Sunflowers Meet Sparsity: A Linear-Vertex Kernel for Weighted Clique-Packing on Sparse Graphs

Bart M. P. Jansen
Eindhoven University of Technology, The Netherlands

Shivesh K. Roy
Eindhoven University of Technology, The Netherlands

Abstract

We study the kernelization complexity of the Weighted H-Packing problem on sparse graphs. For a fixed connected graph H, in the Weighted H-Packing problem the input is a graph G, a vertex-weight function $w: V(G) \to \mathbb{N}$, and positive integers k, t. The question is whether there exist k vertex-disjoint subgraphs $H_1, \ldots, H_k$ of G such that H_i is isomorphic to H for each $i \in [k]$ and the total weight of these $k \cdot |V(H)|$ vertices is at least t. It is known that the (unweighted) H-Packing problem admits a kernel with $\mathcal{O}(k^{|V(H)|-1})$ vertices on general graphs, and a linear kernel on planar graphs and graphs of bounded genus. In this work, we focus on case that H is a clique on $h \geq 3$ vertices (which captures Triangle Packing) and present a linear-vertex kernel for Weighted K_h-Packing on graphs of bounded expansion, along with a kernel with $\mathcal{O}(k^{1+\varepsilon})$ vertices on nowhere-dense graphs for all $\varepsilon > 0$. To obtain these results, we combine two powerful ingredients in a novel way: the Erdős-Rado Sunflower lemma and the theory of sparsity.

2012 ACM Subject Classification Mathematics of computing → Graph algorithms; Theory of computation → Parameterized complexity and exact algorithms; Theory of computation → Packing and covering problems

Keywords and phrases kernelization, weighted problems, graph packing, sunflower lemma, bounded expansion, nowhere dense

Digital Object Identifier 10.4230/LIPIcs.IPEC.2023.29

Funding This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 803421, ReduceSearch).

1 Introduction

Packing and covering problems form an important area in the study of (algorithmic) graph theory [12, 25, 32, 35, 39, 47, 48, 56]. These problems have also been actively studied from the kernelization viewpoint [1, 3, 9, 16, 24, 43, 44, 52]. Roughly speaking, kernelization is a formalization of polynomial-time preprocessing aimed at compressing the instance size in terms of a complexity parameter (see Definition 4 for a formal definition). It is well-known that a decidable parameterized problem has a kernelization algorithm if and only if it is fixed-parameter tractable (FPT) [15]. Having an FPT algorithm for the problem implies that there exists a kernel, but the size of the kernel can be exponential in the parameter. Hence, finding a polynomial (or even linear) kernel is an active area of research in parameterized complexity [1, 2, 5, 6, 10, 11, 13, 16, 26, 27, 37, 45, 46, 53, 55].

For a fixed graph H, the H-Packing problem asks, given a graph G and a positive integer k, whether there are k vertex-disjoint subgraphs $H_1, \ldots, H_k$ of G such that H_i is isomorphic to H for each $i \in [k]$. It is known that H-Packing is NP-hard whenever H has a connected component on at least three vertices [42]. For $H = K_3$ the problem is equivalent to the well-known Triangle Packing problem. An application of the sunflower lemma due to Erdős and Rado [22] gives a $\mathcal{O}(k^{|V(H)|})$ vertex-kernel for H-Packing. This bound was improved to $\mathcal{O}(k^{|V(H)|-1})$ by Abu-Khzam [1]. For restricted graph classes, such as planar graphs and graphs of bounded genus, H-Packing is known to admit a linear kernel [11, §8.4]. Very recently, the problem was shown to have kernels with $\mathcal{O}(k^{1+\varepsilon})$ vertices and edges (for every $\varepsilon > 0$) on every class of *nowhere dense* graphs [4, Theorem 4.1].

When taking $d := |V(H)|$, the H-Packing problem is a special case of d-Set Packing. The latter problem asks, given a family of size-d subsets of a universe U and integer k, whether the family contains k pairwise disjoint sets. Dell and Marx [16] showed that for $d \geq 3$, there does not exist a kernel for d-Set Packing with bit-size $\mathcal{O}(k^{d-\varepsilon})$ for any $\varepsilon > 0$, under the assumption that $\mathsf{NP} \not\subseteq \mathsf{coNP/poly}$. It is a long-standing open problem whether d-Set Packing (or the related d-Hitting Set) admits a kernel with $\mathcal{O}(k)$ universe elements. Even for special cases such as Triangle Packing, no kernels with $\mathcal{O}(k)$ vertices are known on general graphs despite intensive research into linear-vertex kernels for packing problems [9, 24].

In this work, our focus is on the *weighted* variant of H-Packing, which is defined as follows for a fixed graph H.

Weighted H-Packing **Parameter:** k
Input: An undirected graph G, a vertex-weight function $w\colon V(G) \to \mathbb{N}$, and positive integers k, t.
Question: Do there exist k vertex-disjoint subgraphs $H_1, \ldots, H_k$ of G such that H_i is isomorphic to H for each $i \in [k]$ and $\sum_{i\in[k]} \sum_{v\in V(H_i)} w(v) \geq t$?

The use of weights in the problem definition allows the problem to model a larger set of applications, since the weights can be used to capture different profits associated to a solution. Extending tractability horizons to weighted versions of problems is a natural and often challenging direction of research [14, 23, 36, 38, 40, 41].

It is not difficult to extend the sunflower-based kernel with $\mathcal{O}(k^{|V(H)|})$ vertices and edges to work in the weighted setting as well. However, the techniques used in previous papers to obtain (almost) linear kernels for sparse graph classes seem incompatible with the use of weights. For example, the meta-kernelization framework [11] does not apply to weighted problems since they do not have the *finite integer index* property.

Our results

In this work, we focus on the important special case that H is a clique on $h \geq 3$ vertices, which captures the Triangle Packing problem. We prove that Weighted K_h-Packing has a linear-vertex kernel on graph classes of bounded expansion. Classes of bounded expansion generalize planar graphs, bounded-degree graphs, and graphs excluding any fixed graph H as a minor or topological minor. Roughly speaking, a graph class $\mathcal{G}$ has bounded expansion if there exists a function $f\colon \mathbb{N} \to \mathbb{N}$ such that for each $G \in \mathcal{G}$, the edge density of each graph G' that can be obtained from a subgraph of G by contracting disjoint connected vertex sets of diameter at most r, is bounded by $f(r)$. See [51, §5.5] for formal definitions. Our main result reads as follows.

▶ **Theorem 1.** *For each graph class $\mathcal{G}$ of bounded expansion, for each integer $h \geq 3$, Weighted K_h-Packing admits a linear-vertex kernel on graphs from $\mathcal{G}$.*

Our approach extends to provide almost-linear kernels for every nowhere-dense graph class.

