achieve this. Here, we consider the merging of modules instead. Every union $\bigcup_{i \in I} M_i$ of the connected components M_1, \dots, M_k of the edited graph $G^*[M]$ or $\overline{G^*[M]}$ forms a module G^* , while $\bigcup_{i \in I} M_i$ was not a module in the graph G before editing. In this

situation, we say that “the modules M_i , $i \in I$ of G are merged w.r.t. G^* ”. Vertices within a module $\bigcup_{i \in I} M_i$ share the same neighbors in $V \setminus (\bigcup_{i \in I} M_i)$. It is sufficient therefore to adjust the neighbors of certain submodules M_i of M to merge the M_i in a way that resolves the prime module M to obtain G^* . In this setting, it seems natural to edit the modular decomposition tree of a graph directly with the aim of converting it step-by-step into the closest modular decomposition tree of a cograph. To this end, one would like to break up individual prime modules by means of the module merge operation.

The key results of this contribution are that (1) every prime node M can be resolved by a sequence of *pairwise* merges of modules that are children of M in the modular decomposition tree, and (2) optimal cograph editing can be expressed as optimal *pairwise* module merging. To prove these statements, we start with an overview of important properties on cographs and the modular decomposition (Section 2 and 3). In Section 4, we then show that so-called module-preserving edit sets are characterized by resolving any prime node by module-merges. In particular, we show that any graph has an optimal edit set that can be entirely expressed by merging modules that are children of prime modules in the modular decomposition tree. Finally in Section 5, we summarize the results and show how they can be used for establishing efficient heuristics for the cograph editing problem. We provide an exact algorithm that allows to optimally edit a cograph via pairwise module-merges. As by-product, we obtain an FPT algorithm for the case that cograph editing is parameterized by the so-called modular-width [1, 18]. We finish this paper with a short discussion on how the latter method can be used to obtain a simple $O(|V|^2)$ -time heuristic.

2 Basic definitions

We consider simple finite undirected graphs $G = (V, E)$ without loops. The complement \overline{G} of a graph $G = (V, E)$ has vertex set V and edge set $E(\overline{G}) = \{xy \mid x, y \in V, x \neq y, xy \notin E\}$. The notation $G \Delta F$ is used to denote the graph $(V, E \Delta F)$, where Δ denotes the symmetric difference. The disjoint union $G \cup H$ of two distinct graphs $G = (V, E)$ and $H = (W, F)$ is simply the graph $(V \cup W, E \cup F)$. The join $G \oplus H$ of G and H is defined as the graph $(V \cup W, E \cup F \cup \{xy \mid x \in V, y \in W\})$. A graph $H = (W, F)$ is a *subgraph* of a graph $G = (V, E)$, in symbols $H \subseteq G$, if $W \subseteq V$ and $F \subseteq E$. If $H \subseteq G$ and $xy \in F$ if and only if $xy \in E$ for all $x, y \in W$, then H is called an *induced subgraph*. We will often denote an induced subgraph $H = (W, F)$ by $G[W]$. A *connected component* of G is a connected induced subgraph that is maximal w.r.t. inclusion. We write $G \simeq H$ for two isomorphic graphs G and H .

Let $G = (V, E)$ be a graph. The *neighborhood* $N(v)$ of $v \in V$ is defined as $N(v) = \{x \mid vx \in E\}$. If there is a risk of confusion we will write $N_G(v)$ to indicate that the respective neighborhood is taken w.r.t. G . The *degree* $\deg(v)$ of a vertex is defined as $\deg(v) = |N(v)|$.

A *tree* is a connected graph that does not contain cycles. A *path* is a tree where every vertex has degree 1 or 2. A *rooted tree* $T = (V, E)$ is a tree with one distinguished vertex $\rho \in V$. We distinguish two further types of vertices in a tree: the *leaves* which are distinct from the root and are contained in only one edge and the *inner* vertices which are contained in at least two edges. The first inner vertex $\text{lca}(x, y)$ that lies on both unique paths from two vertices x , resp., y to the root, is called *lowest common ancestor* of x and y . We say that a rooted tree T *displays* the *triple* $xy|z$ if x , y , and z are leaves of T and the path from x to y does not intersect the path from z to the root of T .

It is well-known that there is a one-to-one correspondence between (isomorphism classes of) rooted trees on V and so-called hierarchies on V . For a finite set V , a *hierarchy* on V is a subset \mathcal{C} of the power set $\mathcal{P}(V)$ such that (i) $V \in \mathcal{C}$, (ii) $\{x\} \in \mathcal{C}$ for all $x \in V$ and (iii) $p \cap q \in \{p, q, \emptyset\}$ for all $p, q \in \mathcal{C}$.

Theorem 2.1 ([51]). *Let \mathcal{C} be a collection of non-empty subsets of V . Then, there is a rooted tree $T = (W, E)$ on V with $\mathcal{C} = \{L(v) \mid v \in W\}$ if and only if \mathcal{C} is a hierarchy on V .*

3 Cographs and the modular decomposition

3.1 Introduction to cographs

Cographs are defined as the class of graphs formed from a single vertex under the closure of the operations of union and complementation, namely: (i) a single-vertex graph K_1 is a cograph; (ii) the disjoint union $G = (V_1 \cup V_2, E_1 \cup E_2)$ of cographs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ is a cograph; (iii) the complement \overline{G} of a cograph G is a cograph. Condition (ii) can be replaced by the equivalent condition that the join $G_1 \oplus G_2$ is a cograph, since $G_1 \oplus G_2$ is the complement of $\overline{G_1} \cup \overline{G_2}$.

The name cograph originates from *complement reducible graphs*, as by definition, cographs can be “reduced” by stepwise complementation of connected components to totally disconnected graphs [50].

It is well-known that for each induced subgraph H of a cograph G either H is disconnected or its complement \overline{H} is disconnected [4]. This, in particular, allows representing the structure of a cograph $G = (V, E)$ in an unambiguous way as a rooted tree $T = (W, F)$, called *cotree*: If the considered cograph is the single vertex graph K_1 , then output the tree $(\{u\}, \emptyset)$. Else if the given cograph G is connected, create an inner vertex u in the cotree with label “series”, build the complement \overline{G} and add the connected components of \overline{G} as children of u . If G is not connected, then create an inner vertex u in the cotree with label “parallel” and add the connected components of G as children of u . Proceed recursively on the respective connected components that consists of more than one vertex. Eventually, this cotree will have leaf-set $V \subseteq W$ and the inner vertices $u \in W \setminus V$ are labeled with either “parallel” or “series” such that $xy \in E$ if and only if $u = \text{lca}_T(x, y)$ is labeled “series”.

The complement of a path on four vertices P_4 is again a P_4 and hence, such graphs are not cographs. Intriguingly, cographs have indeed a quite simple characterization as P_4 -free graphs, that is, no four vertices induce a P_4 . A number of further equivalent characterizations are given in [4] and Theorem 3.2. Determining whether a graph is a cograph can be done in linear time [5, 8].

3.2 Modules and the modular decomposition

The concept of *modular decompositions* (MD) is defined for arbitrary graphs G and allows us to present the structure of G in the form of a tree that generalizes the idea of cotrees. However, in general much more information needs to be stored at the inner vertices of this tree if the original graph has to be recovered.

The MD is based on the notion of modules. These are also known as autonomous sets [43, 44], closed sets [19], clans [17], stable sets, clumps [2] or externally related sets [27]. A *module* of a given graph $G = (V, E)$ is a subset $M \subseteq V$ with the property that for all vertices in $x, y \in M$ it holds that $N(y) \setminus M = N(x) \setminus M$. Therefore, the vertices

within a given module M are not distinguishable by the part of their neighborhoods that lie “outside” M . We denote with $\text{MD}(G)$ the set of all modules of $G = (V, E)$. Clearly, the vertex set V and the singletons $\{v\}$, $v \in V$ are modules, called *trivial* modules. A graph G is called *prime* if it only contains trivial modules. For a module M of G and a vertex $v \in M$, we define the out_M -neighborhood of v as $N(v) \setminus M$. Since for any two vertices contained in M the out_M -neighborhoods are identical, we can equivalently define $N(v) \setminus M$ as the out_M -neighborhood of the module M , where $v \in M$.

We say that a module M of G is *parallel*, resp., *series* if the induced subgraph $G[M]$, resp., the complement $\overline{G[M]}$ is disconnected. If both $G[M]$ and $\overline{G[M]}$ are connected, then M is called *prime*.

For a graph $G = (V, E)$ let M and M' be disjoint subsets of V . We say that M and M' are adjacent (in G) if each vertex of M is adjacent to all vertices of M' ; the sets are non-adjacent if none of the vertices of M is adjacent to a vertex of M' . Two disjoint modules are either adjacent or non-adjacent [43]. One can therefore define the *quotient graph* G/P for an arbitrary subset $P \subseteq \text{MD}(G)$ of pairwise disjoint modules: G/P has P as its vertex set and $M_i M_j \in E(G/P)$ if and only if M_i and M_j are adjacent in G .

A module M is called *strong* if for any module $M' \neq M$ either $M \cap M' = \emptyset$, or $M \subseteq M'$, or $M' \subseteq M$, i.e., a strong module does not *overlap* any other module. The set of all strong modules $\text{MDs}(G) \subseteq \text{MD}(G)$ thus forms a hierarchy, the so-called *modular decomposition* of G . While arbitrary modules of a graph form a potentially exponential-sized family, the sub-family of strong modules has size $O(|V(G)|)$ [26].

Let $\mathbb{P} = \{M_1, \dots, M_k\}$ be a partition of the vertex set of a graph $G = (V, E)$. If every $M_i \in \mathbb{P}$ is a module of G , then \mathbb{P} is a *modular partition* of G . A non-trivial modular partition $\mathbb{P} = \{M_1, \dots, M_k\}$ that contains only maximal (w.r.t. inclusion) strong modules is a *maximal modular partition*. We denote the (unique) maximal modular partition of G by $\mathbb{P}_{\max}(G)$. We will refer to the elements of $\mathbb{P}_{\max}(G[M])$ as the *the children of M* . This terminology is motivated by the following considerations:

The hierarchical structure of $\text{MDs}(G)$ gives rise to a canonical tree representation of G , which is usually called the *modular decomposition tree* $T_{\text{MDs}}(G)$ [28, 44]. The root of this tree is the trivial module V and its $|V|$ leaves are the trivial modules $\{v\}$, $v \in V$. The set of leaves L_v associated with the subtree rooted at an inner vertex v induces a strong module of G . In other words, each inner vertex v of $T_{\text{MDs}}(G)$ *represents* the strong module L_v . An inner vertex v is then labeled “parallel”, “series”, resp., “prime” if L_v is a parallel, series, resp., prime module. The strong module L_v of the induced subgraph $G[L_v]$ associated to a vertex v labeled “prime” is called prime module. Note, the latter does not imply that the graph $G[L_v]$ is prime, however, in all cases the quotient graph $G[L_v]/\mathbb{P}_{\max}(G[L_v])$ is prime [28]. Similar to cotrees it holds that $xy \in E$ if $u = \text{lca}_{T_{\text{MDs}}(G)}(xy)$ is labeled “series”, and $xy \notin E$ if $u = \text{lca}_{T_{\text{MDs}}(G)}(xy)$ is labeled “parallel”. However, to trace back the full structure of a given graph G from $T_{\text{MDs}}(G)$ one has to store additionally the information of the subgraph $G[L_v]/\mathbb{P}_{\max}(G[L_v])$ in the vertices v labeled “prime”. Although, $\text{MDs}(G) \subseteq \text{MD}(G)$ does not represent all modules, we state the following remarkable fact [12, 43]: Any subset $M \subseteq V$ is a module if and only if $M \in \text{MDs}(G)$ or M is the union of children of non-prime modules. Thus, $T_{\text{MDs}}(G)$ represents at least implicitly all modules of G .

A simple polynomial time recursive algorithm to compute $T_{\text{MDs}}(G)$ is as follows [28]: (1) compute the maximal modular partition $\mathbb{P}_{\max}(G)$; (2) label the root node according to the parallel, series or prime type of G ; (3) for each strong module M of $\mathbb{P}_{\max}(G)$,

compute $T_{\text{MDs}}(G[M])$ and attach it to the root node and proceed with $\mathbb{P}_{\text{max}}(G[M])$. The first polynomial time algorithm to compute the modular decomposition is due to Cowan *et al.* [10], and it runs in $O(|V|^4)$. Improvements are due to Habib and Maurer [27], who proposed a cubic time algorithm, and to Müller and Spinrad [45], who designed a quadratic time algorithm. The first two linear time algorithms appeared independently in 1994 [9, 40]. Since then a series of simplified algorithms has been published, some running in linear time [13, 41, 52], and others in almost linear time [13, 26, 29, 42].

For later reference we give the following lemma.

Lemma 3.1. *Let M be a module of a graph $G = (V, E)$ and $M' \subseteq M$. Then, M' is a module of $G[M]$ if and only if M' is a module of G . If M is a strong module of G , then M' is a strong module of $G[M]$ if and only if M' is a strong module of G . Moreover, if M_1 and M_2 are overlapping modules in G , then $M_1 \setminus M_2$, $M_1 \cap M_2$ and $M_1 \cup M_2$ are also modules in G .*

Proof. The first and the last statement were shown in [43]. We prove the second statement.

Let $M \in \text{MDs}(G)$. Assume that $M' \subseteq M$ is a strong module of $G[M]$. Assume for contradiction that M' is not a strong module of G . Hence M' must overlap some module M'' in G . This module M'' cannot be entirely contained in M as otherwise, M'' and M' overlap in $G[M]$ implying that M' is not a strong module of $G[M]$, a contradiction. But then M and M'' must overlap, contradicting that M is strong in G .

If $M' \subseteq M$ is a strong module of G then it does not overlap any module of G . Since every module of $G[M]$ is also a module of G , there cannot be a module of $G[M]$ that overlaps M' and thus, M' is a strong module of $G[M]$. \square

3.3 Useful properties of modular partitions

First, we briefly summarize the relationship between cographs G and the modular decomposition $\text{MDs}(G)$. While the first three items are from [4, 7], the proof of the fourth item can be found in [3, 30].

Theorem 3.2 ([4, 7, 30]). *Let $G = (V, E)$ be an arbitrary graph. Then the following statements are equivalent.*

1. G is a cograph.
2. G does not contain induced paths on four vertices.
3. $T_{\text{MDs}}(G)$ is the cotree of G and hence, has no inner vertices labeled with “prime”.
4. Define a set $\mathcal{R}(G)$ of triples as follows: For any three vertices $x, y, z \in V$ we add the triple $xy|z$ to $\mathcal{R}(G)$ if either $xz, yz \in E$ and $xy \notin E$ or $xz, yz \notin E$ and $xy \in E$. There is a tree T that displays all triples in $\mathcal{R}(G)$.

For later explicit reference, we summarize in the next theorem several results that we already implicitly referred to in the discussion above.

Theorem 3.3 ([25, 28, 43]). *The following statements are true for an arbitrary graph $G = (V, E)$:*

- (T1) *The maximal modular partition $\mathbb{P}_{\text{max}}(G)$ and the modular decomposition $\text{MDs}(G)$ of G are unique.*

- (T2) Let $\mathbb{P}_{\max}(G[M])$ be the maximal modular partition of $G[M]$, where M denotes a prime module of G and $\mathbb{P}' \subsetneq \mathbb{P}_{\max}(G[M])$ be a proper subset of $\mathbb{P}_{\max}(G[M])$ with $|\mathbb{P}'| > 1$. Then, $\bigcup_{M' \in \mathbb{P}'} M' \notin \text{MD}(G)$.
- (T3) Any subset $M \subseteq V$ is a module of G if and only if M is either a strong module of G or M is the union of children of a non-prime module of G .

Statements (T1) and (T3) are clear. Statement (T2) explains that none of the unions of elements of a maximal modular partition of $G[M]$ are modules of G , whenever M is a prime module of G . Moreover, Statement (T3) can be used to show that all prime modules are strong.

Lemma 3.4. *Let $G = (V, E)$ be an arbitrary graph. Then, every prime module M of G is strong.*

Proof. Let M be a prime module of G . Assume for contradiction that M is not strong in G . Theorem 3.3(T3) implies that M is the union of children of some non-prime module M' . Hence, there is a subset $\mathcal{M} \subsetneq \mathbb{P}_{\max}(G[M'])$ such that $M = \bigcup_{M'_i \in \mathcal{M}} M'_i$. Note that $1 < |\mathcal{M}| < |\mathbb{P}_{\max}(G[M'])|$, since all $M'_i \in \mathbb{P}_{\max}(G[M'])$ are strong and $\bigcup_{M'_i \in \mathbb{P}_{\max}(G[M'])} M'_i = M'$ is non-prime. As M' is non-prime, it is either parallel or series. Since M is a non-trivial union of elements in $\mathbb{P}_{\max}(G[M'])$, $G[M]$ is either disconnected (if M' is parallel) or its complement $\overline{G[M]}$ is disconnected (if M' is series). But then M is non-prime; a contradiction. Thus, M is a strong module of G . □

In what follows, whenever the term “prime module” is used it refers therefore always to a strong module.

3.4 Cograph editing

Given an arbitrary graph we are interested in understanding how the graph can be edited into a cograph. A well-studied problem is the following optimization problem.

Problem 3.5 (Optimal Cograph Editing). Given a graph $G = (V, E)$. Find a set $F \subseteq \binom{V}{2}$ of minimum cardinality such that $H = (V, E \triangle F)$ is a cograph.

