algorithms

MDPI

Article

Expansion Lemma—Variations and Applications to Polynomial-Time Preprocessing

Ashwin Jacob [1], Diptapriyo Majumdar [2] and Venkatesh Raman [3,*]

[1] Department of Computer Science, Ben-Gurion University of the Negev, Beersheva 8410501, Israel
[2] Indraprastha Institute of Information Technology Delhi, New Delhi 110020, India
[3] The Institute of Mathematical Sciences, HBNI, Chennai 600113, India
* Correspondence: vraman@imsc.res.in

Abstract: In parameterized complexity, it is well-known that a parameterized problem is fixed-parameter tractable if and only if it has a kernel—an instance equivalent to the input instance, whose size is just a function of the parameter. The size of the kernel can be exponential or worse, resulting in a quest for fixed-parameter tractable problems with polynomial-sized kernels. The developments in machinery (showing lower bounds for the sizes of the kernels) have led researchers to question what are the asymptotically optimum sizes for the kernels of fixed-parameter tractable problems. In this article, we surveyed a tool called expansion lemma that helps in reducing the size of the kernel. Its early origin was in the form of crown decomposition, i.e., to obtain the linear kernel for the VERTEX COVER problem; the specific lemma was identified as the tool behind the optimal $O(k^2)$ kernel for the undirected feedback vertex set problem. Since then, several variations and extensions of the tool have been discovered. We surveyed them along with their applications in this article.

Keywords: parameterized algorithms; kernelization; expansion lemma; crown decomposition

check for updates

Citation: Jacob, A.; Majumdar, D.; Raman, V. Expansion Lemma—Variations and Applications to Polynomial-Time Preprocessing. *Algorithms* **2023**, *16*, 144. https://doi.org/10.3390/a16030144

Academic Editor: Jesper Jansson

Received: 11 October 2022
Revised: 24 February 2023
Accepted: 25 February 2023
Published: 6 March 2023

1. Introduction

Preprocessing is an age-old technique used for implementing algorithms, especially for difficult problems. In the field of parameterized complexity, preprocessing has gained formal recognition. A parameterized problem involves a parameter in addition to the input instance. The objective of parameterized complexity is to determine whether it is possible to design an algorithm that limits combinatorial explosion to the parameter while keeping the rest of the running time polynomial in the input size. Such an algorithm is known as a fixed-parameter algorithm, and a problem that can be solved by such an algorithm is called fixed-parameter tractable.

It is well-known (see, for example, [1]) that a decidable parameterized problem is fixed-parameter tractable if and only if the input instance can be reduced in the polynomial time to an equivalent instance, whose size is a function of the parameter. Such an equivalent instance is said to be a kernel. The size of the kernel can be exponential or worse, resulting in a quest for fixed-parameter tractable problems with polynomial-sized kernels. Machinery to show the non-existence of polynomial-sized kernels [2,3] and lower bounds for kernel sizes [4] under complex theoretic assumptions make the theory more interesting.

In this article, we surveyed a tool called expansion lemma that helps reduce the size of the kernel. Its early origin was in the form of crown decomposition, used for obtaining the linear kernel for the VERTEX COVER problem [5]; the specific lemma was identified as the tool behind the optimal $O(k^2)$ kernel for the UNDIRECTED FEEDBACK VERTEX SET problem [6]. Since then, several variations and extensions of the tool have been discovered. In this paper, we surveyed them along with their applications. We provided sketches of as many results as possible wherever the main (and likely only) tool for the application was the expansion lemma variation. For several problems, there were several other ideas involved in obtaining the improved kernel beyond the tool. Moreover, wherever the main tool was the expansion lemma, we explained the algorithm in a self-contained way. For others, we indicated how the tool was applied; we often did not go into the other complicated technical details as they distract from the main purpose of the survey.

Organization of the paper. In Section 2, we introduce basic terminologies and notations. We begin with the technical part of the paper in the following three sections. In Section 3, we introduce the crown decomposition technique, basic lemmas, and their application to obtain some improved kernels. In Section 4, we introduce the expansion lemma and its use for several problems in kernelization. Finally, in Section 5, we discuss the recent developments, including generalizations of the (basic) expansion lemma, along with their applications. We conclude with some possible future research directions in Section 6.

2. Preliminaries

2.1. Parameterized Complexity and Kernelization

A parameterized problem L is a subset of $\Sigma^* \times \mathbb{N}$ where Σ is a finite alphabet. Assuming that $k \in \mathbb{N}$ is given in unary, an instance to a parameterized problem is (x, k) where $x \in \Sigma^*$ and $k \in \mathbb{N}$.

Definition 1 (fixed-parameter tractability)**.** *A parameterized problem $L \subseteq \Sigma^* \times \mathbb{N}$ is said to be* fixed-parameter tractable *(or* FPT*) if there exists an algorithm for solving this problem L that input (x, k) runs in $f(k)|x|^{\mathcal{O}(1)}$-time, where $f\mathbb{N} \to \mathbb{N}$ is a computable function.*

Definition 2 (kernelization)**.** *Let $L \subseteq \Sigma^* \times \mathbb{N}$ be a parameterized problem. A* kernelization *for L is a polynomial time ($(|x| + k)^{\mathcal{O}(1)}$-time) procedure that replaces an input instance (x, k) by an input instance (x', k'), such that*

- *$(x, k) \in L$ if and only if $(x', k') \in L$.*
- *$|x'| + k' \leq g(k)$ for some computable function $g : \mathbb{N} \to \mathbb{N}$.*

It is well-known that a decidable parameterized problem is FPT if and only if it admits kernelization [1]. Kernelization can be viewed as a theoretical foundation of *preprocessing heuristics.* We are interested in investigating when a parameterized problem admits kernelization of a small size, particularly a kernel of a polynomial size with as small an upper bound as possible. For other definitions, notations, and details on parameterized complexity and kernelization, please refer to [1,7–11].

2.2. Sets, Numbers, and Graph Theory

We use $\mathbb{N}$ to denote the set of natural numbers. For $r \in \mathbb{N}$, we denote the set $\{1, \ldots, r\}$ by $[r]$. A *Boolean formula* is an expression of binary variables with a logical operator conjunction ($\wedge$), disjunction ($\vee$), and negation ($\neg$). A Boolean formula is in *conjunctive normal form* (or *CNF*) if it is a conjunction of clauses. A *literal* is a Boolean variable in its own form or in its complemented form. Each clause is a disjunction of literal. We usually use $G = (V, E)$ to denote an undirected graph, and $D = (V, A)$ to denote a directed graph. Let $V(G)$ denote the set of vertices of G, and $E(G)$ the set of edges of G. When G is clear from context, let $n = |V(G)|$ and $m = |E(G)|$. Informally, we use n to denote the number of vertices and m to denote the number of edges in the input graph. Given a graph $G = (V, E)$ and $X \subseteq V(G)$, we use $G[X]$ to denote an *induced subgraph* of G; its vertex set is X, and its edge set consists of all edges of $E(G)$ that have both endpoints in X. Let $G - X$ denote the subgraph of G induced on the vertex set $V(G) \setminus X$. A *clique* in an undirected graph is a set of vertices where every pair of vertices are adjacent to each other. An *independent set* is a pairwise non-adjacent set of vertices. A *tournament* is a directed graph T, such that for every pair of vertices $u, v \in V(T)$, one of uv or vu is a directed edge of T. In an undirected graph $G = (V, E)$, a *path* is a set P of distinct vertices (except possibly for the start vertex and the end vertex), such that the vertices of P can be arranged in an order $v_1, \ldots, v_d$, such that for every $1 \leq i \leq d - 1$, $v_i v_{i+1} \in E(G)$. Informally, for all $1 \leq i \leq d - 1$, $v_i \neq v_{i-1}$ for a path.

Similarly, in a directed graph $D = (V, A)$, a *directed path* can be defined in a similar way when for every $i \leq d-1$, $(v_i, v_{i+1}) \in A$. A (directed) *cycle* is a (directed) path that starts and ends at the same vertex. A *connected component* of an undirected graph is a connected subgraph that is not part of any larger connected subgraph. For subsets $X, A, B \subseteq V(G)$, we say X *separates* A and B when no component of $G - X$ contains vertices from both $A \setminus X$ and $B \setminus X$. Given a vertex subset $X \subseteq V(G)$, we use $N_G(X)$ to denote the *open neighborhood* of X and $N_G[X]$ to denote the *closed neighborhood* of X. When graph G is clear from the context, we omit the subscript.

