



HAL
open science

An $O(\log \text{OPT})$ -Approximation for Covering and Packing Minor Models of θ_r

Dimitris Chatzidimitriou, Jean-Florent Raymond, Ignasi Sau, Dimitrios M.
Thilikos

► **To cite this version:**

Dimitris Chatzidimitriou, Jean-Florent Raymond, Ignasi Sau, Dimitrios M. Thilikos. An $O(\log \text{OPT})$ -Approximation for Covering and Packing Minor Models of θ_r . *Algorithmica*, 2018, 80 (4), pp.1330-1356. 10.1007/s00453-017-0313-5 . lirmm-01609998

HAL Id: lirmm-01609998

<https://hal-lirmm.ccsd.cnrs.fr/lirmm-01609998>

Submitted on 4 Oct 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

An $O(\log \text{OPT})$ -Approximation for Covering and Packing Minor Models of θ_r

Dimitris Chatzidimitriou¹ · Jean-Florent Raymond^{2,3} ·
Ignasi Sau⁴ · Dimitrios M. Thilikos^{1,4}

Received: 26 October 2015 / Accepted: 12 April 2017
© The Author(s) 2017. This article is an open access publication

Abstract Given two graphs G and H , we define $\mathbf{v}\text{-cover}_H(G)$ (resp. $\mathbf{e}\text{-cover}_H(G)$) as the minimum number of vertices (resp. edges) whose removal from G produces a graph without any minor isomorphic to H . Also $\mathbf{v}\text{-pack}_H(G)$ (resp. $\mathbf{e}\text{-pack}_H(G)$) is the maximum number of vertex- (resp. edge-) disjoint subgraphs of G that contain a minor isomorphic to H . We denote by θ_r the graph with two vertices and r parallel edges between them. When $H = \theta_r$, the parameters $\mathbf{v}\text{-cover}_H$, $\mathbf{e}\text{-cover}_H$, $\mathbf{v}\text{-pack}_H$, and $\mathbf{e}\text{-pack}_H$ are NP-hard to compute (for sufficiently big values of r). Drawing upon combinatorial results in Chatzidimitriou et al. (Minors in graphs of large θ_r -girth, 2015, [arXiv:1510.03041](https://arxiv.org/abs/1510.03041)), we give an algorithmic proof that if $\mathbf{v}\text{-pack}_{\theta_r}(G) \leq k$, then $\mathbf{v}\text{-cover}_{\theta_r}(G) = O(k \log k)$, and similarly for $\mathbf{e}\text{-pack}_{\theta_r}$ and $\mathbf{e}\text{-cover}_{\theta_r}$. In other words, the class of graphs containing θ_r as a minor has the vertex/edge Erdős–Pósa property, for every positive integer r . Using the algorithmic machinery of our proofs we introduce a unified approach for the design of an $O(\log \text{OPT})$ -approximation algorithm for $\mathbf{v}\text{-pack}_{\theta_r}$, $\mathbf{v}\text{-cover}_{\theta_r}$, $\mathbf{e}\text{-pack}_{\theta_r}$, and $\mathbf{e}\text{-cover}_{\theta_r}$ that runs in $O(n \cdot \log(n) \cdot m)$ steps. Also, we derive several new Erdős–Pósa-type results from the techniques that we introduce.

Keywords Erdős–Pósa property · Packings in graphs · Coverings in graphs · Minor-models of θ_r · Approximation algorithms · Protrusion decomposition

✉ Jean-Florent Raymond
jean-florent.raymond@mimuw.edu.pl

¹ Department of Mathematics, National and Kapodistrian University of Athens, Athens, Greece

² AIGCo Project Team, LIRMM, Université de Montpellier, Montpellier, France

³ Faculty of Mathematics, Informatics and Mechanics, University of Warsaw, Warsaw, Poland

⁴ AIGCo Project Team, CNRS, LIRMM, Montpellier, France

Mathematics Subject Classification 05C70

1 Introduction

All graphs in this paper are undirected, do not have loops but they may contain multiple edges. We denote by θ_r the graph containing two vertices x and y connected by r parallel edges between x and y . Given a graph class \mathcal{C} and a graph G , we call \mathcal{C} -*subgraph* of G any subgraph of G that is isomorphic to some graph in \mathcal{C} . All along this paper, when giving the running time of an algorithm with input some graph G , we agree that $n = |V(G)|$ and $m = |E(G)|$.

Packings and Coverings Paul Erdős and Lajos Pósa, proved in 1965 [12] that there is a function $f: \mathbb{N} \rightarrow \mathbb{N}$ such that for each positive integer k , every graph either contains k vertex-disjoint cycles or it contains $f(k)$ vertices that intersect every cycle in G . Moreover, they proved that the “gap” of this min-max relation is $f(k) = O(k \cdot \log k)$ and that this gap is optimal. This result initiated an interesting line of research on the duality between coverings and packings of combinatorial objects. To formulate this duality, given a class \mathcal{C} of connected graphs, we define $\mathbf{v}\text{-cover}_{\mathcal{C}}(G)$ (resp. $\mathbf{e}\text{-cover}_{\mathcal{C}}(G)$) as the minimum cardinality of a set S of vertices (resp. edges) such that each \mathcal{C} -subgraph of G contains some element of S . Also, we define $\mathbf{v}\text{-pack}_{\mathcal{C}}(G)$ (resp. $\mathbf{e}\text{-pack}_{\mathcal{C}}(G)$) as the maximum number of vertex- (resp. edge-) disjoint \mathcal{C} -subgraphs of G .

We say that \mathcal{C} has the *vertex Erdős–Pósa property* (resp. the *edge Erdős–Pósa property*) if there is a function $f: \mathbb{N} \rightarrow \mathbb{N}$, called *gap function*, such that, for every graph G , $\mathbf{v}\text{-cover}_{\mathcal{C}}(G) \leq f(\mathbf{v}\text{-pack}_{\mathcal{C}}(G))$ (resp. $\mathbf{e}\text{-cover}_{\mathcal{C}}(G) \leq f(\mathbf{e}\text{-pack}_{\mathcal{C}}(G))$). Using this terminology, the original result of Erdős and Pósa says that the set of all cycles has the vertex Erdős–Pósa property with gap $O(k \cdot \log k)$. The general question in this branch of Graph Theory is to detect instantiations of \mathcal{C} which have the vertex/edge Erdős–Pósa property (in short, *v/e-EP-property*) and when this is the case, minimize the gap function f . Several theorems of this type have been proved concerning different instantiations of \mathcal{C} such as odd cycles [24, 30], long cycles [2], and graphs containing cliques as minors [10] (see also [18, 22, 34] for results on more general combinatorial structures and [32, 33] for surveys).

A general class that is known to have the v-EP-property is the class \mathcal{C}_H of the graphs that contain some fixed planar graph H as a minor.¹ This fact was proven by Robertson and Seymour in [35] and the best known general gap is $f(k) = O(k \cdot \log^{O(1)} k)$ due to the results of [8]—see also [14, 15, 17] for better gaps for particular instantiations of H . Moreover, the planarity of H appears to be the right dichotomy, as for non-planar H 's, \mathcal{C}_H does not satisfy the v-EP-property. Besides the near-optimality of the general upper bound of [8], it is open whether the lower bound $\Omega(k \cdot \log k)$ can be matched for the general gap function, while this is indeed the case when $H = \theta_r$ [14].

The question about classes that have the e-EP-property has also attracted some attention (see [2]). According to [9, Exercise 23 of Chapter 7], the original proof of Erdős and Pósa implies that cycles have the e-EP-property with gap $O(k \cdot \log k)$.

¹ A graph H is a *minor* of a graph G if it can be obtained from some subgraph of G by contracting edges.

Moreover, as proved in [31], the class \mathcal{C}_{θ_r} has the \mathbf{e} -EP-property with the (non-optimal) gap $f(k) = O(k^2 \cdot \log^{O(1)} k)$. Interestingly, not much more is known on the graphs H for which \mathcal{C}_H has the \mathbf{e} -EP-property and it is tempting to conjecture that the planarity of H provides again the right dichotomy. Other graph classes that are known to have the \mathbf{e} -EP-property are rooted cycles [29] (here the cycles to be covered and packed are required to intersect some particular set of terminals of G) and odd cycles for the case where G is a 4-edge connected graph [23], a planar graph [26], or a graph embeddable in an orientable surface [24].

Approximation Algorithms The above defined four graph parameters are already quite general when $\mathcal{C} := \mathcal{C}_H$. From the algorithmic point of view, the computation of \mathbf{x} -pack $_{\mathcal{C}_H}$ (for $\mathbf{x} \in \{\mathbf{v}, \mathbf{e}\}$) corresponds to the general family of graph packing problems, while the computation of \mathbf{x} -cover $_{\mathcal{C}_H}$ belongs to the general family of graph modification problems where the modification operation is the removal of vertices/edges (depending on whether $\mathbf{x} = \mathbf{v}$ or $\mathbf{x} = \mathbf{e}$). Interestingly, particular instantiations of $H = \theta_r$ generate known, well studied, NP-hard problems. For instance, asking whether \mathbf{v} -cover $_{\mathcal{C}_{\theta_r}} \leq k$ generates VERTEX COVER for $r = 1$, FEEDBACK VERTEX SET for $r = 2$, and DIAMOND HITTING SET for $r = 3$ [13, 16]. Moreover, asking whether \mathbf{x} -pack $_{\mathcal{C}_{\theta_2}}(G) \geq k$ corresponds to VERTEX CYCLE PACKING [5, 25] and EDGE CYCLE PACKING [6, 27] when $\mathbf{x} = \mathbf{v}$ and $\mathbf{x} = \mathbf{e}$, respectively. Finally, asking whether $|E(G)| - \mathbf{e}$ -cover $_{\mathcal{C}_{\theta_3}}(G) \leq k$ corresponds to the MAXIMUM CACTUS SUBGRAPH.² All parameters keep being NP-complete to compute because the aforementioned base cases can be reduced to the general one by replacing each edge by one of multiplicity $r - 1$.

From the approximation point of view, it was proven in [16] that, when H is a planar graph, there is a randomized polynomial $O(1)$ -approximation algorithm for \mathbf{v} -cover $_H$. For the cases of \mathbf{v} -cover $_{\mathcal{C}_{\theta_r}}$ and \mathbf{v} -pack $_{\mathcal{C}_{\theta_r}}$, $O(\log n)$ -approximations are known for every $r \geq 1$ because of [21] (see also [36]). Moreover, \mathbf{v} -cover $_{\mathcal{C}_{\theta_r}}$ admits a deterministic 9-approximation [13]. Also, about \mathbf{e} -pack $_{\mathcal{C}_{\theta_r}}(G)$ it is known, from [28], that there is a polynomial $O(\sqrt{\log n})$ -approximation algorithm for the case where $r = 2$. Notice also that for $r = 1$, it is trivial to compute \mathbf{e} -cover $_{\mathcal{C}_{\theta_r}}(G)$ in polynomial time. However, to our knowledge, nothing is known about the computation of \mathbf{e} -cover $_{\mathcal{C}_{\theta_r}}(G)$ for $r \geq 3$.

Our Results In this paper we introduce a unified approach for the study of the combinatorial interconnections and the approximability of the parameters \mathbf{v} -cover $_{\mathcal{C}_{\theta_r}}$, \mathbf{e} -cover $_{\mathcal{C}_{\theta_r}}$, \mathbf{v} -pack $_{\mathcal{C}_{\theta_r}}$, and \mathbf{e} -pack $_{\mathcal{C}_{\theta_r}}$. Our main combinatorial result is the following.

Theorem 1 *For every $r \in \mathbb{N}_{\geq 2}$ and every $\mathbf{x} \in \{\mathbf{v}, \mathbf{e}\}$ the graph class \mathcal{C}_{θ_r} has the \mathbf{x} -EP-property with (optimal) gap function $f(k) = O(k \cdot \log k)$.*

² The MAXIMUM CACTUS SUBGRAPH problem asks, given a graph G and an integer k , whether G contains a subgraph with k edges where no two cycles share an edge. It is easy to reduce to this problem the VERTEX CYCLE PACKING problem on cubic graphs which, in turn, can be proved to be NP-complete using a simple variant of the NP-completeness proof of EXACT COVER BY 2-SETS [19].

Our proof is unified and treats simultaneously the covering and the packing parameters for both the vertex and the edge cases. This verifies the optimal combinatorial bound for the case where $x = v$ [14] and optimally improves the previous bound in [31] for the case where $x = e$. Based on the proof of Theorem 1, we prove the following algorithmic result.

Theorem 2 *For every $r \in \mathbb{N}_{\geq 2}$ and every $x \in \{v, e\}$, there is an algorithm that, given a graph G , outputs an $O(\log \text{OPT})$ -approximation of $x\text{-cover}_{C_{\theta_r}}(G)$ and $x\text{-pack}_{C_{\theta_r}}(G)$ in $O(n \cdot \log(n) \cdot m)$ steps when $x = v$ and in $O(m^2 \log n)$ steps when $x = e$.*

Theorem 2 improves the results in [21] for the cases of $v\text{-cover}_{C_{\theta_r}}$ and $v\text{-pack}_{C_{\theta_r}}$ and, to our knowledge, is the first approximation algorithm for $e\text{-cover}_{C_{\theta_r}}$ and $e\text{-pack}_{C_{\theta_r}}$ for $r \geq 3$. We were also able to derive the following Erdős–Pósa-type result with linear gap on graphs of bounded tree-partition width (the definition of this width parameter is given in Sect. 2.2).