We will simply call an edit set of minimum cardinality an *optimal (cograph) edit set*. For later reference we recall Lemma 9 of [35]. It shows that it suffices to solve the cograph editing problem separately for each connected component of G .

Lemma 3.6 ([35]). *Let $G = (V, E)$ be a graph with optimal edit set F . Then $\{x, y\} \in F \setminus E$ implies that x and y are located in the same connected component of G .*

Let $G = (V, E)$ be a graph and F be an arbitrary edit set that transforms G to the cograph $H = (V, E \triangle F)$. If any module of G is a module of H , then F is called *module-preserving*.

Proposition 3.7 ([25]). *Every graph has an optimal module-preserving cograph edit set.*

The importance of module-preserving edit sets lies in the fact that they update either all or none of the edges between any two disjoint modules. It is worth noting that module preserving edit sets do not necessarily preserve the property of modules being strong, i.e., although M might be a strong module in G it needs not to be strong in H .

Definition 3.8. Let $G = (V, E)$ be a graph, F a cograph edit set for G and M be a non-trivial module of G . The induced edit set in $G[M]$ is

$$F[M] := \{\{x, y\} \in F \mid x, y \in M\}.$$

The next result shows that any optimal edit set F can entirely expressed by the union of edits within prime modules and that $F[M]$ is an optimal edit set of $G[M]$ for any module M of G . Hence, if $F[M]$ is not optimal for some module M of G , then F cannot be an optimal edit set for G .

Lemma 3.9 ([25]). *Let $G = (V, E)$ be an arbitrary graph and let M be a non-trivial module of G . If F' is an optimal edit set of the induced subgraph $G[M]$ and F is an optimal edit set of G , then $(F \setminus F[M]) \cup F'$ is an optimal edit set of G . Thus, $|F[M]| = |F'|$.*

Moreover, the optimal cograph editing problem can be solved independently on the prime modules of G .

4 Module merge is the key to cograph editing

Since cographs are characterized by the absence of induced P_4 s, we can interpret every optimal cograph-editing method as the removal of all P_4 s in the input graph with a minimum number of edits. A natural strategy is therefore to detect P_4 s and then to decide which edges must be edited. Optimal edit sets are not necessarily unique, see Figure 1. The computational difficulty arises from the fact that editing an edge of a P_4 can produce new P_4 s in the updated graph. Hence, we cannot expect *a priori* that local properties of G alone will allow us to identify optimal edits.

By Lemma 3.9, on the other hand, it is sufficient to edit within the prime modules. Moreover, as shown in Figure 1, there are strong modules M^* in an optimal edited cograph H that are not modules in G . Hence, instead of editing P_4 s in G , it might suffice to edit the out_{M_i} -neighborhoods for some $M_i \in \mathbb{P}_{\max}(G[M])$ in such a way that they result in the new module M^* in H . The following definitions are important for the concepts of the “module merge process” that we will extensively use in our approach.

Definition 4.1 (Module Merge). Let G and H be arbitrary graphs with $V(H) \subseteq V(G)$ and let $\text{MD}(G)$ and $\text{MD}(H)$ denote their corresponding sets of all modules. Consider a set $\mathcal{M} := \{M_1, M_2, \dots, M_k\} \subseteq \text{MD}(G)$. We say that the modules in \mathcal{M} are *merged* (w.r.t. H) if

- (i) $M_1, \dots, M_k \in \text{MD}(H)$,
- (ii) $M := \bigcup_{i=1}^k M_i \in \text{MD}(H)$, and
- (iii) $M \notin \text{MD}(G)$.

We use the symbols \boxplus and \rightarrow as operations that allows us to illustrate the merge process, that is, we write $M_1 \boxplus \dots \boxplus M_k = \boxplus_{i=1}^k M_i \rightarrow M$, whenever the modules M_1, M_2, \dots, M_k are merged w.r.t. H resulting in the module $M = \bigcup_{i=1}^k M_i$ of H .

The intuition is that the modules M_1 through M_k of G are merged into a single new module M , their union, that is present in H but not in G . This, in particular, already defines all required edits to adjust the neighbors of the vertices in $\bigcup_{i=1}^k M_i$ in G resulting in the module $M = \bigcup_{i=1}^k M_i$ of H . It is easy to verify that \boxplus is commutative in the sense that

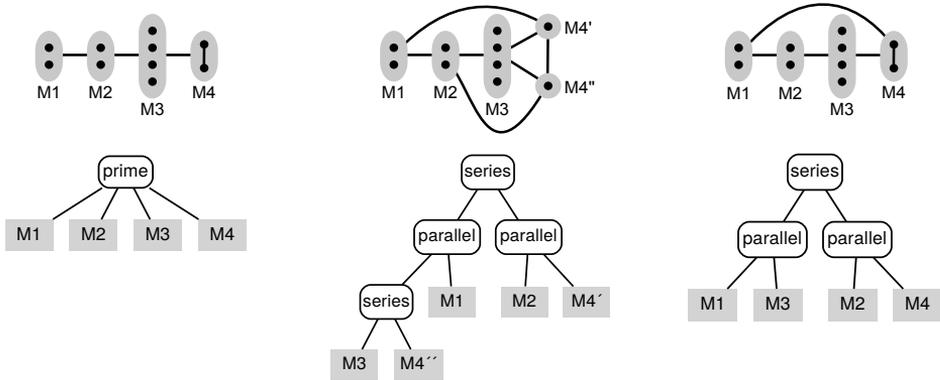


Figure 1: Shown are three graphs G, H_1, H_2 (from left to right). Maximal non-trivial strong modules are indicated by gray ovals in each graph and edges are used to show whether two modules are adjacent or not. The dots/lines within the modules are used to depict the vertices/edges within the modules. The modular decomposition trees up to a certain level are depicted below the respective graphs. This tree differs from the modular decomposition tree of the original graph G, H_1 , and H_2 , respectively, only from the unresolved leaf-nodes (gray boxes).

Left: A non-cograph G is shown. The optimal edit set F has cardinality 4. *Center:* An optimal edited cograph $H_1 = G \triangle F$ is shown, where F is not module-preserving. None of the new strong modules of H_1 that are not modules of G can be expressed as the union of the sets M_1, \dots, M_4 . Hence, none of these modules are the result of a module merge process. *Right:* An optimal edited cograph $H_2 = G \triangle F$ is shown, where F is module-preserving. The new strong modules M_1^*, M_2^* of H_2 that are not modules of G are two parallel modules. They can be written as $M_1^* = M_1 \cup M_3$ and $M_2^* = M_2 \cup M_4$. Hence, they are obtained by merging modules of G , in symbols: $M_1 \boxplus M_3 \rightarrow M_1^*$ and $M_2 \boxplus M_4 \rightarrow M_2^*$. Here we have $F_{H_2}(M_1 \boxplus M_3 \rightarrow M_1^*) = F_{H_2}(M_2 \boxplus M_4 \rightarrow M_2^*) = F = \{\{x, y\} \mid x \in M_1, y \in M_4\}$.

if $M_1 \boxplus M_2 \rightarrow M$, then $M_2 \boxplus M_1 \rightarrow M$. However, \boxplus is not necessarily associative. To see this, consider the example in Figure 2. Although the module M_3^* in H is obtained by merging the modules $\{3\}, \{4\}$ and $\{5\}$, the set $\{3\} \cup \{4\}$ does not form a module in H . Hence, although $\{3\} \boxplus \{4\} \boxplus \{5\} \rightarrow M_3^*$, it does not hold that $\{3\} \boxplus \{4\} \rightarrow M^*$ for any module M^* in H . Thus, we cannot write $(\{3\} \boxplus \{4\}) \boxplus \{5\} \rightarrow M_3^*$.

It follows directly from Definition 4.1 that every new module M of H that is not a module of G can be obtained by merging trivial modules: simply set $M = \bigcup_{x \in M} \{x\}$ and $\boxplus_{x \in M} \{x\} \rightarrow M$ follows immediately. In what follows we will show, however, that each strong module of H that is not a module of G can be obtained by merging the modules that are contained in $\mathbb{P}_{\max}(G[M])$ of some prime module M of G .

When modules M_1, \dots, M_k of G are merged w.r.t. H then all vertices in $M = \bigcup_{h=1}^k M_h$ must have the same out_M -neighbors in H , while at least two vertices $x \in M_i, y \in M_j, 1 \leq i \neq j \leq k$ must have different out_M -neighbors in G . Hence, in order to merge these modules it is necessary to change the out_M -neighbors in G . However, edit operations between vertices *within* M are dispensable for obtaining the module M .

Definition 4.2 (Module Merge Edit). Let $G = (V, E)$ be an arbitrary graph and F be an arbitrary edit set resulting in the graph $H = (V, E \triangle F)$. Let $H' \subseteq H$ be an induced subgraph of H and suppose $M_1, \dots, M_k \in \text{MD}(G)$ are modules that have been merged w.r.t. H' resulting in the module $M = \bigcup_{i=1}^k M_i \in \text{MD}(H')$. We then call

$$F_{H'}(\boxplus_{i=1}^k M_i \rightarrow M) := \{\{x, v\} \in F \mid x \in M, v \in V(H') \setminus M\} \quad (4.1)$$

the *module merge edits* associated with $\boxplus_{i=1}^k M_i \rightarrow M$ w.r.t. H' .

By construction, the edit set $F_{H'}(\boxplus_{i=1}^k M_i \rightarrow M)$ comprises exactly those (non)edges of F that have been edited so that all vertices in M have the same out_M -neighborhood in $H' = (V', E')$. In particular, it contains only (non)edges of F that are not entirely contained in $G[M]$, but entirely contained in H' . Moreover, (non)edges of F that contain a vertex in $V(H')$ and a vertex in $V \setminus V(H')$ are not considered as well.

Let G be an arbitrary graph and F be an optimal edit set that applied to G results in the cograph H . We will show that every optimal module-preserving edit set F can be expressed completely by means of module merge edits. To this end, we will consider the prime modules M of the given graph G (in particular certain children of M that do not share the same out-neighborhood) and adjust their out-neighbors to obtain new modules. Illustrative examples are given in Figure 1 and 2.

We are now in the position to derive the main results, Theorems 4.3–4.7. We begin with showing that each strong module of H that is not a module of G can be obtained by merging some children of a particular chosen prime module of G . Moreover, we prove that any strong module of H that is a module of G must also be strong in G .

Theorem 4.3. *Let $G = (V, E)$ be an arbitrary graph, F an optimal module-preserving cograph edit set, and $H = (V, E \triangle F)$ the resulting cograph. Then, each strong module M^* of H is either a module in G or there exists a prime module P_{M^*} of G that contains M^* and is minimal w.r.t. inclusion, i.e., there is no prime module P'_{M^*} of G with $M^* \subseteq P'_{M^*} \subsetneq P_{M^*}$. In the latter case M^* is obtained by merging some modules in $\mathbb{P}_{\max}(G[P_{M^*}])$.*

Furthermore, if a strong module M^ of H is a module in G , then M^* is a strong module of G .*

Proof. Let M^* be an arbitrary strong module of H that is not a module of G . We show first that for the module M^* there is a prime module P_{M^*} of G with $M^* \subseteq P_{M^*}$ such that there is no other prime module P'_{M^*} of G with $M^* \subseteq P'_{M^*} \subsetneq P_{M^*}$.

Since M^* is a module of H but not of G there are vertices $x \in M^*$ and $y \in V \setminus M^*$ with $\{x, y\} \in F$. Now, let P_{M^*} be the strong module of G containing x and y that is minimal w.r.t. inclusion, that is, there is no other strong module of G that is properly contained in P_{M^*} and that contains x and y . Thus $\{x, y\} \in F[P_{M^*}]$. Lemma 3.9 implies that $F[P_{M^*}]$ is an optimal edit set of $G[P_{M^*}]$. Since P_{M^*} is minimal w.r.t. inclusion it holds that x and y are from distinct children $M_x, M_y \in \mathbb{P}_{\max}(G[P_{M^*}])$. We continue to show that this strong module P_{M^*} is indeed prime. Assume for contradiction, that P_{M^*} is a non-prime module of G . If P_{M^*} is parallel, then editing $\{x, y\}$ would connect the two connected components M_x, M_y of $G[P_{M^*}]$. Then, it follows by Lemma 3.6 that $F[P_{M^*}]$ is not optimal; a contradiction. By similar arguments for the complement $\overline{G[P_{M^*}]}$ it can be shown that P_{M^*} cannot be a series module. Thus P_{M^*} must be prime. Since F is module-preserving, P_{M^*} is module in H . Hence, P_{M^*} and M^* cannot overlap, since M^* is strong in H . However, since $x \in P_{M^*} \cap M^*$ and $y \in P_{M^*}$ but $y \notin M^*$ we have $M^* \subseteq P_{M^*}$.

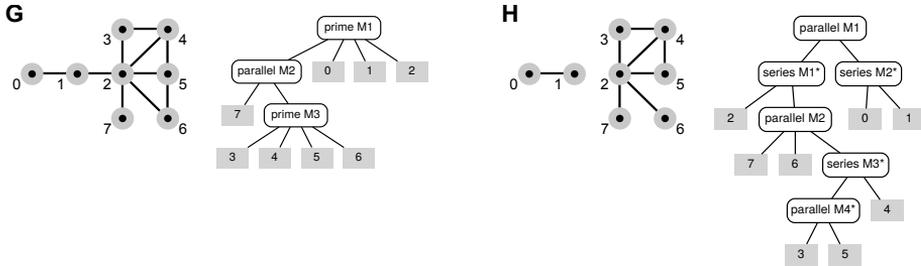


Figure 2: Illustration of the main results. Consider the non-cograph G , the cograph $H = G \triangle F$ and the module-preserving edit set $F = \{\{1, 2\}, \{5, 6\}\}$. The modular decomposition trees are depicted right to the respective graphs.

According to Theorem 4.3, both strong modules M_1 and M_2 of H that are modules of G are also strong modules of G and correspond to the prime module M_1 and the parallel module M_2 in G , respectively. Moreover, each of the new strong modules M_1^*, \dots, M_4^* of H are obtained by merging children of a prime module of G . To be more precise, M_1^* and M_2^* are obtained by merging children of the prime module M_1 of G : $M_2 \sqcup \{2\} \rightarrow M_1^*$ and $\{0\} \sqcup \{1\} \rightarrow M_2^*$ with $F_{H[M_1]}(M_2 \sqcup \{2\} \rightarrow M_1^*) = F_{H[M_1]}(\{0\} \sqcup \{1\} \rightarrow M_2^*) = \{\{1, 2\}\}$. The new strong modules M_3^* and M_4^* are obtained by merging children of the prime module M_3 of G : $\{3\} \sqcup \{5\} \rightarrow M_4^*$ and $\{3\} \sqcup \{4\} \sqcup \{5\} \rightarrow M_3^*$ with $F_{H[M_3]}(\{3\} \sqcup \{5\} \rightarrow M_4^*) = F_{H[M_3]}(\{3\} \sqcup \{4\} \sqcup \{5\} \rightarrow M_3^*) = \{\{5, 6\}\}$. According to Corollary 4.7, the set F can be written as the union of the edit sets used to obtain the new merged modules of H .

It is worth noting that not all strong modules of G remain strong in H (e.g. the prime module M_3) and that there are (non-strong) modules in H (e.g. the module $\{6, 7\}$) that are not obtained by merging children of prime modules of G .

Finally, since P_{M^*} is chosen to be minimal w.r.t. inclusion, there exists in particular no prime module P'_{M^*} of G with $M^* \subseteq P'_{M^*} \subsetneq P_{M^*}$.

We continue to show that M^* is obtained by merging some child modules of P_{M^*} in G , say $M_1, \dots, M_k \in \mathbb{P}_{\max}(G[P_{M^*}])$. Note that we just formally prove the existence of such a subset $\{M_1, \dots, M_k\} \subset \mathbb{P}_{\max}(G[P_{M^*}])$ without explicitly constructing it. To this end, we need to verify the three conditions of Definition 4.1, i.e., (i) $M_1, \dots, M_k \in \text{MD}(H)$, (ii) $M^* := \bigcup_{i=1}^k M_i \in \text{MD}(H)$, and (iii) $M^* \notin \text{MD}(G)$. Since each $M_i \in \mathbb{P}_{\max}(G[P_{M^*}])$ is module of G and F is module-preserving, Condition (i) is always satisfied. Moreover, by assumption $M^* \notin \text{MD}(G)$ and thus Condition (iii) is satisfied.