▶ **Theorem 2.** *For each nowhere-dense graph class $\mathcal{G}$, integer $h \geq 3$, and $\varepsilon > 0$,* Weighted K_h-Packing *admits a kernel with $\mathcal{O}(k^{1+\varepsilon})$ vertices on graphs from $\mathcal{G}$.*

The main idea behind our kernel for bounded-expansion graph classes is as follows. We start with a greedy phase that repeatedly extracts a maximum-weight K_h-subgraph. After having collected $h \cdot k$ such subgraphs, we show that if a solution exists, there is a solution in which each copy of K_h intersects one of the greedily identified subgraphs. This yields an $\mathcal{O}(k)$-sized vertex set P_0 for which we can assume any solution intersects P_0. Then we apply tools of sparsity to enrich P_0 into a slightly larger vertex set P of size $\mathcal{O}(k)$, such that the remaining vertices in G can be partitioned into $\mathcal{O}(k)$ equivalence classes in such a way that all K_h-subgraphs intersecting a class interact with the same set of $\mathcal{O}(1)$ vertices in P. Having this constant bound allows us to apply the sunflower lemma separately on each family of K_h-subgraphs intersecting a given equivalence class, in such a way that having a sunflower of *constant* size suffices to guarantee that one of the corresponding K_h-subgraphs can be avoided when making a solution. In this way, we can shrink each of the $\mathcal{O}(k)$ equivalence classes to $\mathcal{O}(1)$ vertices, giving a linear-vertex kernel. Hence our approach exploits the structural properties of sparse graphs to allow more efficient usage of the sunflower lemma. It easily generalizes to the setting of nowhere dense graph classes by utilizing a different lemma to compute the enriched set $P \supseteq P_0$, giving a bound of $\mathcal{O}(k^{\varepsilon})$ rather than $\mathcal{O}(1)$ on the number of vertices from P that can interact with the copies of K_h in a given equivalence class.

We consider the conceptual simplicity of our kernelization algorithm an appealing feature. Unlike the meta-kernelization framework [11], it does not rely on treewidth-based argumentation and the corresponding notion of protrusion replacement. (The latter yields proofs that a kernelization algorithm exists, without explicitly showing what the algorithm is.) Compared to previous (kernelization) results on sparse graphs [4, 18, 54] we only require a few tools from the sparsity theory based on neighborhood complexity and the closure lemma, and avoid the use of the technical notion of *uniform quasi-wideness*.

In our argumentation, we focus on reducing the number of vertices in the instance to $\mathcal{O}(k)$. Strictly speaking this does not ensure the total encoding size becomes bounded in k, as the weights can be arbitrarily large. However, once the number of vertices is small, the weight-compression technique of Etscheid et al. [23] can be used to get to bound the maximum weight.

Related work

The study of kernelization on restricted graph classes began with the seminal result of Alber et al. [5], who proved that Dominating Set on planar graphs admits a linear kernel. It was later extended to larger graph classes [7, 28, 29, 30, 34]. The tools from sparsity have been extensively studied in the last decades. Dvorák, Král, and Thomas gave an FPT algorithm for deciding first-order properties in classes of graphs with bounded expansion [19], which was later extended to nowhere dense graph classes by Grohe, Kreutzer, and Siebertz [33]. It was also shown in [19] that if a graph class $\mathcal{G}$ is not nowhere dense (is somewhere dense) and is closed under taking subgraphs, then model checking First Order formulae on $\mathcal{G}$ is not FPT parameterized by the length of the formula unless FPT$= W[1]$.

In terms of kernelization, the first systematic study using the modern sparsity framework was started by Drange et al. [18]. They showed that for every fixed positive integer r, the r-Dominating Set problem admits a linear kernel on bounded expansion graphs. They also gave an almost-linear kernel for the standard Dominating Set problem on nowhere dense graphs. Later, Eickmeyer et al. [20] showed that r-Dominating Set admits an almost-linear kernel on nowhere dense graphs. Pilipczuk and Siebertz [54] proved that the r-Independent Set problem admits an almost-linear kernel on every nowhere dense graph class. The above kernelization results were recently unified by Einarson and Reidl [21] and Ahn et al. [4]. Apart from that, the tools from sparsity have also been used by Demaine et al. [17] in real-world graphs.

2 Preliminaries

We use standard notation for graphs and parameterized algorithms. We refer the reader to a textbook [15] for any undefined terms. For positive integers n we define $[n] := \{1, \ldots, n\}$. We consider simple undirected graphs. A graph G has vertex set $V(G)$ and edge set $E(G)$. The open neighborhood of $v \in V(G)$ is $N_G(v) := \{u \mid \{u, v\} \in E(G)\}$, where we omit the subscript G if it is clear from context. For a vertex set $S \subseteq V(G)$ the open neighborhood of S, denoted $N_G(S)$, is defined as $S := \bigcup_{v \in S} N_G(v) \setminus S$. For $S \subseteq V(G)$, the graph induced by S is denoted by $G[S]$. For two vertices x, y in a graph G, an $x - y$ path is a sequence $(x = v_1, \ldots, v_k = y)$ of vertices such that $\{v_i, v_{i+1}\} \in E(G)$ for all $i \in [k-1]$. Furthermore, the vertices $v_2 \ldots, v_{k-1}$ are called the *internal vertices* of the $x - y$ path. We say that a subgraph H of G *intersects* a vertex set $S \subseteq V(G)$ if $V(H) \cap S \neq \emptyset$.

We next state the following lemma due to Erdős-Rado [22]. Before presenting the lemma we define the terminology used in the lemma.

A *sunflower* $\mathcal{S}$ with k sets and *core* X is a collection of sets $S_1, \ldots, S_k$ such that $S_i \cap S_j = X$ for all $i \neq j$, and such that $S_i \setminus X \neq \emptyset$ for all $i \in [k]$. The sets $S_i \setminus X$ are *petals* of the sunflower $\mathcal{S}$.

▶ **Theorem 3** (Sunflower lemma, [15, Theorem 2.25]). *Let $\mathcal{A}$ be a family of sets (without duplicates) over a universe U, such that each set in $\mathcal{A}$ has cardinality exactly d. If $|\mathcal{A}| > d!(k-1)^d$, then $\mathcal{A}$ contains a sunflower with k petals and such a sunflower can be computed in time polynomial in $|\mathcal{A}|, |U|$, and k.*

For completeness, we now give the formal definition of a kernelization. A *parameterized problem* Q is a subset of $\Sigma^* \times \mathbb{N}_+$, where Σ is a finite alphabet.

▶ **Definition 4** (Kernel). *Let $Q, Q' \subseteq \Sigma^* \times \mathbb{N}_+$ be parameterized problems and let $h \colon \mathbb{N}_+ \to \mathbb{N}_+$ be a computable function. A* generalized kernel for Q into Q' of size $h(k)$ *is an algorithm that, on input $(x, k) \in \Sigma^* \times \mathbb{N}_+$, takes time polynomial in $|x| + k$ and outputs an instance (x', k') such that:*
1. *$|x'|$ and k' are bounded by $h(k)$, and*
2. *$(x', k') \in Q'$ if and only if $(x, k) \in Q$.*

The algorithm is a kernel *for Q if $Q = Q'$. It is a* polynomial (generalized) kernel *if $h(k)$ is a polynomial.*

Sparsity. The theory of sparsity was introduced by Nesetril and Ossona de Mendez [49, 50] using the notions of bounded expansion and nowhere denseness. Many important sparse graphs, like classes of bounded treewidth, planar graphs, graphs with bounded genus, apex-minor-free graphs, (topological)-minor free graphs, and graphs of bounded degree have bounded expansion. We refer the reader to the book [51] for a detailed introduction to the topic.