Theorem 3 *Let $t \in \mathbb{N}$. For every $x \in \{v, e\}$, the following holds: if \mathcal{H} is a finite collection of connected graphs and G is a graph of tree-partition width at most t , then $x\text{-cover}_{\mathcal{H}}(G) \leq \alpha \cdot x\text{-pack}_{\mathcal{H}}(G)$, where α is a constant which depends only on t and \mathcal{H} .*

Let $\theta_{r,r'}$ (for some $r, r' \in \mathbb{N}_{\geq 1}$) denote the graph obtained by taking the disjoint union of θ_r and $\theta_{r'}$ and identifying one vertex of θ_r with one of $\theta_{r'}$. Another consequence of our results is that, for every $r, r' \in \mathbb{N}_{\geq 1}$, the class $C_{\theta_{r,r'}}$ has the edge Erdős–Pósa property.

Theorem 4 *For every $r, r' \in \mathbb{N}$, there is a function $f_1^{r,r'} : \mathbb{N}_{\geq 1} \rightarrow \mathbb{N}_{\geq 1}$ such that for every simple graph G where $k = e\text{-pack}_{C_{\theta_{r,r'}}}(G)$, it holds that $e\text{-cover}_{C_{\theta_{r,r'}}}(G) \leq f_1^{r,r'}(k)$.*

Our Techniques Our proofs are based on the notion of *partitioned protrusion* that, roughly, is a tree-structured subgraph of G with small boundary to the rest of G (see Sect. 2.2 for the precise definition). Partitioned protrusions were essentially introduced in [7] by the name *edge-protrusions* and can be seen as the edge-analogue of the notion of protrusions introduced in [4] (see also [3]). Our approach makes strong use of the main result of [7], that is equivalently stated as Proposition 1 in this paper. According to this result, there exists a polynomial algorithm that, given a graph G and an integer k as an input, outputs one of the following: (1) a collection of k edge/vertex disjoint C_{θ_r} -subgraphs of G , (2) a C_{θ_r} -subgraph J with $O(\log k)$ edges, or (3) a large partitioned protrusion of G .

Our approximation algorithm does the following for each $k \leq |V(G)|$. If the first case of the above combinatorial result applies on G , we can safely output a packing of k C_{θ_r} -subgraphs in G . In the second case, we make some progress as we may remove the vertices/edges of J from G and then set $k := k - 1$. In order to deal with the third case, we prove that in a graph G with a sufficiently large partitioned protrusion, we can either find some C_{θ_r} -subgraph with $O(\log k)$ edges (which is the same as in the second

case), or we can replace it by a smaller graph where both $\mathbf{x}\text{-cover}_{\mathcal{H}}$ and $\mathbf{x}\text{-pack}_{\mathcal{H}}$ remain invariant (Lemma 1). The proof that such a reduction is possible is given in Sect. 3 and is based on a suitable dynamic programming encoding of partial packings and coverings that is designed to work on partitioned protrusions.

Notice that the “essential step” in the above procedure is the second case that reduces the packing number of the current graph by 1 to the price of reducing the covering number by $O(\log k)$. This is the main argument that supports the claimed $O(\log \text{OPT})$ -approximation algorithm (Theorem 2) and the corresponding Erdős–Pósa relations in Theorem 1. Finally, Theorem 3 is a combinatorial implication of Lemma 1 and Theorem 4 follows by combining Theorem 3 with the results of Ding and Oporowski in [11].

Organization of the Paper In Sect. 2 we provide all concepts and notation that we use in our proofs. Section 3 contains the proof of Lemma 1, which is the main technical part of the paper. The presentation and analysis of our approximation algorithm is done in Sect. 4, where Theorem 1 and Theorem 2 are proven. Section 5 contains the proofs of Theorem 3 and Theorem 4. Finally, we summarize our results and provide several directions for further research in Sect. 6.

2 Preliminaries

2.1 Basic Definitions

Let $\mathbf{t} = (x_1, \dots, x_l) \in \mathbb{N}^l$ and $\chi, \psi : \mathbb{N} \rightarrow \mathbb{N}$. We say that $\chi(n) = O_{\mathbf{t}}(\psi(n))$ if there exists a computable function $\phi : \mathbb{N}^l \rightarrow \mathbb{N}$ such that $\chi(n) = O(\phi(\mathbf{t}) \cdot \psi(n))$. For every $i, j \in \mathbb{N}$, we denote by $\llbracket i, j \rrbracket$ the interval of $\{i, \dots, j\}$.

Graphs All graphs in this paper are undirected, loopless, and may have multiple edges. For this reason, a graph G is represented by a pair (V, E) where V is its vertex set, denoted by $V(G)$ and E is its edge multi-set, denoted by $E(G)$. The *edge-multiplicity* of an edge of G is the number of times it appears in $E(G)$. We set $n(G) = |V(G)|$ and $m(G) = |E(G)|$. If \mathcal{H} is a finite collection of connected graphs, we set $n(\mathcal{H}) = \sum_{H \in \mathcal{H}} n(H)$, $m(\mathcal{H}) = \sum_{H \in \mathcal{H}} m(H)$, and $\mathbf{U}\mathcal{H} = \bigcup_{G \in \mathcal{H}} G$. The degree of a vertex is the number of vertices adjacent to it. We denote by $\delta(G)$ the *minimum degree* of G .

Let $\mathbf{x} \in \{\mathbf{v}, \mathbf{e}\}$ where, in the rest of this paper, \mathbf{v} will be interpreted as “vertex” and \mathbf{e} will be interpreted as “edge”. Given a graph G , we denote by $A_{\mathbf{x}}(G)$ the set of vertices or edges of G depending on whether $\mathbf{x} = \mathbf{v}$ or $\mathbf{x} = \mathbf{e}$, respectively.

Given a graph H and a graph J that are both subgraphs of the same graph G , we define the subgraph $H \cap G J$ of G as the graph $(V(H) \cap V(J), E(H) \cap E(J))$.

Given a graph G and a set $S \subseteq V(G)$, such that all vertices of S have degree 2 in G , we define $\text{diss}(G, S)$ as the graph obtained from G after we dissolve in it all vertices in S , i.e., replace each maximal path whose internal vertices are in S with an edge whose endpoints are the endpoints of the path.

A *rooted tree* is a pair (T, s) where T is a tree and $s \in V(T)$ is a distinguished vertex called the *root* of T . If $v \in V(T)$, a vertex $u \in V(T)$ is a *descendant* of v in (T, s) if v lies on the (unique) path from u to s . Note that v is a descendant of itself. We denote by $\text{children}_{(T,s)}(v)$ the set of children of v , which are the vertices that are both neighbors and descendants of v .

Given a graph G and an edge $e = (u, v) \in E(G)$, the *subdivision* of e is defined as the operation of removing e and adding a new vertex w and the edges (w, u) and (w, v) . A graph H is called a *subdivision* of G , if it can be obtained by performing a series of subdivisions in G .

Minors and Topological Minors Given two graphs G and H , we say that H is a *minor* of G if there exists some function $\phi : V(H) \rightarrow 2^{V(G)}$ such that

- for every $v \in V(H)$, $G[\phi(v)]$ is connected;
- for every two distinct $v, u \in V(H)$, $\phi(v) \cap \phi(u) = \emptyset$; and
- for every edge $e = \{v, u\} \in E(H)$ of multiplicity l , there are at least l edges in G with one endpoint in $\phi(v)$ and the other in $\phi(u)$.

We say that H is a *topological minor* of G if there exists some collection \mathcal{P} of paths in G and an injection $\phi : V(H) \rightarrow V(G)$ such that

- no path in \mathcal{P} has an internal vertex that belongs to some other path in \mathcal{P} ;
- $\phi(V(H))$ is the set of endpoints of the paths in \mathcal{P} ; and
- for every two distinct $v, u \in V(H)$, $\{v, u\}$ is an edge of H of multiplicity l if and only if there are l paths in \mathcal{P} between $\phi(v)$ and $\phi(u)$.

Given a graph H , we define by $\text{ex}(H)$ the set of all topologically-minor minimal graphs that contain H as a minor. Notice that the size of $\text{ex}(H)$ is upper-bounded by some function of $m(H)$ and that H is a minor of G if it contains a member of $\text{ex}(H)$ as a topological minor. An *H-minor model* of G is any minimal subgraph of G that contains a member of $\text{ex}(H)$ as a topological minor.

Subdivisions The subdivision of an edge $\{u, v\}$ of a graph is the operation that introduces a new vertex w to the graph adjacent to u and v and deletes the edge. A graph G is a subdivision of H if it can be obtained from H by repeatedly subdividing edges. If G is a graph and \mathcal{H} is a finite collection of connected graphs, an \mathcal{H} -subdivision of G is a subgraph M of G that is a subdivision of a graph, denoted by \hat{M} , that is isomorphic to a member of \mathcal{H} . Clearly, the vertices of \hat{M} are vertices of G and its edges correspond to paths in G between their endpoints such that internal vertices of a path do not appear in any other path. We refer to the vertices of \hat{M} in G as the *branch vertices* of the \mathcal{H} -subdivision M , whereas internal vertices of the paths between branch vertices will be called *subdivision vertices* of M . Note that we are here dealing with what is sometimes called a topological minor model. A graph which contains no \mathcal{H} -subdivision is said to be \mathcal{H} -free. Notice that G has an \mathcal{H} -subdivision iff G contains a graph of \mathcal{H} as a topological minor.

Packings and Coverings An \mathbf{x} - \mathcal{H} -packing of size k of a graph G is a collection \mathcal{P} of k pairwise \mathbf{x} -disjoint \mathcal{H} -subdivisions of G . Given an $\mathbf{x} \in \{\mathbf{v}, \mathbf{e}\}$, we define $\mathbf{P}_{\mathbf{x}, \mathcal{H}}^{\geq k}(G)$

as the set of all x - \mathcal{H} -packings of G of size at least k . An x - \mathcal{H} -covering of a graph G is a set $C \subseteq A_x(G)$ such that $G \setminus C$ does not contain any \mathcal{H} -subdivision. We define $C_{x,\mathcal{H}}^{\leq k}(G)$ as the set of all x - \mathcal{H} -coverings of G of size at most k . We finally define

$$x\text{-cover}_{\mathcal{H}}(G) = \min \left\{ k \mid C_{x,\mathcal{H}}^{\leq k}(G) \neq \emptyset \right\}$$

and

$$x\text{-pack}_{\mathcal{H}}(G) = \max \left\{ k \mid P_{x,\mathcal{H}}^{\geq k}(G) \neq \emptyset \right\}.$$

These numbers are well-defined: G always has a packing of size 0 (the empty packing) and a covering of size $|A_x(G)|$ (which contains all vertices/edges of G). It is easy to observe that, for every graph G and every finite collection of connected graphs \mathcal{H} , the following inequalities hold:

$$\begin{aligned} v\text{-cover}_{\mathcal{H}}(G) &\leq e\text{-cover}_{\mathcal{H}}(G), & v\text{-pack}_{\mathcal{H}}(G) &\leq e\text{-pack}_{\mathcal{H}}(G), \\ v\text{-pack}_{\mathcal{H}}(G) &\leq v\text{-cover}_{\mathcal{H}}(G), & e\text{-pack}_{\mathcal{H}}(G) &\leq e\text{-cover}_{\mathcal{H}}(G). \end{aligned}$$

Notice that the definition of these invariants differs from that given in the introduction.

2.2 Boundaried Graphs

Informally, a boundaried graph will be used to represent a graph that has been obtained by “dissecting” a larger graph along some of its edges, where the boundary vertices correspond to edges that have been cut.

Boundaried Graphs A boundaried graph $\mathbf{G} = (G, B, \lambda)$ is a triple consisting of a graph G , where B is a set of vertices of degree one (called *boundary*) and λ is a bijection from B to a subset of $\mathbb{N}_{\geq 1}$. The edges with at least one endpoint in B are called *boundary edges*. We define $E^s(G)$ as the subset of $E(G)$ of boundary edges. We stress that instead of $\mathbb{N}_{\geq 1}$ we could choose any other set of symbols to label the vertices of B . We denote the set of labels of \mathbf{G} by $\Lambda(\mathbf{G}) = \lambda(B)$. Given a finite collection of connected graphs \mathcal{H} , we say that a boundaried graph \mathbf{G} is \mathcal{H} -free if $G \setminus B$ is \mathcal{H} -free.

Two boundaried graphs \mathbf{G}_1 and \mathbf{G}_2 are *compatible* if $\Lambda(\mathbf{G}_1) = \Lambda(\mathbf{G}_2)$. Let now $\mathbf{G}_1 = (G_1, B_1, \lambda_1)$ and $\mathbf{G}_2 = (G_2, B_2, \lambda_2)$ be two compatible boundaried graphs. We define the graph $\mathbf{G}_1 \oplus \mathbf{G}_2$ as the non-boundaried graph obtained by first taking the disjoint union of G_1 and G_2 , then, for every $i \in \Lambda(\mathbf{G}_1)$, identifying $\lambda_1^{-1}(i)$ with $\lambda_2^{-1}(i)$, and finally dissolving all resulting identified vertices. Suppose that e is an edge of $G = \mathbf{G}_1 \oplus \mathbf{G}_2$ that was created after dissolving the vertex resulting from the identification of a vertex v_1 in B_1 and a vertex v_2 in B_2 and that e_i is the boundary edge of G_i that has v_i as endpoint, for $i = 1, 2$.