It remains to show that Condition (ii) is satisfied. To this end, we show that there are modules M_1, \dots, M_k of G (without explicitly constructing them) such that $M^* = \bigcup_{i=1}^k M_i$. We prove this by showing that each module from P_{M^*} is either completely contained in, or disjoint from M^* . First, note that $M^* \neq P_{M^*}$, since M^* is not a module of G . Second, M^* cannot overlap any $M_i \in \mathbb{P}_{\max}(G[P_{M^*}])$, since M_i is a module of H and M^* is strong in H . We continue to show that there is no $M_i \in \mathbb{P}_{\max}(G[P_{M^*}])$ such that $M^* \subseteq M_i$. Assume for contradiction that there is a module $M_i \in \mathbb{P}_{\max}(G[P_{M^*}])$ with $M^* \subseteq M_i$. Note that M_i cannot be prime in G , as otherwise $M^* \subseteq M_i = P'_{M^*} \subsetneq P_{M^*}$, contradicting the minimality of P_{M^*} . Moreover, M^* cannot overlap any $M_j^i \in \mathbb{P}_{\max}(G[M_i])$, since M^* is strong in H and any M_j^i is a module of H , since F is module-

preserving. Furthermore, since M_i is non-prime in G for any subset $\{M_1^i, \dots, M_l^i\} \subsetneq \mathbb{P}_{\max}(G[M_i])$ it holds that the set $M' = \bigcup_{j=1}^l M_j^i$ is a module of G (cf. Theorem 3.3(T3)). Since M^* is no module of G it cannot be a union of elements in $\mathbb{P}_{\max}(G[M_i])$. Note that this especially implies that $M^* \neq M_i$ and $M^* \neq M_j^i$ for all $M_j^i \in \mathbb{P}_{\max}(G[M_i])$. Now it follows, that $M^* \subset M_j^i$ for some $M_j^i \in \mathbb{P}_{\max}(G[M_i])$. Repeating the latter arguments and since G is finite, there must be a minimal set M_b^a with $M^* \subset M_b^a \subset \dots \subset M_j^i \subset M_i$. Now we apply the latter arguments again and obtain that $M^* \subset M' \in \mathbb{P}_{\max}(G[M_b^a])$ which is not possible, since M_b^a is chosen to be the minimal module that contains M^* . Thus, there is no $M_i \in \mathbb{P}_{\max}(G[P_{M^*}])$ such that $M^* \subseteq M_i$.

Now, since $M^* \neq P_{M^*}$, and M^* does not overlap any $M_i \in \mathbb{P}_{\max}(G[P_{M^*}])$, and there is no $M_i \in \mathbb{P}_{\max}(G[P_{M^*}])$ such that $M^* \subseteq M_i$, there must be a set $\{M_1, \dots, M_k\} \subsetneq \mathbb{P}_{\max}(G[P_{M^*}])$ such that $M^* = \bigcup_{i=1}^k M_i$. Thus, Condition (ii) is satisfied and therefore M^* is obtained by merging modules in $\mathbb{P}_{\max}(G[P_{M^*}])$.

Hence, any strong module of H is either a module of G or obtained by merging the children of a prime module of G .

Finally, assume that there is a strong module M^* in H that is a module of G . Assume that M^* is not strong in G . Then there is a module M in G that overlaps M^* . Since F is module-preserving, M is a module in H and thus, M overlaps M^* in H ; a contradiction. Thus, any strong module M^* of H that is also a module of G must be strong in G . \square

Theorem 4.3 allows us to give the following definitions that we will use in the subsequent part.

Definition 4.4. Let $G = (V, E)$ be an arbitrary graph, F an optimal module-preserving cograph edit set, and $H = (V, E \triangle F)$ the resulting cograph. Let M^* be a strong module of H but no module of G .

We denote by P_{M^*} the prime module of G that contains M^* and is minimal w.r.t. inclusion, i.e., there is no prime module P'_{M^*} of G with $M^* \subseteq P'_{M^*} \subsetneq P_{M^*}$. Furthermore, we denote by $\mathcal{C}(M^*) \subset \mathbb{P}_{\max}(G[P_{M^*}])$ the set of children of P_{M^*} that satisfies $\bigcup_{M_i \in \mathcal{C}(M^*)} M_i = M^*$.

The next result provides a characterization of module-preserving edit sets by means of module merge of the children of prime modules.

Theorem 4.5. Let $G = (V, E)$ be an arbitrary graph, F an optimal cograph edit set, and $H = (V, E \triangle F)$ the resulting cograph. Then F is module-preserving for G if and only if each new strong module M^* of H that is not a module of G is obtained by merging the modules in $\mathcal{C}(M^*) \subset \mathbb{P}_{\max}(G[P_{M^*}])$, in symbols $\boxplus_{M_i \in \mathcal{C}(M^*)} M_i \rightarrow M^*$.

Proof. If F is an optimal and module-preserving edit-set for G , we can apply Theorem 4.3.

For the converse, assume for contraposition that F is not module-preserving. Then, there is a module M_i in G that is not a module in H . Hence, there is a vertex $z \in V \setminus M_i$ and two vertices $x, y \in M_i$ such that $xz \in E(H)$ and $yz \notin E(H)$ and thus, either $\{x, z\} \in F$ or $\{y, z\} \in F$. There are two cases, either $xy \in E(H)$ or $xy \notin E(H)$. Since H is a cograph we can apply Theorem 3.2 and conclude that either $yz|x \in \mathcal{R}(H)$ or $xz|y \in \mathcal{R}(H)$. Assume that $xz|y \in \mathcal{R}(H)$ and let T be the cotree of H . Since T displays $xz|y$, the strong module M^* of H located at the $\text{lca}_T(x, z)$ contains the vertices x and z but not y . Moreover, since there is an edit $\{x, z\}$ or $\{y, z\}$ in F there is a strong prime module P_{M^*} in G that contains x, y, z and is minimal w.r.t. inclusion. Note, $M_i \neq P_{M^*}$ since

$x, y \in M_i$ and $z \notin M_i$. Moreover, since M_i is a module in G , but none of the unions of the children of P_{M^*} is a module of G (cf. Theorem 3.3(T3)), we can conclude that $M_i \subseteq M'$, where M' is a child of P_{M^*} in G . Since P_{M^*} is the minimal prime module that contains x, y, z and there is an edit $\{x, z\}$ or $\{y, z\}$ in F , the vertex z must be located in a module different from the module M' that contains both x and y . Thus, $z \notin M'$. Therefore, there is no module in G that contains x and z but not y . Thus, M^* is no module of G . Since there is no module in G that contains x and z but not y , the set M^* cannot be written as the union of children of any strong prime module P_{M^*} and thus, M^* is not obtained by merging modules of $\mathbb{P}_{\max}(G[P_{M^*}])$. The case $yz|x \in \mathcal{R}(H)$ is shown analogously. \square

Combining the latter results, it can be shown that for every graph G there is always an optimal edit set such that the resulting cograph H contains all modules of G and any newly created strong module M^* of H is obtained by merging the respective modules in $\mathcal{C}(M^*)$.

Theorem 4.6. *Any graph $G = (V, E)$ has an optimal edit-set F such that each strong module M^* in $H = (V, E \triangle F)$ that is not a module of G is obtained by merging modules in $\mathbb{P}_{\max}(G[P_{M^*}])$, where P_{M^*} is a prime module of G .*

Proof. Proposition 3.7 implies that any graph has a module-preserving optimal edit set. Hence, we can apply Theorem 4.5 to derive the statement. \square

Finally, the following result shows that each module-preserving edit set can indeed be derived by considering the module merge edits only.

Theorem 4.7. *Let $G = (V, E)$ be an arbitrary graph, F an optimal module-preserving cograph edit set, $H = (V, E \triangle F)$ the resulting cograph, and \mathcal{M} the set of all strong modules of H that are no modules of G . Then,*

$$F = \bigcup_{M^* \in \mathcal{M}} (F_{H[P_{M^*}]}(\bigsqcup_{M_i \in \mathcal{C}(M^*)} M_i \rightarrow M^*)).$$

Proof. We set $F^* = \bigcup_{M^* \in \mathcal{M}} (F_{H[P_{M^*}]}(\bigsqcup_{M_i \in \mathcal{C}(M^*)} M_i \rightarrow M^*))$. Clearly, it holds that $F^* \subseteq F$. It remains to show that, $F \subseteq F^*$. First, observe, that every edit $\{x, y\} \in F$ is between distinct children $M_x, M_y \in \mathbb{P}_{\max}(G[P_{M^*}])$ of a prime module P_{M^*} of G . To see this, let P_{M^*} be a strong module of G such that x and y are in distinct children $M_x, M_y \in \mathbb{P}_{\max}(G[P_{M^*}])$ and assume for contradiction that P_{M^*} is non-prime in G . Let $F' := \bigcup_{M_i \in \mathbb{P}_{\max}(G[P_{M^*}])} F[M_i]$. Since P_{M^*} is non-prime in G it follows that F' is an edit set for $G[P_{M^*}]$, that is, $G[P_{M^*}] \triangle F'$ is a cograph. But $|F'| < |F[P_{M^*}]|$; contradicting Lemma 3.9. Thus, every edit $\{x, y\} \in F$ is between distinct children $M_x, M_y \in \mathbb{P}_{\max}(G[P_{M^*}])$ of a prime module P_{M^*} of G .

Assume that $\{x, y\} \in F$, but $\{x, y\} \notin F^*$. By the latter arguments, there is a prime module P_{M^*} of G with $x \in M_x$ and $y \in M_y$ and $M_x, M_y \in \mathbb{P}_{\max}(G[P_{M^*}])$. Now let M'_x be the strong module of H that contains x but not y and that is maximal w.r.t. inclusion. Since F is module-preserving, M_x is a module in H . Moreover, since M'_x is a strong module of H , the modules M'_x and M_x do not overlap in H . Therefore, either $M_x \subsetneq M'_x$ or $M'_x \subseteq M_x$. We show first that the case $M_x \subsetneq M'_x$ is not possible. Assume for contradiction, that $M_x \subsetneq M'_x$. Thus, there is a vertex $z \in M'_x \setminus M_x$. Since P_{M^*} is prime in G and $M_x \in \mathbb{P}_{\max}(G[P_{M^*}])$, we can apply Theorem 3.3(T2) and conclude that there is no other module than M_x in G that entirely contains M_x but not y . Since

$M_x \subsetneq M'_x \subsetneq P_{M^*}$ it follows that M'_x is a new strong module of H and therefore, by Theorem 4.3, obtained by merging modules $M_1, \dots, M_k \in \mathcal{C}(M'_x) \subsetneq \mathbb{P}_{\max}(G[P_{M^*}])$. But then $\{x, y\} \in F_{H[P_{M^*}]}(\bigsqcup_{M_i \in \mathcal{C}(M'_x)} M_i \rightarrow M'_x) \subseteq F^*$; contradicting that $\{x, y\} \notin F^*$. Hence, $M'_x \subseteq M_x$. Similarly, $M'_y \subseteq M_y$ for the strong module M'_y of H that contains y but not x and that is maximal w.r.t. inclusion.

Consider now the strong module M^* of H that is identified with the lowest common ancestor of the modules $\{x\}$ and $\{y\}$ within the cotree of H . Then, there are distinct children in $\mathbb{P}_{\max}(H[M^*])$, containing x and y , respectively. Since M'_x is the strong module of H that contains x but not y and that is maximal w.r.t. inclusion, we have $M'_x \in \mathbb{P}_{\max}(H[M^*])$. Analogously, $M'_y \in \mathbb{P}_{\max}(H[M^*])$.

Both, M_x as well as M_y are modules in H and G . Since F is module-preserving, either all or none of the edges between M_x and M_y are edited. Since $\{x, y\} \in F$ we have, therefore, $\{x', y'\} \in F$ for all $x' \in M'_x \subseteq M_x$ and $y' \in M'_y \subseteq M_y$. Let $F' := \{\{x', y'\} \mid x' \in M'_x, y' \in M'_y\}$. By the latter argument $F' \neq \emptyset$ and $F' \subseteq F$.

Note, the subgraphs $H[M'_x]$ and $H[M'_y]$ are cographs. Since M^* is either a parallel or a series module in H , we have either (i) $H[M'_x \cup M'_y] = H[M'_x] \cup H[M'_y]$ or (ii) $H[M'_x \cup M'_y] = H[M'_x] \oplus H[M'_y]$, respectively. Since F' comprises the edits $\{x', y'\}$ between *all* vertices $x' \in M'_x$ and $y' \in M'_y$, the graph $H[M'_x \cup M'_y] \triangle F'$ is in case (i) the graph $H[M'_x] \cup H[M'_y]$ and in case (ii) $H[M'_x] \oplus H[M'_y]$. By definition, in both cases $H[M'_x \cup M'_y] \triangle F'$ is a cograph. Note that F' did not change the out $_{M'_x \cup M'_y}$ -neighborhood and thus, the graph $H[M^*] \triangle F' = G[M^*] \triangle (F[M^*] \setminus F')$ is a cograph as well. Since $\{x, y\} \in F' \cap F[M^*]$ it holds that $|F[M^*] \setminus F'| < |F[M^*]|$. But then, $F[M^*]$ is not optimal, and therefore, by Lemma 3.9 the set F is not optimal; a contradiction.

In summary, there exists no edit $\{x, y\} \in F$ with $\{x, y\} \notin F^*$. Hence, $F \subseteq F^*$ and the statement follows. \square

From an algorithmic perspective, Theorem 4.7 implies that it is sufficient to correctly determine the set of strong modules of a resulting cograph H that are no modules of the given graph G . Afterwards, the module-preserving edit set F is obtained by taking all the edits needed for the corresponding module merge operations. On the other hand, by Theorem 4.6 it is ensured that such a closest cograph H that contains all modules of G always exists.

5 Pairwise module merge and algorithmic issues

So far, we have shown that for an arbitrary graph $G = (V, E)$ there is an optimal module-preserving edit set F that transforms G into the cograph $H = (V, E \triangle F)$ (cf. Theorem 4.6). Moreover, this edit set F can be expressed in terms of edits derived by module merge operations on the strong modules of H that are no modules of G (cf. Theorem 4.7). In what follows, we show that there is an explicit order in which these individual merge operations can be consecutively applied to G such that all intermediate edit-steps result in graphs that contain all modules of G , and, moreover, all new strong modules produced in this edit-step are preserved in any further step. In Section 5.1, we show that an optimal edit set can always be obtained by a series of “ordered” pairwise merge operations. In Section 5.2, we show that the latter “order”-condition can even be relaxed and that particular modules can be pairwise merged in an arbitrary order to obtain an optimal edited graph.

The next Lemma shows that the number of edits in an optimal edit set F can be expressed as the sum of individual edits based on the \boxplus -operator to obtain the strong modules

in a cograph $H = G \triangle F$ that are no modules in G .

Lemma 5.1. *Let $G = (V, E)$ be a graph, F an optimal module-preserving cograph edit-set, and $H = (V, E \triangle F)$ the resulting cograph. Let $\mathcal{M} = \{M_1^*, \dots, M_n^*\}$ be the set of all strong modules of H that are no modules of G and assume that the elements in \mathcal{M} are partially ordered w.r.t. inclusion, i.e., $M_i^* \subseteq M_j^*$ implies $i \leq j$.*

Let $M^ \in \mathcal{M}$. We set $F_{M^*} := \{\{x, v\} \in F \mid x \in M^*, v \in P_{M^*} \setminus M^*\}$, that is, the set $F_{M^*} \subseteq F$ comprises all edits in F that are used to obtain the module M^* within $G[P_{M^*}]$.*

Furthermore, we set $\sigma_{M_1^} = F_{M_1^*}$ and $\sigma_{M_i^*} = F_{M_i^*} \setminus (\bigcup_{j=1}^{i-1} F_{M_j^*})$, $2 \leq i \leq n$. Then*

$$F = \bigcup_{i=1}^n \sigma_{M_i^*} \text{ and, thus, } |F| = \sum_{i=1}^n |\sigma_{M_i^*}|.$$

Moreover, for each intermediate graph $G_j = G \triangle (\bigcup_{i=1}^j \sigma_{M_i^})$ and any $M_i^* \in \mathcal{M}$ with $i - 1 \leq j$ we have*

$$G_j[M_i^*] = H[M_i^*].$$

In each step j the induced subgraphs $G_j[M_i^]$ are already cographs for all sets M_i^* with $i - 1 \leq j$ and hence $F[M_i^*] \setminus \bigcup_{k=1}^j \sigma_{M_k^*} = \emptyset$, for all $i - 1 \leq j$.*

Proof. By Theorem 4.3, for each $M^* \in \mathcal{M}$ there is an inclusion-minimal prime module P_{M^*} in G and a set of children $\mathcal{C}(M^*) \subseteq \mathbb{P}_{\max}(G[P_{M^*}])$ such that $\bigsqcup_{M_i \in \mathcal{C}(M^*)} M_i \rightarrow M^*$. Thus, P_{M^*} and $\mathcal{C}(M^*)$ exists and $\mathcal{C}(M^*)$ is not empty.

Now, we show that $|F|$ can be expressed by the sum of the size of the edits in $\sigma_{M_i^*}$. To this end, observe that by Theorem 4.7, $F = \bigcup_{M^* \in \mathcal{M}} (F_{H[P_{M^*}]}(\bigsqcup_{M_i \in \mathcal{C}(M^*)} M_i \rightarrow M^*))$. Thus, $F = \bigcup_{M^* \in \mathcal{M}} F_{M^*}$. By construction of $\sigma_{M_i^*}$ it holds first that $\bigcup_{i=1}^n \sigma_{M_i^*} = \bigcup_{i=1}^n F_{M_i^*}$ and second that $\sigma_{M_i^*} \cap \sigma_{M_j^*} = \emptyset$ for all $i \neq j$. Hence, $F = \bigcup_{i=1}^n \sigma_{M_i^*}$ and thus, $|F| = \sum_{i=1}^n |\sigma_{M_i^*}|$.