We will need some basic notation and tools for sparse graphs from earlier work [18, 20, 54] to prove our results. Let G be a graph and $X \subseteq V(G)$ be a subset of vertices. For a vertex $v \in V(G) \setminus X$ and a positive integer r, we define the *r-projection* of v onto X as the set of all the vertices $w \in X$, for which there is a $v - w$ path in G of length at most r whose internal vertices do not belong to X. The r-projection of v onto a set X is denoted by $M_r(v, X)$.

Now we are ready to state the lemmas. The following lemma says that any vertex set $X \subseteq V(G)$ of a bounded expansion graph G can be "closed" to a set $\widehat{X}$ whose size is asymptotically the same as $|X|$, such that the r-projection of any vertex outside $\widehat{X}$ onto $\widehat{X}$ has constant size.

▶ **Lemma 5** (Closure lemma, [18] Lemma 2.2)**.** *Let $\mathcal{G}$ be a class of bounded expansion. There exists a polynomial-time algorithm that, given a graph $G \in \mathcal{G}$, a non-negative integer r, and a set $X \subseteq V(G)$, computes a vertex-set $\widehat{X}$ with the following properties.*
1. *$X \subseteq \widehat{X} \subseteq V(G)$,*
2. *$|\widehat{X}| = \mathcal{O}(|X|)$, and*
3. *$|M_r(v, \widehat{X})| \leq \alpha \in \mathcal{O}(1)$, for each $v \in V(G) \setminus \widehat{X}$.*

We note that in the above, the $\mathcal{O}(\cdot)$ notation also hides the factors depending on r and the graph class $\mathcal{G}$.

We also cite the corresponding closure lemma for nowhere dense graphs due to Eickmeyer et al. [20].

▶ **Lemma 6** (Closure lemma for nowhere dense graphs, [20, 54])**.** *Let $\mathcal{G}$ be a nowhere dense class of graphs. There are a function $f_{cl} \colon \mathbb{N} \times \mathbb{R} \to \mathbb{N}$ and a polynomial-time algorithm that, given a graph $G \in \mathcal{G}$, a non-negative integer r, a set $X \subseteq V(G)$, and $\varepsilon > 0$, computes a vertex-set $\widehat{X}$ with the following properties.*
1. *$X \subseteq \widehat{X} \subseteq V(G)$,*
2. *$|\widehat{X}| = f_{cl}(r, \varepsilon) \cdot |X|^{1+\varepsilon}$, and*
3. *$|M_r(v, \widehat{X})| \leq f_{cl}(r, \varepsilon) \cdot |X|^{\varepsilon}$, for each $v \in V(G) \setminus \widehat{X}$.*

In [18], Drange et al. proved that the number of distinct r-projections on a vertex set $X \subseteq V(G)$ of a bounded expansion graph G is linear in the cardinality of X.

▶ **Lemma 7** ([18, Lemma 2.3])**.** *Let $\mathcal{G}$ be a class of bounded expansion and let r be a non-negative integer. Let $G \in \mathcal{G}$ be a graph and $X \subseteq V(G)$. Then*

$$|\{Y : Y = M_r(v, X) \text{ for some } v \in V(G) \setminus X\}| \leq c \cdot |X|,$$

for some constant c depending only on r and the graph class $\mathcal{G}$.

A similar bound exists for nowhere dense graphs.

▶ **Lemma 8** ([20, Theorem 3])**.** *Let $\mathcal{G}$ be a nowhere dense class of graphs. There is a function $f_{nbr} \colon \mathbb{N} \times \mathbb{R} \to \mathbb{N}$ such that for every non-negative integer r, real $\varepsilon > 0$, graph $G \in \mathcal{G}$, and vertex set $X \subseteq V(G)$, we have*

$$|\{Y : Y = M_r(v, X) \text{ for some } v \in V(G) \setminus X\}| \leq f_{nbr}(r, \varepsilon) \cdot |X|^{1+\varepsilon},$$

for some constant c depending only on r and the graph class $\mathcal{G}$.

3 Kernelization for Weighted K_h-Packing on Sparse Graphs

In this section we present our kernels for Weighted K_h-Packing. We start by introducing some problem-specific terminology that will be useful to streamline our arguments.

A *solution* to an instance (G, w, k, t) of Weighted K_h-Packing is a sequence of vertex-disjoint subgraphs $H_1, \ldots, H_k$ of G such that H_i is isomorphic to K_h for each $i \in [k]$ and $\sum_{i \in [k]} \sum_{v \in V(H_i)} w(v) \geq t$.

▶ **Definition 9** (P-bound solution and solution confined to $\mathcal{H}$). *Let (G, w, k, t) be an instance of* WEIGHTED K_h-PACKING. *For a vertex set $P \subseteq V(G)$, a solution $H_1, \ldots, H_k$ of (G, w, k, t) is P-bound if $V(H_i) \cap P \neq \emptyset$ for all $i \in [k]$.*

For a collection $\mathcal{H}$ of subgraphs isomorphic to K_h in G, a solution $H_1, \ldots, H_k$ of (G, w, k, t) is said to be confined to $\mathcal{H}$ if $H_i \in \mathcal{H}$ for all $i \in [k]$.

We now show how the sunflower lemma can be combined with the theory of sparsity to get a linear-vertex kernel for WEIGHTED K_h-PACKING on bounded expansion graph classes.

▶ **Theorem 1.** *For each graph class $\mathcal{G}$ of bounded expansion, for each integer $h \geq 3$,* WEIGHTED K_h-PACKING *admits a linear-vertex kernel on graphs from $\mathcal{G}$.*

Proof. Let (G, w, k, t) be an instance of WEIGHTED K_h-PACKING with $G \in \mathcal{G}$. We refer to a subgraph H_i of G isomorphic to K_h as a *copy* of K_h. In the following proof, we will treat such H_i both as a subgraph of G and as a vertex subset of G, depending on which is more convenient. Our kernelization algorithm performs the following steps.

Algorithm.

1. Compute a greedy packing $\mathcal{P}$ of up to hk vertex-disjoint copies $H_1, \ldots, H_{hk}$ of K_h in G such that H_1 is a maximum-weighted copy of K_h in G, and for each $i \in \{2, \ldots, hk\}$, the copy H_i is a maximum-weighted copy of K_h in the graph $G - \left(\bigcup_{j=1}^{i-1} V(H_j)\right)$. While following the above greedy procedure, if it is not possible to pack hk disjoint copies of K_h then we obtain a maximal packing.
2. Let $P_0 := V(\mathcal{P})$. Invoke the algorithm of Lemma 5 with G, $r = 2$, and $P_0 \subseteq V(G)$ to obtain a vertex set P such that:
   a. $P_0 \subseteq P \subseteq V(G)$,
   b. $|P| = \mathcal{O}(|P_0|) = \mathcal{O}(k)$, and
   c. $|M_2(v, P)| \leq \alpha \in \mathcal{O}(1)$ for each $v \in V(G) \setminus P$,

   where α is a constant depending on r and the graph class $\mathcal{G}$.
3. Partition the vertices of $V(G) \setminus P$ into equivalence classes $C_1, \ldots, C_m$ based on their 2-projection onto the set P, i.e., for every equivalence class C_i, and for every pair of distinct vertices $x, y \in C_i$, we have $M_2(x, P) = M_2(y, P)$. (Due to Lemma 7, we have $m = \mathcal{O}(|P|) = \mathcal{O}(k)$.)
4. Let $\mathcal{H}$ be the set family containing the (vertex sets of) all copies of K_h in G.
   - For each equivalence class C_i of Step 3, do the following.