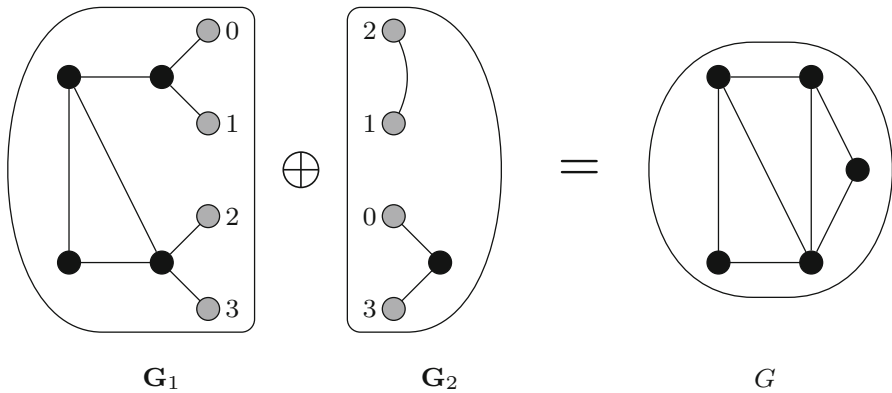


Fig. 1 Gluing graphs together: $G = G_1 \oplus G_2$

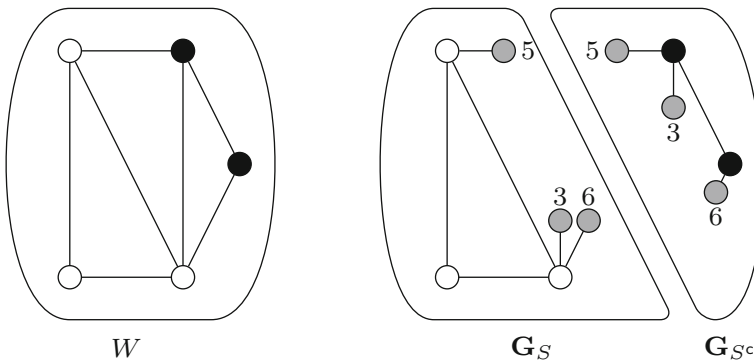


Fig. 2 Cutting a graph: (G_S, G_{S^c}) is an S -splitting of W , where S consists of all the white vertices

Figure 1 shows the result of the operation \oplus on two graphs. Boundaries are drawn in gray and their labels are written next to them. The graphs G_1 and G_2 on this picture are compatible as $\Lambda(G_1) = \Lambda(G_2) = \{0, 1, 2, 3\}$.

For every $t \in \mathbb{N}_{\geq 1}$, we denote by \mathcal{B}_t all boundaried graphs whose boundary is labeled by numbers in $\llbracket 1, t \rrbracket$. Given a boundaried graph $\mathbf{G} = (G, B, \lambda)$ and a subset S of $V(G)$ such that all vertices in S have degree 2 in G , we define $\text{diss}(\mathbf{G}, S)$ as the graph $\hat{\mathbf{G}} = (\hat{G}, B, \lambda)$ where $\hat{G} = \text{diss}(G, S)$.

Let W be a graph and S be a non-empty subset of $V(W)$. An S -splitting of W is a pair $(\mathbf{G}_S, \mathbf{G}_{S^c})$ consisting of two boundaried graphs $\mathbf{G}_S = (G_S, B_S, \lambda_S)$ and $\mathbf{G}_{S^c} = (G_{S^c}, B_{S^c}, \lambda_{S^c})$ that can be obtained as follows: First, let W^+ be the graph obtained by subdividing in W every edge with one endpoint in S and the other in $V(W) \setminus S$ and let B be the set of created vertices. Let λ be any bijection from B to a subset of $\mathbb{N}_{\geq 1}$. Then $G_S = W^+[S \cup B]$, $G_{S^c} = W^+ \setminus S$, $B_S = B_{S^c} = B$, and $\lambda_S = \lambda_{S^c} = \lambda$. Notice that there are infinite such pairs, depending on the numbers that will be assigned to the boundaries of \mathbf{G}_S and \mathbf{G}_{S^c} . Moreover, keep in mind that all the boundary edges of G_S are non-loop edges with exactly one endpoint in B and the same holds for the boundary edges of G_{S^c} . An example of a splitting is given in Fig. 2, where boundaries are depicted by gray vertices.

We say that $\mathbf{G}' = (G', B', \lambda')$ is a *boundaried subgraph* of $\mathbf{G} = (G, B, \lambda)$ if G' is a subgraph of G , $B' \subseteq B \cap V(G')$ and $\lambda' = \lambda|_{B'}$. On the other hand, \mathbf{G} is a subgraph of a (non-boundaried) graph H if $\mathbf{G} = \mathbf{H}_S$ for some S -splitting $(\mathbf{H}_S, \mathbf{H}_{S^c})$, where $S \subseteq V(H)$.

If H is a graph, G is a subgraph of H , and $\mathbf{F} = (F, B, \lambda)$ is a boundaried subgraph of H , we define $G \cap_H \mathbf{F}$ as follows. Let $S = V(G) \cap (V(F) \setminus B)$ and let G^+ be the graph obtained by subdividing once every edge of G that has one endpoint in S and the other in $V(G) \setminus S$. We call B' the set of created vertices and let $G' = G^+[S \cup B']$. Then G' is a subgraph of F where $B' \subseteq B$. For every $v \in B'$, we set $\lambda'(v) = \lambda(v)$, which is allowed according to the previous remark. Then $G \cap_H \mathbf{F} = (G', B', \lambda')$. Observe that $G \cap_H \mathbf{F}$ is part of an S -splitting of G (but not necessarily of H).

Given two boundaried graphs $\mathbf{G}' = (G', B', \lambda')$ and $\mathbf{G} = (G, B, \lambda)$, we say that they are *isomorphic* if there is an isomorphism from G' to G that respects the labelings of B and B' , i.e., maps every vertex $x \in B'$ to $\lambda^{-1}(\lambda'(x)) \in B$. Given a boundaried graph $\mathbf{G} = (G, B, \lambda)$, we denote $n(\mathbf{G}) = n(G) - |B|$ and $m(\mathbf{G}) = m(G)$.

Given a boundaried graph $\mathbf{G} = (G, B, \lambda)$ and an $\mathbf{x} \in \{\mathbf{v}, \mathbf{e}\}$, we set $A_{\mathbf{x}}(\mathbf{G}) = V(G) \setminus B$ or $A_{\mathbf{x}}(\mathbf{G}) = E(G)$, depending on whether $\mathbf{x} = \mathbf{v}$ or $\mathbf{x} = \mathbf{e}$.

Partial Structures Let \mathcal{H} be a family of graphs. A boundaried subgraph \mathbf{J} of a boundaried graph \mathbf{G} is a *partial \mathcal{H} -subdivision* if there is a boundaried graph \mathbf{H} which is compatible with \mathbf{G} and a boundaried subgraph \mathbf{J}' of \mathbf{H} which is compatible with \mathbf{J} such that $\mathbf{J} \oplus \mathbf{J}'$ is an \mathcal{H} -subdivision of $\mathbf{G} \oplus \mathbf{H}$. Intuitively, this means that \mathbf{J} can be extended into an \mathcal{H} -subdivision in some larger graph. In this case, the \mathcal{H} -subdivision $\mathbf{J} \oplus \mathbf{J}'$ is said to be an *extension* of \mathbf{J} .

Similarly, for every $p \in \mathbb{N}_{\geq 1}$, a collection of boundaried subgraphs $\mathcal{J} = \{\mathbf{J}_1, \dots, \mathbf{J}_p\}$ of a graph \mathbf{G} is a *partial \mathbf{x} - \mathcal{H} -packing* if there is a boundaried graph \mathbf{H} which is compatible with \mathbf{G} and a collection of boundaried subgraphs $\{\mathbf{J}'_1, \dots, \mathbf{J}'_p\}$ of \mathbf{H} such that $\{\mathbf{J}_1 \oplus \mathbf{J}'_1, \dots, \mathbf{J}_p \oplus \mathbf{J}'_p\}$ is an \mathbf{x} - \mathcal{H} -packing of $\mathbf{G} \oplus \mathbf{H}$. The obtained packing is said to be an *extension* of the partial packing \mathcal{J} . A partial packing is *\mathcal{H} -free* if none of its members (the boundaried graphs $\{\mathbf{J}_1, \dots, \mathbf{J}_p\}$) has an \mathcal{H} -subdivision. Observe that if the graphs in \mathcal{H} are connected, then every partial subdivision of an \mathcal{H} -free partial \mathbf{x} - \mathcal{H} -packing in \mathbf{G} must contain at least one boundary vertex of \mathbf{G} .

Partitions and Protrusions A *rooted tree-partition* of a graph G is a triple $\mathcal{D} = (\mathcal{X}, T, s)$ where (T, s) is a rooted tree and $\mathcal{X} = \{X_t\}_{t \in V(T)}$ is a partition of $V(G)$ where either $n(T) = 1$ or for every $\{x, y\} \in E(G)$, there exists an edge $\{t, t'\} \in E(T)$ such that $\{x, y\} \subseteq X_t \cup X_{t'}$ (see also [11, 20, 37]). Given an edge $f = \{t, t'\} \in E(T)$, we define E_f as the set of edges of G with one endpoint in X_t and the other in $X_{t'}$. Notice that all edges in E_f are non-loop edges. The *width* of \mathcal{D} is defined as

$$\max\{|X_t|_{t \in V(T)}\} \cup \{|E_f|_{f \in E(T)}\}.$$

The *tree-partition width* of G is the minimum width over all tree-partitions of G and will be denoted by $\mathbf{tpw}(G)$. The elements of \mathcal{X} are called *bags*.

In order to decompose graphs along edge cuts, we introduce the following edge-counterpart of the notion of (*vertex-*)*protrusion* introduced in [3, 4].

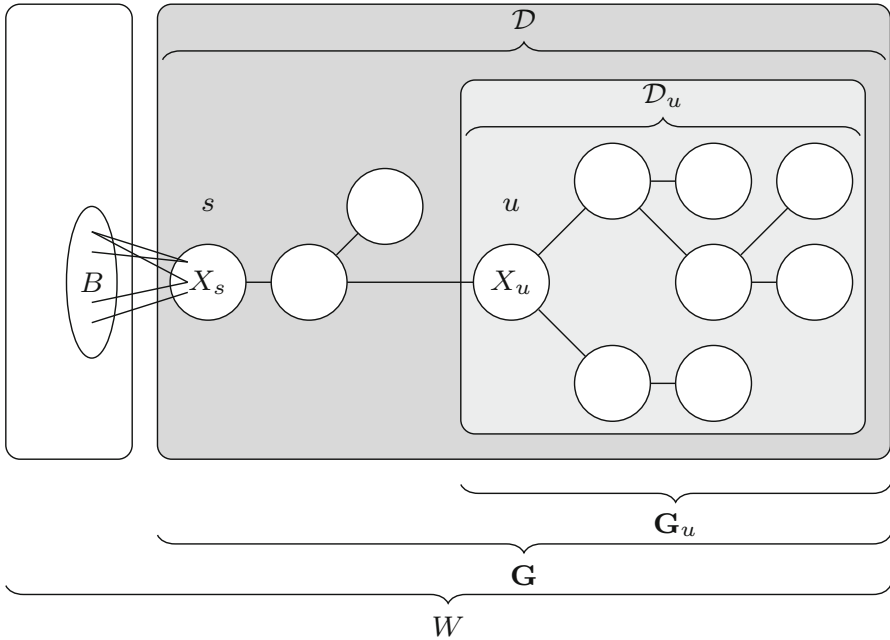


Fig. 3 The partitioned protrusion $\mathbf{P}_u = (\mathbf{G}_u, \mathcal{D}_u)$ as a substructure of $(\mathbf{G}, \mathcal{D})$. White circles represent bags of the rooted tree-partitions. Only the edges between B and X_s are depicted

Given a rooted tree-partition $\mathcal{D} = (\mathcal{X}, T, s)$ of G and a vertex $i \in V(T)$, we define T_i as the subtree of T induced by the descendants of i (including i) and

$$V_i = \bigcup_{h \in V(T_i)} X_h, \text{ and } G_i = G[V_i].$$

Let W be a graph and $t \in \mathbb{N}_{\geq 1}$. A pair $\mathbf{P} = (\mathbf{G}, \mathcal{D})$ is a t -partitioned protrusion of W if there exists an $S \subseteq V(W)$ such that

- $\mathbf{G} = (G, B, \lambda)$ is a boundaried graph where $\mathbf{G} \in \mathcal{B}_t$ and $\mathbf{G} = \mathbf{G}_S$ for some S -splitting $(\mathbf{G}_S, \mathbf{G}_{S^c})$ of W and
- $\mathcal{D} = (\mathcal{X}, T, s)$ is a rooted tree-partition of $G \setminus B$ of width at most t , where X_s are the neighbors in G of the vertices in B .