A vertex v of G is a *cut vertex* if $G - \{v\}$ has more connected components than G. Graph G is *bi-connected* if $|V(G) \geq 3$, and it has no cut vertex. Graph G is a *block graph* if each of its maximal bi-connected components is a clique. A *tree* is a connected undirected graph with no cycle (or equivalently, a connected undirected acyclic graph). A *forest* is a graph, every connected component of which is a tree. A *directed acyclic graph* is a directed graph with no directed cycles. A *topological ordering* for a directed acyclic graph is a linear ordering of vertices, such that for every directed edge uv, vertex u comes before v in the ordering. An undirected graph $G = (V, E)$ is said to be *bipartite* if $V(G) = A \uplus B$, such that for every $uv \in E(G)$, $u \in A$ if and only if $v \in B$. We call A and B the bipartitions of G. Another alternate characterization is that a bipartite graph has no odd cycle. We use $G = (A \uplus B, E)$ to denote a bipartite graph such that A and B are its bipartitions and E is the set of its edges. A set $M \subseteq E(G)$ is said to be a *matching* of a graph if for every two edges $uv, xy \in M$, $\{u, v\} \cap \{x, y\} = \emptyset$. A vertex $u \in V(G)$ is said to be *matched* by a matching M, if there is an edge of M incident on u. A matching M of G is said to *saturate* A if $A \subseteq V(M)$. A matching is said to be *perfect matching* if all vertices are matched by M. A cycle C in a graph is said to be an *induced cycle* if $G[C]$ has no edge other than the edges of the cycle. A graph is said to be a *chordal graph* if it has no induced cycle of a length of at least four. An *out-tree* is a directed graph where each vertex has an in-degree of (at most) 1, and the underlying (undirected) graph is a tree. An *out-forest* is a disjoint union of out-trees. A graph is said to be *k-colorable* if there is a function $\lambda : V(G) \rightarrow [k]$ for some $k \in \mathbb{N}$ such that for every $uv \in E(G)$, $\lambda(u) \neq \lambda(v)$. Such a function λ is called a *proper coloring* of G. It is well-known that a bipartite graph is 2-colorable. Graph G is said to be *factor-critical* if for every $u \in V(G)$, the graph $G - u$ has a perfect matching. For other terminologies of the graph theory, we use standard graph-theoretic notations by Diestel [12].

We use the following well-known proposition several times in the paper.

Proposition 1 (Hall's Theorem [13])**.** *Let $G = (V = A \uplus B, E)$ be a bipartite graph. Then, G has a matching saturating A if and only if for every $X \subseteq A$, $|N(X)| \geq |X|$.*

Given a bipartite graph $G = (A \uplus B, E)$, a set $X \subseteq A$ is said to be a *hall set* if $|N(X)| < |X|$. A hall set is said to be *minimal* if none of its proper subsets is a hall set. Given a bipartite graph $G = (A \uplus B, E)$, a minimal hall set can be computed in polynomial time using Proposition 1.

Before we state the following proposition, we need a few notations. For a graph $G = (V, E)$, suppose that $S \subseteq V(G)$ and a set $\mathcal{C}$ of vertex-disjoint subgraphs of $G - S$, let $\mathcal{N}_G(S, \mathcal{C}) = \{H \in \mathcal{C} \mid E(S, V(H)) \neq \emptyset\}$.

Proposition 2 (Gallai–Edmonds structure theorem [14,15])**.** *For every graph $G = (V, E)$, there are disjoint subsets $X, Y, Z \subseteq V(G)$, such that*

(i) $X \cup Y \cup Z = V(G)$;
(ii) $Y = N_G(X)$;
(iii) *Every connected component of $G[X]$ is factor-critical;*
(iv) *$G[Z]$ has perfect matching; and*
(v) *For every nonempty subset $Y' \subseteq Y$, it holds that $|\mathcal{N}_G(Y', \mathcal{C})| \geq |Y'|$.*

Furthermore, such X, Y, Z can be computed in polynomial time.

2.3. Graph Parameters

We introduce different graph parameters that we will use in this paper. A subset $S \subseteq V(G)$ is a *vertex cover* of G if for every edge $uv \in E(G)$, $\{u, v\} \cap S \neq \emptyset$. It follows that for an undirected graph G, if S is a vertex cover, then $V(G) \setminus S$ is an independent set (and vice versa). A subset $S \subseteq V(G)$ is called a *cluster vertex deletion set* of G if every connected component of $G - S$ is a clique. A subset $S \subseteq V(G)$ is a *feedback vertex set* of G if $G - S$ has no cycle.

The following graph parameter requires a more non-trivial definition.

Definition 3 (Pathwidth [16–18])**.** *A* path decomposition *of graph G is a pair (T, χ) in which $T = (V_T, E_T)$ is a path and $\chi = \{\chi_i \mid i \in V_T\}$ is a family of subsets of $V(G)$ called* bags*, such that*

(i) $\cup_{i \in V_T} \chi_i = V(G)$.

(ii) *for each edge $uv \in E(G)$, there is $i \in V_T$ such that $u, v \in \chi_i$.*

(iii) *for each vertex $u \in V(G)$, the set of nodes $\{i \in V_T \mid u \in \chi_i\}$ induces a connected subgraph of T.*

The value $\max_{i \in V_T}\{|\chi_i| - 1\}$ is called the *width* of the path decomposition (T, χ). The *pathwidth* of a graph is the minimum width taken over all possible tree decompositions of G. It is not very hard to understand that the pathwidth of a path (or a linear forest) is 1. Similarly, the pathwidth of a cycle is 2, and the pathwidth of a caterpillar is also 1. Moreover, the pathwidth of a clique with n vertices is $n - 1$. On an informal note, the pathwidth of a graph is a measure of how close a graph is to a path.

We begin the technical part of the survey with a structure and a lemma that could be considered a precursor to the expansion lemma.

3. Crown Decomposition and Applications

In this section, we define a related technique called "crown decomposition" and its application to some problems. In Sections 3.1–3.5, we discuss how crown decomposition techniques are used to obtain kernels for VERTEX COVER, $(n - k)$-COLORING, MAXIMUM SATISFIABILITY, $(n - k)$-LIST COLORING, and LONGEST CYCLE problems, respectively. After that, in Section 3.6, we discuss a generalization of the crown decomposition technique recently introduced by Chen et al. [19].

In the context of the VERTEX COVER problem, a well-known reduction rule to discard the pendant (degree one) vertices is to pick their neighbors (and delete them and their neighbors) in the solution [1].

A crown introduced by Fellows [20] generalizes the structure's underlying pendant vertices.

Definition 4 (Crown decomposition)**.** *Given a graph $G = (V, E)$, a* crown decomposition *of G is a partition of $V(G) = C \uplus H \uplus R$ such that*

- *$C \neq \emptyset$;*
- *C is an independent set;*
- *There is no edge between a vertex of C and a vertex of R, i.e., H separates C from R; and*
- *There exists a matching of size $|H|$ among the edges $E \cap (H \times C)$.*

Set C can be seen as a crown placed on the head H of the remaining part (body) R. If G has a pendant vertex x with its (unique) neighbor y, then $C = \{x\}$, $H = \{y\}$ and $R = V \setminus \{x, y\}$ constitute the crown decomposition.

Lemma 1 (Crown Lemma-1, [5,20])**.** *Let G be a graph without isolated vertices with at least $3k + 1$ vertices. Then there is a polynomial-time algorithm that either*

- *Finds a matching of size $k + 1$ in G; or*
- *Finds a crown decomposition of G.*

Fellows [20] applied Lemma 1 for kernelization in the following way. First, they observe that if there is a matching of size $k+1$, then the instance is a trivial yes or no instance for some problems. Moreover, in the other case, the head of the crown can be argued to be in some optimum solution (as it hits a large number of obstructions) and, hence, can be deleted, resulting in a reduction rule.

The following variation of Lemma 1 can be useful sometimes. This lemma also serves as a precursor to the more general expansion lemma described in the next section.