     Let $\mathcal{H}_i \subseteq \mathcal{H}$ be the (vertex sets of) copies of K_h in G that contain a vertex of C_i.
     while $|\mathcal{H}_i| > h!(h\alpha + 1)^h$ **do**
       Apply Theorem 3 to obtain a sunflower $\mathcal{S} \subseteq \mathcal{H}_i$ with $h\alpha + 2$ copies of K_h.
       Let $S_r \in \mathcal{S}$ be a copy of K_h with minimum weight. Remove S_r from $\mathcal{H}$ and $\mathcal{H}_i$.
     end while
5. Let $\mathcal{H}' \subseteq \mathcal{H}$ be the (reduced) set obtained after Step 4. Define $G' := G[V(\mathcal{H}')]$ and output (G', w, k, t) as the result of the kernelization.

   (Note that in principle, the encoding size of the weight function can be unbounded in terms of k, which can be resolved by a standard application of the weight reduction technique by Frank and Tardos [31] as explained by Etscheid et al. [23].)

This concludes the description of the algorithm.

Analysis. It is easy to observe that since we treat h as a constant, the above algorithm takes polynomial time: each step of the algorithm takes polynomial time and each step is applied a polynomial number of times. We now prove the correctness of the algorithm. Towards this, we first prove the following claim which says that any solution of (G, w, k, t) which is not already P-bound can be converted to a solution where the number of copies of K_h intersecting with P is strictly larger.

▷ Claim 10. If $\mathcal{H}^*$ is a solution of (G, w, k, t) and $S_f \in \mathcal{H}^*$ such that $V(S_f) \cap P = \emptyset$, then there exists a set $S_j \in \mathcal{P}$ such that $(\mathcal{H}^* \setminus \{S_f\}) \cup \{S_j\}$ is a solution of (G, w, k, t).

Proof. First note that, by the construction of the packing $\mathcal{P}$ (in Step 1 of the algorithm), if $|\mathcal{P}| < hk$ then the constructed set $\mathcal{P}$ is an inclusion-maximal packing, implying that all copies of H in G intersect $V(\mathcal{P})$ and therefore P. Under the stated assumptions, as S_f is a copy of K_h which is completely contained in the graph $G - P$, we have $|\mathcal{P}| = hk$. Moreover, for every $S_p \in \mathcal{P}$, it holds that $w(S_p) \geq w(S_f)$ since the copy S_f was available to choose in the iteration when the algorithm selected S_f, while the algorithm selects a maximum-weight copy at every step.

Since $\mathcal{H}^*$ is a packing of k copies of K_h, $S_f \in \mathcal{H}^*$, and $V(S_f) \cap P = \emptyset$, at most $h(k-1)$ vertices of $\mathcal{H}^*$ can intersect with the hk copies of K_h from the packing $\mathcal{P}$. Hence, there is at least one copy of K_h say $S_j \in \mathcal{P}$ such that $V(S_j) \cap V(\mathcal{H}^*) = \emptyset$. Moreover, we have $w(S_j) \geq w(S_f)$ as $S_j \in \mathcal{P}$. Thus the set $(\mathcal{H}^* \setminus \{S_f\}) \cup \{S_j\}$ is a solution of (G, w, k, t). ◁

Next, using the above claim we prove that if there is a solution to (G, w, k, t) then there is a P-bound solution.

▷ Claim 11. If (G, w, k, t) has a solution, then (G, w, k, t) has a P-bound solution.

Proof. Let $\mathcal{H}^*$ be a solution of (G, w, k, t). If $V(S_f) \cap P \neq \emptyset$ for all $S_f \in \mathcal{H}^*$ then by Definition 9, the set $\mathcal{H}^*$ is a P-bound solution. Otherwise, while there exists a set $S_f \in \mathcal{H}^*$ such that $V(S_f) \cap P = \emptyset$, we use Claim 10 to obtain a set $S_j \in \mathcal{P}$ such that $(\mathcal{H}^* \setminus \{S_f\}) \cup \{S_j\}$ is a solution of (G, w, k, t) with strictly fewer copies of K_h (than in $\mathcal{H}^*$) which are disjoint from the set P. ◁

Finally, in the following claim we prove that if (G, w, k, t) has a P-bound solution (which is guaranteed due to the above claim) then removal of the set S_r in Step 4 of the algorithm is safe.

▷ Claim 12. Suppose Step 4 of the above algorithm removes the set S_r from $\mathcal{H}$. If (G, w, k, t) has a P-bound solution which is confined to $\mathcal{H}$, then (G, w, k, t) has a P-bound solution which is confined to $\mathcal{H} \setminus \{S_r\}$.

Proof. Let $\mathcal{H}^* \subseteq \mathcal{H}$ be a P-bound solution of (G, w, k, t) which is confined to $\mathcal{H}$. If $S_r \notin \mathcal{H}^*$, then $\mathcal{H}^*$ is also a P-bound solution which is confined to $\mathcal{H} \setminus \{S_r\}$. Therefore assume that $S_r \in \mathcal{H}^*$. Let $\mathcal{S} := \{S_1, \ldots, S_{h\alpha+2}\}$ be the sunflower found in Step 4 of the above algorithm when it removes $S_r \in \mathcal{S}$ from $\mathcal{H}$, let X be its core, and let C_i be the equivalence class it considered when it found the sunflower. We now show that there exists a "free" set $S_f \in \mathcal{S} \setminus \{S_r\}$ such that the set $(\mathcal{H}^* \setminus \{S_r\}) \cup \{S_f\}$ is a solution of (G, w, k, t) which is confined to $\mathcal{H} \setminus \{S_r\}$. Towards this, we first derive the following: some set S_f of the sunflower $\mathcal{S} \setminus \{S_r\}$ is disjoint from $V(\mathcal{H}^* \setminus \{S_r\})$.

$$\exists S_f \in \mathcal{S} \setminus \{S_r\} \colon V(S_f) \cap V(\mathcal{H}^* \setminus \{S_r\}) = \emptyset. \tag{1}$$

Assume for a contradiction that there does not exist such a set S_f. First, note that since $S_r \in \mathcal{H}^*$, the core X of the sunflower $\mathcal{S}$ is contained in S_r, and $\mathcal{H}^*$ is a collection of vertex-disjoint copies of K_h, we have $X \cap V(\mathcal{H}^* \setminus \{S_r\}) = \emptyset$. Hence the copies of K_h in the set $\mathcal{H}^* \setminus \{S_r\}$ only intersect with petals of the sunflower $\mathcal{S} \setminus \{S_r\}$. Moreover, as the petals of a sunflower are pairwise disjoint, each copy of K_h from the set $\mathcal{H}^* \setminus \{S_r\}$ can intersect with at most h petals of the sunflower $\mathcal{S} \setminus \{S_r\}$.