We say that a t -partitioned protrusion $(\mathbf{G}, \mathcal{D})$ (with $\mathcal{D} = (\{X_u\}_{u \in V(T)}, T, s)$) of a graph W is \mathcal{H} -free if \mathbf{G} is \mathcal{H} -free. Recall that in the rooted tree-partition \mathcal{D} , for every $u \in V(T)$, we denote by V_u the union of the bags indexed by descendants of u (including u). For every vertex $u \in V(T)$, we define the t -partitioned protrusion \mathbf{P}_u of W as a pair $\mathbf{P}_u = (\mathbf{G}_u, \mathcal{D}_u)$, where $\mathcal{D}_u = (\{X_v\}_{v \in V_{T_u}}, T_u, u)$ and \mathbf{G}_u is defined as \mathbf{G}_{V_u} for some V_u -splitting $(\mathbf{G}_{V_u}, \mathbf{G}_{V_u^c})$ of W . Informally speaking, the rooted tree-partition \mathcal{D}_u is the part of \mathcal{D} corresponding to the subgraph of \mathbf{G} induced by vertices in the bags that are descendants of u . Figure 3 provides an example of partitioned protrusions. We choose the labeling function of \mathbf{G}_s so that it is the same as the one

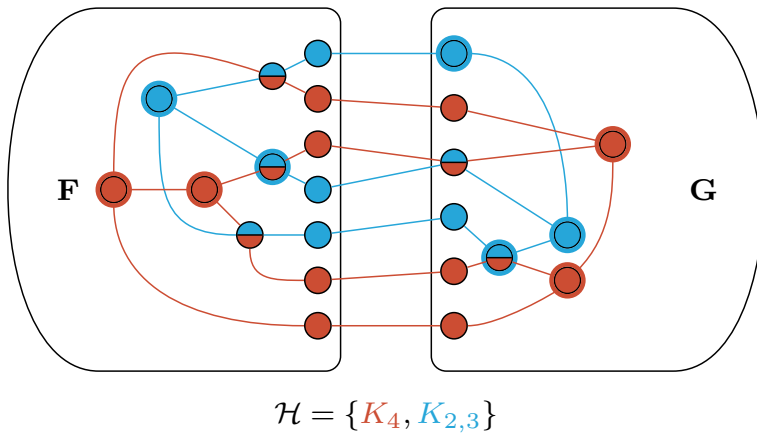
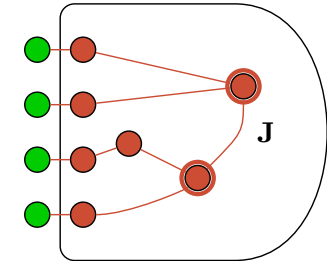


Fig. 4 A member of $\mathbf{P}_{\mathbf{e}, \mathcal{H}}^{\geq 1}(\mathbf{F} \oplus \mathbf{G})$. Branch vertices are *circled*

Fig. 5 A partial subdivision from the packing of Fig. 4. Branch vertices are *circled*



of \mathbf{G} , i.e., $\mathbf{G}_s = \mathbf{G}$. Notice that the labelings of all other \mathbf{G}_u 's are arbitrary. For every $u \in V(T)$ we define

$$\mathcal{G}_u = \{\mathbf{G}_l \mid l \in \text{children}_{(T,s)}(u)\}.$$

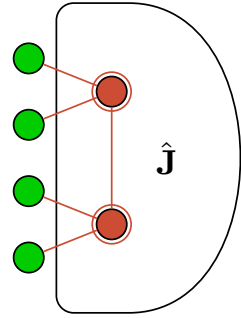
2.3 Encodings, Signatures, and Folios

In this section we introduce tools that we will use to sort boundaried graphs depending on the subdivisions that are realizable inside.

Encodings Let \mathcal{H} be a family of graphs, let $t \in \mathbb{N}_{\geq 1}$, and let $\mathbf{x} \in \{\mathbf{v}, \mathbf{e}\}$. Recall that if M is an \mathcal{H} -subdivision in G , we denote by \hat{M} a graph isomorphic to some graph of \mathcal{H} such that M is a subdivision of \hat{M} (see Sect. 2.1). If $\mathbf{G} = (G, B, \lambda) \in \mathcal{B}_t$ is a boundaried graph and $S \subseteq A_{\mathbf{x}}(\mathbf{G})$, we define $\mu_{\mathcal{H}}^{\mathbf{x}}(\mathbf{G}, S)$ as the collection of all sets $\{(\mathbf{J}_1, L_1), \dots, (\mathbf{J}_\sigma, L_\sigma)\}$ such that

- (i) $\{\mathbf{J}_1, \dots, \mathbf{J}_\sigma\}$ is a partial \mathbf{x} - \mathcal{H} -packing of $\mathbf{G} \setminus S$ of size σ and
- (ii) $L_i = V(\hat{M}_i) \cap V(G)$, where \hat{M}_i is an extension of \mathbf{J}_i , for every $i \in \llbracket 1, \sigma \rrbracket$.

Fig. 6 The compression of the partial packing of Fig. 5:
 $\mathbf{J} = \kappa(\mathbf{J}, L)$



In other words, L_i contains branch vertices of the partial subdivision \mathbf{J}_i for every $i \in \llbracket 1, \sigma \rrbracket$ (see Figs. 4, 5). The set $\mu_{\mathcal{H}}^{\mathbf{x}}(\mathbf{G}, S)$ encodes all different restrictions in \mathbf{G} of partial \mathbf{x} - \mathcal{H} -packings that avoid the set S . Given a boundaried graph $\mathbf{G} = (G, B, \lambda)$ and a set $L \subseteq V(G)$ such that every vertex of $V(G) \setminus L$ has degree 2 in G , we define $\kappa(\mathbf{G}, L)$ as the boundaried graph obtained from \mathbf{G} by dissolving every vertex of $V(G) \setminus L$, i.e., $\kappa(\mathbf{G}, L) = (\text{diss}(\mathbf{G}, V(G) \setminus L), B, \lambda)$. In the definition of κ we assume that the boundary vertices of $\kappa(\mathbf{G}, L)$ remain the same as in \mathbf{G} while the other vertices are treated as new vertices (see Fig. 6).

This allows us to introduce the following notation aimed at representing, intuitively, the essential part of each partial packing.

$$\hat{\mu}_{\mathcal{H}}^{\mathbf{x}}(\mathbf{G}, S) = \{\hat{\mathcal{J}} = \{\hat{\mathbf{J}}_1, \dots, \hat{\mathbf{J}}_{\sigma}\} = \{\kappa(\mathbf{J}_1, L_1), \dots, \kappa(\mathbf{J}_{\sigma}, L_{\sigma})\} \mid \{\mathbf{J}_1, L_1, \dots, \mathbf{J}_{\sigma}, L_{\sigma}\} \in \mu_{\mathcal{H}}(\mathbf{G}, S)\}.$$

Isomorphisms If $\mathbf{G} = (G, B, \lambda)$ and $\mathbf{G}' = (G', B', \lambda')$ are two compatible boundaried graphs in \mathcal{B}_t , $S \in V(G)$, and $S' \in V(G')$, we say that a member $\hat{\mathcal{J}}$ of $\hat{\mu}_{\mathcal{H}}^{\mathbf{x}}(\mathbf{G}, S)$ and a member $\hat{\mathcal{J}}'$ of $\hat{\mu}_{\mathcal{H}}^{\mathbf{x}}(\mathbf{G}', S')$ are *isomorphic* if there is a bijection between them such that paired elements are isomorphic. We also say that $\hat{\mu}_{\mathcal{H}}^{\mathbf{x}}(\mathbf{G}, S)$ and $\hat{\mu}_{\mathcal{H}}^{\mathbf{x}}(\mathbf{G}', S')$ are *isomorphic* if there is a bijection between them such that paired elements are isomorphic.

We now come to the point where we can define, for every boundaried graph, a signature encoding all the possible partial packings that can be realized in this graph.

Signatures and Folios For every $y \in \mathbb{N}$, we set

$$\text{sig}_{\mathcal{H}}^{\mathbf{x}}(\mathbf{G}, y) = \{\hat{\mu}_{\mathcal{H}}^{\mathbf{x}}(\mathbf{G}, S), S \subseteq A_{\mathbf{x}}(G), |S| = y\}$$

and, given two compatible t -boundaried graphs \mathbf{G} and \mathbf{G}' and a $y \in \mathbb{N}$, we say that $\text{sig}_{\mathcal{H}}^{\mathbf{x}}(\mathbf{G}, y)$ and $\text{sig}_{\mathcal{H}}^{\mathbf{x}}(\mathbf{G}', y)$ are *isomorphic* if there is a bijection between them such that paired elements are isomorphic.

Finally, for $\rho \in \mathbb{N}$, we set

$$\text{folio}_{\mathcal{H}, \rho}(\mathbf{G}) = (\text{sig}_{\mathcal{H}}^{\mathbf{v}}(\mathbf{G}, 0), \dots, \text{sig}_{\mathcal{H}}^{\mathbf{v}}(\mathbf{G}, \rho), \text{sig}_{\mathcal{H}}^{\mathbf{e}}(\mathbf{G}, 0), \dots, \text{sig}_{\mathcal{H}}^{\mathbf{e}}(\mathbf{G}, \rho)).$$

Given two t -boundaried graphs \mathbf{G} and \mathbf{G}' , a $\rho \in \mathbb{N}$, and a finite collection of connected graphs \mathcal{H} , we say that $\mathbf{G} \simeq_{\mathcal{H},\rho} \mathbf{G}'$ if \mathbf{G} and \mathbf{G}' are compatible, neither \mathbf{G} nor \mathbf{G}' contains an \mathcal{H} -subdivision, and the elements of $\text{folio}_{\mathcal{H},\rho}(\mathbf{G})$ and $\text{folio}_{\mathcal{H},\rho}(\mathbf{G}')$ are position-wise isomorphic.

3 The Reduction

The purpose of this section is to prove the following lemma.

Lemma 1 *There exists a function $f_2: \mathbb{N}^2 \rightarrow \mathbb{N}$ and an algorithm that, given a positive integer t , a finite collection \mathcal{H} of connected graphs where $h = m(\mathcal{H})$, and a t -partitioned protrusion $\mathbf{P} = (\mathbf{G}, (\mathcal{X}, T, s))$ of a graph W with $n(\mathbf{G}) > f_2(h, t)$, outputs either*

- an \mathcal{H} -subdivision of W with at most $f_2(h, t)$ edges or
- a graph W' such that

$$\begin{aligned} \mathbf{x}\text{-pack}_{\mathcal{H}}(W') &= \mathbf{x}\text{-pack}_{\mathcal{H}}(W), \\ \mathbf{x}\text{-cover}_{\mathcal{H}}(W') &= \mathbf{x}\text{-cover}_{\mathcal{H}}(W), \text{ and} \\ n(W') &< n(W). \end{aligned}$$

Furthermore, this algorithm runs in time $O_{t,h}(n(T))$. Moreover, this algorithm can be enhanced so that given a \mathbf{x} - \mathcal{H} -packing (resp. \mathbf{x} - \mathcal{H} -covering) in W' then it outputs a same size \mathbf{x} - \mathcal{H} -packing (resp. \mathbf{x} - \mathcal{H} -covering) in W .

In other words we can, in linear time, either find a *small* \mathcal{H} -subdivision, or reduce the graph to a smaller one where the parameters of packing and covering stay the same. The main lines of the proof are the following. We consider the equivalence relation $\simeq_{\mathcal{H},t}$ on \mathcal{H} -free t -partitioned protrusions. Informally, $\simeq_{\mathcal{H},t}$ relates the partitioned protrusions in which the same partial packings are realizable. The first step is to show that this relation has a finite number of equivalence classes. Recall that each vertex i of the tree T of the partitioned protrusion \mathbf{P} defines a partitioned protrusion \mathbf{P}_i . We then show that if T is too large (i.e. it has a long path or a vertex of large degree), then for several vertices, such partitioned protrusions belong to the same equivalence class of $\simeq_{\mathcal{H},t}$. We finally show that in this case, some of these vertices can be deleted without changing the packing and covering numbers.

Before giving the proof of Lemma 1, we need to prove several intermediate results. In the sequel, unless stated otherwise, we assume that $\mathbf{x} \in \{v, e\}$, $t \in \mathbb{N}_{\geq 1}$ and that \mathcal{H} is a finite collection of connected graphs. We set $h = m(\mathcal{H})$. Recall that, for a t -partitioned protrusion $(\mathbf{G}, (\mathcal{X}, T, s))$ of a graph W , and $v \in V(T)$, we denote by \mathcal{G}_v the set of all boundaried graphs of the form \mathbf{G}_{V_u} , for some V_u -splitting $(\mathbf{G}_{V_u}, \mathbf{G}_{V_u^c})$ of W , where u is a child of v . Informally, these are the boundaried graphs induced by the bags of the subtrees of T rooted at the children of v .

Lemma 2 *There are two functions $f_3: \mathbb{N}^2 \rightarrow \mathbb{N}$ and $f_4: \mathbb{N}^2 \rightarrow \mathbb{N}$ such that, for every graph W and every t -partitioned protrusion $(\mathbf{G}, (\mathcal{X}, T, s))$ of W , if \mathcal{P} is an \mathcal{H} -free partial \mathbf{x} - \mathcal{H} -packing in \mathbf{G} then:*

- (a) The partial subdivisions of graphs in \mathcal{H} that are contained in \mathcal{P} have in total at most $f_3(h, t)$ branch vertices.
- (b) \mathcal{P} intersects at most $f_4(h, t)$ graphs of \mathcal{G}_s .

Proof Proof of (a) First, note that any \mathcal{H} -free partial \mathbf{x} - \mathcal{H} -packing in \mathbf{G} has cardinality at most t , because each partial subdivision it contains must use a boundary edge of \mathbf{G} , and two distinct subdivisions of the same packing are (at least) edge-disjoint. Also, each of these partial subdivisions contains at most $\max_{H \in \mathcal{H}} n(H) \leq h$ branch vertices. Consequently, for every \mathcal{H} -free partial \mathbf{x} - \mathcal{H} -packing in \mathbf{G} , the number of branch vertices of graphs of \mathcal{H} it induces in \mathbf{G} is at most $t \cdot h$. Hence the function $f_3(h, t) := t \cdot h$ upper-bounds the amount of branch vertices each \mathcal{H} -free partial packing can contain.