Lemma 2 (Crown Lemma-2, [5,11,21])**.** *Let $G = (V = A \uplus B, E)$ be a bipartite graph with no isolated vertices in B, and $|B| \geq |A|$. Then there is a polynomial time algorithm that either*

- *Finds a matching saturating A; or*
- *Finds a crown decomposition (C, H, R) of G, with $C \subseteq A$ and $H \subseteq B$.*

If there is no matching saturating A, then by Hall's theorem (Proposition 1), there exists a minimal hall set $C \subseteq A$, such that $|N(C)| < |C|$. A minimal hall set gives rise to a crown decomposition; see [11] for proof.

We exemplify these lemmas using some examples.

3.1. Vertex Cover

We now describe a summary of a kernel of $3k$ vertices for VERTEX COVER (defined in Box 1) provided by Chor et al. [5] where the earliest application of Lemma 1 can be seen. The isolated vertices of the input of graph G can be safely removed. Then, they invoke the algorithm in Lemma 1, which either gives a matching of size $k+1$ in G or a crown decomposition of G if G has more than $3k$ vertices. In the former case, it can be concluded that the input instance is a non-instance as any vertex cover of G needs at least $k+1$ vertices to cover the edges of the matching.

Box 1. Vertex Cover Problem.

VERTEX COVER (VC)
Input: *An undirected graph $G = (V, E)$ and an integer k.*
Parameter: *k.*
Question: *Is there a set $S \subseteq V(G)$ of, at most, k vertices for every $uv \in E(G)$, $u \in S$, or $v \in S$?*

Hence, they assume that the algorithm returns a crown decomposition of G with crown C and head H. Let M be the matching of H into C that saturates H. Any vertex cover that covers the edges in $G[H \cup C]$ has to contain at least $|M| = |H|$ many vertices. Since M saturates H and C is an independent set, the set H covers all of the edges in $G[H \cup C]$. Hence, we can safely reduce the problem instance to $(G \setminus (H \cup C), k - |H|)$. Hence, we conclude that we can reduce the VERTEX COVER instance as long as $|V(G)| > 3k$. Hence, VERTEX COVER has a kernel of $3k$ vertices. This bound has been improved to a kernel with $2k - \log k$ vertices by Lampis et al. [22] using linear programming techniques. See Li and Zhu [23] for another $2k$-vertex kernel for VERTEX COVER.

3.2. $(n-k)$-Coloring

Given an undirected graph, determining whether its vertices can be properly colored with k colors is NP-hard even for $k = 3$. So a kernel or an FPT algorithm is not possible unless P = NP. However, a related problem is whether graph G on n vertices can be colored properly with $n - k$ colors, which is the same as whether k colors can be saved. The formal definition is as follows.

This problem defined in Box 2 admits an FPT algorithm and a $3k$ vertex kernel using Lemma 1 by Chor et al. [5] (also see [11] for more details). The idea is to apply Lemma 2 to the complement graph. If it contains a matching of size k, then those k pairs of non-edges in G can help us save k colors. If not, crown C is a clique in G needs $|C|$ colors, and all vertices in C are adjacent to every vertex of $V \setminus (H \cup C)$. Furthermore, the matching saturating H helps to color vertices in H using colors used in C. Thus, we can remove $H \cup C$ and recurse on the remaining vertices. This process stops if we discover that we have already used more than $(n-k)$ colors, so the input is a non-instance or that there are only $3k$ vertices. Recently, Li et al. [24] have improved this result by giving a $(2+\varepsilon)k$ vertex kernel for this problem. They provided a new structure called 'fractal critical crown' of a graph, we use Proposition 2 and the notion of a factor-critical crown (see Section 3.6 for definition) to obtain their results.

Box 2. $(n-k)$-Coloring Problem.

$(n-k)$-COLORING
Input: *An undirected graph* $G = (V, E)$ *and an integer* k*.*
Parameter: k.
Question: *Is there a coloring* $\Xi : V \to \{1, 2, \dots |V| - k\}$ *such that* $\Xi(x) \neq \Xi(y)$ *whenever* $xy \in E$*?*

3.3. Maximum Satisfiability

We formally define the MAXIMUM SATISFIABILITY problem below.

It is folklore that such a CNF formula has a truth assignment satisfying at least $m/2$ clauses, which implies that we can assume that $m \leq 2k$, as otherwise, the instance is a YES instance. It is also easy to see (by a greedy algorithm that appropriately sets an unassigned variable in each clause in the sequence) that the formula ϕ is satisfiable, i.e., a yes-instance for MAXIMUM SATISFIABILITY (defined in Box 3) for if $n \geq m$.

Box 3. Maximum Satisfiability Problem.

MAXIMUM SATISFIABILITY
Input: *A Conjunctive Normal Form (CNF) formula* ϕ *with* n *variables and* m *clauses, and a nonnegative integer* k*.*
Parameter: k.
Question: *Does* ϕ *have a truth assignment satisfying at least k clauses?*

Thus, it can be assumed that $n < m < 2k$. Lokshtanov [25] showed (as below) using Lemma 2, where n can be made to be less than k. The idea is to construct a bipartite graph with bipartition A and B where A corresponds to the variables in the formula, and B corresponds to the clauses in the formula. There is an edge between $x \in A$ and a $c \in B$ if x appears in clause c.

If $n \geq k$, then from Lemma 2, if we have a matching saturating A, then clauses in the matching can be satisfied by setting the variables in the matching appropriately. Thus, we will be able to satisfy at least k clauses. Otherwise, it can be argued that the crown (C, H, R) can be used to satisfy the clauses represented by H by appropriately setting the variables represented by C, and removing those clauses and variables. See [11] for more details.

3.4. $(n-k)$-List Coloring

Banik et al. [26] use Lemma 2 to obtain an FPT algorithm for $(n-k)$-LIST COLORING; a generalization of the problem is presented at Box 2 in Section 3.2, where there is a reduction rule that binds the number of colors over all the lists to n. The problem is formally defined below.

Here, they create an auxiliary bipartite graph with vertices $V(G)$ on one side, $\mathcal{C}$, and the colors in every vertex list on the other side, adding edges between a vertex and a color if its list contains it. After that, they invoke Lemma 2. If there is a matching saturating $\mathcal{C}$, the number of colors is bounded by $|V(G)| = n$. Otherwise, we have a crown decomposition (C, H, R) with $C \subseteq \mathcal{C}$ and $H \subseteq V(G)$. Since there is a matching M from H to C saturating

H and $N(C) = H$, we can safely color the vertices of H using the colors C according to the matching M and reduce the instance. Thus, the number of colors is reduced to n. Further ideas are used to obtain an FPT algorithm for this problem defined in Box 4.

Box 4. $(n-k)$-List Coloring Problem.

$(n-k)$-List Coloring
Input: *A graph* $G = (V, E)$*, an integer* k *and for every* $u \in V(G)$*, a set* $L(u)$ *of exactly* $(n-k)$ *colors.*
Parameter: k.
Question: *Is there a proper coloring of* G *such that for all* $u \in V(G)$*, the color assigned to* u *is in* $L(u)$*?*

3.5. Longest Cycle Parameterized by Vertex Cover Size

Bodlaender et al. [27] showed that the Longest Cycle (vc) as defined in Box 5 admits a kernel with $\mathcal{O}(k^2)$ vertices. Here, they construct an auxiliary bipartite graph with pairs of vertices on one side and $I = V(G) \setminus S$ on the other side with edges added if both vertices in a pair are adjacent to a vertex in I. We now apply Lemma 2. If there is a matching saturating I, the overall number of vertices is $\mathcal{O}(k^2)$. Otherwise, we have a crown decomposition (C, H, R) with $C \subseteq I$ and $H \subseteq S$. Let M match from H to C, saturating H. We claim that for any vertex $v \in C$ not saturated by M, we can reduce the instance to $(G - v, k, \ell, S)$. Suppose a solution cycle C' in G contains v. Since I is an independent set, the two neighbors of v in C are $s_1, s_2 \in S$. There is another vertex $v' \in I$ adjacent to both s_1 and s_2 as there is an edge in the matching M corresponding to v' and the pair $(s_1, s_2) \in H$. This gives us a cycle C'' by replacing v with v' in $G - v$.

Box 5. Longest Cycle Parameterized by Vertex Cover.