Since all the $h\alpha + 1$ petals of the sunflower $\mathcal{S} \setminus \{S_r\}$ intersect with the set $V(\mathcal{H}^* \setminus \{S_r\})$ and a single copy of K_h from the set $\mathcal{H}^* \setminus \{S_r\}$ can hit at most h petals, the number of copies of K_h from $\mathcal{H}^* \setminus \{S_r\}$ intersecting with the petals of sunflower $\mathcal{S} \setminus \{S_r\}$ is at least $\alpha + 1$. Let $\mathcal{H}^*_{\mathcal{S}} := \{H_{i_1}, \ldots, H_{i_\ell}\} \subseteq \mathcal{H}^*$ be the set containing copies of K_h from the set $\mathcal{H}^* \setminus \{S_r\}$ which intersect with a petal of sunflower $\mathcal{S} \setminus \{S_r\}$. We have $\ell \geq \alpha + 1$.

We will prove that for each $q \in [\ell]$, there is a path P_q in G of length at most 2 that starts in a vertex of equivalence class C_i, ends in a vertex of $P \cap V(H_{i_q})$, and does not intersect any other vertex of P. Towards this end, let p_q be an arbitrary vertex of $V(H_{i_q}) \cap P$, which exists since the solution $\mathcal{H}^*$ is P-bound. By choice of $\mathcal{H}^*_{\mathcal{S}}$, the set $V(H_{i_q})$ intersects some petal $S_z \setminus X$ for $z \neq r$ of the sunflower $\mathcal{S} \setminus \{S_r\}$; let $x_q \in V(H_{i_q}) \cap (S_z \setminus X)$ be a vertex at which the sets intersect, and note that $\{x_q, p_q\} \in E(G)$ since they are both contained in the common clique H_{i_q}. Each set S_z contains a vertex from equivalence class C_i by the specification of Step 4, so there is a vertex $c_z \in S_z \cap C_i$. We have $\{c_z, x_q\} \in E(G)$ since these vertices are contained in the common clique S_z. Observe that $c_z \notin P$ since the equivalence classes partition the vertex set $V(G) \setminus P$. Now, if $x_q \notin P$ then the path (c_z, x_q, p_z) is the desired path P_q; if $x_q \in P$ then we take (c_z, x_q) as the path P_q. Note that in both cases, P_q is indeed a path in G on at most 2 edges starting in C_i and ending in a vertex of $P \cap V(H_{i_q})$.

Hence for each $H_{i_q} \in \mathcal{H}^*_{\mathcal{S}}$, there exists a vertex in the equivalence class C_i that can reach a vertex of $P \cap V(H_{i_q})$ by a path of length at most 2 whose internal vertices do not belong to P. By definition of the equivalence classes C_i, if one vertex in C_i has such a path to P, then *all* vertices of C_i have such a path. As $\ell \geq \alpha + 1$ and the copies in $\mathcal{H}^*_{\mathcal{S}}$ are disjoint, for any $v \in C_i$ we have $|M_2(v, P)| \geq \ell \geq \alpha + 1$. This contradicts that $|M_2(v, P)| \leq \alpha$ which was ensured by Step 2 of the above algorithm. Hence we establish (1).

Now we continue with the remaining proof of Claim 12. As there is a set $S_f \in \mathcal{S} \setminus \{S_r\}$ of the sunflower $\mathcal{S} \setminus \{S_r\}$ with $V(S_f) \cap V(\mathcal{H}^* \setminus \{S_r\}) = \emptyset$ and $w(S_f) \geq w(S_r)$ by our choice of S_r in Step 4 of the above algorithm, the set $\widetilde{\mathcal{H}} := (\mathcal{H}^* \setminus \{S_r\}) \cup \{S_f\}$ is a solution of (G, w, k, t). Note that if $V(S_f) \cap P \neq \emptyset$ then the set $\widetilde{\mathcal{H}}$ is also a P-bound solution. Otherwise we invoke Claim 10 (with the set $\widetilde{\mathcal{H}}$, and $S_f \in \widetilde{\mathcal{H}}$) to obtain a K_h-copy $S_j \in \mathcal{P}$ of the packing $\mathcal{P}$ such that the set $(\widetilde{\mathcal{H}} \setminus \{S_f\}) \cup \{S_j\}$ is a P-bound solution of (G, w, k, t). Note that the latter solution is also confined to $\mathcal{H} \setminus \{S_r\}$, since the copy $S_j \in \mathcal{P}$ was added to the set $\mathcal{H}$ at the initialization and can never be removed: it does not occur in any $\mathcal{H}_i$ since its vertex set is fully contained in P; it does not intersect any equivalence class of $V(G) \setminus P$. This concludes the proof of Claim 12. ◁

It follows from the preceding two claims that the output instance (G', t, k, w) is equivalent to the input (G, w, k, t). Since the output is an induced subgraph of the input, one direction is trivial. For the other direction, if the input instance has a solution, it has a P-bound solution by Claim 11. Then by Claim 12 and induction, there is a solution confined to $\mathcal{H}'$, the final state of the variable $\mathcal{H}$. Since G' contains all copies of K_h contained in $\mathcal{H}'$, this proves the output instance also has a solution.

We conclude the proof of Theorem 1 by giving a bound on the number vertices of the reduced graph G'.

▷ Claim 13. $|V(G')| = \mathcal{O}(k)$.

Proof. Note that $|P| = \mathcal{O}(k)$ by Step 2. It follows that G' contains at most $\mathcal{O}(k)$ vertices which belong to P. To prove the claim, we show that the number of vertices of $V(G') \setminus P$ is also bounded by $\mathcal{O}(k)$.

For each equivalence class C_i of $V(G) \setminus P$, let $\mathcal{H}'_i$ denote the contents of $\mathcal{H}_i$ upon termination of the algorithm. The while-loop of Step 4 ensures that $|\mathcal{H}'_i| \leq h!(h\alpha + 1)^h$.

For each $v \in V(G') \setminus P$, by definition of $G' = G[V(\mathcal{H}')]$ there exists an equivalence class C_i of $V(G) \setminus P$ and a copy $H_j \in \mathcal{H}'_i$, such that $v \in V(H_j)$. Hence $V(G') \setminus P$ is contained in $\bigcup_i \bigcup_{H_j \in \mathcal{H}'_i} V(H_j)$. Since there are $\mathcal{O}(k)$ choices for i by Lemma 7, while $|\mathcal{H}'_i| \leq h!(h\alpha + 1)^h = \mathcal{O}(1)$, while each copy H_j also consists of $\mathcal{O}(1)$ vertices, it follows that $|V(G') \setminus P| = \mathcal{O}(k)$. This concludes the proof. ◁

This concludes the proof of Theorem 1. ◀

The argument for nowhere-dense graphs is almost identical.

▶ **Theorem 2.** *For each nowhere-dense graph class $\mathcal{G}$, integer $h \geq 3$, and $\varepsilon > 0$, WEIGHTED K_h-PACKING admits a kernel with $\mathcal{O}(k^{1+\varepsilon})$ vertices on graphs from $\mathcal{G}$.*

Proof. For WEIGHTED K_h-PACKING on a nowhere dense graph class $\mathcal{C}$, one can use the same approach. Let $\varepsilon' := \frac{\varepsilon}{h+1}$. In the algorithm, we use Lemma 6 for value ε' instead of Lemma 5; this means that the closure set P has size $\mathcal{O}(k^{1+\varepsilon'})$ rather than $\mathcal{O}(k)$, and that the bound α on $|M_2(v, P)|$ becomes $\mathcal{O}(k^{\varepsilon'})$ rather than $\mathcal{O}(1)$. For the analysis, we use Lemma 8 for value ε' instead of Lemma 7, which means the number of equivalence classes of $V(G) \setminus P$ becomes $\mathcal{O}(k^{1+\varepsilon'})$ rather than $\mathcal{O}(k)$. The rest of the algorithm and its correctness proof is identical.