By construction, \mathcal{M} is partially ordered w.r.t. inclusion. We want to show that $G_j[M_i^*] = H[M_i^*]$ for all $i - 1 \leq j$. To this end, we show that $F[M_i^*] \setminus \bigcup_{k=1}^j \sigma_{M_k^*} = \emptyset$, in which case after each step j there are no more edits left to modify an edge between vertices within M_i^* . We show first that the latter is satisfied for all $1 \leq i \leq n$ and a fixed $j = i - 1$. Assume for contradiction that $\{x, y\} \in F[M_i^*] \setminus \bigcup_{k=1}^{i-1} \sigma_{M_k^*}$ and thus, $x, y \in M_i^*$. Since $\{x, y\} \in F = \bigcup_{k=1}^n F_{M_k^*}$, there must be a module $M_\ell^* \in \mathcal{M}$ such that $\{x, y\} \in F_{M_\ell^*}$. By construction, $F_{M_\ell^*}$ contains only the edits that affect the out $_{M_\ell^*}$ -neighborhood. Thus, w.l.o.g. we can assume that $x \in M_\ell^*$ and $y \notin M_\ell^*$. Since M_ℓ^* and M_i^* are strong modules, they do not overlap, and therefore, $M_\ell^* \subsetneq M_i^*$. However, since \mathcal{M} is partially ordered, we can conclude that $\ell < i$ and therefore, $\{x, y\} \in \bigcup_{k=1}^{i-1} \sigma_{M_k^*}$. Hence, $\{x, y\} \notin F[M_i^*] \setminus \bigcup_{k=1}^{i-1} \sigma_{M_k^*}$; a contradiction. Thus, $F[M_i^*] \setminus \bigcup_{k=1}^{i-1} \sigma_{M_k^*} = \emptyset$ for all $1 \leq i \leq n$. But then, clearly $F[M_i^*] \setminus \bigcup_{k=1}^j \sigma_{M_k^*} = \emptyset$ holds for any $j \geq i - 1$. Thus, $G_j[M_i^*] = H[M_i^*]$ for all $i - 1 \leq j$. \square

The following Lemma shows that, given the explicit order $\mathcal{M} = \{M_1^*, \dots, M_n^*\}$ from Lemma 5.1, in which the edits are applied to the graph G , the intermediate graphs G_i retain all modules of G and also all new modules M_j^* , $j \leq i$.

Lemma 5.2. *Let $G = (V, E)$ be an arbitrary graph, F an optimal module-preserving cograph edit set, and $H = (V, E \triangle F)$ the resulting cograph. Moreover, let $\mathcal{M} = \{M_1^*, \dots, M_n^*\}$ be the partially ordered (w.r.t. inclusion) set of all strong modules of H that are no modules of $G_0 := G$, and choose $\sigma_{M_i^*}$, $F_{M_i^*}$ and the intermediate graphs G_i , $1 \leq i \leq n$ as in Lemma 5.1.*

Then, any module M' of G is a module of G_i and the set M_j^ is a module of G_i for $1 \leq i \leq n$ and any $j \leq i$.*

Proof. First note that $\sigma_{M_i^*}$ affects only modules that are entirely contained in $P_{M_i^*}$ and only their out-neighbors within $P_{M_i^*}$. Moreover $M_j^* \subseteq M_i^*$ implies that $P_{M_j^*} \subseteq P_{M_i^*}$. The partial ordering of the elements in \mathcal{M} implies that $P_{M_i^*}$ remains a module in G_i .

Before we prove the main statement, we show first that the following statement is satisfied:

Claim 1. *For every M' with $M_i^* \subsetneq M' \subsetneq P_{M_i^*}$ we have $M' \neq M_j^* \in \mathcal{M}$, $j \leq i$ and M' cannot be a module of G .*

Proof of Claim 1. Let M' be an arbitrary set with $M_i^* \subsetneq M' \subsetneq P_{M_i^*}$. By the partial order of the elements in \mathcal{M} we immediately observe that $M' \neq M_j^* \in \mathcal{M}$ for any $j \leq i$. Now assume for contradiction that M' is a module of G . Note, all elements in $\mathbb{P}_{\max}(G[P_{M_i^*}])$ are strong modules of G , and thus, do not overlap the module M' . Moreover, since $P_{M_i^*}$ is prime in G , we can apply Theorem 3.3(T2) and conclude that the union of elements of any proper subset $\mathbb{P}' \subsetneq \mathbb{P}_{\max}(G[P_{M_i^*}])$ with $|\mathbb{P}'| > 1$ is not a module of G . Taken the latter arguments together and because $M' \subsetneq P_{M_i^*}$, we have $M' \subseteq M_\ell \in \mathbb{P}_{\max}(G[P_{M_i^*}])$ for some ℓ . Hence, $M_i^* \subsetneq M' \subseteq M_\ell$. However, since M_i^* is the union of some children $\mathbb{P}' \subseteq \mathbb{P}_{\max}(G[P_{M_i^*}])$ of $P_{M_i^*}$ it follows that $M_\ell \subseteq M_i^*$; a contradiction. This proves Claim 1. \triangleleft

We continue with proving the main statement by induction over i . Since $G_0 = G$, the statement is satisfied for G_0 . We continue to show that the statement is satisfied for G_{i+1} under the assumption that it is satisfied for G_i .

For further reference, we note that $P_{M_{i+1}^*}$ is a module of G_i , since $P_{M_{i+1}^*}$ is a module of G and by induction assumption. Moreover, $P_{M_{i+1}^*}$ remains a module of G_{i+1} , since $G_{i+1} = G_i \triangle \sigma_{M_{i+1}^*}$ and $\sigma_{M_{i+1}^*}$ does not affect the $\text{out}_{P_{M_{i+1}^*}}$ -neighborhood. Furthermore, M_{i+1}^* is a module of H and thus, of $H[P_{M_{i+1}^*}]$. Since $\sigma_{M_{i+1}^*}$ contains all such edits to adjust M_{i+1}^* to a module in $H[P_{M_{i+1}^*}]$, we can conclude that M_{i+1}^* is a module in $G_{i+1}[P_{M_{i+1}^*}]$. Therefore, Lemma 3.1 implies that M_{i+1}^* is a module of G_{i+1} .

Now, let M' be an arbitrary module of G . We proceed to show that M' is a module of G_{i+1} . By induction assumption, each module M' of G is a module of G_i . Since F is module-preserving, M' is also a module of H . Hence, $M' \in \text{MD}(G) \cap \text{MD}(G_i) \cap \text{MD}(H)$. Moreover, by Claim 1 the case $M_{i+1}^* \subsetneq M' \subsetneq P_{M_{i+1}^*}$ cannot occur for any module M' of G .

Note, the module M' cannot overlap $P_{M_{i+1}^*}$, since $P_{M_{i+1}^*}$ is strong in G . Hence, for M' one of the following three cases can occur: either $P_{M_{i+1}^*} \subseteq M'$, $P_{M_{i+1}^*} \cap M' = \emptyset$, or $M' \subsetneq P_{M_{i+1}^*}$. In the first two cases, M' remains a module of G_{i+1} , since $\sigma_{M_{i+1}^*}$ contains only edits between vertices within $P_{M_{i+1}^*}$, and thus, the $\text{out}_{M'}$ -neighborhood is not affected. Therefore, assume that $M' \subsetneq P_{M_{i+1}^*}$. The module M' cannot overlap M_{i+1}^* ,

since M_{i+1}^* is strong in H . As shown above, the case $M_{i+1}^* \subsetneq M' \subsetneq P_{M_{i+1}^*}$ cannot occur, and thus we have either (1) $M' \subseteq M_{i+1}^*$, or (2) $M_{i+1}^* \cap M' = \emptyset$.

Case (1): Since $\sigma_{M_{i+1}^*}$ affects only the $\text{out}_{M_{i+1}^*}$ -neighborhood, there is no edit between vertices in M' and $M_{i+1}^* \setminus M'$ and, moreover, $G_{i+1}[M_{i+1}^*] = G_i[M_{i+1}^*]$. By assumption, M' is a module of G_i . Thus, M' is a module in any induced subgraph of G_i that contains M' and hence, in particular in $G_i[M_{i+1}^*]$. Hence, M' is a module of $G_{i+1}[M_{i+1}^*]$. Now, we can apply Lemma 3.1 and conclude that M' is also a module of G_{i+1} .

Case (2): Assume for contradiction that M' is no module of G_{i+1} . Thus, there must be an edge $xy \in E(G_{i+1})$, $x \in M', y \in V \setminus M'$ such that for some other vertex $x' \in M'$ we have $x'y \notin E(G_{i+1})$. Since M' is a module of G_i it must hold that $\{x, y\} \in \sigma_{M_{i+1}^*}$ or $\{x', y\} \in \sigma_{M_{i+1}^*}$. Since $x, x' \notin M_{i+1}^*$ and each edit in $\sigma_{M_{i+1}^*}$ affects a vertex within M_{i+1}^* , we can conclude that $y \in M_{i+1}^*$. Now, by construction of $F_{M_{i+1}^*}$ and since $M' \subsetneq P_{M_{i+1}^*}$, all edits between vertices of M_{i+1}^* and M' are entirely contained in $F_{M_{i+1}^*}$. But this implies that none of the sets $\sigma_{M_\ell^*}$ with $\ell > i + 1$ contains $\{x, y\}$ or $\{x', y\}$. Hence, it holds that $xy \in E(H)$ and $x'y \notin E(H)$, which implies that M' is no module of H ; a contradiction.

Therefore, each module M' of G is a module of G_{i+1} .

We proceed to show that $M_j^* \in \mathcal{M}$ is a module of G_{i+1} for all $j \leq i + 1$. As we have already shown this for $j = i + 1$, we proceed with $j < i + 1$. By induction assumption, each module M_j^* is a module of G_i for all $j < i + 1$. Note, the module M_j^* cannot overlap $P_{M_{i+1}^*}$, since M_j^* is strong in H and $P_{M_{i+1}^*}$ is a module of H , because F is module-preserving. Hence, for M_j^* one of the following three cases can occur: either $P_{M_{i+1}^*} \subseteq M_j^*$, $P_{M_{i+1}^*} \cap M_j^* = \emptyset$, or $M_j^* \subsetneq P_{M_{i+1}^*}$. In the first two cases, M_j^* remains a module of G_{i+1} , since $\sigma_{M_{i+1}^*}$ contains only edits between vertices within $P_{M_{i+1}^*}$, and thus, the $\text{out}_{M_j^*}$ -neighborhood is not affected. Therefore, assume that $M_j^* \subsetneq P_{M_{i+1}^*}$. The module M_j^* cannot overlap M_{i+1}^* , since both are strong in H . Due to the partial ordering of the elements in \mathcal{M} , the case $M_{i+1}^* \subsetneq M_j^*$ cannot occur. Hence there are two cases, either (A) $M_j^* \subseteq M_{i+1}^*$, or (B) $M_{i+1}^* \cap M_j^* = \emptyset$.

Case (A): Since $\sigma_{M_{i+1}^*}$ affects only the $\text{out}_{M_{i+1}^*}$ -neighborhood, there is no edit between vertices in M_j^* and $M_{i+1}^* \setminus M_j^*$. By analogous arguments as in Case (1), we can conclude that M_j^* remains a module of $G_{i+1}[M_{i+1}^*]$. Lemma 3.1 implies that M_j^* is also a module of G_{i+1} .

Case (B): Assume for contradiction that M_j^* is no module of G_{i+1} . Thus, there must be an edge $xy \in E(G_{i+1})$, $x \in M_j^*, y \in V \setminus M_j^*$ such that for some other vertex $x' \in M_j^*$ we have $x'y \notin E(G_{i+1})$. Since M_j^* is a module of G_i it must hold that $\{x, y\} \in \sigma_{M_{i+1}^*}$ or $\{x', y\} \in \sigma_{M_{i+1}^*}$. Now, we can argue analogously as in Case (2) and conclude that $xy \in E(H)$ and $x'y \notin E(H)$, which implies that M_j^* is no module of H ; a contradiction.

Therefore, each module $M_j^*, j \leq i + 1$ is a module of G_{i+1} . □

The latter two Lemmata show that there exists an explicit order, in which all new modules M_i^* of H can be constructed such that whenever a module M_i^* is produced step i the induced subgraph $G_{i-1}[M_i^*]$ is already a cograph and, moreover, is not edited any further in subsequent steps.

5.1 Pairwise module-merge

Regarding Lemma 5.1, each module M_i^* is created by applying the remaining edits $\sigma_{M_i^*} \subseteq F_{M_i^*}$ of the module merge $\sqcup_{M' \in \mathcal{C}(M_i^*)} M' \rightarrow M_i^*$ to the previous intermediate graph G_{i-1} . Now, there might be linear many modules in $\mathcal{C}(M_i^*)$ which have to be merged at once to create M_i^* . However, from an algorithmic point of view the module M_i^* is not known in advance. Hence, in each step, for a given prime module M of G an editing algorithm has to choose one of the exponentially many sets from the power set $\mathcal{P}(\mathbb{P}_{\max} G[M])$ to determine which new module M_i^* have to be created. For an algorithmic approach, however, it would be more convenient to only merge modules in a pairwise manner, since then only quadratic many combinations of choosing two elements of $\mathbb{P}_{\max} G[M]$ have to be considered in each step.

The aim of this section is to show that for each of the n steps of creating one of the new strong modules $\mathcal{M} = \{M_1^*, \dots, M_n^*\}$ of H it is possible to replace the merge operation $\sqcup_{M' \in \mathcal{C}(M_i^*)} M' \rightarrow M_i^*$ with a series of pairwise merge operations.

Before we can state this result we have to define the following partition of strong modules of a resulting cograph H that are no modules of a given graph G .

Definition 5.3. Let $G = (V, E)$ be an arbitrary graph, F a module-preserving cograph edit set, and $H = (V, E \triangle F)$ the resulting cograph. Moreover, let $M^* \in \mathcal{M}$ be a strong module of H that is no module of G and consider the partitions $\mathbb{P}_{\max}(H[M^*]) = \{\widetilde{M}_1, \dots, \widetilde{M}_k\}$ and $\mathcal{C}(M^*) = \{\widehat{M}_1, \dots, \widehat{M}_l\}$. We define with $\mathcal{X}(M^*) = \{M_0, \dots, M_n\}$ the set of modules that contains the maximal (w.r.t. inclusion) modules of $\mathbb{P}_{\max}(H[M_i^*]) \cup \mathcal{C}(M_i^*)$ as follows

$$\begin{aligned} \mathcal{X}(M^*) := & \{\widetilde{M}_i \in \mathbb{P}_{\max}(H[M^*]) \mid \exists \widehat{M}_j \in \mathcal{C}(M^*) \text{ s.t. } \widehat{M}_j \subseteq \widetilde{M}_i\} \\ & \cup \{\widehat{M}_j \in \mathcal{C}(M^*) \mid \exists \widetilde{M}_i \in \mathbb{P}_{\max}(H[M^*]) \text{ s.t. } \widetilde{M}_i \subseteq \widehat{M}_j\}. \end{aligned}$$

Note that for technical reasons the index of the elements in \mathcal{X} starts with 0.

Furthermore, assume that $\mathcal{M} = \{M_1^*, \dots, M_n^*\}$ is a partially ordered (w.r.t. inclusion) set of all strong modules of H that are no modules of G . For each $M_i^* \in \mathcal{M}$ let $\mathcal{X}(M_i^*) = \{M_{i,0}, \dots, M_{i,l_i}\}$ and set $M_i^*(j) = \bigcup_{k=0}^j M_{i,k}$ for all $1 \leq i \leq n$ and $1 \leq j \leq l_i$. Then, we denote with

$$\mathcal{N}(\mathcal{M}) = \{N_1^* = M_1^*(1), \dots, N_m^* = M_n^*(l_n)\}$$

the set of all such $M_i^*(j)$. In particular, we assume that $\mathcal{N}(\mathcal{M})$ is ordered as follows: if $N_k^* = M_i^*(j)$ and $N_l^* = M_{i'}^*(j')$, then $k < l$ if and only if either $i < i'$, or $i = i'$ and $j < j'$, i.e., within $\mathcal{N}(\mathcal{M})$ the elements $M_i^*(j)$ are ordered first w.r.t. i , and second w.r.t. j .

Although, we have already shown by Theorem 4.5 that any new strong module $M^* \in \mathcal{M}$ of H can be obtained by merging the modules from $\mathcal{C}(M^*)$, we will see in the following that M^* can also be obtained by merging the modules from $\mathcal{X}(M^*)$. In particular, we will see that if all elements in $\mathcal{X}(M^*)$ are already modules of the intermediate graph G^* , then we can use any order of the elements within $\mathcal{X}(M^*)$ and successively merge them in a pairwise manner to construct M^* . As a consequence of doing pairwise module merges we obtain in each step an intermediate module $N^* \in \mathcal{N}(\mathcal{M})$.

To see the intention to use the partition $\mathcal{X}(M^*)$ instead of $\mathcal{C}(M^*)$ observe the following. Due to the order of the elements in \mathcal{M} , the modules M_1^*, \dots, M_n^* are constructed

from bottom to top, i.e., when module M^* is processed then all child modules from $\mathbb{P}_{\max}(H[M^*])$ are already constructed. So, instead of obtaining M^* by merging $\mathcal{C}(M^*)$ we can indeed obtain M^* also by merging $\mathbb{P}_{\max}(H[M^*])$. However, it might be the case that a non-trivial subset $\bigcup_{i \in I} \widetilde{M}_i = \widehat{M}_j$ for some j , e.g., if \widehat{M}_j is a (strong) prime module of G but not a strong module of H . But also in this case, we have to assure that \widehat{M}_j remains a module of H . In particular, we do not want to destroy \widehat{M}_j by merging the elements from $\mathbb{P}_{\max}(H[M^*])$ in the incorrect order. Thus, we choose $\widehat{M}_j \in \mathcal{X}(M^*)$ and do not include the individual $\widetilde{M}_i, i \in I$ into $\mathcal{X}(M^*)$.