Longest Cycle (vc)
Input: *graph* G*, integers* k *and* ℓ *and a vertex cover* S *of* G *such that* $|S| = k$*.*
Parameter: k.
Question: *Does* G *contain a cycle on* ℓ *vertices?*

3.6. Generic Crown Decomposition and Its Application to Packing and Covering Problems

Chen et al. [19] introduced a general notion of crowns and used it to obtain an $\mathcal{O}(k)$ vertex kernel for T_r-Packing problem (for each fixed integer r) where T_r is a tree with r edges.

It is folklore that many covering problems in graphs can be formulated as instances of d-Hitting Set defined as follows.

Similarly, many packing problems in graphs can be formulated as instances of d-Set Packing, defined as follows.

It is folklore that the d-Hitting Set (defined in Box 6) and d-Set Packing (defined in Box 7) problems can be formulated as a dominating set problem and their duals in bipartite graphs that are defined as follows.

Box 6. d-Hitting Set Problem.

d-HITTING SET
Input: *A family $\mathcal{F}$ of sets over a universe U, where each set in $\mathcal{F}$ has a size of, at most, d, and an integer k.*
Parameter: k.
Question: *Does there exist a subset $X \subseteq U, |X| \leq k$, such that X contains at least one element from each set in $\mathcal{F}$?*

Box 7. d-Set Packing Problem.

d-SET PACKING
Input: *A family $\mathcal{F}$ of sets over a universe U, where each set in $\mathcal{F}$ has a size of, at most, d, and an integer k.*
Parameter: k.
Question: *Does there exist a subset $\mathcal{F}' \subseteq \mathcal{F}, |\mathcal{F}'| = k$ such that every element in U is contained in at most one set in $\mathcal{F}'$?*

We formally state the definition of a *generic crown decomposition* below.

Definition 5 ([19]). *Let $G = (V = A \uplus B, E)$ be a bipartite graph such that for all $x \in B$, $deg_G(x) = d$. Then, a* generic crown decomposition *(C, H, X) of G is given by a* head *$H \subseteq A$ and a* crown *$C \subseteq (A \uplus B) \setminus (H \cup C)$ satisfying the following properties*

- *H is a separator in G, one collection of connected components being C.*
- *$N(H) \supseteq C \uplus B$.*
- *There is an injective mapping $M : H \to C \cap B$, such that*
  - *For all $x \in H$, $xM(x) \in E(G)$, and*
  - *Each $v \in A$ is a neighbor of, at most, one $x \in M(H)$.*
- *$X = (A \uplus B) \setminus (H \cup C)$ are the remaining vertices.*

The authors proved that (G, k) is a YES instance for d-RED/BLUE DOMINATING SET (or d-RED/BLUE DISTANCE-3 PACKING, resp.) if and only if $(G - (H \cup C), k)$ is YES instance where (C, H, X) is a *generic crown decomposition* of G.

The authors devised a framework for obtaining crown rules for covering and packing problems in graphs. They defined a notion of crown decomposition specific to the problem. They then reduced the problem to d-RED/BLUE DOMINATING SET (defined in Box 8) or d-RED/BLUE DISTANCE-3 PACKING (defined in Box 9) where the problem-specific crown decomposition corresponded to a generic crown decomposition. From the safety of the reduction corresponding to the generic crown decomposition, the safety of the problem-specific crown decomposition follows.

Box 8. Red-Blue Dominating Set Problem.

d-RED/BLUE DOMINATING SET
Input: *A bipartite graph $G = (R \uplus B, E)$, where $\forall x \in B : deg(x) = d$, and an integer k.*
Parameter: k.
Question: *Does there exist a subset $X \subseteq R, |X| \leq k$ such that X such that $N(X) = B$?*

Box 9. Distance-3 Red-Blue Set Packing Problem.

d-Red/Blue Distance-3 Packing
Input: *A bipartite graph* $G = (R \uplus B, E)$, *where* $\forall x \in B : deg(x) = d$, *and an integer* k.
Parameter: k.
Question: *Does there exist a subset* $X \subseteq B, |X| \geq k$ *such that each* $x \in R$ *is a neighbor of, at most, one* $x \in X$?

The authors demonstrate this framework in the context of the following covering and packing problems.

The dual version of the above problem is defined as follows.

They [19] define the *r-tree crown decomposition* as follows.

Definition 6 ([19])**.** *An* r-tree crown decomposition *of a graph* $G = (V, E)$ *is a partitioning of* $V(G) = C \uplus H \uplus X$ *satisfying the following properties.*

- H *(the head) separates* C *from* X *in* G.
- $G[C]$ *(*C *is the crown) induces no subgraph that is a tree with* r *edges.*
- *There are* r *injective mappings* $\pi_1, \dots, \pi_r$ *from* H *to* C *called* witness functions *such that for all* $i \neq j$, $\pi_i(H) \cap \pi_j(H) = \emptyset$ *and for each* $v \in H$, *the vertex set* $\{p\} \cup \{\pi_i(v) \mid 1 \leq i \leq r\}$ *forms a tree with* r *edges in* G.

Given an instance (G, k) of r-Edge-Tree-Packing (see Box 10), the authors construct a bipartite graph $G' = (R \uplus B, E')$ as follows. Let $R = V$ and B be the set of all trees with r edges present in G. Add an edge $uv \in E', u \in R, v \in B$ if and only if the vertex u is part of the tree corresponding to v.

Box 10. r-Edge-Tree Packing Problem.

r-Edge-Tree-Packing
Input: *An undirected graph* G *and an integer* k.
Parameter: k
Question: *Is there a set of at least* k *pairwise disjointed collections of trees with* r *edges in* G?

Let us focus on an r-tree crown decomposition (C, H, X) in G. The witness functions $\pi_1, \dots, \pi_r$ from H to C correspond to a mapping M from H to some of the trees in B; in particular $v \in H$ is mapped to the tree $\{v, \pi_1(v), \dots, \pi_r(v)\}$. From the definition of $\pi_i, i \in [r]$, one can conclude that M is a matching and $M(H)$ is a packing. Thus, (C', H, X') is a generic crown decomposition in G' where $C' = C \uplus M(H)$ and $X' = V(G') \setminus (H \cup C')$. Since (G', k) is a YES instance of r-red/blue distance-3 packing (see Box 9) if and only if $(G' - (H \cup C'), k - |H|)$ is a YES instance of r-red/blue distance-3 packing, one can conclude that (G, k) is a YES instance of r-edge-tree-packing if and only if $(G - (H \cup C), k - |H|)$ is a YES instance of r-edge-tree-packing.

Using the above reduction, the authors provided a kernel with at most $(r^2 + 1)(r + 1)k$ vertices for r-edge-tree-packing. The authors also devised a methodology to transfer a kernelization result from a packing problem to the corresponding covering problem with a similar size (see [19] for more details). Thus, a kernel with at most $(r^2 + 1)(r + 1)k$ vertices for r-edge-tree-covering (see Box 11) also follows.

Box 11. r-Edge-Tree-Covering Problem.

r-EDGE-TREE-COVERING
Input: *An undirected graph G and an integer k.*
Parameter: k
Question: *Is there a set S of, at most, k vertices, such that $G - S$ has no tree with r edges as a subgraph?*

3.7. Other Examples

More examples of the application of crown decompositions and some variants include

- A kernel with $\mathcal{O}(k^3)$ vertices for TRIANGLE PACKING (packing vertex disjoint triangles as subgraphs) [28];
- A kernel with $\mathcal{O}(k^2)$ vertices for k-INTERNAL SPANNING TREE (finding a spanning tree with at least k internal vertices) [29];
- A linear-sized kernel for several variants of VERTEX COVER [30];
- A $(2d-1)k^{d-1}+k$ sized kernel for d-HITTING SET [31];
- A $\mathcal{O}(k^{d-1})$ sized kernel for d-SET PACKING [32]; and
- A kernel with $7k$ vertices for P_3-PACKING (packing vertex disjoint P_3s as subgraphs) [33]. Later, Xiao and Khu [34] improved this result into a kernel with $5k$ vertices using the crown decomposition with some critical observations.

4. (Basic) Expansion Lemma and Applications

In this section, we provide the definitions of 'q-expansion' and the expansion lemma that generalize Hall's theorem (Proposition 1). In the following subsections, we give a summary of how expansion lemma has been used for several problems, e.g., p-COMPONENT ORDER CONNECTIVITY, FEEDBACK VERTEX SET, CLUSTER VERTEX DELETION, and several other theoretical graph problems.