Proof of (b) Let ζ be the maximum multiplicity of an edge in a graph of \mathcal{H} . Because of (a), every \mathcal{H} -free partial \mathbf{x} - \mathcal{H} -packing \mathcal{P} in \mathbf{G} has at most $f_3(h, t)$ branch vertices of graphs of \mathcal{H} , so at most $f_3(h, t)$ graphs of \mathcal{G}_s may contain such vertices. Besides, \mathcal{P} might also contain paths free of branch vertices linking pairs of branch vertices. Since there are at most $(f_3(h, t))^2$ such pairs and no pair will need to be connected with more than $\zeta \leq h$ distinct paths, it follows that at most $(f_3(h, t))^2 \cdot h$ graphs of \mathcal{G}_s contain vertices from these paths. Therefore, every \mathcal{H} -free partial \mathbf{x} - \mathcal{H} -packing intersects at most $f_3(h, t) + (f_3(h, t))^2 \cdot h =: f_4(h, t)$ graphs of \mathcal{G}_s . \square

Lemma 3 *There is a function $f_5: \mathbb{N}^2 \rightarrow \mathbb{N}$ such that the image of the function $\hat{\mu}_{\mathcal{H}}^{\mathbf{x}}$, when its domain is restricted to*

$$\{(\mathbf{G}, S), \mathbf{G} \text{ is } \mathcal{H}\text{-free and } S \subseteq A_{\mathbf{x}}(\mathbf{G})\},$$

has size upper-bounded by $f_5(h, t)$.

Proof Let \mathbf{G} be an \mathcal{H} -free t -boundaried graph and let $S \subseteq A_{\mathbf{x}}(\mathbf{G})$. From Lemma 2(a), every \mathcal{H} -free partial \mathbf{x} - \mathcal{H} -packing in \mathbf{G} contains at most $f_3(h, t)$ branch vertices. This partial packing can in addition use at most t boundary vertices. Let $\mathcal{C}_{h,t}$ be the class of all $(\leq t)$ -boundaried graphs on at most $f_3(h, t) + t$ vertices. Clearly the size of this class is a function depending on h and t only. Recall that the elements of the set $\hat{\mu}_{\mathcal{H}}^{\mathbf{x}}(\mathbf{G}, S)$ are obtained from partial \mathbf{x} - \mathcal{H} -packings by dissolving internal vertices of the paths linking branch vertices, hence every element of $\hat{\mu}_{\mathcal{H}}^{\mathbf{x}}(\mathbf{G}, S)$ is a t -boundaried graph of \mathcal{B}_t having at most $f_3(h, t) + t$ vertices. Therefore, for any \mathcal{H} -free t -boundaried graph \mathbf{G} and subset $S \subseteq A_{\mathbf{x}}(\mathbf{G})$, we have $\hat{\mu}_{\mathcal{H}}^{\mathbf{x}}(\mathbf{G}, S) \subseteq \mathcal{C}_{h,t}$. As a consequence, the image of the function $\hat{\mu}_{\mathcal{H}}^{\mathbf{x}}$ when restricted to \mathcal{H} -free t -boundaried graphs $\mathbf{G} \in \mathcal{B}_t$ (and subsets $S \subseteq A_{\mathbf{x}}(\mathbf{G})$) is a subset of the power set of $\mathcal{C}_{h,t}$, so its size is upper-bounded by a function (which we call f_5) that depends only on h and t . \square

Corollary 1 *There is a function $f_6: \mathbb{N}^2 \rightarrow \mathbb{N}$ such that the relation $\simeq_{\mathcal{H},t}$ partitions \mathcal{H} -free t -boundaried graphs into at most $f_6(h, t)$ equivalence classes.*

The following lemma follows directly from the definition of $\hat{\mu}_{\mathcal{H}}^{\mathbf{x}}$.

Lemma 4 *Let $\mathbf{F}, \mathbf{G} \in \mathcal{B}_t$ be two compatible t -boundaried graphs and let $k \in \mathbb{N}$. Then we have:*

$$\mathbf{P}_{\mathbf{x}, \mathcal{H}}^{\geq k}(\mathbf{F} \oplus \mathbf{G}) \neq \emptyset \iff \exists \hat{\mathcal{J}} \in \hat{\mu}_{\mathcal{H}}^{\mathbf{x}}(\mathbf{G}, \emptyset), \mathbf{P}_{\mathbf{x}, \mathcal{H}}^{\geq k}(\mathbf{F} \oplus \mathbf{U}\hat{\mathcal{J}}) \neq \emptyset.$$

The choice of the definition of the relation \simeq is justified by the following lemma. Roughly speaking, it states that we can replace a t -partitioned protrusion of a graph with any other $\simeq_{\mathcal{H}, t}$ -equivalent t -partitioned protrusion without changing the covering and packing number of the graph. The reduction algorithm that we give after this lemma relies on this powerful property.

Lemma 5 (Protrusion replacement) *Let $\mathbf{F}, \mathbf{G}, \mathbf{G}' \in \mathcal{B}_t$ be three compatible boundaried graphs such that $\mathbf{G} \simeq_{\mathcal{H}, t} \mathbf{G}'$. For every $k \in \mathbb{N}$, we have:*

- (i) $\mathbf{P}_{\mathbf{x}, \mathcal{H}}^{\geq k}(\mathbf{F} \oplus \mathbf{G}) \neq \emptyset \iff \mathbf{P}_{\mathbf{x}, \mathcal{H}}^{\geq k}(\mathbf{F} \oplus \mathbf{G}') \neq \emptyset$ and
- (ii) $\mathbf{C}_{\mathbf{x}, \mathcal{H}}^{\leq k}(\mathbf{F} \oplus \mathbf{G}) \neq \emptyset \iff \mathbf{C}_{\mathbf{x}, \mathcal{H}}^{\leq k}(\mathbf{F} \oplus \mathbf{G}') \neq \emptyset.$

Proof Notice that, by definition of $\simeq_{\mathcal{H}, t}$, the boundaried graphs \mathbf{G} and \mathbf{G}' are \mathcal{H} -free. *Proof of item (i), “ \Rightarrow ”.* Let \mathcal{M} be an \mathbf{x} - \mathcal{H} -packing of size at least k in $\mathbf{F} \oplus \mathbf{G}$, whose set of branch vertices is L . We define

$$\begin{aligned} \mathbf{J}_{\mathbf{F}} &= (\mathbf{U}\mathcal{M}) \cap_{\mathbf{F} \oplus \mathbf{G}} \mathbf{F}, \\ \mathbf{J}_{\mathbf{G}} &= (\mathbf{U}\mathcal{M}) \cap_{\mathbf{F} \oplus \mathbf{G}} \mathbf{G}, \text{ and} \\ \hat{\mathbf{J}}_{\mathbf{G}} &= \bigcup_{M \in \mathcal{M}} \kappa(M \cap_{\mathbf{F} \oplus \mathbf{G}} \mathbf{G}, L \cap V(\mathbf{G})). \end{aligned}$$

Note that $\hat{\mathbf{J}}_{\mathbf{G}} \in \hat{\mu}_{\mathcal{H}}^{\mathbf{x}}(\mathbf{G}, \emptyset)$ and that $\mathbf{F} \oplus \hat{\mathbf{J}}_{\mathbf{G}}$ has an \mathbf{x} - \mathcal{H} -packing of size at least k (cf. Lemma 4). By definition of \simeq , there is a bijection ψ between $\hat{\mu}_{\mathcal{H}}^{\mathbf{x}}(\mathbf{G}, \emptyset)$ and $\hat{\mu}_{\mathcal{H}}^{\mathbf{x}}(\mathbf{G}', \emptyset)$. Let $\hat{\mathbf{J}}_{\mathbf{G}'}$ be the image of $\hat{\mathbf{J}}_{\mathbf{G}}$ by ψ . Since $\hat{\mathbf{J}}_{\mathbf{G}}$ and $\hat{\mathbf{J}}_{\mathbf{G}'}$ are isomorphic, $\mathbf{F} \oplus \hat{\mathbf{J}}_{\mathbf{G}'}$ also has an \mathbf{x} - \mathcal{H} -packing of size at least k . By Lemma 4, this implies that such a packing exists in $\mathbf{F} \oplus \mathbf{G}'$ as well. The direction “ \Leftarrow ” is symmetric as \mathbf{G} and \mathbf{G}' play the same role.

Proof of item (ii), “ \Rightarrow ”. Let $C \subseteq A_{\mathbf{x}}(\mathbf{F} \oplus \mathbf{G})$ be a minimum \mathbf{x} - \mathcal{H} -covering of $\mathbf{F} \oplus \mathbf{G}$ of size at most k . Let $S = C \cap A_{\mathbf{x}}(\mathbf{G})$. Since we assume that \mathbf{G} is \mathcal{H} -free and that C is minimum, we can also assume that $|S| \leq t$ (otherwise we could get a smaller covering by taking the t boundary vertices/edges of \mathbf{G}). By our assumption that $\mathbf{G} \simeq_{\mathcal{H}, t} \mathbf{G}'$, there is an isomorphism between $\text{sig}_{\mathcal{H}}^{\mathbf{x}}(\mathbf{G}, |S|)$ and $\text{sig}_{\mathcal{H}}^{\mathbf{x}}(\mathbf{G}', |S|)$. Let $S' \subseteq A_{\mathbf{x}}(\mathbf{G}')$ be a set such that $\hat{\mu}_{\mathcal{H}}^{\mathbf{x}}(\mathbf{G}, S)$ is sent to $\hat{\mu}_{\mathcal{H}}^{\mathbf{x}}(\mathbf{G}', S')$ by this isomorphism. Then observe that every partial packing \mathcal{J}' of $\mathbf{G}' \setminus S'$, such that $(\mathbf{F} \setminus C) \oplus (\mathbf{U}\mathcal{J}')$ has an \mathcal{H} -subdivision, can be translated into a partial packing \mathcal{J} of $\mathbf{G} \setminus S$ such that $(\mathbf{F} \setminus C) \oplus (\mathbf{U}\mathcal{J})$ also has such a subdivision, in the same way as in the proof of item (i) above. As C is a cover, this would lead to contradiction. Therefore $\hat{\mu}_{\mathcal{H}}^{\mathbf{x}}(\mathbf{G}, S)$ does not contain such a partial packing. As a consequence, $(C \cap A_{\mathbf{x}}(\mathbf{F})) \cup S'$ is a covering of $\mathbf{F} \oplus \mathbf{G}'$ of size at most k . As in the previous case, the proof of direction “ \Leftarrow ” comes from the symmetry in the statement. □

Lemma 5 can be rewritten as follows.

Corollary 2 *Under the assumptions of Lemma 5, we have $\mathbf{x}\text{-pack}(\mathbf{F} \oplus \mathbf{G}) = \mathbf{x}\text{-pack}(\mathbf{F} \oplus \mathbf{G}')$ and $\mathbf{x}\text{-cover}(\mathbf{F} \oplus \mathbf{G}) = \mathbf{x}\text{-cover}(\mathbf{F} \oplus \mathbf{G}')$.*

Recall that $f_6(h, t)$ denotes the number of equivalence classes of $\simeq_{\mathcal{H},t}$ among boundaried graphs of \mathcal{B}_t . For every $h, t \in \mathbb{N}$, let

$$f_7(h, t) = f_6(h, t) \cdot f_4(h, t) \quad \text{and} \quad f_2(h, t) = 2ht^3 \cdot (f_3(h, t))^{f_6(h,t)+1}.$$

Let us give some intuition about these definitions. The first remark is an application of the pigeonhole principle.

Observation 1 *In a collection of more than $f_3(h, t)$ \mathcal{H} -free boundaried graphs of \mathcal{B}_t , there is one that is equivalent (w.r.t. $\simeq_{\mathcal{H},t}$) to $f_4(h, t)$ other graphs of the collection.*

Observation 2 *If (T, s, \mathcal{X}) is a rooted tree-partition of a graph G with the following properties:*

- (T, \mathcal{X}) has width at most t ;
- T has height at most $f_6(h, t)$; and
- T has degree at most $f_3(h, t) + 1$,

then G has at most $f_2(h, t)$ vertices, and every \mathcal{H} -subdivision of G has at most $f_2(h, t)$ edges.

Proof The above assumptions imply that T has at most $(f_3(h, t))^{f_6(h,t)+1}$ vertices. Every bag of (T, \mathcal{X}) contains at most t vertices of G , therefore G has at most $(f_3(h, t))^{f_6(h,t)+1} \cdot t \leq f_2(h, t)$ vertices. Also, every bag induces a subgraph with at most $t(t - 1)/2$ multiedges (i.e. without counting multiplicities), and for every edge f of T we have $|E_f| \leq t$, hence every bag contributes for at most $t^2 + t$ multiedges. Therefore G has at most $(t + t^2) (f_3(h, t))^{f_6(h,t)+1}$ multiedges. Now, observe that an edge of G is used at most h times by an \mathcal{H} -subdivision (if it is a multiple edge), since every path connecting two branch vertices of a subdivision uses a given edge at most once. We deduce that an \mathcal{H} -subdivision of G contains at most $h \cdot (t + t^2) (f_3(h, t))^{f_6(h,t)+1} \leq f_2(h, t)$ edges. □

The two next lemmas are the main tools used in the proof of Lemma 1. Under different conditions, they provide either a small subdivision, or a reduced graph. Lemma 6 considers the case where a vertex of T has high degree, whereas Lemma 7 deals with the situation where T has a long path.

Lemma 6 *Let $\mathbf{P} = (\mathbf{G}, (\mathcal{X}, T, s))$ be a t -partitioned protrusion of a graph W , and let $u \in V(T)$ be a vertex with more than $f_3(h, t)$ children such that for every $v \in \text{children}_{(T,s)}(u)$, we have $m(\mathbf{G}_v) \leq f_6(h, t)$. Then, either*

- W contains an \mathcal{H} -subdivision M with at most $f_6(h, t)$ edges or
- there exists a graph W' such that

$$\begin{aligned} \text{x-pack}_{\mathcal{H}}(W') &= \text{x-pack}_{\mathcal{H}}(W), \\ \text{x-cover}_{\mathcal{H}}(W') &= \text{x-cover}_{\mathcal{H}}(W), \text{ and} \\ n(W') &< n(W). \end{aligned}$$

Moreover, there is an algorithm that, given such \mathbf{P} , W , and u , returns either M or W' as above in $O_{h,t}(1)$ steps.