Before we can continue, we have to show that $\mathcal{X}(M^*)$ as given in Definition 5.3 is indeed a partition of M^* .

Proposition 5.4. *Let $G = (V, E)$ be an arbitrary graph, F a module-preserving cograph edit set, and $H = (V, E \triangle F)$ the resulting cograph. Moreover, let M^* be a strong module of H that is no module of G and consider the partitions $\mathbb{P}_{\max}(H[M^*]) = \{\widetilde{M}_1, \dots, \widetilde{M}_k\}$ and $\mathcal{C}(M^*) = \{\widehat{M}_1, \dots, \widehat{M}_l\}$. Then $\mathcal{X}(M^*)$ is a partition of M^* . As a consequence, for each $M \in \mathcal{X}(M^*)$ there are index sets $I \subseteq \{1, \dots, k\}$ and $J \subseteq \{1, \dots, l\}$ such that $M = \bigcup_{i \in I} \widetilde{M}_i$ and $M = \bigcup_{j \in J} \widehat{M}_j$.*

Proof. First note that all $\widetilde{M}_i \in \mathbb{P}_{\max}(H[M^*])$ are strong modules of H . Moreover, all $\widehat{M}_j \in \mathcal{C}(M^*)$ are strong modules of G . Since F is module-preserving it follows that none of the elements $\widetilde{M}_i \in \mathbb{P}_{\max}(H[M^*])$ overlap any $\widehat{M}_j \in \mathcal{C}(M^*)$, and vice versa. Hence, for each $\widetilde{M}_i \in \mathbb{P}_{\max}(H[M^*])$ there are three distinct cases: Either $\widetilde{M}_i \subseteq \widehat{M}_j$, or $\widehat{M}_j \subsetneq \widetilde{M}_i$, or $\widetilde{M}_i \cap \widehat{M}_j = \emptyset$ for all $\widehat{M}_j \in \mathcal{C}(M^*)$. Now, since $\mathbb{P}_{\max}(H[M^*])$ and $\mathcal{C}(M^*)$ are partitions of M^* it follows for each $x \in M^*$ that x is contained in exactly one $\widetilde{M}_i \in \mathbb{P}_{\max}(H[M^*])$ and exactly one $\widehat{M}_j \in \mathcal{C}(M^*)$ and either $\widetilde{M}_i \subseteq \widehat{M}_j$ or $\widehat{M}_j \subsetneq \widetilde{M}_i$. By construction of $\mathcal{X}(M^*)$ then either $\widetilde{M}_i = \widehat{M}_j \in \mathcal{X}(M^*)$; or $\widetilde{M}_i \in \mathcal{X}(M^*)$ and $\widehat{M}_j \notin \mathcal{X}(M^*)$; or $\widetilde{M}_i \notin \mathcal{X}(M^*)$ and $\widehat{M}_j \in \mathcal{X}(M^*)$. Thus, $\mathcal{X}(M^*)$ is a partition of M^* . \square

Using the partitions $\mathcal{X}(M^*)$, $M^* \in \mathcal{M}$ we now show that there is a sequence of pairwise module merge operations that construct the intermediate modules $N_j^* \in \mathcal{N}(\mathcal{M})$ while keeping all modules from G as well as all previous modules $N_i^*, i < j$.

Lemma 5.5. *Let $G = (V, E)$ be an arbitrary graph, F an optimal module-preserving cograph edit set, $H = (V, E \triangle F)$ the resulting cograph and $\mathcal{M} = \{M_1^*, \dots, M_n^*\}$ be the partially ordered (w.r.t. inclusion) set of all strong modules of H that are no modules of G .*

For each $M_i^ \in \mathcal{M}$ let $\mathcal{X}(M_i^*) = \{M_{i,0}, \dots, M_{i,l_i}\}$ and assume that $\mathcal{N} := \mathcal{N}(\mathcal{M}) = \{N_1^*, \dots, N_m^*\}$. Note, each N_i^* coincides with some $M_i^*(j) = \bigcup_{k=0}^j M_{i,k}$. We define $F_{M_i^*(j)} \subseteq F$ as the set*

$$F_{M_i^*(j)} := \{\{x, v\} \in F \mid x \in M_i^*(j), v \in P_{M_i^*} \setminus M_i^*(j)\}.$$

Furthermore, set $G'_0 = G$ and for each $1 \leq l \leq m$ define $G'_l = G'_{l-1} \triangle \theta_l$ with

$$\theta_l = \begin{cases} \emptyset, & \text{if } N_l^* \text{ is a module of } G'_{l-1} \\ F_{N_l^*} \setminus \bigcup_{k=1}^{l-1} \theta_k, & \text{otherwise.} \end{cases}$$

If N_l^* is no module of G'_{l-1} , then θ_l contains exactly those edits that affect the out-neighborhood of $N_l^* = M_i^*(j)$ within $G[P_{M_i^*}]$ that have not been used so far.

The following statements are true for the intermediate graphs G'_l , $1 \leq l \leq m$:

1. Any set N_k^* is a module of G'_l for all $k \leq l$.
2. Any module M' of G is a module of G'_l , i.e., $\bigcup_{k=1}^l \theta_k$ is module-preserving.
3. Either $G'_{l-1} \simeq G'_l$, or there are two modules $M_1, M_2 \in G'_{l-1}$ such that $M_1 \boxplus M_2 \rightarrow N_l^*$ is a pairwise module merge w.r.t. G'_l .

Proof. Before we start to prove the statements, we will first show

Claim 1. For each $1 \leq l \leq m$ it holds that N_l^* is a module of H .

Proof of Claim 1. By construction $N_l^* = M_i^*(j) = \bigcup_{k=0}^j M_{i,k}$ for some $1 \leq i \leq n$ and $1 \leq j \leq l_i$ with $M_{i,k} \in \mathcal{X}(M_i^*)$. Moreover, for each $M_{i,k}$ it holds either that $M_{i,k} \in \mathbb{P}_{\max} H[M_i^*]$ or $M_{i,k}$ is a union of elements in $\mathbb{P}_{\max} H[M_i^*]$. Therefore, N_l^* is a union of elements in $\mathbb{P}_{\max} H[M_i^*]$. Since M_i^* is a strong non-prime module of H , Theorem 3.3(T3) implies that each union of elements in $\mathbb{P}_{\max} H[M_i^*]$ is a module of H and therefore, N_l^* is a module of H , which proves Claim 1. \triangleleft

We proceed to prove Statements 1 and 2 for each intermediate graph G'_l by induction over l . Since $G'_0 = G$, the Statements 1 and 2 are satisfied for G'_0 . We continue to show that Statements 1 and 2 are satisfied for G'_{l+1} under the assumption that they are satisfied for G'_l .

We start to prove Statement 1. First assume that N_{l+1}^* is already a module of G'_l . Then, by construction it holds that $\theta_{l+1} = \emptyset$ and therefore, $G'_l = G'_{l+1}$. Now, by induction assumption, it holds that all modules of G and all modules $N_k^* \in \mathcal{N}$, $k \leq l$ are modules of $G'_l = G'_{l+1}$. Hence, all modules $N_k^* \in \mathcal{N}$, $k \leq l+1$ are modules of G'_{l+1} . Hence, if N_{l+1}^* is already a module of G'_l , then Statement 1 is satisfied for G'_{l+1} .

Now assume that N_{l+1}^* is not a module of G'_l . For the proof of Statement 1, we show first

Claim 2. N_{l+1}^* is a module of G'_{l+1} .

Proof of Claim 2. By construction it holds that $N_{l+1}^* = M_i^*(j)$ for some $1 \leq i \leq n$ and $1 \leq j \leq l_i$. Note that $P_{M_i^*}$ is a module of G and therefore, by induction assumption it is a module of G'_l . Since $\theta_{l+1} \subseteq F_{M_i^*(j)}$ did only affect the out-neighborhood within the prime module $P_{M_i^*}$ of G it follows that $P_{M_i^*}$ is a module of G'_{l+1} . Moreover, it holds that $F_{M_i^*(j)} \subseteq \bigcup_{k=1}^{l+1} \theta_k$. Note that $F_{M_i^*(j)}$ contains all those edits that affect the out-neighborhood within the prime module $P_{M_i^*}$ of G . Hence, for all $x \in M_i^*(j)$ and all $y \in P_{M_i^*} \setminus M_i^*(j)$ it holds that $xy \in E(H)$ if and only if $xy \in E(G'_{l+1})$. The latter arguments then imply that $M_i^*(j)$ is a module of G'_{l+1} and therefore, N_{l+1}^* is a module of G'_{l+1} . This proves Claim 2. \triangleleft

Now, we proceed with showing

Claim 3. N_k^* , $k \leq l$ is a module of G'_{l+1} .

Proof of Claim 3. Let $N_k^* = M_{i'}^*(j')$ and $N_{l+1}^* = M_i^*(j)$. By induction assumption it holds that N_k^* is a module of G'_l . By the ordering of elements in \mathcal{N} it holds that $i' \leq i$ and by the ordering of elements in \mathcal{M} it then follows that $P_{M_{i'}^*} \subseteq P_{M_i^*}$ or $P_{M_{i'}^*} \cap P_{M_i^*} = \emptyset$.

If $P_{M_{i'}^*} \cap P_{M_i^*} = \emptyset$ then N_k^* is not affected by the edits in θ_{l+1} since they are all within $P_{M_i^*}$ and thus, N_k^* remains a module of G'_{l+1} .

Now consider the case $P_{M_{i'}^*} \subseteq P_{M_i^*}$. For later reference, we show

Claim 3'. $N_k^* \subseteq N_{l+1}^*$ or $N_k^* \cap N_{l+1}^* = \emptyset$.

Proof of Claim 3'. If $i' = i$, then $j' < j$ and by construction, $M_{i'}^*(j') \subseteq M_i^*(j)$ which implies that $N_k^* \subseteq N_{l+1}^*$. Assume now that $i' < i$ and thus, $N_k^* = M_{i'}^*(j') \subseteq M_{i'}^*$. Since M_i^* and $M_{i'}^*$ are strong modules of H they cannot overlap. Therefore, and due to the ordering of the elements in \mathcal{M} it follows that either $M_{i'}^* \subset M_i^*$ or $M_{i'}^* \cap M_i^* = \emptyset$. If $M_{i'}^* \cap M_i^* = \emptyset$, then $N_k^* \cap N_{l+1}^* = \emptyset$. If $M_{i'}^* \subset M_i^*$, then there is a module $M' \in \mathbb{P}_{\max}(H[M_i^*])$ such that $M_{i'}^* \in M'$, since M_i^* and $M_{i'}^*$ are strong modules of H . Furthermore, the set $M_i^*(j)$ is a union of elements in $\mathcal{X}(M_i^*)$ and for each $M_{i,h} \in \mathcal{X}(M_i^*)$ it holds that either $M_{i,h} \in \mathbb{P}_{\max}(H[M_i^*])$ or $M_{i,h}$ is the union of elements in $\mathbb{P}_{\max}(H[M_i^*])$. Hence, it follows that either $M' \subseteq M_i^*(j)$ or $M' \cap M_i^*(j) = \emptyset$. If $M' \cap M_i^*(j) = \emptyset$, then $M_{i'}^*(j') \cap M_i^*(j) = \emptyset$ and hence, $N_k^* \cap N_{l+1}^* = \emptyset$. If, on the other hand, $M' \subseteq M_i^*(j)$, then $M_{i'}^*(j') \subseteq M_i^*(j)$ and thus, $N_k^* \subseteq N_{l+1}^*$. Therefore, in all cases we have either $N_k^* \subseteq N_{l+1}^*$ or $N_k^* \cap N_{l+1}^* = \emptyset$, which proves Claim 3'. \diamond

By Claim 3', we are left with the following two cases.

Case $N_k^* \subseteq N_{l+1}^*$. Since θ_{l+1} did not effect edges within N_{l+1}^* it holds that $G'_l[N_{l+1}^*] \simeq G'_{l+1}[N_{l+1}^*]$. By induction assumption, N_k^* is a module of G'_l and hence, of $G'_l[N_{l+1}^*] = G'_l[M_i^*(j)]$. Thus, N_k^* is a module of $G'_{l+1}[M_i^*(j)]$. Now, since N_{l+1}^* is a module of G'_{l+1} and by Lemma 3.1 it follows that N_k^* is a module of G'_{l+1} .

Case $N_k^* \cap N_{l+1}^* = \emptyset$. Recall that $N_k^* = M_{i'}^*(j')$ and $N_{l+1}^* = M_i^*(j)$ by the fact that $i' \leq i$. Moreover, as shown in the proof of Claim 2, we have $F_{M_{i'}^*(j')} \subseteq \bigcup_{k=1}^{l+1} \theta_k$. Therefore, for all $x \in M_i^*(j)$ and all $y \in M_{i'}^*(j')$ it holds that $xy \in E(H)$ if and only if $xy \in E(G'_{l+1})$. Now let $y, y' \in M_{i'}^*(j')$ and $x \notin M_{i'}^*(j')$. Since $M_{i'}^*(j')$ is a module of H , xy as well as xy' are either both edges H or both are non-edges in H .

If $x \in M_i^*(j)$, then there are no further edits $F \setminus F_{M_i^*(j)}$ that may affect any of these edges, since $F_{M_i^*(j)} \subseteq \bigcup_{k=1}^{l+1} \theta_k$. Thus, $xy \in E(G'_{l+1})$ if and only if $xy' \in E(G'_{l+1})$.

If $x \notin M_i^*(j)$, then xy as well as xy' are not affected by θ_{l+1} . Hence, $xy' \in E(G'_{l+1})$ if and only if $xy' \in E(G'_l)$. By induction assumption, $M_{i'}^*(j')$ is a module of G'_l and hence, $xy \in E(G'_l)$ if and only if $xy' \in E(G'_l)$ and therefore, $xy \in E(G'_{l+1})$ if and only if $xy' \in E(G'_{l+1})$. Hence, $N_k^* = M_{i'}^*(j')$ is a module of G'_{l+1} , which proves Claim 3. \triangleleft

By Claim 1, 2 and 3, Statement 1 is satisfied for G'_{l+1} . We continue to prove Statement 2 and assume that M' is a module of G and by induction assumption M' is a module of G'_l .

Again, let $N_{l+1}^* = M_i^*(j)$ and consider the module $P_{M_i^*}$ of G . Since $P_{M_i^*}$ is strong in G , it cannot overlap M' . Thus, either $M' \cap P_{M_i^*} = \emptyset$, or $P_{M_i^*} \subseteq M'$, or $M' \subset P_{M_i^*}$.

If $M' \cap P_{M_i^*} = \emptyset$ or $P_{M_i^*} \subseteq M'$ then M' is not affected by the edits in θ_{l+1} since they are all within $P_{M_i^*}$ and thus, M' remains a module of G'_{l+1} .

Hence, we only have to consider the case $M' \subset P_{M_i^*}$. We show

Claim 4. *Either $M' \subseteq N_{l+1}^*$ or $M' \cap N_{l+1}^* = \emptyset$.*

Proof of Claim 4. Note again, that the set $M_i^*(j)$ is a union of elements in $\mathcal{X}(M_i^*)$ and for each $M_{i,h} \in \mathcal{X}(M_i^*)$ it holds that either $M_{i,h} \in \mathbb{P}_{\max}(G[P_{M_i^*}])$ or $M_{i,h}$ is the union of elements in $\mathbb{P}_{\max}(G[P_{M_i^*}])$. Hence, $M_i^*(j)$ is a union of elements in $\mathbb{P}_{\max}(G[P_{M_i^*}])$. Theorem 3.3(T2) implies that no union of elements in $\mathbb{P}_{\max}(G[P_{M_i^*}])$ of the prime module $P_{M_i^*}$ is a module of G and thus, $M_i^*(j)$ cannot be a proper subset of M' . Therefore, either $M' \subseteq M_i^*(j)$ or $M' \cap M_i^*(j) = \emptyset$ or M' and $M_i^*(j)$ overlap. However, the latter case cannot occur, since then M' would either overlap one of the strong modules in $\mathbb{P}_{\max}(G[P_{M_i^*}])$ or be a union of elements in $\mathbb{P}_{\max}(G[P_{M_i^*}])$. Thus, in all cases either $M' \subseteq N_{l+1}^*$ or $M' \cap N_{l+1}^* = \emptyset$, which proves Claim 4. \triangleleft

Now the same argumentation that was used to show Statement 1 can be used to show Statement 2. Thus, Statement 2 is satisfied for G'_{l+1} .