Definition 7 (q-Expansion)**.** *Let $G = (V = A \uplus B, E)$ be a bipartite graph with vertex bipartition (A, B). For a positive integer q, a set of edges $M \subseteq E(G)$ is called a* q-expansion of A into B *if*

- *Every vertex of A is incident to exactly q edges of M, and*
- *M Saturates exactly $q|A|$ vertices in B.*

For $q = 1$, a q-expansion is matching. We now have expansion lemma, which was introduced by Thomassé [6].

Lemma 3 (q-Expansion Lemma [6])**.** *Let $q \geq 1$ be a positive integer and $G = (V = A \uplus B, E)$ be a bipartite graph such that*

- *$|B| \geq q|A|$, and*
- *There are no isolated vertices in B.*

Then, there exist nonempty vertex sets $X \subseteq A$ and $Y \subseteq B$, such that

1. *There is a q-expansion of X into Y, and*
2. *$N_G(Y) \subseteq X$.*

Furthermore, sets X and Y can be found in polynomial time in the size of G (see Figure 1 for an illustration of $q = 2$).

When $q = 1$, note that the sets Y, X, and $V(G) \setminus (X \cup Y)$ form a crown decomposition with crown, head, and body, respectively. Hence, the expansion lemma can be seen as a generalization of Lemma 2. In particular, it allows us to work with objects that need to be hit that can be of size more than 2. It gives a way to "encode" disjoint objects of size $q+1$ such that any solution must hit these objects. We use this to argue that there is a solution containing X, noting that every object Y is part of can be transformed into a solution containing X.

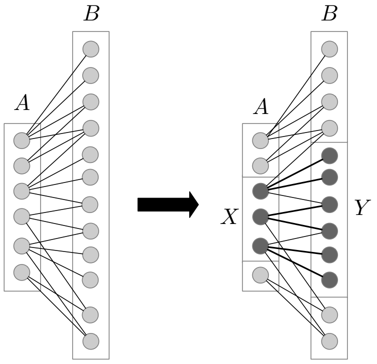

Figure 1. An illustration of Lemma 3 with $q = 2$.

We illustrate this with a few examples below. To encode objects in this way, we usually construct an auxiliary bipartite graph where we apply the expansion lemma.

4.1. *p-Component Order Connectivity*

In p-COMPONENT ORDER CONNECTIVITY, the goal is to find a subset of k vertices in a given graph, such that every component of the remaining graph has a size of, at most, p for a fixed integer of p. More formally, note that when $p = 1$, $G - S$ is an independent set. Thus, p-COMPONENT ORDER CONNECTIVITY is a generalization of VERTEX COVER (see Box 1). We explain how a kernel with $\mathcal{O}(p^3k)$ vertices can be obtained for this problem. Note that improved kernels are available for this problem using stronger versions of the expansion lemma (see Section 5), we give an $\mathcal{O}(k^3p)$ kernel to illustrate a simple application of Lemma 3. The results in the literature are improvements of this result using some new tools (available in Section 5).

We can obtain an approximate solution X of, at most, $(p+1)k$ by repeatedly finding connected subsets of size $p+1$, adding it to X and deleting it from G (if $|X| > (p+1)k$, then it is a no-instance). We can safely remove all components in $G - X$ with a size of, at most, p, as its vertices cannot be part of any optimal solution. We now use an auxiliary bipartite graph Q with X on one side and components of $G - X$ on the other side. We add an edge between a vertex $u \in X$ and a component D in $G - X$ if u has a neighbor in D. If the number of components in $G - X$ is more than $p(p+1)k$, then the conditions for Lemma 3 are satisfied with $q = p$. Thus, we obtain a head $H \subseteq X$ and a crown C, which is a subset of connected components in $G - X$, such that there is a p-expansion of H into C and $N_Q(C) \subseteq H$. Our reduction rule, similar to that of the VERTEX COVER, includes all of H into the solution, removes H and components of C and drops k by $|H|$. We again aim to claim that for an optimal solution S, the set $S' = (S \setminus (H \cup C)) \cup H$ is also an optimal solution. The p-expansion gives a collection of $|H|$ disjointed connected sets of sizes $(p+1)$ that any solution needs to hit. Hence, $|S'| \leq |S|$. We use that $N_Q(C) \subseteq H$ and the graph on C contains only components with a size of, at most, p to claim that S' is indeed a solution. This is because all of the connected sets of sizes more than p where vertices in components of C are part of must contain a vertex in H. These are hit by S'. Thus, the number of connected components in $G - X$ is bounded by $p(p+1)k$. Each of these components has a size of, at most, p. Hence, we obtain a kernel with $\mathcal{O}(p^3k)$ vertices for this problem (defined in Box 12).

Box 12. p-Component Order Connectivity Problem.

p-COMPONENT ORDER CONNECTIVITY
Input: *Graph G and an integer k.*
Parameter: k.
Question: *Does G contain a subset S of vertices, $|S| \leq k$, such that every connected component of $G - S$ has a size of, at most, p?*

4.2. Undirected Feedback Vertex Set

Thomassé [6] gave a kernel with $\mathcal{O}(k^2)$ vertices for the FEEDBACK VERTEX SET. The main idea is to use the expansion lemma to bind the maximum degree of the graph to $\mathcal{O}(k)$.

Suppose a vertex v in the graph has a degree more than $11k$. In polynomial time, one can determine [11,35], where v is part of at least $k+1$ cycles with v as the only intersection (in which case v is forced into the solution) or find a set S_v with a size of, at most, $3k$, which hits all of the cycles passing through v. The next step is to bind the number of components in the graph $G - (\{v\} \cup S_v)$ that has a vertex adjacent to v. Firstly, note that v is adjacent to exactly one vertex in such components; otherwise, there is a cycle containing v not intersecting S_v. Such components can be disregarded in this graph, which can contain cycles. Note that the number of such components is at most k as they form disjointed cycles.

Consider the auxiliary bipartite graph where one partition is S_v and the other partition is the remaining components of $G - (\{v\} \cup S_v)$ and add edges if a vertex in S_v is incident to a vertex in the component. If the number of components is more than $6k$, then Lemma 3 can be used with $q = 2$ to find subsets with each vertex in S_v forming a 2-expansion. Note that since v is adjacent to these components, this provides us with a set of cycles containing v corresponding to each 2-expansion. Using the presence of these cycles, it can be proven that any cycles hit by vertices in these components can be hit by either v or all the vertices in the expansion subset of S_v. The authors encode this by creating disjoint 2-cycles containing v and each vertex in the expansion subset of S_v if not already present.

This bounds the number of such connected components to $2|S_v| \leq 6k$. Thus, the number of vertices adjacent to v is bounded by $6k + |S_v| + k \leq 10k$. At most, k of these vertices can have 2 edges to v as we can delete v if it is a part of the $k+1$ cycle intersecting only at v. Thus, the degree of v is bounded by $11k$.

From other reduction rules, it is possible to prove that the graph has a minimum degree of three. Finally, in this case, it can be proved that FEEDBACK VERTEX SET (defined in Box 13) admits a kernel with $\mathcal{O}(k\Delta)$ vertices where Δ is the maximum degree of G. Thus, we have a kernel with $\mathcal{O}(k^2)$ vertices as $\Delta \leq 11k$.

Recently, Iwata [36] obtained a kernel with $\mathcal{O}(k^2)$ vertices in $\mathcal{O}(m+n)$ time.

Box 13. Undirected Feedback Vertex Set Problem.

FEEDBACK VERTEX SET (FVS)
Input: *An undirected graph $G = (V, E)$ and an integer k.*
Parameter: k.
Question: *Is there a set $S \subseteq V(G)$ of, at most, k vertices, such that $G - S$ is acyclic?*

4.3. Cluster Vertex Deletion

The CLUSTER VERTEX DELETION problem can be formulated as a 3-HITTING SET (see Box 6 with $d = 3$) problem with $U = V(G)$ and $\mathcal{F}$ being the vertex sets of all induced paths on three vertices (P_3s) of G. Abu-Khzam [31] obtained a kernel with $\mathcal{O}(k^2)$ elements for the 3-HITTING SET. This kernel can be adapted to obtain kernels with $\mathcal{O}(k^2)$ vertices for CLUSTER VERTEX DELETION.