As in the proof of Claim 13, we can bound the number of vertices in the resulting graph G' using the insight that every vertex of G' is either contained in P, or belongs to some copy H_j of K_h that remains in a set $\mathcal{H}'_i$ for some equivalence class C_i of $V(G) \setminus P$. The key insight is again that $|\mathcal{H}'_i| \leq h!(h\alpha + 1)^h$ due to the application of the Sunflower lemma.

Hence the number of vertices in the reduced instance G' is bounded as follows:

$$\begin{aligned}
|V(G')| \leq& |P| + |\bigcup_i \bigcup_{H_j \in \mathcal{H}'_i} V(H_j)| \\
\leq& \mathcal{O}(k^{1+\varepsilon'}) + \mathcal{O}(k^{1+\varepsilon'} \cdot h!(h\alpha + 1)^h \cdot h) \\
\leq& \mathcal{O}(k^{1+\varepsilon'}) + \mathcal{O}(k^{1+\varepsilon'} \cdot h! 2^h h^h k^{\varepsilon' \cdot h} \cdot h) && \text{since } h\alpha + 1 \leq 2h\alpha \\
\leq& \mathcal{O}(k^{1+\varepsilon' + \varepsilon' \cdot h}) = \mathcal{O}(k^{1+\varepsilon'(h+1)}). && \text{since } h \in \mathcal{O}(1)
\end{aligned}$$

Since we chose $\varepsilon' = \frac{\varepsilon}{h+1}$, the number of vertices in the kernel is indeed bounded by $\mathcal{O}(k^{1+\varepsilon})$, as required. ◀

4 Conclusions

We have shown that for a fixed complete graph K_h, the WEIGHTED K_h-PACKING problem admits a linear-vertex kernel on bounded-expansion graphs and an almost-linear kernel on nowhere-dense graphs. Whether there is a linear-vertex kernel for the associated WEIGHTED K_h-HITTING problem is an interesting problem for further study. In this problem, the input consists of a graph G, weight function $w \colon V(G) \to \mathbb{N}$, and integers k, t; the question is whether there is a vertex set of size at most k and weight at most t that intersects all

K_h-subgraphs of G. In the unweighted setting, the kernelization complexity of packing problems typically matches that of the related hitting problem [8, 24]. In the weighted setting, the situation seems different and we do not know how to extend our techniques to WEIGHTED K_h-HITTING.

To illustrate the difficulty of hitting over packing in the presence of weights, observe the following. If $C_i \subseteq V(G)$ is a vertex subset such that all copies of K_h which intersect C_i also intersect a vertex set P_i of size $\mathcal{O}(1)$, then it effectively means that any packing of disjoint copies of K_h uses $\mathcal{O}(1)$ vertices of C_i, so that only a limited redundancy is needed in terms of which vertices of C_i are preserved in the kernel. But note that in the same scenario, a solution to WEIGHTED K_h-HITTING can contain up to k vertices from C_i: even though the K_h-subgraphs through C_i can be intersected by the vertex set P_i of *size* $\mathcal{O}(1)$, the weight of these vertices may be much larger than the weight of k vertices from C_i hitting the same subgraphs. Hence solutions to the hitting problem may select more than a constant number of vertices from C_i, which leads to having to store more vertices in the kernel.

References

1 Faisal N. Abu-Khzam. An improved kernelization algorithm for r-set packing. *Inf. Process. Lett.*, 110(16):621–624, 2010. doi:10.1016/j.ipl.2010.04.020.

2 Faisal N. Abu-Khzam. A kernelization algorithm for d-hitting set. *J. Comput. Syst. Sci.*, 76(7):524–531, 2010. doi:10.1016/j.jcss.2009.09.002.

3 Akanksha Agrawal, Daniel Lokshtanov, Diptapriyo Majumdar, Amer E. Mouawad, and Saket Saurabh. Kernelization of cycle packing with relaxed disjointness constraints. *SIAM J. Discret. Math.*, 32(3):1619–1643, 2018. doi:10.1137/17M1136614.

4 Jungho Ahn, Jinha Kim, and O-joung Kwon. Unified almost linear kernels for generalized covering and packing problems on nowhere dense classes. *CoRR*, abs/2207.06660, 2022. doi:10.48550/arXiv.2207.06660.

5 Jochen Alber, Michael R. Fellows, and Rolf Niedermeier. Polynomial-time data reduction for dominating set. *J. ACM*, 51(3):363–384, 2004. doi:10.1145/990308.990309.

6 Noga Alon, Gregory Z. Gutin, Eun Jung Kim, Stefan Szeider, and Anders Yeo. Solving max-r-sat above a tight lower bound. *Algorithmica*, 61(3):638–655, 2011. doi:10.1007/s00453-010-9428-7.

7 Noga Alon and Shai Gutner. Kernels for the dominating set problem on graphs with an excluded minor. *Electron. Colloquium Comput. Complex.*, TR08-066, 2008. arXiv:TR08-066.

8 Stéphane Bessy, Marin Bougeret, Dimitrios M. Thilikos, and Sebastian Wiederrecht. Kernelization for graph packing problems via rainbow matching. *CoRR*, abs/2207.06874, 2022. doi:10.48550/arXiv.2207.06874.

9 Stéphane Bessy, Marin Bougeret, Dimitrios M. Thilikos, and Sebastian Wiederrecht. Kernelization for graph packing problems via rainbow matching. In Nikhil Bansal and Viswanath Nagarajan, editors, *Proceedings of the 2023 ACM-SIAM Symposium on Discrete Algorithms, SODA 2023, Florence, Italy, January 22-25, 2023*, pages 3654–3663. SIAM, 2023. doi:10.1137/1.9781611977554.ch139.

10 Daniel Binkele-Raible, Henning Fernau, Fedor V. Fomin, Daniel Lokshtanov, Saket Saurabh, and Yngve Villanger. Kernel(s) for problems with no kernel: On out-trees with many leaves. *ACM Trans. Algorithms*, 8(4):38:1–38:19, 2012. doi:10.1145/2344422.2344428.

11 Hans L. Bodlaender, Fedor V. Fomin, Daniel Lokshtanov, Eelko Penninkx, Saket Saurabh, and Dimitrios M. Thilikos. (meta) kernelization. *J. ACM*, 63(5):44:1–44:69, 2016. doi:10.1145/2973749.

12 Marthe Bonamy, Edouard Bonnet, Hugues Déprés, Louis Esperet, Colin Geniet, Claire Hilaire, Stéphan Thomassé, and Alexandra Wesolek. Sparse graphs with bounded induced cycle packing number have logarithmic treewidth. In Nikhil Bansal and Viswanath Nagarajan, editors, *Proceedings of the 2023 ACM-SIAM Symposium on Discrete Algorithms, SODA 2023, Florence, Italy, January 22-25, 2023*, pages 3006–3028. SIAM, 2023. doi:10.1137/1.9781611977554.ch116.