Finally, we prove Statement 3. To this end, assume that $G'_l \not\cong G'_{l+1}$ and that N_{l+1}^* is no module of G'_l . We show that there are modules $M_1, M_2 \in G'_l$ with $M_1 \boxplus M_2 \rightarrow N_{l+1}^*$ being a pairwise module merge w.r.t. G'_{l+1} . Clearly, Items (ii) and (iii) of Definition 4.1 are satisfied, since N_{l+1}^* is a module of G'_{l+1} but no module of G'_l . It remains to show that there are two modules $M_1, M_2 \in G'_l$ with $M_1 \cup M_2 = N_{l+1}^*$ and $M_1, M_2 \in G'_{l+1}$, i.e., Item (i) of Definition 4.1 is satisfied. Note, $N_{l+1}^* = M_i^*(j)$ for some i and $j \geq 1$. Assume first that $j = 1$. Then, $M_i^*(1) = M_{i,0} \cup M_{i,1}$ with $M_{i,0}, M_{i,1} \in \mathcal{X}(M_i^*)$. For each $M_{i,h}$ it holds that $M_{i,h} \in \mathbb{P}_{\max}(H[P_{M_i^*}])$ or $M_{i,h} \in \mathbb{P}_{\max}(G[P_{M_i^*}])$. If $M_{i,h} \in \mathbb{P}_{\max}(G[P_{M_i^*}])$ then $M_{i,h}$ is a module of G and by Statement 2, a module of G'_l and G'_{l+1} . If $M_{i,h}$ is no module of G , then $M_{i,h} \in \mathbb{P}_{\max}(H[P_{M_i^*}])$ is a new strong module of H . Therefore, there exists a $k < i$ such that $M_{i,h} = M_k^*$. Since $M_k^* = M_k^*(l_k)$ and by the ordering of elements in \mathcal{N} it holds that $M_k^*(l_k) = N_{k'}^*$ for some $k' \leq l$. Thus, by Statement 1, all $M_{i,h}$ and therefore, $M_{i,0}$ and $M_{i,1}$ are modules of G'_l and G'_{l+1} .

Now, assume that $N_{l+1}^* = M_i^*(j)$ with $j > 1$. Then, $M_i^*(j) = M_i^*(j-1) \cup M_{i,j}$. By the same argumentation as before, it holds that $M_{i,j}$ is a module of G'_l and G'_{l+1} . Moreover, by Statement 1, $M_i^*(j-1) = N_l^*$ is a module of G'_l and G'_{l+1} .

Thus, there are modules M_1, M_2 of G'_l and G'_{l+1} with $M_1 \cup M_2 = N_{l+1}^*$. Moreover, since for all $\{x, y\} \in \theta_{l+1}$ it holds that either $x \in N_{l+1}^*$ and $y \in P_{M_i^*} \setminus N_{l+1}^*$, or vice versa, it follows that there are no additional edits contained in θ_{l+1} besides the edits of the module merge $M_1 \boxplus M_2 \rightarrow N_{l+1}^*$ that transforms G'_l into G'_{l+1} . \square

We are now in the position to derive the main result of this section that shows that optimal pairwise module-merge is always possible.

Theorem 5.6 (Pairwise Module-Merge). *For an arbitrary graph $G = (V, E)$ and an optimal module-preserving cograph edit set F with $H = (V, E \triangle F)$ being the resulting cograph there exists a sequence of pairwise module merge operations that transforms G into H .*

Proof. Set $\mathcal{M} = \{M_1^*, \dots, M_n^*\}$, $\mathcal{N} = \{N_1^*, \dots, N_m^*\}$, $\mathcal{X}(M_i^*) = \{M_{i,0}, \dots, M_{i,l_i}\}$, as well as θ_k and G'_k for all $1 \leq k \leq m$ as in Lemma 5.5. Again, we set $G_0 := G$

and $H' := G_m$. By Lemma 5.5 for each $1 \leq k \leq m$ there is a pairwise module merge $M_1 \uplus M_2 \rightarrow N_k^*$ that transforms G_{k-1} to G_k . Thus, there exists a sequence of module merge operations that transforms G to some graph H' .

In what follows, we will show that $\bigcup_{k=1}^m \theta_k = F$ and therefore $H' \simeq H$, from which we can conclude the statement. For simplicity, we put $F' := \bigcup_{k=1}^m \theta_k$.

We start with showing

Claim 1. $F' \subseteq F$.

Proof of Claim 1. Note first that by construction it holds that $\theta_k \cap \theta_l = \emptyset$ for all $k \neq l$ and therefore, $F' = \bigcup_{k=1}^m \theta_k = \biguplus_{k=1}^m \theta_k$. By construction of θ it holds that $\theta_k \subseteq F$ for all $1 \leq k \leq m$. Hence, $F' \subseteq F$. ◁

Before we show that $F = F'$, we will prove

Claim 2. All strong modules of H are modules of H' .

Proof of Claim 2. Lemma 5.5(1) implies that all modules M' of G are modules of H' . Moreover, Lemma 5.5(2) implies that all $N_k^* \in \mathcal{N}$ are modules of H' . Since for all $M_i^* \in \mathcal{M}$ it holds that $M_i^* = M_i^*(l_i) = N_k^*$ for some $1 \leq k \leq m$, the set M_i^* is a module of H' . Since each strong module of H is either a module of G or a new module $M_i^* \in \mathcal{M}$, all strong modules of H are modules of H' . ◁

We continue to show

Claim 3. $F' \subsetneq F$ is not possible.

Proof of Claim 3. By Claim 1, $F' \subseteq F$. Thus assume for contradiction that $F' \neq F$. Since F is an optimal edit set and $F' \subsetneq F$ it follows that H' is not a cograph. Thus, there exist a prime module M in H' that contains no other prime module.

We will now show that M is a module of H and that all $M_i \in \mathbb{P}_{\max}(H[M])$ are modules of H' . Therefore, consider the strong module P_M of H that entirely contains M and that is minimal w.r.t. inclusion. Since P_M is strong in H it is, by Claim 2, also a module of H' . Moreover, each module $M_i \in \mathbb{P}_{\max}(H[P_M])$ is strong in H and, again by Claim 2, a module of H' as well. If $P_M = M$, then M is a module of H and we are done. Assume now that $M \subsetneq P_M$. Note that since M and all $M_i \in \mathbb{P}_{\max}(H[P_M])$ are modules of H' and M is strong in H' it holds that M does not overlap any $M_i \in \mathbb{P}_{\max}(H[P_M])$. Moreover, $M \not\subseteq M_i$ since otherwise M_i would have been chosen instead of P_M . Thus, $M = \bigcup_{i \in I} M_i$ is the union of some elements M_i in $\mathbb{P}_{\max}(H[P_M])$. Since P_M is a non-prime module of H it follows by Theorem 3.3(T3) that M is a module of H . Since H is a cograph, the children $M_i \in \mathbb{P}_{\max}(H[P_M])$ of the non-prime module P_M are the connected components of either $H[P_M]$ (if P_M is parallel) or its complement $\overline{H[P_M]}$ (if P_M is series). Since $M = \bigcup_{i \in I} M_i$ is the union of some elements in $\mathbb{P}_{\max}(H[P_M])$ and $H[M] \subseteq H[P_M]$, we can conclude that $H[M]$, resp., its complement $\overline{H[M]}$, has as its connected components $M_i, i \in I$. Thus, $\mathbb{P}_{\max}(H[M]) \subset \mathbb{P}_{\max}(H[P_M])$. Hence, all $M_i, i \in I$ are strong modules in H and, by the discussion above, all M_i are modules of H' .

Since all $M_i \in \mathbb{P}_{\max}(H[M])$ are modules of H' and all $M'_j \in \mathbb{P}_{\max}(H'[M])$ are strong in H' , it holds that no $M_i \in \mathbb{P}_{\max}(H[M])$ can overlap any $M'_j \in \mathbb{P}_{\max}(H'[M])$. Therefore, if $M_i \cap M'_j \neq \emptyset$ then either $M'_j \subsetneq M_i$ or $M_i \subseteq M'_j$ for any i and j . If $M'_j \subsetneq M_i$ then M_i must be the union of some elements in $\mathbb{P}_{\max}(H'[M])$. However, since

M is prime in H' no union of elements in $\mathbb{P}_{\max}(H'[M])$, besides M itself, is a module of H' (cf. Theorem 3.3(T2)). Thus, M_i cannot be a module of H' ; a contradiction. Hence, $M_i \subseteq M'_j$ and therefore, each M'_j is the union of some elements in $\mathbb{P}_{\max}(H'[M])$. Note that this holds for any $M'_j \in \mathbb{P}_{\max}(H'[M])$, i.e., there are distinct sets $I_1, \dots, I_{|\mathbb{P}_{\max}(H'[M])|}$ with $I_j \subsetneq \{1, \dots, |\mathbb{P}_{\max}(H'[M])|\}$ such that $M'_j = \bigcup_{i \in I_j} M_i$. Hence, all M'_j are modules of H .

Since, M is prime in H' and M did not contain any other prime module, it holds that all $H'[M'_j]$ are cographs. Moreover, since all M'_j are modules in H and M is prime in H' it holds that there are at least two distinct $M'_k, M'_l \in \mathbb{P}_{\max}(H'[M])$ with $xy \in E(H')$ if and only if $xy \notin E(H)$. Thus, $F'' = \{\{x, y\} \mid x \in M'_k, y \in M'_l\} \subseteq F$. Now, since all $H'[M'_j]$ are cographs it holds that $H'[M'_k \cup M'_l]$ is a cograph.

Now, consider the graph $H'' = G \triangle F \setminus F''$, and in particular the subgraph $H''[M] = G[M] \triangle F[M] \setminus F''$. Again, since all $H'[M'_j]$ with $M'_j \in \mathbb{P}_{\max}(H'[M])$ are cographs it holds that $H[M'_j] \simeq H'[M'_j] \simeq H''[M'_j]$. By construction of F'' for the previously chosen M'_k and M'_l it holds that $H'[M'_k \cup M'_l] \simeq H''[M'_k \cup M'_l]$ as well as $H[M \setminus (M'_k \cup M'_l)] \simeq H''[M \setminus (M'_k \cup M'_l)]$ is a cograph. Moreover, since for all $x \in M'_k \cup M'_l$ and all $y \in M \setminus (M'_k \cup M'_l)$ we have $xy \in E(H)$ if and only if $xy \in E(H'')$ it holds that $H''[M]$ is a cograph as well. Note that $F'' \subseteq F[M]$ and $F'' \neq \emptyset$ and therefore, $|F[M] \setminus F''| < |F[M]|$. But then, since $F[M] \setminus F''$ is an edit set for $G[M]$ and by Lemma 3.9 the set F is not optimal; a contradiction. Thus, F' cannot be a proper subset of F , which proves Claim 3. \square

Claim 1 and 3 immediately imply that $F = F'$. In particular, we have

$$F' = \bigcup_{i=1}^n \bigcup_{j=1}^{l_i} \theta'_{M_i^*(j)} = \bigcup_{i=1}^n \bigcup_{j=1}^{l_i} \theta_{M_i^*(j)} = F. \quad \square$$

It can easily be seen by the latter results that each of the modules in $\mathcal{N}(M) = \{N_1^*, \dots, N_m^*\}$ that is created by a pairwise module merge is either already a module of G , or a union of elements from $\mathbb{P}_{\max}(G[M])$ of some prime module M of G .

5.2 A modular-decomposition-based heuristic for cograph editing

Although the (decision version of the) optimal cograph-editing problem is NP-complete [38, 39], it is fixed-parameter tractable (FPT) [6, 39, 49]. However, the best-known runtime for an FPT-algorithm is $\mathcal{O}(4.612^k + |V|^{4.5})$, where the parameter k denotes the number of edits. These results are of little use for practical applications, because the parameter k can become quite large. An exact algorithm that runs in $\mathcal{O}(3^{|V|}|V|)$ -time is introduced in [53]. Moreover, approximation algorithms are described in [16, 46]. In the following we provide an alternative exact algorithm for the cograph-editing problem based on pairwise module-merge. The virtue of this algorithm is that it can be adopted very easily to design a cograph-editing heuristic.

Algorithm 1 contains two points at which the choice of a particular module or a particular pair of modules affects performance and efficiency. First, the function `get-module-pair()` returns two modules of \mathcal{P} in the correct order of the sequence of pairwise module merge operations that transforms G into H (cf. Theorem 5.6). Second, subroutine `get-module-pair-edit()` is used to compute the edits needed to merge

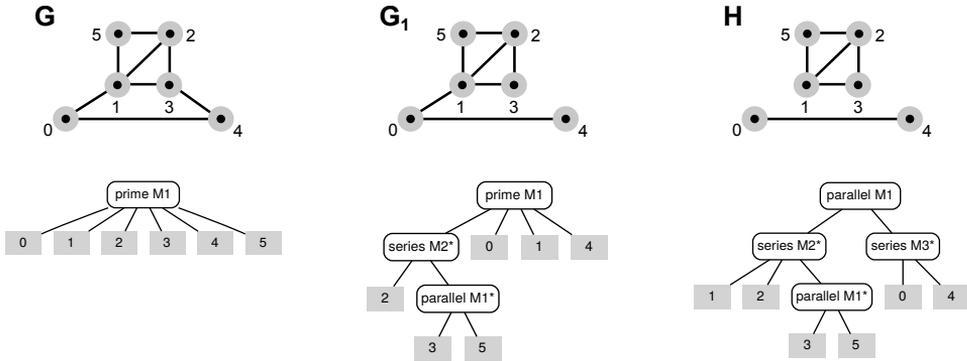


Figure 3: Illustration of Lemma 5.1–5.5, Theorem 5.6 and the exact algorithm. Consider the non-cograph G , the cograph $H = G \triangle F$ and the optimal module-preserving edit set $F = \{\{0, 1\}, \{3, 4\}\}$. The modular decomposition trees are depicted below the respective graphs.

Let $\mathcal{M} = \{M_1^*, M_2^*, M_3^*\}$ be the inclusion-ordered set of strong modules of H that are no modules of G . For all modules $M_i^* \in \mathcal{M}$ the inclusion-minimal module $P_{M_i^*}$ is the prime module M_1 in G .

In compliance with Lemma 5.2 we start with constructing the module M_1^* . By definition $F_{M_1^*} = \{\{3, 4\}\} = \sigma_{M_1^*}$. and we obtain $G_1 = G \triangle \sigma_{M_1^*}$. Thus, $\{3\} \boxplus \{5\} \rightarrow M_1^*$ w.r.t. G_1 . Next, we continue with M_2^* . By construction, $F_{M_2^*} = \{\{0, 1\}, \{3, 4\}\}$ and $\sigma_{M_2^*} = F_{M_2^*} \setminus F_{M_1^*} = \{\{0, 1\}\}$. We then obtain $G_2 = G_1 \triangle \sigma_{M_2^*} = H$. Thus, $\boxplus_{M_i \in \mathcal{C}(M_2^*)} M_i \rightarrow M_2^*$ w.r.t. $G_2 = H$. The module M_3^* is now obtained for free, since $F_{M_3^*} = \{\{0, 1\}, \{3, 4\}\}$ and $\sigma_{M_3^*} = F_{M_3^*} \setminus (F_{M_1^*} \cup F_{M_2^*}) = \emptyset$.

In compliance with Lemma 5.5, i.e., when considering pairwise module merge only, we start with constructing the module $M_1^*(1)$. Here, $\mathcal{X}(M_1^*) = \{M_0 = \{3\}, M_1 = \{5\}\}$ and $M_1^*(1) = \{3, 5\} = M_1^*$. By definition, $F_{M_1^*(1)} = \{\{3, 4\}\} = \theta_{M_1^*(1)}$ and we obtain $G_{1,1} = G_1 = G \triangle \theta_{M_1^*(1)}$. Thus, $\{3\} \boxplus \{5\} \rightarrow M_1^*$ w.r.t. $G_{1,1} = G_1$. Next, we continue with $M_2^*(1)$ and $M_2^*(2)$. Here, $\mathcal{X}(M_2^*) = \{M_0 = \{1\}, M_1 = \{2\}, M_2 = M_1^*\}$ and $M_2^*(1) = \{1\} \cup \{2\}$ and $M_2^*(2) = \{1, 2, 3, 5\} = M_2^*$. By definition $\theta_{M_2^*(1)} = F_{M_2^*(1)} \setminus F_{M_1^*(1)} = \{\{0, 1\}\}$ comprises the edits to obtain the new module $\{1, 2\}$. Thus, $\{1\} \boxplus \{2\} \rightarrow M_2^*(1)$ w.r.t. $G_{2,1}$. Then, since $F_{M_2^*(2)} = F_{M_2^*} = \{\{0, 1\}, \{3, 4\}\}$, we obtain $\theta_{M_2^*(2)} = F_{M_2^*(2)} \setminus (F_{M_1^*(1)} \cup \theta_{M_2^*(1)}) = \emptyset$. Thus, there are no edits left to apply in order to derive at H , since $G_{2,1} = G_{2,2} = G_2 = H$. Again, the module M_3^* is now obtained for free. In all steps, we obtained the new modules by merging pairs of existing modules.

Algorithm 1 Pairwise Module Merge

```

1: INPUT: A graph  $G = (V, E)$ .
2:  $G^* \leftarrow G$ ;
3:  $F^* \leftarrow \emptyset$ ;
4:  $\text{MDs}(G) \leftarrow \text{compute-modular-decomposition}(G)$ .
5:  $P_1, \dots, P_m$  be the prime modules of  $G$  that are partially ordered w.r.t. inclusion, i.e.,  $P_i \subseteq P_j$  implies  $i \leq j$ .
6: for  $p = 1, \dots, m$  do
7:    $\mathcal{P}_p \leftarrow \mathbb{P}_{\max}(G[P_p])$ 
8:   while  $G^*[P_p]$  is not a cograph do
9:      $M_i, M_j \leftarrow \text{get-module-pair}(\mathcal{P}_p)$ . {according to Theorem 5.6}
10:    if  $M_i \cup M_j$  is no module of  $G^*$  then
11:       $\theta \leftarrow \text{get-module-pair-edit}(M_i \uplus M_j \rightarrow N \text{ w.r.t. } G[P_p])$  {according to  $\theta_l$  in Lemma 5.5}
12:       $G^* \leftarrow G^* \Delta \theta$ 
13:    end if
14:     $\mathcal{P}_p \leftarrow \mathcal{P}_p \setminus \{M_i, M_j\} \cup \{N\}$ 
15:  end while
16: end for
17: OUTPUT:  $H = G^*$ ;

```

the modules M_i and M_j to a new module such that these edits affect only the vertices within P_p (cf. Lemma 5.5).