Proof As u has more than $f_3(h, t)$ children, it contains a collection of $d = f_4(h, t) + 1$ children v_1, \dots, v_d , such that $\mathbf{G}_{v_1} \simeq_{\mathcal{H},t} \mathbf{G}_{v_i}$ for every $i \in [2, d]$ (by Observation 1). Let us now assume that \mathbf{G}_u is \mathcal{H} -free. Since every $\text{x}\mathcal{H}$ -packing of W will intersect at most $f_4(h, t)$ bags of children of u (by Lemma 2(b)), we can safely delete one of the $f_4(h, t) + 1$ equivalent subgraphs mentioned above. We use the following algorithm in order to find such a bag to delete or a small \mathcal{H} -subdivision.

1. Let A be an array of $f_6(h, t)$ counters initialized to 0, each corresponding to a distinct equivalence class of $\simeq_{\mathcal{H},t}$;
2. Pick a vertex $v \in \text{children}_{(T,s)}(u)$ that has not been considered yet;
3. If G_v contains an \mathcal{H} -subdivision \mathcal{M} , then return \mathcal{M} and exit;
4. Otherwise, increment the counter of A corresponding to the equivalence class of \mathbf{G}_u by one;
5. If this counter reaches $d + 1$, return v , otherwise go back to Line 2.

Notice that the subdivision returned in Line 3 has size at most $t \cdot f_6(h, t)$ (as we assume that $m(G_v) \leq f_6(h, t)$) and that the vertex returned in Line 5 has the desired property. By Corollary 1, the relation $\simeq_{\mathcal{H},t}$ has at most $f_6(h, t)$ equivalence classes, thus the main loop will be run at most $f_6(h, t) \cdot f_4(h, t) + 1$ times (by the pigeonhole principle). Eventually, Lines 3 and 4 can be performed in $O_{h,t}(1)$ -time given that G_v has size bounded by a function of h and t .

In the end, we return $W' = W \setminus V(G_v)$ if the algorithm outputs v and \mathcal{M} otherwise. □

Lemma 7 *There is an algorithm that, given a t -partitioned protrusion $\mathbf{P} = (\mathbf{G}, (\mathcal{X}, T, s))$ of a graph W and a vertex $u \in V(T)$ such that*

- u has height exactly $f_6(h, t)$ in (T, s) ,
- the graph of \mathbf{G}_u is \mathcal{H} -free, and
- T_u has maximum degree at most $f_3(h, t)$,

outputs a graph W' such that

$$\begin{aligned} \text{x-pack}_{\mathcal{H}}(W') &= \text{x-pack}_{\mathcal{H}}(W), \\ \text{x-cover}_{\mathcal{H}}(W') &= \text{x-cover}_{\mathcal{H}}(W), \text{ and} \\ n(W') &< n(W). \end{aligned}$$

Moreover, this algorithm runs in $O_{h,t}(1)$ -time.

Proof As in the proof of Observation 2, we use the fact that every \mathcal{H} -subdivision of \mathbf{G}_u uses every (multi)edge at most h times. A consequence is that the boundaried subgraph of \mathbf{G}_u obtained by setting the multiplicity of every edge e to the minimum of h and the multiplicity of e contains an \mathcal{H} -subdivision iff \mathbf{G}_u does. As the number of vertices and edges of this subgraph is bounded by a function of h and t , we can therefore check in $O_{h,t}(1)$ -time if \mathbf{G}_u contains an \mathcal{H} -subdivision. If one is found, it has at most $f_2(h, t)$ edges (Observation 2) and we are done.

Let us now consider the case where \mathbf{G}_u is \mathcal{H} -free. By definition of the vertex u , there is a path of $f_6(h, t) + 1$ vertices from a leaf of T_u to u . Let us arbitrarily choose, for every vertex v of this path, a V_v -splitting $(\mathbf{G}_{G_v}, \mathbf{G}_{G_v^c})$ of \mathbf{G} . By definition of $f_6(h, t)$ (the number of equivalence classes in $\simeq_{\mathcal{H},t}$ in \mathcal{B}_t), there are two distinct vertices v, w on this path such that $\mathbf{G}_v \simeq_{\mathcal{H},t} \mathbf{G}_w$. As mentioned above, the number edges of \mathbf{G}_u is bounded by a function of h and t , hence finding these two vertices can be done in $O_{h,t}(1)$ -time. Let us assume without loss of generality that w is a descendant of v . Let \mathbf{H} be the boundaried graph such that $W = \mathbf{H} \oplus \mathbf{G}_{G_v}$ and let $W' = \mathbf{H} \oplus \mathbf{G}_{G_w}$. By Corollary 2, we have $\mathbf{x}\text{-pack}_{\mathcal{H}}(W') = \mathbf{x}\text{-pack}_{\mathcal{H}}(W)$ and $\mathbf{x}\text{-cover}_{\mathcal{H}}(W') = \mathbf{x}\text{-cover}_{\mathcal{H}}(W)$. Furthermore, the graph W' is clearly smaller than W . \square

We are now ready to prove Lemma 1.

Proof of Lemma 1 Observe that since $n(G) > f_2(h, t)$ and each bag of (\mathcal{X}, T, s) contains at most t vertices, we have

$$n(T) > f_2(h, t)/t = (f_6(h, t) \cdot f_4(h, t))^{f_6(h, t)}.$$

Therefore, T has either diameter more than $f_6(h, t)$ or a vertex of degree more than $f_6(h, t) \cdot f_4(h, t)$.

Let us consider the following procedure.

1. By a DFS on (T, s) , compute the height of each vertex of T and find (if it exists) a vertex v of degree more than $f_3(h, t) + 1$ and height at most $f_6(h, t) - 1$ that has minimum height.
2. If such a vertex v is found, then apply the algorithm of Lemma 6 on \mathbf{P} and v , and return the obtained result.
3. Otherwise, find a vertex u of height exactly $f_6(h, t)$ in (T, s) and then apply the algorithm of Lemma 7 on \mathbf{P} and (T_u, u) and return the obtained result.

Observe that since $n(G) > f_2(h, t)$, Observation 2 implies that either T has diameter more than $f_6(h, t)$, or it contains a vertex of degree more than $f_3(h, t) + 1$. Therefore, the vertex u of line 3 always exists in the case where no vertex of high degree is found in line 1. The correctness of this algorithm follows from Lemmas 6 and 7. The DFS done in the first step takes time $O(n(T))$ and the rest of the algorithm takes time $O_{h,t}(1)$ according to the aforementioned lemmata.

The fact that a $\mathbf{x}\mathcal{H}$ -packing of W' can be lifted to an equal size $\mathbf{x}\mathcal{H}$ -packing of W follows easily when Lemma 6 is applied, as then W' is a subgraph of W . In the case that Lemma 7 is applied, the new $\mathbf{x}\mathcal{H}$ -packing is obtained because of Corollary 2 and the supporting protrusion replacement Lemma 5, whose proof indicates which part of

the $x\mathcal{H}$ -packing of W' should be replaced with what in order to obtain an equal size $x\mathcal{H}$ -packing of W . □

4 From the Erdős–Pósa Property to Approximation

For the purposes of this section we define $\Theta_r = \text{ex}(\theta_r)$. We need the following that is one of the main results in [7, Theorem 3.2].

Proposition 1 *There is an algorithm that, with input three positive integers $r \geq 2$, w, z and a connected m -edge graph W , where $m \geq z > r \geq 2$, outputs one of the following:*

- a Θ_r -subdivision of W with at most z edges,
- a $(2r - 2)$ -partitioned protrusion $(\mathbf{G}, \mathcal{D})$ of W , where $\mathbf{G} = (G, B, \lambda)$ and such that G is a connected graph and $n(\mathbf{G}) > w$, or
- an H -minor model of W for some graph H with $\delta(H) \geq \frac{1}{r-1} 2^{\frac{z-5r}{4r(2w+1)}}$,

in $O_r(m)$ steps.

4.1 A Lemma on Reduce or Progress

The proof of the next lemma combines Proposition 1 and Lemma 1.

Lemma 8 (Reduce or progress) *There is an algorithm that, with input $x \in \{v, e\}$, $r \in \mathbb{N}_{\geq 2}$, $k \in \mathbb{N}$ and an n -vertex graph W , outputs one of the following:*

- a Θ_r -subdivision of W with at most $O_r(\log k)$ edges;
- a graph W' where

$$\begin{aligned} x\text{-cover}_{\mathcal{H}}(W') &= x\text{-cover}_{\mathcal{H}}(W), \\ x\text{-pack}_{\mathcal{H}}(W') &= x\text{-pack}_{\mathcal{H}}(W), \text{ and} \\ n(W') &< n(W); \text{ or} \end{aligned}$$

- an H -minor model in W , for some graph H with $\delta(H) \geq k(r + 1)$,

in $O_r(m)$ steps. Moreover, this algorithm can be enhanced so that, when the second case applies, given a $x\mathcal{H}$ -packing (resp. $x\mathcal{H}$ -covering) in W' then it outputs a same size $x\mathcal{H}$ -packing (resp. $x\mathcal{H}$ -covering) in W .

Proof We set $t = 2r - 2$, $w = f_2(h, t)$, $z = 2r(w - 1) \log(k(r + 1)(r - 1)) + 5r$, and $h = m(\Theta_r)$. Observe that $z = O_r(\log k)$ and $h, t, w = O_r(1)$. Also observe that our choice for variable z ensures that $2^{\frac{z-5r}{2r(w-1)}} / (r - 1) = k(r + 1)$.

By applying the algorithm of Proposition 1 to r, w, z , and W , we obtain in $O_r(m(W))$ -time either:

- a Θ_r -subdivision in W of at most z edges (first case),
- a $(2r - 2)$ -edge-protrusion Y of W with extension $> w$ (second case), or

– an H -minor model M in W , for some graph H with $\delta(H) \geq k(r + 1)$ (third case).

In the first case, we return the obtained Θ_r -subdivision. In the second case, by applying the algorithm of Lemma 1 on Y , we get in $O(n(W))$ -time either a Θ_r -subdivision of W on at most $w = O_r(1)$ vertices, or a graph W' where, for $\mathbf{x} \in \{\mathbf{v}, \mathbf{e}\}$, $\mathbf{x}\text{-cover}_{\mathcal{H}}(W') = \mathbf{x}\text{-cover}_{\mathcal{H}}(W)$, $\mathbf{x}\text{-pack}_{\mathcal{H}}(W') = \mathbf{x}\text{-pack}_{\mathcal{H}}(W)$ and $n(W') < n(W)$. In the third case, we return the minor model M .

In each of the above cases, we get after $O(m)$ steps either a minor model of a graph with minimum degree more than $k(r + 1)$, a Θ_r -subdivision in W with at most z edges, or an equivalent graph of smaller size. \square

It might not be clear yet to what purpose the minor model of a graph of degree more than $k(r + 1)$ output by the algorithm of Lemma 8 can be used. An answer is given by the following lemmata, which state that such a graph contains a packing of at least k minor models of Θ_r . These lemmata will be used in the design of the approximation algorithms in Sect. 4.2.

Lemma 9 *There is an algorithm that, given $k, r \in \mathbb{N}_{\geq 1}$ and a graph G with $\delta(G) \geq kr$, returns a member of $\mathbf{P}_{\mathbf{e}, \Theta_r}^{\geq k}(G)$ in G in $O(m)$ steps.*

Proof Starting from any vertex u , we grow a maximal path P in G by iteratively adding to P a vertex that is adjacent to the previously added vertex but does not belong to P . Since $\delta(G) \geq kr$, any such path will have length at least $kr + 1$. At the end, all the neighbors of the last vertex v of P belong to P (otherwise P could be extended). Since v has degree at least kr , v has at least kr neighbors in P . Let w_0, \dots, w_{kr-1} be an enumeration of the kr first neighbors of v in the order given by P , starting from u . For every $i \in \llbracket 0, k - 1 \rrbracket$, let S_i be the subgraph of G induced by v and the subpath of P starting at w_{ir} and ending at $w_{(i+1)r-1}$. Observe that for every $i \in \llbracket 0, k - 1 \rrbracket$, S_i contains a Θ_r -subdivision and that the intersection of every pair of graphs from $\{S_i\}_{i \in \llbracket 0, k-1 \rrbracket}$ is $\{v\}$. Hence P contains a member of $\mathbf{P}_{\mathbf{e}, \Theta_r}^{\geq k}(G)$, as desired. Every edge of G is considered at most once in this algorithm, yielding to a running time of $O(m)$ steps. \square

Corollary 3 *There is an algorithm that, given $r \in \mathbb{N}_{\geq 1}$ and a graph G with $\delta(G) \geq r$, returns a Θ_r -subdivision in G in $O(m)$ -steps.*

Observe that the previous lemma only deals with edge-disjoint packings. An analogue of Lemma 9 for vertex-disjoint packings can be proved using Proposition 2, to the price of a worse time complexity.

Proposition 2 (Theorem 12 of [1]) *Given $k, r \in \mathbb{N}_{\geq 1}$ and an input graph G such that $\delta(G) \geq k(r + 1) - 1$, a partition (V_1, \dots, V_k) of $V(G)$ satisfying $\forall i \in \llbracket 1, k \rrbracket, \delta(G[V_i]) \geq r$ can be found in $O(n^c)$ steps for some $c \in \mathbb{N}$.*

Lemma 10 *There is an algorithm that, given $k, r \in \mathbb{N}_{\geq 1}$ and a graph G with $\delta(G) \geq k(r + 1) - 1$, outputs a member of $\mathbf{P}_{\mathbf{v}, \Theta_r}^{\geq k}(G)$ in $O(n^c + m)$ steps, where c is the constant of Proposition 2.*

Proof After applying the algorithm of Proposition 2 on G to obtain in $O(n^c)$ -time k graphs $G[V_1], \dots, G[V_k]$, we extract a Θ_r -subdivision from each of them using Corollary 3. \square

4.2 Approximation Algorithms

As Θ_r -subdivisions are exactly the subgraph minimal graphs that contain Θ_r as a minor, a graph has a Θ_r - \mathbf{x} -packing of size k iff it contains k \mathbf{x} -disjoint subgraphs, each containing θ_r as a minor. Therefore, Theorem 1 is a direct combinatorial consequence of the following.