13 Jianer Chen, Iyad A. Kanj, and Weijia Jia. Vertex cover: Further observations and further improvements. *J. Algorithms*, 41(2):280–301, 2001. `doi:10.1006/jagm.2001.1186`.

14 Miroslav Chlebík and Janka Chlebíková. Crown reductions for the minimum weighted vertex cover problem. *Discret. Appl. Math.*, 156(3):292–312, 2008. `doi:10.1016/j.dam.2007.03.026`.

15 Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015. `doi:10.1007/978-3-319-21275-3`.

16 Holger Dell and Dániel Marx. Kernelization of packing problems. In Yuval Rabani, editor, *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2012, Kyoto, Japan, January 17-19, 2012*, pages 68–81. SIAM, 2012. `doi:10.1137/1.9781611973099.6`.

17 Erik D. Demaine, Felix Reidl, Peter Rossmanith, Fernando Sánchez Villaamil, Somnath Sikdar, and Blair D. Sullivan. Structural sparsity of complex networks: Bounded expansion in random models and real-world graphs. *J. Comput. Syst. Sci.*, 105:199–241, 2019. `doi:10.1016/j.jcss.2019.05.004`.

18 Pål Grønås Drange, Markus Sortland Dregi, Fedor V. Fomin, Stephan Kreutzer, Daniel Lokshtanov, Marcin Pilipczuk, Michal Pilipczuk, Felix Reidl, Fernando Sánchez Villaamil, Saket Saurabh, Sebastian Siebertz, and Somnath Sikdar. Kernelization and sparseness: the case of dominating set. In Nicolas Ollinger and Heribert Vollmer, editors, *33rd Symposium on Theoretical Aspects of Computer Science, STACS 2016, February 17-20, 2016, Orléans, France*, volume 47 of *LIPIcs*, pages 31:1–31:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016. `doi:10.4230/LIPIcs.STACS.2016.31`.

19 Zdenek Dvorák, Daniel Král, and Robin Thomas. Testing first-order properties for subclasses of sparse graphs. *J. ACM*, 60(5):36:1–36:24, 2013. `doi:10.1145/2499483`.

20 Kord Eickmeyer, Archontia C. Giannopoulou, Stephan Kreutzer, O-joung Kwon, Michal Pilipczuk, Roman Rabinovich, and Sebastian Siebertz. Neighborhood complexity and kernelization for nowhere dense classes of graphs. In Ioannis Chatzigiannakis, Piotr Indyk, Fabian Kuhn, and Anca Muscholl, editors, *44th International Colloquium on Automata, Languages, and Programming, ICALP 2017, July 10-14, 2017, Warsaw, Poland*, volume 80 of *LIPIcs*, pages 63:1–63:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017. `doi:10.4230/LIPIcs.ICALP.2017.63`.

21 Carl Einarson and Felix Reidl. A general kernelization technique for domination and independence problems in sparse classes. In Yixin Cao and Marcin Pilipczuk, editors, *15th International Symposium on Parameterized and Exact Computation, IPEC 2020, December 14-18, 2020, Hong Kong, China (Virtual Conference)*, volume 180 of *LIPIcs*, pages 11:1–11:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. `doi:10.4230/LIPIcs.IPEC.2020.11`.

22 P. Erdös and R. Rado. Intersection theorems for systems of sets. *Journal of the London Mathematical Society*, s1-35(1):85–90, 1960. `doi:10.1112/jlms/s1-35.1.85`.

23 Michael Etscheid, Stefan Kratsch, Matthias Mnich, and Heiko Röglin. Polynomial kernels for weighted problems. *J. Comput. Syst. Sci.*, 84:1–10, 2017. `doi:10.1016/j.jcss.2016.06.004`.

24 Fedor V. Fomin, Tien-Nam Le, Daniel Lokshtanov, Saket Saurabh, Stéphan Thomassé, and Meirav Zehavi. Subquadratic kernels for implicit 3-hitting set and 3-set packing problems. *ACM Trans. Algorithms*, 15(1):13:1–13:44, 2019. `doi:10.1145/3293466`.

25 Fedor V. Fomin, Daniel Lokshtanov, Neeldhara Misra, Geevarghese Philip, and Saket Saurabh. Quadratic upper bounds on the erdős-pósa property for a generalization of packing and covering cycles. *J. Graph Theory*, 74(4):417–424, 2013. `doi:10.1002/jgt.21720`.

26 Fedor V. Fomin, Daniel Lokshtanov, Neeldhara Misra, and Saket Saurabh. Planar f-deletion: Approximation, kernelization and optimal FPT algorithms. In *53rd Annual IEEE Symposium on Foundations of Computer Science, FOCS 2012, New Brunswick, NJ, USA, October 20-23, 2012*, pages 470–479. IEEE Computer Society, 2012. `doi:10.1109/FOCS.2012.62`.

27 Fedor V. Fomin, Daniel Lokshtanov, Saket Saurabh, and Dimitrios M. Thilikos. Bidimensionality and kernels. In Moses Charikar, editor, *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2010, Austin, Texas, USA, January 17-19, 2010*, pages 503–510. SIAM, 2010. `doi:10.1137/1.9781611973075.43`.

28 Fedor V. Fomin, Daniel Lokshtanov, Saket Saurabh, and Dimitrios M. Thilikos. Linear kernels for (connected) dominating set on H-minor-free graphs. In Yuval Rabani, editor, *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2012, Kyoto, Japan, January 17-19, 2012*, pages 82–93. SIAM, 2012. doi:10.1137/1.9781611973099.7.

29 Fedor V. Fomin, Daniel Lokshtanov, Saket Saurabh, and Dimitrios M. Thilikos. Linear kernels for (connected) dominating set on graphs with excluded topological subgraphs. In Natacha Portier and Thomas Wilke, editors, *30th International Symposium on Theoretical Aspects of Computer Science, STACS 2013, February 27 - March 2, 2013, Kiel, Germany*, volume 20 of *LIPIcs*, pages 92–103. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2013. doi:10.4230/LIPIcs.STACS.2013.92.

30 Fedor V. Fomin, Daniel Lokshtanov, Saket Saurabh, and Dimitrios M. Thilikos. Bidimensionality and kernels. *SIAM J. Comput.*, 49(6):1397–1422, 2020. doi:10.1137/16M1080264.

31 András Frank and Éva Tardos. An application of simultaneous diophantine approximation in combinatorial optimization. *Comb.*, 7(1):49–65, 1987. doi:10.1007/BF02579200.

32 Prachi Goyal, Neeldhara Misra, Fahad Panolan, and Meirav Zehavi. Deterministic algorithms for matching and packing problems based on representative sets. *SIAM J. Discret. Math.*, 29(4):1815–1836, 2015. doi:10.1137/140981290.

33 Martin Grohe, Stephan Kreutzer, and Sebastian Siebertz. Deciding first-order properties of nowhere dense graphs. *J. ACM*, 64(3):17:1–17:32, 2017. doi:10.1145/3051095.