Lemma 5.7. *Let $\mathcal{P}(G)$ be the set of all strong prime modules of G and suppose that Algorithm 1 is applied on the graph G with $n = |V(G)|$. If $\text{get-module-pair}()$ is an “oracle” that always returns the correct pair M_i and M_j and $\text{get-module-pair-edit}()$ returns the correct edit set θ , then Algorithm 1 computes an optimally edited cograph H in $O(m\Lambda h(G)) \leq O(n^2 h(G))$ time, where m denotes the number of strong prime modules in G and $\Lambda = \max_{P \in \mathcal{P}(G)} |\mathbb{P}_{\max}(G[P])|$ is the size of the largest maximal strong partition among all prime modules $P \in \mathcal{P}(G)$, and $h(G)$ is the maximal cost for evaluating $\text{get-module-pair}()$ and $\text{get-module-pair-edit}()$.*

Proof. The correctness of Algorithm 1 follows directly from Lemma 5.5 and Theorem 5.6.

The modular decomposition tree of a graph $G = (V, E)$ can be computed in linear-time, i.e., $O(|V| + |E|) \leq O(n^2)$ with $n = |V(G)|$, see [9, 13, 40, 41, 52]. It yields the partial order P_1, \dots, P_m of the prime modules of G (line 5) in time $O(n)$ by depth first search. Then, we have to resolve each of the m prime modules and in each step in the worst case all modules have to be merged stepwisely, resulting in an effort of $O(|\mathbb{P}_{\max}(G[P_p])|)$ merging steps in each iteration. Since $m \leq n$ and $\Lambda \leq n$ we obtain $O(n^2 h(G))$ as an upper bound. \square

In practice, the exact computation of the optimal editing requires exponential effort. To be more precise, we show now the complexity $h(G)$ as in Lemma 5.7 using a naive brute-force method. Given a prime module P with $\lambda = |\mathbb{P}_{\max}(G[P])|$ child modules there are $\binom{\lambda}{2}$ possibilities for selecting the first module pair that has to be merged. After merging those two modules there are at most $\lambda - 1$ modules left from which possibly two more have to be merged. In general in the i -th merging step there are at most $\binom{\lambda-i}{2}$ possible merge pairs left. This process have to repeat at most $(\lambda - 4)$ times, since any module with less than four child modules cannot be prime. In the worst case this adds up to $\prod_{i=4}^{\lambda} \binom{i}{2} = \prod_{i=4}^{\lambda} \frac{i!}{2!(i-2)!} = \prod_{i=4}^{\lambda} \frac{i \cdot (i-1)}{2}$ merge sequences per prime module of G which

gives $O((\lambda!)^2)$ executions of `get-module-pair()` per prime module in G . Finding the optimal edit set for one merge operation of two modules $M_1, M_2 \in \mathbb{P}_{\max}(G[P])$ requires checking the $2^{\lambda-2}$ combinations to add or remove edges to adjust the out_{M_1} - and out_{M_2} -neighbors w.r.t. to the remaining $\lambda - 2$ modules. Therefore, for each of the remaining modules $M \in \mathbb{P}_{\max}(G[P]) \setminus \{M_1, M_2\}$ there are either only edges or only non-edges between the vertices from M and $M_1 \cup M_2$. In summary, for a given prime module P the graph $G[P]$ can be optimally edited to a cograph in $O((\lambda!)^2 2^\lambda)$ time. Therefore, with $\Lambda = \max_P |\mathbb{P}_{\max}(G[P])|$ being the size of the largest maximal strong partition among all prime modules P of G , it follows that $h(G) \in O((\Lambda!)^2 2^\Lambda)$. We note in passing that Λ is always less than or equal to the maximum degree in the modular decomposition tree, which is also known as *modular-width* [1, 18]. Hence, the latter findings together with Lemma 5.7 imply the following

Observation 5.8. The optimal cograph editing problem parameterized by the modular-width k can be solved in $O((k!)^2 2^k |V|^2)$ time and thus, it is in FPT.

Practical heuristics for `get-module-pair()` and `get-module-pair-edit()` can be implemented to run in polynomial time. In particular, as a main result, we can observe that it is always possible to find an optimal edit set by stepwisely merging only *pairs* of modules. Based on this, we provide in the following several strategies to improve the runtime of these heuristics.

A simple greedy strategy yields a heuristic with $O(|V|^3)$ time complexity as follows: In each call of `get-module-pair()` select the pair (M_i, M_j) in \mathcal{P} where the edit set that adjusts the out_{M_i} - and out_{M_j} -neighbors so that the $\text{out}_{M_i \cup M_j}$ -neighborhood becomes identical in $G^*[P_p]$ has minimum cardinality. This minimum edit set can be obtained from `get-module-pair-edit()` by adjusting only the out-neighbors of the smaller module to be identical to the out-neighbors of the larger module. The pseudocode for this heuristic is given in Algorithm 2 which is, in fact, a natural extension of the exact Algorithm 1. A detailed numerical evaluation will be discussed elsewhere.

Lemma 5.9. Algorithm 2 outputs a cograph and has a time complexity of $O(|V|^3)$.

Proof. First we show that Algorithm 2 constructs a cograph. To this end we show that in each iteration of the main *for*-loop (Lines 16 to 41) the corresponding prime module P_p is edited such that the resulting subgraph $G^*[P_p]$ is a cograph and P_p is still a module of G^* .

Due to the processing order of the prime modules P_1, \dots, P_m constructed in Line 4, we may assume that, upon processing a prime module P_p , the induced subgraphs $G^*[M], M \in \mathbb{P}_{\max}(G[P_p])$ are already cographs and all M are modules of G^* . This holds in particular for the prime modules that do not contain any other prime module in the input graph G and which, therefore, are processed first. Hence, it suffices to show that if all $G^*[M], M \in \mathbb{P}_{\max}(G[P_p])$, are already cographs and all M are modules in G^* , then executing the p -th iteration of the *for*-loop results in an updated intermediate graph G' with $G'[P_p]$ being a cograph and P_p as well as all modules $M \in \mathbb{P}_{\max}(G[P_p])$ remain modules of G' .

In Line 17, we define $\mathcal{P} = \mathbb{P}_{\max}(G[P_p])$ and therefore, by assumption, all $G^*[M], M \in \mathcal{P}$ are cographs and all M are modules of G^* . In particular, the two sets M_i and M_j that are chosen first (in Line 20) are already cographs. Moreover, since M_i and M_j are modules of G^* it follows that $G^*[M_i \cup M_j]$ is either the disjoint union $G^*[M_i] \cup G^*[M_j]$ or the join $G^*[M_i] \oplus G^*[M_j]$ of $G^*[M_i]$ and $G^*[M_j]$. Thus, $G^*[M_i \cup M_j]$ is already a cograph and none of the edges within $M_i \cup M_j$ is edited further. It remains to show that

Algorithm 2 Pairwise Module Merge Heuristic

```

1: INPUT: A graph  $G = (V, E)$ .
2:  $G^* \leftarrow G$ ;
3:  $\text{MDs}(G) \leftarrow \text{compute-modular-decomposition}(G)$ .
4:  $P_1, \dots, P_m$  be the prime modules of  $G$  that are partially ordered w.r.t. inclusion, i.e.,  $P_i \subseteq P_j$  implies  $i \leq j$ .
5:  $A \leftarrow$  zero initialized  $|\text{MDs}(G)| \times |\text{MDs}(G)|$  matrix
6:  $B \leftarrow$  zero initialized  $|\text{MDs}(G)| \times |\text{MDs}(G)| \times |\text{MDs}(G)|$  matrix
7:  $\triangleright$ Lines 8 to 15: Initialize  $A$  where the entries  $A_{ij}$  store the number  $|V \setminus \{M_i \cup M_j\}|$  of vertices that need to be adjusted to merge the modules  $M_i$  and  $M_j$ . Initialize  $B$  s.t.  $B_{ijk} = 1$  iff  $M_i$  and  $M_j$  have different out-neighborhoods w.r.t.  $M_k$ 
8: for each  $\{M_i, M_j, M_k\} \in \binom{\text{MDs}(G)}{3}$  with  $M_i, M_j, M_k$  being children of one and the same prime module  $P$  do
9:   if  $\text{out}_{M_i} \cap M_k \neq \text{out}_{M_j} \cap M_k$  then  $B_{ijk}, B_{jik} \leftarrow 1$  end if
10:  if  $\text{out}_{M_i} \cap M_j \neq \text{out}_{M_k} \cap M_j$  then  $B_{ikj}, B_{kij} \leftarrow 1$  end if
11:  if  $\text{out}_{M_j} \cap M_i \neq \text{out}_{M_k} \cap M_i$  then  $B_{jki}, B_{kji} \leftarrow 1$  end if
12:   $A_{ij}, A_{ji} \leftarrow A_{ij} + |M_k| \cdot B_{ijk}$ 
13:   $A_{ik}, A_{ki} \leftarrow A_{ik} + |M_j| \cdot B_{ikj}$ 
14:   $A_{jk}, A_{kj} \leftarrow A_{jk} + |M_i| \cdot B_{jki}$ 
15: end for
16: for  $p = 1, \dots, m$  do
17:   $\mathcal{P} \leftarrow \mathbb{P}_{\max}(G[P_p])$ 
18:  while  $|\mathcal{P}| > 1$  do
19:     $\theta \leftarrow \emptyset$  ( $\theta$  denotes the set of (non)edges that will be edited)
20:    select two distinct modules  $M_i$  and  $M_j$  from  $\mathcal{P}$  with  $|M_i| \geq |M_j|$  that have a minimum value of  $A_{ij} * |M_j|$ .
21:     $\triangleright$ Line 22 to 26: Compute the edits for adjusting the  $\text{out}_{M_i \cup M_j}$ -neighborhood s.t.  $M_j$  has the same out-neighborhood as  $M_i$  within  $G[P_p]$ . Note, since  $P_p$  is a module of  $G$ ,  $M_j$  and  $M_i$  have the same out-neighbors in  $G$  after editing.
22:    if  $A_{ij} \neq 0$ , i.e.,  $M_i \cup M_j$  is no module of  $G^*$  then
23:      for each  $M_k \in \mathcal{P} \setminus \{M_i, M_j\}$  do
24:        if  $B_{ijk} = 1$  then  $\theta \leftarrow \theta \cup \{xy \mid x \in M_j, y \in M_k\}$  end if
25:      end for
26:      end if
27:       $\triangleright$ Line 28 to 30: Adjust in  $A$  the number of edits needed for merging the new module  $M_i \cup M_j$  with some  $M_k$ 
28:      for each  $M_k \in \mathcal{P} \setminus \{M_i, M_j\}$  do
29:         $A_{ik}, A_{ki} \leftarrow A_{ik} - |M_j| \cdot B_{ikj}$ 
30:      end for
31:       $\triangleright$ Line 32 to 34: Adjust in  $A$  the number of edits needed for merging two modules  $M_k$  and  $M_l$ 
32:      for each  $\{M_k, M_l\} \in \binom{\mathcal{P} \setminus \{M_i, M_j\}}{2}$  do
33:         $A_{kl}, A_{lk} \leftarrow A_{kl} + |M_j| \cdot B_{kli} - |M_j| \cdot B_{klj}$ 
34:      end for
35:      remove the  $j$ -th row and column  $A$ 
36:      remove the  $j$ -th layer in all 3 dimensions of  $B$ 
37:      in  $\mathcal{P}$  replace  $M_i$  with  $M_i \cup M_j$ 
38:       $\mathcal{P} \leftarrow \mathcal{P} \setminus \{M_j\}$ 
39:       $G^* \leftarrow G^* \Delta \theta$ 
40:    end while
41:  end for
42: OUTPUT:  $H = G^*$ ;

```

applying the edits constructed in Line 24 result in the (new) merged module $M_i \cup M_j$ of $G^* \Delta \theta$. Note, if $M_i \cup M_j$ is already a module of G^* then Lines 22 to 26 are not executed and therefore, $\theta = \emptyset$, which implies that $M_i \cup M_j$ remains a module of $G^* \Delta \theta$. On the other hand, if $M_i \cup M_j$ is no module of G^* then the *for*-loop in Lines 12 to 26 iterates over all modules M_k in $\mathcal{P} \setminus \{M_i, M_j\}$ and adjusts the edges between M_j and M_k to be in accordance to the edges between M_i and M_k . Note that all those edits are within P_p . In particular, the $\text{out}_{M_i \cup M_j}$ -neighborhood was adjusted only between vertices from M_j and vertices from $P_p \setminus (M_i \cup M_j)$. After applying these edits, $M_i \cup M_j$ is therefore a module in $G^*[P_p] \Delta \theta$. In particular, the out_{P_p} -neighborhood has not changed and P_p is therefore a module of G^* as well as of $G^* \Delta \theta$. Then, it follows by Lemma 3.1 that $M_i \cup M_j$ is a module in $G^* \Delta \theta$. To see that also all $M_k \in \mathcal{P} \setminus \{M_i, M_j\}$ remain modules in $G^* \Delta \theta$ note first that \mathcal{P} is a partition of P_p and second, that only edges between M_j and M_k are edited for some $M_k \in \mathcal{P} \setminus \{M_i, M_j\}$. Moreover, if a (non)edge between M_j and M_k is edited, then all (non)edges $\{xy \mid x \in M_j, y \in M_k\}$ between M_j and M_k are edited. Thus all $M_k \in \mathcal{P} \setminus \{M_i, M_j\}$ remain modules of $G^*[P_p] \Delta \theta$ and therefore modules $G^* \Delta \theta$.

Now consider the prime module P_{p+1} that is processed in the next iteration of the main *for*-loop. It can be easily seen that for P_{p+1} we also have: $G^*[M], M \in \mathbb{P}_{\max}(G[P_{p+1}])$ is a cograph and all M are modules of G^* , since all prime modules of G that are subsets of P_{p+1} are already processed, and therefore, are all those M are non-prime modules of G^* and form cographs $G^*[M]$. Hence, by the same argumentation as before, $G^*[P_{p+1}]$ is edited to a cograph by the next execution of the main *for*-loop. Thus, after processing all prime modules of G the final graph H is a cograph.

Next, we show that Algorithm 2 has a time complexity of $O(|V|^3)$. Creating the modular decomposition in Line 3 can be done in linear time by the algorithms presented in, e.g., [13, 41, 52]. Note that “linear” in this context means linear in the number of edges, i.e., $O(|V| + |E|) \in O(|V|^2)$. Initializing the matrices A and B (Lines 8 to 15) requires time $O(|V|^3)$ since the corresponding *for*-loop iterates over every ordered set of 3 strong modules of G and there are at most $O(|V|)$ such modules. Moreover, checking if the out-neighborhoods of two modules M_i and M_j w.r.t. a third module M_k are identical (the *if*-statements in Lines 9 to 11) can be done in constant time by checking the adjacencies between three arbitrary vertices, exactly one from each of the three modules. For the remaining Lines 16 to 41 we can consider how often the inner *while*-loop (Lines 18 to 40) is executed. Therefore, note that within each execution always two modules are merged and there are $O(n)$ of those merge operations at most. This can most easily be seen by considering the matrix A which has $\text{MDs}(G)$ rows and columns at first with $|\text{MDs}(G)| < |V|$. Each row, respectively each column, of A represents a module that is possibly selected for merging. Moreover, within each iteration of the *while*-loop, the matrix A is reduced by one row, respectively one column. This leads to no more than $|V|$ many executions of the *while*-loop. Selecting the two modules M_i and M_j in Line 20 requires $O(|V|^2)$ time. Although, the *for*-loop in Lines 23 to 25 is executed $O(|V|)$ times and each partial edit set that is computed in Line 24 might contain more than $O(|V|)$ many edits, the whole edit set θ (constructed within Lines 23 to 25) contains no more than $O(|V|^2)$ edits. Thus, executing Lines 12 to 26 requires $O(|V|^2)$ time at most. Adjusting the matrix A is done in two steps. Lines 28 to 30 iterates over $O(|V|)$ many modules M_k and Lines 32 to 34 iterates over $O(|V|^2)$ many pairs of modules (M_k, M_l) . Shrinking the matrices A and B in Lines 35 and 36 can technically be done in time $O(|V|)$ if we use a labeling function $l: \mathbb{N} \times \mathbb{N}$ to index the values within the matrices, i.e., instead of reading A_{ij} we read $A_{l(i),l(j)}$. Then

we just have to relabel those indices, i.e., $l(x) \leftarrow l(x) + 1$ for all $x > j$. In that way we do not have to remove anything from A or B . Line 37 and 38 can also be done in $O(|V|)$ time and applying the edits in Line 39 requires at most $O(|V|^2)$ time. In summary, executing a single iteration of the main *for*-loop requires $O(|V|^2)$ time, which yields a total time complexity of $O(|V|^3)$. \square

The heuristic as given in Algorithm 2 is deterministic and therefore lacks of a randomization component which would be helpful in order to sample solutions and construct a consensus cograph. However, randomization can be introduced easily by selecting a pair of modules M_i and M_j in line 20 with a probability inversely correlated with the value of $A_{ij} \cdot |M_j|$. Moreover, with probability $p = |M_i|/(|M_i| + |M_j|)$ the edits $\{xy \mid x \in M_j, y \in M_k\}$ can be selected in line 24 and otherwise $\{xy \mid x \in M_i, y \in M_k\}$ with probability $1 - p$.