Theorem 5 *There is a function $f_8: \mathbb{N} \rightarrow \mathbb{N}$ and an algorithm that, with input $\mathbf{x} \in \{\mathbf{v}, \mathbf{e}\}$, $r \in \mathbb{N}_{\geq 2}$, $k \in \mathbb{N}$, and an n -vertex graph W , outputs either a \mathbf{x} - Θ_r -packing of W of size k or an \mathbf{x} - Θ_r -covering of W of size at most $f_8(r) \cdot k \cdot \log k$. Moreover, this algorithm runs in $O(m^2)$ steps if $\mathbf{x} = \mathbf{e}$ and in $O_r(n^c + n \cdot m)$ steps if $\mathbf{x} = \mathbf{v}$, where c is the constant from Proposition 2.*

Proof Let $f_8(r) : \mathbb{N} \rightarrow \mathbb{N}$ be a function such that each Θ_r -subdivision output by the algorithm of Lemma 8 has size at most $f_8(r) \cdot \log k$. We consider the following procedure.

1. $G := W ; P := \emptyset$;
2. Apply the algorithm of Lemma 8 on (\mathbf{x}, r, k, G) :
 - Progress: if the output is a Θ_r -subdivision M , let $G := G \setminus A_{\mathbf{x}}(M)$ and $P = P \cup \{M\}$;
 - Win: if the output is a H -minor model M in W for some graph H with $\delta(H) \geq k(r + 1)$, apply the algorithm of Lemma 9 (if $\mathbf{x} = \mathbf{e}$) or the one of Lemma 10 (if $\mathbf{x} = \mathbf{v}$) to H to obtain a member of $\mathbf{P}_{\mathbf{x}, \Theta_r}^{\geq k}(H)$. Using M , translate this packing into a member of $\mathbf{P}_{\mathbf{x}, \Theta_r}^{\geq k}(W)$ and return this new packing;
 - Reduce: otherwise, the output is a graph G' . Then let $G := G'$.
3. If $|P| = k$ then return P which is a member of $\mathbf{P}_{\mathbf{x}, \Theta_r}^{\geq k}(W)$;
4. If $n(W) = 0$ then return P which is in this case a member of $\mathbf{C}_{\mathbf{x}, \Theta_r}^{\leq f_8(r) \log k}(W)$;
5. Otherwise, go back to Line 2.

This algorithm clearly returns the desired result. Furthermore, the loop is executed at most $|A_{\mathbf{x}}(W)|$ times (as an element of $A_{\mathbf{x}}(W)$ is deleted each time we reach the *Progress* case) and each call to the algorithm of Lemma 8 takes $O(m(W))$ steps. When the algorithm reaches the “Win” case (which can happen at most once), the calls to the algorithm of Lemma 9 (if $\mathbf{x} = \mathbf{e}$) or the one of Lemma 10 (if $\mathbf{x} = \mathbf{v}$), respectively, take $O(m(H))$ and $O((n(H))^c)$ steps. Therefore, in total, this algorithm terminates in $O(m^2)$ steps if $\mathbf{x} = \mathbf{e}$ and in $O(n^c + n \cdot m)$ steps if $\mathbf{x} = \mathbf{v}$. □

Observe that if the algorithm of Theorem 5 reaches the “Win” case, then the input graph is known to contain an \mathbf{x} - Θ_r -packing of size at least k . As a consequence, if we are only interested in the existence of a packing or covering, the call to the algorithm of Lemmas 9 or 10 is not necessary.

Corollary 4 *There is an algorithm that, with input $\mathbf{x} \in \{\mathbf{v}, \mathbf{e}\}$, $r \in \mathbb{N}_{\geq 2}$, $k \in \mathbb{N}$, and a graph W , outputs 0 only if W has an \mathbf{x} - Θ_r -packing of size k or 1 only if W has an \mathbf{x} - Θ_r -covering of size at most $f_8(r) \cdot k \cdot \log k$. Furthermore this algorithm runs in $O_r(n \cdot m)$ steps when $\mathbf{x} = \mathbf{v}$ and in $O_r(m^2)$ steps when $\mathbf{x} = \mathbf{e}$.*

We now conclude this section with the proof of Theorem 2. For the same reason as in the proof of Theorem 1 at the beginning of this section, we can deal with Θ_r -subdivisions instead of dealing with subgraphs that contain θ_r as a minor.

Proof of Theorem 2 Let us call **A** the algorithm of Corollary 4. Let $k_0 \in \llbracket 1, n(W) \rrbracket$ be an integer such that $\mathbf{A}(x, r, k_0, W) = 1$ and $\mathbf{A}(x, r, k_0 - 1, W) = 0$, and let us show that the value $f_8(r) \cdot k_0 \log k_0$ is an $O(\log OPT)$ -approximation of $\mathbf{x}\text{-pack}_{\Theta_r}(W)$ and $\text{cover}_{\Theta_r}(W)$.

First, notice that for every $k > \mathbf{x}\text{-pack}_{\Theta_r}(W)$, the value returned by $\mathbf{A}(x, r, k, W)$ is 1. Symmetrically, for every k such that $f_8(r) \cdot k \log k < \mathbf{x}\text{-cover}_{\Theta_r}(W)$, the value of $\mathbf{A}(x, r, k, W)$ is 0. Therefore, the value k_0 is such that:

$$k_0 - 1 \leq \mathbf{x}\text{-pack}_{\Theta_r}(W) \text{ and } \mathbf{x}\text{-cover}_{\Theta_r}(W) \leq f_8(r) \cdot k_0 \log k_0.$$

As every minimal covering must contain at least one vertex or edge (depending on whether $\mathbf{x} = \mathbf{v}$ or $\mathbf{x} = \mathbf{e}$) of each element (i.e. subdivision) of a maximal packing $\mathbf{x}\text{-pack}_{\Theta_r}(W) \leq \mathbf{x}\text{-cover}_{\Theta_r}(W)$, we have the following two equations:

$$\mathbf{x}\text{-pack}_{\Theta_r}(W) \leq f_8(r) \cdot k_0 \log k_0 \leq (\mathbf{x}\text{-pack}_{\Theta_r}(W) + 1) \log(\mathbf{x}\text{-pack}_{\Theta_r}(W) + 1) \tag{1}$$

$$\mathbf{x}\text{-cover}_{\Theta_r}(W) \leq f_8(r) \cdot k_0 \log k_0 \leq (\mathbf{x}\text{-cover}_{\Theta_r}(W) + 1) \log(\mathbf{x}\text{-cover}_{\Theta_r}(W) + 1). \tag{2}$$

Dividing (1) by $\mathbf{x}\text{-pack}_{\Theta_r}(W)$ and (2) by $\text{cover}_{\Theta_r}(W)$, we get:

$$1 \leq \frac{k_0 \log k_0}{\mathbf{x}\text{-pack}_{\Theta_r}(W)} \leq \log(\mathbf{x}\text{-pack}_{\Theta_r}(W) + 1) + \frac{\log \mathbf{x}\text{-pack}_{\Theta_r}(W)}{\mathbf{x}\text{-pack}_{\Theta_r}(W)} = O(\log(\mathbf{x}\text{-pack}_{\Theta_r}(W))), \text{ and}$$

$$1 \leq \frac{k_0 \log k_0}{\mathbf{x}\text{-cover}_{\Theta_r}(W)} \leq \log(\mathbf{x}\text{-cover}_{\Theta_r}(W) + 1) + \frac{\log \mathbf{x}\text{-cover}_{\Theta_r}(W)}{\mathbf{x}\text{-cover}_{\Theta_r}(W)} = O(\log(\mathbf{x}\text{-cover}_{\Theta_r}(W))).$$

Therefore the value $k_0 \log k_0$ is both an $O(\log OPT)$ -approximation of $\mathbf{x}\text{-pack}_{\Theta_r}(W)$ and $\text{cover}_{\Theta_r}(W)$. The value k_0 can be found by performing a binary search in the interval $\llbracket 1, n \rrbracket$, with $O(\log n)$ calls to Algorithm **A**. Hence, our approximation algorithm runs in $O(n \cdot \log(n) \cdot m)$ steps when $\mathbf{x} = \mathbf{v}$ and in $O(m^2 \cdot \log(n))$ steps when $\mathbf{x} = \mathbf{e}$. □

5 Erdős–Pósa Property and Tree-Partition Width

Using the machinery introduced in Sect. 3, we are able to prove a slightly more general version of Theorem 3.

Theorem 6 *Let $t \in \mathbb{N}$. For every $\mathbf{x} \in \{\mathbf{v}, \mathbf{e}\}$, the following holds: if \mathcal{H} is a finite collection of connected graphs and G is a graph of tree-partition width at most t , then*

$\mathbf{x}\text{-cover}_{\mathcal{H}}(G) \leq \alpha \cdot \mathbf{x}\text{-pack}_{\mathcal{H}}(G)$, where α is a constant which depends only on t and \mathcal{H} .

Proof Let $k = \mathbf{x}\text{-pack}_{\mathcal{H}}(G)$. This proof is similar to that of Theorem 5 (progress or reduce). We start with $P := \emptyset$ and $G_0 := G$ and repeatedly apply Lemma 1 to G_i , which provides the existence of either a smaller graph G_{i+1} with the same parameters $\mathbf{x}\text{-cover}_{\mathcal{H}}$ and $\mathbf{x}\text{-pack}_{\mathcal{H}}$ (reduce), or that of an \mathcal{H} -subdivision M with at most $f_2(h, t)$ edges, in which case we set $G_{i+1} := G_i \setminus A_x(M)$ and $P := P \cup \{M\}$ (progress) and continue. Notice that while the algorithm of Lemma 1 requires a t -partitioned protrusion, it is enough here to know that such a partitioned protrusion exists as we do not actually run the algorithm. This is the case because the considered graphs are subgraph of G , which has tree-partition width at most t , so we can consider a t -partitioned protrusion containing the whole graph. We stop this process when the current graph has at most $f_2(h, t)$ vertices. Let G_j be this graph. In the end, P contains at most k \mathcal{H} -subdivisions. Therefore, $C := \bigcup A_x(P) \cup A_x(G_j)$ is an $\mathbf{x}\text{-}\mathcal{H}$ -covering of G of size $(k + 1) \cdot f_2(h, t)$, as required. \square

Theorem 3 can be obtained from Theorem 6 by considering packing and covering of the graphs in $\bigcup_{H \in \mathcal{H}} \mathbf{ex}(H)$, where \mathcal{H} is the family of graphs mentioned in the statement of Theorem 3.

We define $\Theta_{r,r'} = \mathbf{ex}(\theta_{r,r'})$. The rest of this section is devoted to the proof of Theorem 4. Prior to this, we need to introduce a result of Ding et al. [11].

Tree-partition width has been studied in [11, 20, 37]. In particular, the authors of [11] characterized the classes of graphs of bounded tree-partition width in terms of excluded topological minors. The statement of this result requires additional definitions.

Walls, Fans, Paths, and Stars The n -wall is the graph with vertex set $[1, n]$ and whose edge set is:

$$\begin{aligned} & \{(i, j), (i, j + 1)\}, 1 \leq i, j \leq n \\ & \cup \{(2i - 1, 2j + 1), (2i, 2j + 1)\}, 1 < 2i \leq n \text{ and } 1 \leq 2j + 1 \leq n \\ & \cup \{(2i, 2j), (2i + 1, 2j)\}, 1 \leq 2i < n \text{ and } 1 \leq 2j \leq n \}. \end{aligned}$$

The 7-wall is depicted in Fig. 7. The n -fan is the graph obtained by adding a dominating vertex to a path on n vertices. A collection of paths is said to be *independent* if two paths of the collection never share interior vertices. The n -star is the graph obtained by replacing every edge of $K_{1,n}$ with n independent paths of two edges. The n -path is the graph obtained by replacing every edge of an n -edge path with n independent paths of two edges. Examples of these graphs are depicted in Fig. 7. The *wall number* (resp. *fan number*, *star number*, and *path number*) of a graph G is defined as the largest integer k such that G contains a minor model of a k -wall (resp. of a k -fan, of a k -star, of a k -path), or infinity is no such integer exists. Let $\gamma(G)$ denote the maximum of the wall number, fan number, star number, and path number of a graph G .

We need the following result.