34 Shai Gutner. Polynomial kernels and faster algorithms for the dominating set problem on graphs with an excluded minor. In Jianer Chen and Fedor V. Fomin, editors, *Parameterized and Exact Computation, 4th International Workshop, IWPEC 2009, Copenhagen, Denmark, September 10-11, 2009, Revised Selected Papers*, volume 5917 of *Lecture Notes in Computer Science*, pages 246–257. Springer, 2009. doi:10.1007/978-3-642-11269-0_20.

35 Penny E. Haxell, Alexandr V. Kostochka, and Stéphan Thomassé. Packing and covering triangles in K 4-free planar graphs. *Graphs Comb.*, 28(5):653–662, 2012. doi:10.1007/s00373-011-1071-9.

36 Bart M. P. Jansen. Kernelization for maximum leaf spanning tree with positive vertex weights. *J. Graph Algorithms Appl.*, 16(4):811–846, 2012. doi:10.7155/jgaa.00279.

37 Bart M. P. Jansen. Turing kernelization for finding long paths and cycles in restricted graph classes. In Andreas S. Schulz and Dorothea Wagner, editors, *Algorithms - ESA 2014 - 22th Annual European Symposium, Wroclaw, Poland, September 8-10, 2014. Proceedings*, volume 8737 of *Lecture Notes in Computer Science*, pages 579–591. Springer, 2014. doi:10.1007/978-3-662-44777-2_48.

38 Bart M. P. Jansen, Shivesh Kumar Roy, and Michal Wlodarczyk. On the hardness of compressing weights. In Filippo Bonchi and Simon J. Puglisi, editors, *46th International Symposium on Mathematical Foundations of Computer Science, MFCS 2021, August 23-27, 2021, Tallinn, Estonia*, volume 202 of *LIPIcs*, pages 64:1–64:21. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPIcs.MFCS.2021.64.

39 Naonori Kakimura, Ken-ichi Kawarabayashi, and Dániel Marx. Packing cycles through prescribed vertices. *J. Comb. Theory, Ser. B*, 101(5):378–381, 2011. doi:10.1016/j.jctb.2011.03.004.

40 Eun Jung Kim, Stefan Kratsch, Marcin Pilipczuk, and Magnus Wahlström. Flow-augmentation III: complexity dichotomy for boolean csps parameterized by the number of unsatisfied constraints. In Nikhil Bansal and Viswanath Nagarajan, editors, *Proceedings of the 2023 ACM-SIAM Symposium on Discrete Algorithms, SODA 2023, Florence, Italy, January 22-25, 2023*, pages 3218–3228. SIAM, 2023. doi:10.1137/1.9781611977554.ch122.

41 Eun Jung Kim, Marcin Pilipczuk, Roohani Sharma, and Magnus Wahlström. On weighted graph separation problems and flow-augmentation. *CoRR*, abs/2208.14841, 2022. doi:10.48550/arXiv.2208.14841.

42 David G. Kirkpatrick and Pavol Hell. On the complexity of general graph factor problems. *SIAM J. Comput.*, 12(3):601–609, 1983. doi:10.1137/0212040.

43 Stefan Kratsch. On polynomial kernels for integer linear programs: Covering, packing and feasibility. In Hans L. Bodlaender and Giuseppe F. Italiano, editors, *Algorithms - ESA 2013 - 21st Annual European Symposium, Sophia Antipolis, France, September 2-4, 2013. Proceedings*, volume 8125 of *Lecture Notes in Computer Science*, pages 647–658. Springer, 2013. doi:10.1007/978-3-642-40450-4_55.

44 Stefan Kratsch and Vuong Anh Quyen. On kernels for covering and packing ilps with small coefficients. In Marek Cygan and Pinar Heggernes, editors, *Parameterized and Exact Computation - 9th International Symposium, IPEC 2014, Wroclaw, Poland, September 10-12, 2014. Revised Selected Papers*, volume 8894 of *Lecture Notes in Computer Science*, pages 307–318. Springer, 2014. doi:10.1007/978-3-319-13524-3_26.

45 Stefan Kratsch and Magnus Wahlström. Compression via matroids: A randomized polynomial kernel for odd cycle transversal. *ACM Trans. Algorithms*, 10(4):20:1–20:15, 2014. doi:10.1145/2635810.

46 Stefan Kratsch and Magnus Wahlström. Representative sets and irrelevant vertices: New tools for kernelization. *J. ACM*, 67(3):16:1–16:50, 2020. doi:10.1145/3390887.

47 Daniel Lokshtanov, Amer E. Mouawad, Saket Saurabh, and Meirav Zehavi. Packing cycles faster than erdos-posa. *SIAM J. Discret. Math.*, 33(3):1194–1215, 2019. doi:10.1137/17M1150037.

48 Dániel Marx. Chordless cycle packing is fixed-parameter tractable. In Fabrizio Grandoni, Grzegorz Herman, and Peter Sanders, editors, *28th Annual European Symposium on Algorithms, ESA 2020, September 7-9, 2020, Pisa, Italy (Virtual Conference)*, volume 173 of *LIPIcs*, pages 71:1–71:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.ESA.2020.71.

49 Jaroslav Nesetril and Patrice Ossona de Mendez. Grad and classes with bounded expansion i. decompositions. *Eur. J. Comb.*, 29(3):760–776, 2008. doi:10.1016/j.ejc.2006.07.013.

50 Jaroslav Nesetril and Patrice Ossona de Mendez. On nowhere dense graphs. *Eur. J. Comb.*, 32(4):600–617, 2011. doi:10.1016/j.ejc.2011.01.006.

51 Jaroslav Nesetril and Patrice Ossona de Mendez. *Sparsity - Graphs, Structures, and Algorithms*, volume 28 of *Algorithms and combinatorics*. Springer, 2012. doi:10.1007/978-3-642-27875-4.

52 Christophe Paul, Anthony Perez, and Stéphan Thomassé. Conflict packing yields linear vertex-kernels for k -fast, k -dense RTI and a related problem. In Filip Murlak and Piotr Sankowski, editors, *Mathematical Foundations of Computer Science 2011 - 36th International Symposium, MFCS 2011, Warsaw, Poland, August 22-26, 2011. Proceedings*, volume 6907 of *Lecture Notes in Computer Science*, pages 497–507. Springer, 2011. doi:10.1007/978-3-642-22993-0_45.

53 Marcin Pilipczuk, Michal Pilipczuk, Piotr Sankowski, and Erik Jan van Leeuwen. Network sparsification for steiner problems on planar and bounded-genus graphs. *ACM Trans. Algorithms*, 14(4):53:1–53:73, 2018. doi:10.1145/3239560.

54 Michal Pilipczuk and Sebastian Siebertz. Kernelization and approximation of distance-r independent sets on nowhere dense graphs. *Eur. J. Comb.*, 94:103309, 2021. doi:10.1016/j.ejc.2021.103309.

55 Stéphan Thomassé. A $4k^2$ kernel for feedback vertex set. *ACM Trans. Algorithms*, 6(2):32:1–32:8, 2010. doi:10.1145/1721837.1721848.

56 Meirav Zehavi. Parameterized approximation algorithms for packing problems. *Theor. Comput. Sci.*, 648:40–55, 2016. doi:10.1016/j.tcs.2016.08.004.