An even simpler (but probably less accurate) heuristic with time complexity $O(|V|^2)$ can be obtained by randomly selecting the next pair of modules M_i and M_j that have to be merged. Such a procedure would not require the computation of the matrices A and B at all. Nevertheless, this $O(|V|^2)$ -time heuristic requires that computing the edit set θ can be done in $O(|V|)$ time. However, this is possible if we only track the $O(|V|)$ many edits on the corresponding quotient graph $G^*[P_p]/\mathbb{P}_{\max}(G[P_p])$ and recover the $O(|V|^2)$ many individual edits from that only once in a single post-processing step at the end.

Cograph editing heuristics based on the destruction of P4s requires $O(|V|^4)$ time merely for enumerating all P4s. Thus, using module merges as editing operation may lead to significantly faster cograph editing heuristics.

ORCID iDs

Adrian Fritz  <https://orcid.org/0000-0002-9853-5577>

Marc Hellmuth  <https://orcid.org/0000-0002-1620-5508>

Peter F. Stadler  <https://orcid.org/0000-0002-5016-5191>

Nicolas Wieseke  <https://orcid.org/0000-0002-9538-7564>

References

- [1] F. N. Abu-Khzam, S. Li, C. Markarian, F. Meyer auf der Heide and P. Podlipyan, Modular-width: An auxiliary parameter for parameterized parallel complexity, in: M. Xiao and F. A. Rosamond (eds.), *Frontiers in Algorithmics*, Springer International Publishing, Cham, volume 10336 of *Lecture Notes in Computer Science*, 2017 pp. 139–150, doi:10.1007/978-3-319-59605-1_13.
- [2] A. Blass, Graphs with unique maximal clumpings, *J. Graph Theory* **2** (1978), 19–24, doi: 10.1002/jgt.3190020104.
- [3] S. Böcker and A. W. M. Dress, Recovering symbolically dated, rooted trees from symbolic ultrametrics, *Adv. Math.* **138** (1998), 105–125, doi:10.1006/aima.1998.1743.
- [4] A. Brandstädt, V. B. Le and J. P. Spinrad, *Graph Classes: A Survey*, SIAM Monographs on Discrete Mathematics and Applications, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1999, doi:10.1137/1.9780898719796.
- [5] A. Bretscher, D. Corneil, M. Habib and C. Paul, A simple linear time LexBFS cograph recognition algorithm, *SIAM J. Discrete Math.* **22** (2008), 1277–1296, doi:10.1137/060664690.

- [6] L. Cai, Fixed-parameter tractability of graph modification problems for hereditary properties, *Inform. Process. Lett.* **58** (1996), 171–176, doi:10.1016/0020-0190(96)00050-6.
- [7] D. G. Corneil, H. Lerchs and L. Steward Burlingham, Complement reducible graphs, *Discrete Appl. Math.* **3** (1981), 163–174, doi:10.1016/0166-218x(81)90013-5.
- [8] D. G. Corneil, Y. Perl and L. K. Stewart, A linear recognition algorithm for cographs, *SIAM J. Comput.* **14** (1985), 926–934, doi:10.1137/0214065.
- [9] A. Cormier and M. Habib, A new linear algorithm for modular decomposition, in: S. Tison (ed.), *Trees in Algebra and Programming – CAAP '94*, Springer, Berlin, volume 787 of *Lecture Notes in Computer Science*, pp. 68–84, 1994, doi:10.1007/bfb0017474, proceedings of the Nineteenth International Colloquium held in Edinburgh, April 11 – 13, 1994.
- [10] D. D. Cowan, L. O. James and R. G. Stanton, Graph decomposition for undirected graphs, in: F. Hoffman, R. B. Levow and R. S. D. Thomas (eds.), *Proceedings of the Third Southeastern Conference on Combinatorics, Graph Theory and Computing*, Utilitas Mathematica, Winnipeg, 1972 pp. 281–290, held at Florida Atlantic University, Boca Raton, Fla., February 28 – March 2, 1972.
- [11] C. Crespelle, Linear-time minimal cograph editing, 2019, preprint.
- [12] E. Dahlhaus, J. Gustedt and R. M. McConnell, Efficient and practical modular decomposition, in: *SODA '97: Proceedings of the Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1997 pp. 26–35, held in New Orleans, LA, January 5 – 7, 1997.
- [13] E. Dahlhaus, J. Gustedt and R. M. McConnell, Efficient and practical algorithms for sequential modular decomposition, *J. Algorithms* **41** (2001), 360–387, doi:10.1006/jagm.2001.1185.
- [14] R. Dondi, N. El-Mabrouk and M. Lafond, Correction of weighted orthology and paralogy relations – Complexity and algorithmic results, in: M. Frith and C. Storm Pedersen (eds.), *Algorithms in Bioinformatics*, Springer, Cham, volume 9838 of *Lecture Notes in Computer Science*, 2016 pp. 121–136, doi:10.1007/978-3-319-43681-4_10, proceedings of the 16th International Workshop (WABI 2016) held at Aarhus University, Aarhus, August 22 – 24, 2016.
- [15] R. Dondi, M. Lafond and N. El-Mabrouk, Approximating the correction of weighted and unweighted orthology and paralogy relations, *Alg. Mol. Biol.* **12** (2017), Article no. 4 (15 pages), doi:10.1186/s13015-017-0096-x.
- [16] R. Dondi, G. Mauri and I. Zoppis, Orthology correction for gene tree reconstruction: Theoretical and experimental results, *Procedia Comput. Sci.* **108** (2017), 1115–1124, doi:10.1016/j.procs.2017.05.047.
- [17] A. Ehrenfeucht, H. N. Gabow, R. M. McConnell and S. J. Sullivan, An $O(n^2)$ divide-and-conquer algorithm for the prime tree decomposition of two-structures and modular decomposition of graphs, *J. Algorithms* **16** (1994), 283–294, doi:10.1006/jagm.1994.1013.
- [18] J. Gajarský, M. Lampis and S. Ordyniak, Parameterized algorithms for modular-width, in: G. Gutin and S. Szeider (eds.), *Parameterized and Exact Computation*, Springer International Publishing, Cham, volume 8246 of *Lecture Notes in Computer Science*, 2013 pp. 163–176, doi:10.1007/978-3-319-03898-8_15, revised selected papers from the 8th International Symposium (IPEC 2013) held in Sophia Antipolis, September 4 – 6, 2013.
- [19] T. Gallai, Transitiv orientierbare Graphen, *Acta Math. Acad. Sci. Hungar.* **18** (1967), 25–66, doi:10.1007/bf02020961.
- [20] Y. Gao, D. R. Hare and J. Nastos, The cluster deletion problem for cographs, *Discrete Math.* **313** (2013), 2763–2771, doi:10.1016/j.disc.2013.08.017.

- [21] M. Geiß, J. Anders, P. F. Stadler, N. Wieseke and M. Hellmuth, Reconstructing gene trees from Fitch’s xenology relation, *J. Math. Biol.* **77** (2018), 1459–1491, doi:10.1007/s00285-018-1260-8.
- [22] M. Geiß, E. Chávez, M. González Laffitte, A. López Sánchez, B. M. R. Stadler, D. I. Valdivia, M. Hellmuth, M. Hernández Rosales and P. F. Stadler, Best match graphs, *J. Math. Biol.* **78** (2019), 2015–2057, doi:10.1007/s00285-019-01332-9.
- [23] M. Geiß, M. Hellmuth and P. F. Stadler, Reciprocal best match graphs, *J. Math. Biol.* **80** (2020), 865–953, doi:10.1007/s00285-019-01444-2.
- [24] S. Guillemot, F. Havet, C. Paul and A. Perez, On the (non-)existence of polynomial kernels for P_l -free edge modification problems, *Algorithmica* **65** (2013), 900–926, doi:10.1007/s00453-012-9619-5.
- [25] S. Guillemot, C. Paul and A. Perez, On the (non-)existence of polynomial kernels for P_l -free edge modification problems, in: V. Raman and S. Saurabh (eds.), *Parameterized and Exact Computation*, Springer, Berlin, volume 6478 of *Lecture Notes in Computer Science*, pp. 147–157, 2010, doi:10.1007/978-3-642-17493-3_15, proceedings of the 5th International Symposium (IPEC 2010) held in Chennai, December 13 – 15, 2010.
- [26] M. Habib, F. De Montgolfier and C. Paul, A simple linear-time modular decomposition algorithm for graphs, using order extension, in: T. Hagerup and J. Katajainen (eds.), *Algorithm Theory – SWAT 2004*, Springer, Berlin, volume 3111 of *Lecture Notes in Computer Science*, pp. 187–198, 2004, doi:10.1007/978-3-540-27810-8_17.
- [27] M. Habib and M. C. Maurer, On the X -join decomposition for undirected graphs, *Discrete Appl. Math.* **1** (1979), 201–207, doi:10.1016/0166-218x(79)90043-x.
- [28] M. Habib and C. Paul, A survey of the algorithmic aspects of modular decomposition, *Comput. Sci. Rev.* **4** (2010), 41–59, doi:10.1016/j.cosrev.2010.01.001.
- [29] M. Habib, C. Paul and L. Viennot, partition refinement techniques: An interesting algorithmic tool kit, *Internat. J. Found. Comput. Sci.* **10** (1999), 147–170, doi:10.1142/s0129054199000125.
- [30] M. Hellmuth, M. Hernandez-Rosales, K. T. Huber, V. Moulton, P. F. Stadler and N. Wieseke, Orthology relations, symbolic ultrametrics, and cographs, *J. Math. Biol.* **66** (2013), 399–420, doi:10.1007/s00285-012-0525-x.
- [31] M. Hellmuth, P. F. Stadler and N. Wieseke, The mathematics of xenology: Di-cographs, symbolic ultrametrics, 2-structures and tree-representable systems of binary relations, *J. Math. Biol.* **75** (2017), 199–237, doi:10.1007/s00285-016-1084-3.
- [32] M. Hellmuth and N. Wieseke, On symbolic ultrametrics, cotree representations, and cograph edge decompositions and partitions, in: D. Xu, D. Du and D. Du (eds.), *Computing and Combinatorics*, Springer International Publishing, Cham, volume 9198 of *Lecture Notes in Computer Science*, pp. 609–623, 2015, doi:10.1007/978-3-319-21398-9_48, proceedings of the 21st International Conference (COCOON 2015) held in Beijing, August 4 – 6, 2015.
- [33] M. Hellmuth and N. Wieseke, From sequence data including orthologs, paralogs, and xenologs to gene and species trees, in: P. Pontarotti (ed.), *Evolutionary Biology: Convergent Evolution, Evolution of Complex Traits, Concepts and Methods*, Springer, Cham, chapter 21, pp. 373–392, 2016, doi:10.1007/978-3-319-41324-2_21.
- [34] M. Hellmuth and N. Wieseke, On tree representations of relations and graphs: Symbolic ultrametrics and cograph edge decompositions, *J. Comb. Optim.* **36** (2018), 591–616, doi:10.1007/s10878-017-0111-7.

- [35] M. Hellmuth, N. Wieseke, M. Lechner, H.-P. Lenhof, M. Middendorf and P. F. Stadler, Phylogenomics with paralogs, *Proc. Natl. Acad. Sci.* **112** (2015), 2058–2063, doi:10.1073/pnas.1412770112.
- [36] M. Lafond, R. Dondi and N. El-Mabrouk, The link between orthology relations and gene trees: a correction perspective, *Algorithms Mol. Biol.* **11** (2016), Article no. 4 (13 pages), doi:10.1186/s13015-016-0067-7.
- [37] M. Lafond and N. El-Mabrouk, Orthology relation and gene tree correction: complexity results, in: M. Pop and H. Touzet (eds.), *Algorithms in Bioinformatics*, Springer, Heidelberg, volume 9289 of *Lecture Notes in Computer Science*, 2015 pp. 66–79, doi:10.1007/978-3-662-48221-6_5, proceedings of the 15th International Workshop (WABI 2015) held at Georgia Technological Institute, Atlanta, GA, September 10 – 12, 2015.
- [38] Y. Liu, J. Wang, J. Guo and J. Chen, Cograph editing: Complexity and parametrized algorithms, in: B. Fu and D.-Z. Du (eds.), *Computing and Combinatorics*, Springer-Verlag, Berlin, Heidelberg, volume 6842 of *Lecture Notes in Computer Science*, 2011 pp. 110–121, doi:10.1007/978-3-642-22685-4_10.
- [39] Y. Liu, J. Wang, J. Guo and J. Chen, Complexity and parameterized algorithms for cograph editing, *Theoret. Comput. Sci.* **461** (2012), 45–54, doi:10.1016/j.tcs.2011.11.040.
- [40] R. M. McConnell and J. P. Spinrad, Linear-time modular decomposition and efficient transitive orientation of comparability graphs, in: *SODA '94: Proceedings of the Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1994 pp. 536–545, held in Arlington, Virginia, January 23 – 25, 1994.
- [41] R. M. McConnell and J. P. Spinrad, Modular decomposition and transitive orientation, *Discrete Math.* **201** (1999), 189–241, doi:10.1016/s0012-365x(98)00319-7.
- [42] R. M. McConnell and J. P. Spinrad, Ordered vertex partitioning, *Discrete Math. Theor. Comput. Sci.* **4** (2000), 45–60, <https://www.dmtcs.org/dmtcs-ojs/index.php/dmtcs/article/view/113.1.html>.
- [43] R. H. Möhring, Algorithmic aspects of the substitution decomposition in optimization over relations, set systems and Boolean functions, *Ann. Oper. Res.* **4** (1985), 195–225, doi:10.1007/bf02022041.
- [44] R. H. Möhring and F. J. Radermacher, Substitution decomposition for discrete structures and connections with combinatorial optimization, in: R. E. Burkard, R. A. Cuninghame-Green and U. Zimmermann (eds.), *Algebraic and Combinatorial Methods in Operations Research*, North-Holland, Amsterdam, volume 95 of *North-Holland Mathematics Studies*, pp. 257–355, 1984, doi:10.1016/s0304-0208(08)72966-9, proceedings of the Workshop on Algebraic Structures in Operations Research.
- [45] J. H. Müller and J. Spinrad, Incremental modular decomposition, *J. Assoc. Comput. Mach.* **36** (1989), 1–19, doi:10.1145/58562.59300.
- [46] A. Natanzon, R. Shamir and R. Sharan, Complexity classification of some edge modification problems, *Discrete Appl. Math.* **113** (2001), 109–128, doi:10.1016/s0166-218x(00)00391-7, selected papers from the 12th Workshop on Graph-Theoretic Concepts in Computer Science (WG '99).
- [47] N. Nøjgaard, N. El-Mabrouk, D. Merkle, N. Wieseke and M. Hellmuth, Partial homology relations – Satisfiability in terms of di-cographs, in: L. Wang and D. Zhu (eds.), *Computing and Combinatorics*, Springer International Publishing, Cham, volume 10976 of *Lecture Notes in Computer Science*, 2018 pp. 403–415, doi:10.1007/978-3-319-94776-1_34, proceedings of the 24th International Conference (COCOON 2018) held in Qing Dao, China, July 2 – 4, 2018.

- [48] T. Ohtsuki, H. Mori, T. Kashiwabara and T. Fujisawa, On minimal augmentation of a graph to obtain an interval graph, *J. Comput. System Sci.* **22** (1981), 60–97, doi:10.1016/0022-0000(81)90022-2.
- [49] F. Protti, M. Dantas da Silva and J. L. Szwarcfiter, Applying modular decomposition to parameterized cluster editing problems, *Theory Comput. Syst.* **44** (2009), 91–104, doi:10.1007/s00224-007-9032-7.
- [50] D. Seinsche, On a property of the class of n -colorable graphs, *J. Comb. Theory Ser. B* **16** (1974), 191–193, doi:10.1016/0095-8956(74)90063-x.
- [51] C. Semple and M. Steel, *Phylogenetics*, volume 24 of *Oxford Lecture Series in Mathematics and Its Applications*, Oxford University Press, Oxford, UK, 2003.
- [52] M. Tedder, D. Corneil, M. Habib and C. Paul, Simpler linear-time modular decomposition via recursive factorizing permutations, in: L. Aceto, I. Damgård, L. A. Goldberg, M. M. Halldórsson, A. Ingólfssdóttir and I. Walukiewicz (eds.), *Automata, Languages and Programming, Part I*, Springer, Berlin Heidelberg, volume 5125 of *Lecture Notes in Computer Science*, pp. 634–645, 2008, doi:10.1007/978-3-540-70575-8_52, proceedings of the 35th International Colloquium (ICALP 2008) held in Reykjavik, July 7 – 11, 2008.
- [53] W. Timothy J. White, M. Ludwig and S. Böcker, Exact and heuristic algorithms for Cograph Editing, 2018, preprint, [arXiv:1711.05839v3](https://arxiv.org/abs/1711.05839v3) [cs.DS].