Proposition 3 [11] *There is a function $f_9: \mathbb{N} \rightarrow \mathbb{N}$ such that every graph G satisfies $\mathbf{tpw}(G) \leq f_9(\gamma(G))$.*

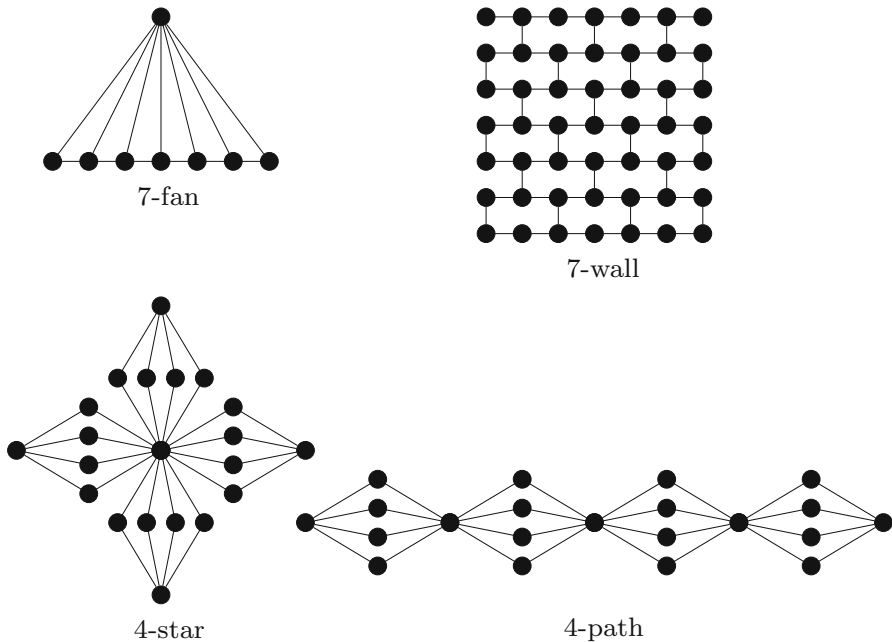


Fig. 7 Unavoidable patterns of graphs of large tree-partition width

In other words, for every integer k , every graph of large enough tree-partition width contains a minor model of one of the following graphs: the k -wall, the k -fan, the k -path, or the k -star.

Notice that for every $r, r' \in \mathbb{N}, r' \leq r$, the graph $\theta_{r,r'}$ is a minor of the following graphs: the r -path, the r -star, the $(r + r' + 1)$ -fan, and the r -wall (for $r \geq 6$). Hence, every graph of large enough tree-partition width contains a $\Theta_{r,r'}$ -subdivision. This can easily be generalized to edge-disjoint packings, as follows.

Lemma 11 *For every $r, r' \in \mathbb{N}, r' \leq r$, and every $k \in \mathbb{N}_{\geq 1}$, every graph G satisfying $\gamma(G) \geq k(r + r' + 2) - 1$ contains an $\mathbf{e}\text{-}\Theta_{r,r'}$ -packing of size k .*

Using Proposition 3, we get the following corollary.

Corollary 5 *For every $r, r' \in \mathbb{N}, r' \leq r$, and every $k \in \mathbb{N}_{\geq 1}$, every graph G satisfying $\mathbf{tpw}(G) \geq f_9(k(r + r' + 2) - 1)$ contains an $\mathbf{e}\text{-}\Theta_{r,r'}$ -packing of size k .*

We are now able to give the proof of Theorem 4. As in the proof of Theorem 1 at the beginning of Sect. 4.2, we can consider $\Theta_{r,r'}$ -subdivisions instead of subgraphs containing $\theta_{r,r'}$ as a minor.

Proof of Theorem 4 According to Corollary 5, for every $k \in \mathbb{N}$, there is a number t_k such that every graph G with $\mathbf{e}\text{-cover}_{\Theta_{r,r'}}(G) = k$ satisfies $\mathbf{tpw}(G) \leq t_k$. Indeed, such a graph does not contain a packing of $k + 1$ $\Theta_{r,r'}$ -subdivisions. Then by Theorem 3 the value $\mathbf{e}\text{-cover}_{\Theta_{r,r'}}(G)$ is bounded above by $f_2(h, t_k) \cdot \mathbf{e}\text{-pack}_{\Theta_{r,r'}}(G)$, and this concludes the proof. \square

6 Concluding Remarks

The main algorithmic contribution of this paper is a $\log(\text{OPT})$ -approximation algorithm for the parameters $v\text{-pack}_{\theta_r}$, $v\text{-cover}_{\theta_r}$, $e\text{-pack}_{\theta_r}$, and $e\text{-cover}_{\theta_r}$, for every positive integer r . This improves the results of [21] in the case of vertex packings and coverings and is the first approximation algorithm for the parameters $e\text{-pack}_{\theta_r}$ and $e\text{-cover}_{\theta_r}$ for general r . Our proof uses a reduction technique of independent interest, which is not specific to the graph θ_r and can be used for any other (classes of) graphs.

On the combinatorial side, we optimally improved the gap of the edge-Erdős–Pósa property of minor models of θ_r for every r . Also, we were able to show that every class of graphs has the (edge and vertex) Erdős–Pósa property in graphs of bounded tree-partition width, with linear gap. An other outcome of this work is that minor models of $\theta_{r,r'}$ have the edge-Erdős–Pósa. Recall that prior to this work, the only graphs for which this was known were θ_r 's.

As mentioned in [31], the planarity of a graph H is a necessary condition for the minor models of H to have the edge-Erdős–Pósa property. However, little is known on which planar graphs have this property and with which gap. This is the first direction of research that we want to highlight here. Also, the question of an approximation algorithm can be asked for packing and covering the minor models of different graphs. It was proved in [8] that the gap of the vertex-Erdős–Pósa property of minor models of every planar graph is $O(k \text{ polylog } k)$. It would be interesting to check if these results can be used to derive a $\text{polylog}(\text{OPT})$ -approximation for vertex packing and covering minor models of any planar graph.

Notice that all our results are strongly exploiting Lemma 1 that holds for every finite collection \mathcal{H} of connected graphs. Actually, what is missing in order to have an overall generalization of all of our results, is an extension of Proposition 1 where Θ_r is replaced by any finite collection \mathcal{H} of connected planar graphs. This is an interesting combinatorial problem even for particular instantiations of \mathcal{H} .

Acknowledgements This research have been supported by the Warsaw Centre of Mathematics and Computer Science (Jean-Florent Raymond), by the (Polish) National Science Centre Grant PRELUDIUM 2013/11/N/ST6/02706 (Jean-Florent Raymond), and by the E.U. (European Social Fund—ESF) and Greek national funds through the Operational Program “Education and Lifelong Learning” of the National Strategic Reference Framework (NSRF)—Research Funding Program: “Thales. Investing in knowledge society through the European Social Fund” (Dimitris Chatzidimitriou and Dimitrios M. Thilikos).

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

1. Bazgan, C., Tuza, Z., Vanderpooten, D.: Efficient algorithms for decomposing graphs under degree constraints. *Discrete Appl. Math.* **155**(8), 979–988 (2007)
2. Birmelé, E., Bondy, J.A., Reed, B.A.: The Erdős–Pósa property for long circuits. *Combinatorica* **27**, 135–145 (2007)
3. Bodlaender, H.L., Fomin, F.V., Lokshantov, D., Penninkx, E., Saurabh, S., Thilikos, D.M.: (Meta) kernelization. *CoRR*, abs/0904.0727 (2009)

4. Bodlaender, H.L., Fomin, F.V., Lokshtanov, D., Penninkx, E., Saurabh, S., Thilikos, D.M.: (Meta) kernelization. In: Proceedings of the 2009 50th Annual IEEE Symposium on Foundations of Computer Science (FOCS), pp. 629–638, Washington, DC, USA, 2009. IEEE Computer Society (2009)
5. Bodlaender, H.L., Thomassé, S., Yeo, A.: Kernel bounds for disjoint cycles and disjoint paths. *Theoret. Comput. Sci.* **412**(35), 4570–4578 (2011)
6. Caprara, A., Panconesi, A., Rizzi, R.: Packing cycles in undirected graphs. *J. Algorithms* **48**(1), 239–256 (2003)
7. Chatzidimitriou, D., Raymond, J.-F., Sau, I., Thilikos, D.M.: Minors in graphs of large θ_r -girth. *Eur. J. Combin.* [arXiv:1510.03041](https://arxiv.org/abs/1510.03041) (2015, Accepted)
8. Chekuri, C., Chuzhoy, J.: Large-treewidth graph decompositions and applications. In: Proceedings of the Forty-fifth Annual ACM Symposium on Theory of Computing (STOC), pp. 291–300, ACM, New York (2013)
9. Diestel, R.: *Graph Theory*, vol. 173 of Graduate Texts in Mathematics, third edition. Springer, Heidelberg (2005)
10. Diestel, R., Ichi Kawarabayashi, K., Wollan, P.: The Erdős–Pósa property for clique minors in highly connected graphs. *J. Comb. Theory Ser. B* **102**(2), 454–469 (2012)
11. Ding, G., Oporowski, B.: On tree-partitions of graphs. *Discrete Math.* **149**(1–3), 45–58 (1996)
12. Erdős, P., Pósa, L.: On independent circuits contained in a graph. *Can. J. Math.* **17**, 347–352 (1965)
13. Fiorini, S., Joret, G., Pietropaoli, U.: Hitting diamonds and growing cacti. In: Proceedings of Integer Programming and Combinatorial Optimization. vol. 6080 of Lecture Notes in Computer Science, pp. 191–204. Springer, Berlin (2010)
14. Fiorini, S., Joret, G., Sau, I.: Optimal Erdős–Pósa property for pumpkins. Manuscript (2013)
15. Fiorini, S., Joret, G., Wood, D.R.: Excluded forest minors and the Erdős–Pósa property. *Comb. Probab. Comput.* **22**(5), 700–721 (2013)
16. Fomin, F.V., Lokshtanov, D., Misra, N., Saurabh, S.: Planar F-deletion: approximation, kernelization and optimal FPT-algorithms. In: Proceedings of the IEEE 53rd Annual Symposium on Foundations of Computer Science (FOCS), pp. 470–479 (2012)
17. Fomin, F.V., Lokshtanov, D., Misra, N., Philip, G., Saurabh, S.: Quadratic upper bounds on the Erdős–Pósa property for a generalization of packing and covering cycles. *J. Graph Theory* **74**(4), 417–424 (2013)
18. Geelen, J., Kabell, K.: The Erdős–Pósa property for matroid circuits. *J. Comb. Theory Ser. B* **99**(2), 407–419 (2009)
19. Golovach, P.A.: Personal communication (2015)
20. Halin, R.: Tree-partitions of infinite graphs. *Discrete Math.* **97**(1–3), 203–217 (1991)
21. Joret, G., Paul, C., Sau, I., Saurabh, S., Thomassé, S.: Hitting and harvesting pumpkins. *SIAM J. Discrete Math.* **28**(3), 1363–1390 (2014)
22. Kakimura, N., Kawarabayashi, K.-i., Kobayashi, Y.: Erdős–Pósa property and its algorithmic applications: parity constraints, subset feedback set, and subset packing. In: Proceedings of the Twenty-third Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), pp. 1726–1736. SIAM (2012)
23. Kawarabayashi, K.-I., Kobayashi, Y.: Edge-disjoint odd cycles in 4-edge-connected graphs. In: Proceedings of the 29th International Symposium on Theoretical Aspects of Computer Science, (STACS), February 29th–March 3rd, 2012, Paris, France, pp. 206–217 (2012)
24. Kawarabayashi, K.-I., Nakamoto, A.: The Erdős–Pósa property for vertex- and edge-disjoint odd cycles in graphs on orientable surfaces. *Discrete Math.* **307**(6), 764–768 (2007)
25. Kloks, T., Lee, C., Liu, J.: New algorithms for k -face cover, k -feedback vertex set, and k -disjoint cycles on plane and planar graphs. In: Proceedings of Graph-Theoretic Concepts in Computer Science. vol. 2573 of Lecture Notes in Computer Science, pp. 282–295. Springer, Berlin (2002)
26. Král', D., Voss, H.-J.: Edge-disjoint odd cycles in planar graphs. *J. Comb. Theory Ser. B* **90**(1), 107–120 (2004)
27. Krivelevich, M., Nutov, Z., Salavatipour, M.R., Yuster, J.V., Yuster, R.: Approximation algorithms and hardness results for cycle packing problems. *ACM Trans. Algorithms* **3**(4), (2007). doi:[10.1145/1290672.1290685](https://doi.org/10.1145/1290672.1290685)
28. Krivelevich, M., Nutov, Z., Yuster, R.: Approximation algorithms for cycle packing problems. In: Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), pp. 556–561, SIAM, Philadelphia (2005)
29. Pontecorvi, M., Wollan, P.: Disjoint cycles intersecting a set of vertices. *J. Comb. Theory Ser. B* **102**(5), 1134–1141 (2012)

30. Rautenbach, D., Reed, B.: The Erdős–Pósa property for odd cycles in highly connected graphs. *Combinatorica* **21**, 267–278 (2001)
31. Raymond, J.-F., Sau, I., Thilikos, D.M.: An edge variant of the Erdős–Pósa property. CoRR, abs/1311.1108 (2013)
32. Raymond, J.-F., Thilikos, D. M.: Recent results on the Erdős–Pósa property. *Discrete Appl. Math.* (2017). doi:[10.1016/j.dam.2016.12.025](https://doi.org/10.1016/j.dam.2016.12.025)
33. Reed, Bruce A: *Tree Width and Tangles: A New Connectivity Measure and Some Applications*. Cambridge University Press, Cambridge (1997)
34. Reed, B.A., Robertson, N., Seymour, P.D., Thomas, R.: Packing directed circuits. *Combinatorica* **16**(4), 535–554 (1996)
35. Robertson, N., Seymour, P.D.: Graph minors. V. Excluding a planar graph. *J. Comb. Theory Ser. B* **41**(2), 92–114 (1986)
36. Salavatipour, M.R., Verstraete, J.: Disjoint cycles: integrality gap, hardness, and approximation. In: *Proceedings of Integer Programming and Combinatorial Optimization*. volume 3509 of *Lecture Notes in Computer Science*, pp. 51–65. Springer, Berlin (2005)
37. Seese, D.: Tree-partite graphs and the complexity of algorithms. In: *Proceedings of Fundamentals of Computation Theory*. volume 199 of *Lecture Notes in Computer Science*, pp. 412–421. Springer, Berlin (1985)