Fomin et al. [37] obtained a kernel with $\mathcal{O}(k^{5/3})$ vertices using Lemma 3 and more sophisticated techniques. We will see a brief overview of this kernel here.

Bounding the number of cliques. We give a simple argument to show how Lemma 3 can be used to bound the number of cliques to $6k$.

First, find a maximal collection $\mathcal{P}$ of the vertex disjoint P_3 in G. If $|\mathcal{P}| > k$, we return a no-instance as there is no optimal cluster vertex deletion set with a size of, at most, k. Otherwise, set S as the subset of vertices in $\mathcal{P}$. Note that $|S| \leq 3k$. We use an auxiliary bipartite graph Q with S on one side and components (cliques) of $G - S$ on the other side. Add an edge between a vertex $u \in S$ and a component D in $G - S$ if u has a neighbor in D. Remove isolated cliques, if any. If the number of remaining cliques is more than $6k$, then the conditions are satisfied for Lemma 3 with $q = 2$. This provides a head $H \subseteq S$ and a crown C, which is a subset of components in $G - S$, such that there is a 2-expansion from H to C and $N_Q(C) \subseteq H$. The reduction rule, similar to that of VERTEX COVER, includes all of H into the solution, removes H and the components of C, and drops k by $|H|$. This is because it is easy to show that for any optimal solution X, the set $X' = X \setminus (H \cup C) \cup H$ is also an optimal solution. The 2-expansion gives a collection of $|H|$ disjointed P_3s that any solution needs to hit. Hence, $|X'| \leq |X|$. Then they use that $N_Q(C) \subseteq H$ and that the graph on C is a cluster graph to claim that X' is indeed a solution. This is because all P_3s containing vertices in components of C have to contain a vertex in H, and these are covered by X'. Thus, the total number of cliques in $G - S$ is bounded by $6k$.

A marking procedure and a weak form of crown decomposition. After bounding the number of cliques in $G - S$ by $6k$, the authors [37] performed a marking procedure; this procedure involved associating an edge in $G - S$ with each vertex in S, such that exactly one endpoint of the edge is adjacent to that vertex. If $k + 1$ edges are associated with a vertex in S, it can be concluded that this vertex must be in the solution as it is the intersection of the $k + 1$ P_3 and reduces the instance accordingly. Otherwise, if $k^{2/3}$ vertices of S roughly have $k^{2/3}$ edges associated with them, call the run 'successful' and mark and delete these vertices. This marking procedure is repeated until a successful run is obtained.

Let U be the set of all marked vertices and M be the set of endpoints of the edges marked in the unsuccessful final run. Let $L = S \setminus U$. The sets $U, V(G) \setminus S$, and L can be viewed as a weak form of the crown decomposition with U being the head, $V - S$ being the crown, and L being the body. When the crown decomposition is applied to obtain kernels, usually it can be concluded that there is a cluster vertex deletion set that contains all of the head, none of the crown, and all of the edges of the crown within the head. Similarly, in the weak crown decomposition, it can be concluded that *most* of the vertices of the head U go into a solution, *almost none* of the vertices of the crown $V(G) \setminus S$ is in the solution, and *almost all* of the edges from $V(G) \setminus S$ are within U.

Identifying good/bad cliques and bounding remaining vertices. Nonetheless, there is no straightforward reduction rule for this weak crown decomposition that the authors obtain. Toward this, they classify the cliques in $G - S$ into good cliques and bad cliques (based on the weak form of crown decomposition). Based on its properties, the number of vertices of bad cliques can be bounded to $\mathcal{O}(k^{5/3})$. For a vertex $s \in S$ and a good clique C, the authors [37] identify a small subset of the vertices of C based on the number of neighbors of C in s. Then they show that good cliques exhibit vertex-cover-like behaviors; any solution either contains s or the small side of a good clique. This property can be exploited by using Lemma 3 to bound the number of vertices of the good cliques to $\mathcal{O}(k^{5/3})$ as well. This completes a sketch of subquadratic kernel for CLUSTER VERTEX DELETION as defined in Box 14.

Box 14. Cluster Vertex Deletion Problem.

Cluster Vertex Deletion (cvd)
Input: *Graph G and an integer k.*
Parameter: *k.*
Question: *Does G contain a subset S of, at most, k vertices, such that every connected component of $G \setminus S$ is a clique?*

4.4. Other Deletion Problems

In this section, we list some other deletion problems where the expansion lemma was used to obtain kernels.

Here, d is a fixed integer. There have been several previous works on d-PVC with $d = 4$, $d = 5$, etc. Cerveny and Suchy [38] provided parameterized algorithms for 4-PVC and 5-PVC (assign $d = 4$ and $d = 5$ respectively in Box 15). Later, Cerveny et al. [39] obtained a polynomial kernel for d-Path Vertex Cover with $\mathcal{O}(k^2)$ vertices for $d = 4$ and $d = 5$ using Lemma 3. They also obtained a polynomial kernel for d-Path Vertex Cover with $k^4 d^{\mathcal{O}(d)}$ vertices and edges for every fixed d.

Box 15. d-Path Vertex Cover Problem.

d-Path Vertex Cover (d-PVC)
Input: *An undirected graph $G = (V, E)$ and an integer k.*
Parameter: k
Question: *Is there a set $S \subseteq V(G)$ of, at most, k vertices, such that $G - S$ has no path of length d?*

Let $|S| = \ell$. Majumdar et al. [40] used Lemma 3 to obtain a kernel with $\mathcal{O}(\ell^5)$ vertices for VC-2-Mod. Note that the connected components of graphs $G - S$ include isolated vertices, paths, and cycles. Lemma 3 is used to bind the number of odd-cycle components to $\mathcal{O}(\ell^3)$. An auxiliary bipartite graph is created with sets of size three in S that form an independent set on one side, and the odd cycle components on the other side, adding edges based on a notion called blocking set. After applying Lemma 3 with $q = 4$, each 3-sized independent set in the expansion has 4 private odd cycles. It can be proved that it is safe to delete one of the four odd cycle components resulting in an instance with $O(\ell^3)$ many odd cycles (without increasing the size of S). Finally, a kernel with $\mathcal{O}(\ell^5)$ vertices can be obtained with additional arguments for VC-2-Mod as defined in Box 16.

Box 16. Vertex Cover parameterized by degree-2-modulator.

VC-2-Mod
Input: *A graph $G = (V, E)$, a set $S \subseteq V(G)$ such that every vertex of $G - S$ has a degree of, at most, 2, and an integer k.*
Parameter: $|S|$
Question: *Does G have a vertex cover with a size of, at most, k?*

Philip et al. [41] provided a kernel with $\mathcal{O}(k^2)$ vertices for Pathwidth-One Vertex Deletion. Cygan et al. [42] used Lemma 3 to give a simpler kernel with $\mathcal{O}(k^2)$ vertices for this above problem as defined in Box 17.

Box 17. Pathwidth-One Vertex Deletion Problem.

PATHWIDTH-ONE VERTEX DELETION
Input: *An undirected graph $G = (V, E)$ and an integer k.*
Parameter: k.
Question: *Is there a set $S \subseteq V(G)$ of, at most, k vertices, such that $G - S$ has a pathwidth of, at most, one?*

Kim and Kwon [43] initiated the study of BLOCK GRAPH VERTEX DELETION problem from parameterized complexity perspective and provided a kernel with $\mathcal{O}(k^9)$ vertices. Later, Agrawal et al. [44] improved it by giving a kernel with $\mathcal{O}(k^4)$ vertices using Lemma 3.

Jansen et al. [45] provided a kernel with $\mathcal{O}(k^{161} \log^{58} k)$ edges for CHORDAL VERTEX DELETION. Later, Agrawal et al. [46] improved their results using Lemma 3 into a kernel with $\mathcal{O}(k^{12} \log^2 k)$ edges. In addition, Agrawal et al. [46] used the notion of the *independence degree* of a vertex to obtain this result.

Misra et al. [47] initiated the study of the EVEN CYCLE TRANSVERSAL problem from a parameterized complexity perspective. They proved the problem to be FPT and gave a kernel with $\mathcal{O}(k^2)$ vertices. Their kernelization algorithm also uses Lemma 3 as a crucial tool to obtain such a kernel upper bound.

Minch and van Leeuwen [48] provided a kernel with $\mathcal{O}(k^3)$ vertices for OUT-FOREST VERTEX DELETION SET. Later, Agrawal et al. [49] used Lemma 3 to obtain an improved kernel with $\mathcal{O}(k^2)$ vertices.

Misra et al. [50] studied the INDEPENDENT FEEDBACK VERTEX SET problem from a parameterized complexity perspective. This problem is a variation of FEEDBACK VERTEX SET (defined in Box 13) with additional constraints imposed for the feedback vertex set. They use Lemma 3 to obtain a kernel with $\mathcal{O}(k^3)$ vertices and edges.

A common theme for the problems PATHWIDTH-ONE VERTEX DELETION (Box 17), BLOCK GRAPH VERTEX DELETION (Box 18), CHORDAL VERTEX DELETION (Box 19), EVEN CYCLE TRANSVERSAL (Box 20), OUT-FOREST VERTEX DELETION SET (Box 21), and INDEPENDENT FEEDBACK VERTEX SET (Box 22) is that they all have some infinite cycles as forbidden subgraphs. The application of Lemma 3 (expansion lemma) for all of these problems is inspired by the application by Thomassé [6] for FEEDBACK VERTEX SET. The q-expansion from the lemma is used to identify sets of cycles that pairwise intersect in one or two vertices, which has to be hit by the solution. This is used to bind a degree measure for vertices v of G (the measure depends on the problem) by bounding the number of components associated with v.

Box 18. Block Graph Vertex Deletion Problem.

BLOCK GRAPH VERTEX DELETION
Input: *An undirected graph $G = (V, E)$ and an integer k.*
Parameter: k.
Question: *Is there a set $S \subseteq V(G)$ of, at most, k vertices, such that $G - S$ is a block graph?*

Box 19. Chordal Vertex Deletion Problem.

CHORDAL VERTEX DELETION
Input: *An undirected graph $G = (V, E)$ and an integer k.*
Parameter: k.
Question: *Is there a set $S \subseteq V(G)$ of, at most, k vertices, such that $G - S$ has no induced cycle of a length of (at least) four?*

Box 20. Even Cycle Transversal Problem.

EVEN CYCLE TRANSVERSAL
Input: *An undirected graph $G = (V, E)$ and an integer k.*
Parameter: k.
Question: *Is there a set $S \subseteq V(G)$ of, at most, k vertices, such that $G - S$ has no even cycle?*

Box 21. Out-Forest Vertex Deletion Problem.

OUT-FOREST VERTEX DELETION SET
Input: *A directed graph $D = (V, A)$ and a positive integer k.*
Parameter: k.
Question: *Is there a set $S \subseteq V(D)$ of, at most, k vertices, such that $D - S$ is an out-forest?*

Box 22. Independent Feedback Vertex Set Problem.

INDEPENDENT FEEDBACK VERTEX SET
Input: *An undirected graph $G = (V, E)$ and an integer k.*
Parameter: k.
Question: *Is there a set $S \subseteq V(G)$ of, at most, k vertices, such that S is an independent set and $G - S$ is acyclic?*

4.5. Graph Packing Problems

The expansion lemma is also used on packing problems. We list a few below.

CYCLE PACKING is a well-studied problem in parameterized complexity and kernelization. The DISJOINT CYCLE PACKING problem asks if there are at least k cycles in a graph that are pairwise disjoint. It is well-known that the DISJOINT CYCLE PACKING is FPT and admits no polynomial kernelization unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$ [51]. After that, Agrawal et al. [52] introduced relaxation criteria in disjointedness constraints and defined the following two problems PAIRWISE t-DISJOINT CYCLE PACKING and ALMOST t-DISJOINT CYCLE PACKING for a fixed integer t.

Note that if $t = 1$, then ALMOST t-DISJOINT CYCLE PACKING (Box 23) is same as VERTEX DISJOINT CYCLE PACKING. On the other hand, if $t = 0$, then PAIRWISE t-DISJOINT CYCLE PACKING is the same as VERTEX DISJOINT CYCLE PACKING problem. Agrawal et al. [52] proved that when $t = 1$, PAIRWISE t-DISJOINT CYCLE PACKING (defined in Box 24) admits a kernel with $\mathcal{O}(k^4 \log k)$ vertices. This makes crucial use of Lemma 3. If $t = |V(G)|$, then PAIRWISE t-DISJOINT CYCLE PACKING is polynomial-time solvable.

Box 23. Almost t-Disjoint Cycle Packing Problem.

ALMOST t-DISJOINT CYCLE PACKING
Input: *An undirected graph G and an integer k.*
Parameter: k.
Question: *Are there at least k distinct cycles $C_1, \ldots, C_k$ in G, such that every vertex of G appears in, at most, t of these cycles?*

Box 24. Pairwise t-Disjoint Cycle Packing Problem.

PAIRWISE t-DISJOINT CYCLE PACKING (t-PDCP)
Input: *An undirected graph G and an integer k.*
Parameter: *k.*
Question: *Are there at least k distinct cycles* $C_1, \ldots, C_k$ *in G such that for every* $i \neq j$, $|V(C_i) \cap V(C_j)| \leq t$*?*

A simple application of Lemma 3 can obtain a kernelization with $\mathcal{O}(k^2)$ vertices. Fomin et al. [37] provided an improved kernel with $\mathcal{O}(k^{5/3})$ vertices for 3-PATH PACKING. They used Lemma 3 to obtain this result. Later, Bessy et al. [53] introduced a new technique called *rainbow matching* and improved this result into a kernel for 3-PATH PACKING (Box 25) with $\mathcal{O}(k)$ vertices.

Box 25. 3-*Path Packing Problem.*

3-PATH PACKING
Input: *An undirected graph* $G = (V, E)$ *and an integer k.*
Parameter: *k*
Question: *Are there k pairwise vertex disjoints inducing* P_3 *in G?*

4.6. Other Problems

In this section, we discuss some applications of the expansion lemma other than hitting or packing problems in graphs.

Lokshtanov et al. [54] gave a kernel for SET SPLITTING (defined in Box 26) parameterized by k with at most $2k$ sets and k elements. They use a notion of strong cut-sets in a hypergraph, which is closely related to crowns in graphs.

Box 26. Set Splitting Problem.

SET SPLITTING
Input: *A universe* $\mathcal{U}$*, a family* $\mathcal{F}$ *of subsets of* $\mathcal{U}$*, and a positive integer k.*
Parameter: *k.*
Question: *Is there a bipartition of* $\mathcal{U}$ *such that at least k sets have non-empty intersection with both parts?*

Fomin et al. [55] use Lemma 3 with $q = 2$ to give a $3k$-vertex kernel for MAXIMUM INTERNAL SPANNING TREE as defined in Box 27.

Box 27. Maximum Internal Spanning Tree Problem.

MAXIMUM INTERNAL SPANNING TREE
Input: *Graph G and a positive integer k.*
Parameter: *k.*
Question: *Is there a spanning tree of G with at least k internal vertices?*

5. Recent Developments

In this section, we discuss recent developments, including several variations of the expansion lemma along with their applications. Such new variations have appeared recently. Some of those variants are generalizations, and some of them have stronger and special implications. This section is primarily devoted to the notions of stronger expansion lemma, double expansion lemma, balanced expansion lemma, balanced crown decomposition, and additive expansion lemma (see the relevant subsections for details).

5.1. Weighted q-Expansion Lemma

We introduced p-COMPONENT ORDER CONNECTIVITY in Section 4.1 and explained how a kernel with $\mathcal{O}(p^3k)$ vertices can be obtained. In this section, we explain how to obtain a kernel with $\mathcal{O}(p^2k)$ vertices for the same problem using a weighted version of the expansion lemma. The weighted version of the expansion lemma was introduced by Kumar and Lokshtanov [56], who gave a kernel with $2pk$ vertices of this problem. A variant of this lemma was also proposed by Xiao [57], who also gave an $\mathcal{O}(pk)$ kernel for this problem.

We use a simpler version of the weighted expansion lemma from [11]. We start with some definitions.

Definition 8 (Weighted q-Expansion)**.** *Consider a bipartite graph $G = (V = A \uplus B, E)$ with a weight function $w : V(G) \to \{1, \dots, W\}$ for an integer W. Given a subset $X \subseteq V(G)$, let $w(X) = \sum_{x \in X} w(x)$. For an integer $q \geq 1$, a function $f : B \to A$ is called a* weighted q-expansion of A into B *if*

- *for every $b \in B$, $f(b) \in N(b)$, and*
- *for every $a \in A$, $w(f^{-1}(a)) \geq q - W + 1$.*

When $W = 1$, a weighted q-expansion corresponding to a q-expansion for $a \in A$, $w(f^{-1}(a)) \geq q$, and $w(f^{-1}(a))$ is the number of neighbors of a in B.

Lemma 4 (Weighted expansion lemma [11])**.** *Let q and W be positive integers. Consider a bipartite graph $G = (V = A \uplus B, E)$ with weight function $w : V(G) \to \{1, \dots, W\}$ such that*

1. *$w(B) \geq q|A|$; and*
2. *There are no isolated vertices in B.*

Then there exist non-empty vertex sets $X \subseteq A$, $Y \subseteq B$ and a function $f : Y \to X$ such that

- *f is a weighted q-expansion of X into Y, and*
- *No vertex in Y has a neighbor outside X, i.e., $N(Y) \subseteq X$.*

Furthermore, the sets X and Y and the function f can be found in time $\mathcal{O}(mn^{1.5}W^{2.5})$.

We first provide a kernel with $\mathcal{O}(p^2k)$ vertices for p-COMPONENT ORDER CONNECTIVITY for the problem (discussed in [11]) as a precursor to the kernel with $2pk$ vertices by Kumar and Lokshtanov [56]. Recall the auxiliary bipartite graph in Section 4.1 with modulator X on one side and $\mathcal{C}$, the connected components of $G - X$, in the other one. For each connected component in $\mathcal{C}$, we assign its weight as the number of vertices in the component. Note that these weights are at most p. For all the vertices in X, we assign weight 1.

We now apply Lemma 4 with $q = 2p - 1$ to obtain a subset $H \subseteq X$ and a subset $C \subseteq \mathcal{C}$ such that there is a weighted q-expansion of H into C. For any optimal solution S, we look at the set $S' = S \setminus (H \cup C) \cup H$. The weighted q-expansion gives us disjoint connected sets in the auxiliary bipartite graph, each of whose weight is at least $q - p + 1 \geq p$. Thus, we can conclude that the weighted q-expansion represents a connected set of size at least $p + 1$ in G, which any solution needs to hit. Similar to the argument in Section 4.1, we can prove that S' is an optimal solution. Thus, we have a reduction rule as long as Lemma 4 is applicable. If this is not the case, we have $w(\mathcal{C}) \leq q|X| \leq (2p - 1)(p + 1)k$ in the auxiliary bipartite graph. From the conclusion that the number of vertices in $G - X$ is $w(\mathcal{C})$, we have a $\mathcal{O}(p^2k)$ kernel.

Note that in the kernel with $\mathcal{O}(p^3k)$ vertices for p-COMPONENT ORDER CONNECTIVITY described in Section 4.1, we were not able to capture the sizes of the components in the auxiliary bipartite graph. We manage to do so using weighted expansions.

Kumar and Lokshtanov [56] use some further ideas and linear programming to improve the kernel size to $2pk$.

5.2. Stronger Expansion Lemma and Double Expansion Lemma

Fomin et al. [37] defined the following stronger notion of a q-expansion. Let $G = (V = A \uplus B, E)$ be a bipartite graph. Let $q > 0$ be an integer, $\hat{A} \subseteq A$, and $\hat{B} \subseteq B$. We say that there is a *stronger q-expansion* of $\hat{A}$ into $\hat{B}$ if for all subsets $X \subseteq \hat{A}$, we have $|N_G(X) \cap \hat{B}| \geq q|X|$. Using this notion, they prove the following version of the expansion lemma, which does not need the conditions that $|B| \geq q|A|$ or that there are no isolated vertices in B.

Lemma 5 (Stronger Expansion Lemma [37])**.** *Let q be a positive integer, and $G = (V = A \uplus B, E)$ be a bipartite graph. Then there exist (possibly empty) sets $\hat{A} \subseteq A$ and $\hat{B} \subseteq B$ such that*

1. *$\hat{A}$ has a stronger q-expansion into $\hat{B}$,*
2. *$N_G(\hat{B}) \subseteq \hat{A}$, and*
3. *$|B \setminus \hat{B}| \leq q|A \setminus \hat{A}|$.*

Moreover, the sets $\hat{A}$ and $\hat{B}$ can be computed in polynomial time. See Figure 2 for an illustration with $q = 2$.

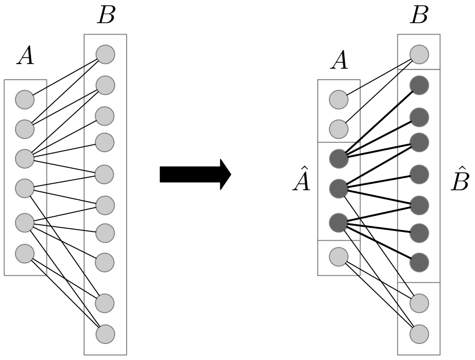

Figure 2. An illustration of Lemma 5 with $q = 2$.

This lemma is different from Lemma 3 as it does not require that $|B| \geq q|A|$ and there are no isolated vertices in B. Moreover, note that this lemma does not demand that $\hat{A}$ and $\hat{B}$ must be non-empty. If $\hat{B} = \emptyset$ for some graph G, then $\hat{A} = \emptyset$ as there is a stronger q-expansion from $\hat{A}$ to $\hat{B}$. Moreover, since $|B \setminus \hat{B}| \leq q|A \setminus \hat{A}|$, we have $|B| \leq q|A|$. Therefore, if $|B| > q|A|$, then $\hat{B} \neq \emptyset$.

Note that in Lemma 3 (expansion lemma), $|Y| = q|X|$. However, in a stronger expansion lemma, it could be that $|\hat{B}| > q|\hat{A}|$. The intuition in applying a stronger expansion lemma for a problem instance is that some vertices of $\hat{B}$ could be safely eliminated from the input graph. Note that since $|B \setminus \hat{B}| \leq q|A \setminus \hat{A}|$, to bound the size of B as a function of $|A|$, we might need to remove some of the vertices of $\hat{B}$. Observe that for every $u \in \hat{A}$, there is at least q edges incident. Thus, by keeping exactly $q|\hat{A}|$ vertices in $\hat{B}$, the property of a q-expansion from $\hat{A}$ is maintained. Thus, the other vertices from $\hat{B}$, which only has neighbors to $\hat{A}$, could be deleted.

The following problem Arc Disjoint 4-Cycle Packing provides an intuition to use the above lemma.

Babu et al. [58] first proved that for every fixed $r \geq 3$, Arc Disjoint r-Cycle Packing is NP-complete in directed graphs of girth r, and this NP-completeness result holds for every $r \geq 4$ when the input graph is bipartite. After that, they provide a kernel with $\mathcal{O}(k^2)$ vertices and $\mathcal{O}(k^3)$ arcs for $r = 4$, i.e., for the Arc Disjoint 4-Cycle Packing (see Box 28) problem using a stronger expansion lemma (Lemma 5). A brief summary of the kernelization algorithm by Babu et al. [58] works as follows.

1. Initially, the authors apply some preprocessing rules to the input graph D and ensure that every arc (u, v) of D is contained in at most $(4k - 4)$ distinct 4-cycles, the pairwise intersection of which is $\{u, v\}$.

2. Let $\mathcal{X}$ be a collection of the maximal set of 4-cycles, such that for every pair of 4-cycles in $\mathcal{X}$, there is at most one common vertex. The authors prove that if $|\mathcal{X}| > 16k^2$, then the input is a YES instance.
3. So, it can be assumed that $|\mathcal{X}| \leq 16k^2$. Let $\mathcal{P}$ be the set of all 3 paths that are contained in the 4-cycle in $\mathcal{X}$. As $|\mathcal{X}| \leq 16k^2$, $|\mathcal{P}|$ is $\mathcal{O}(k^2)$. What is left is to bind the number of vertices in $V(D) \setminus V(\mathcal{P})$.
4. The authors construct an auxiliary bipartite graph H with bipartitions A and B where $A = \mathcal{P}$ and $B = V(D) \setminus V(\mathcal{P})$. For $u \in B$ and $\{x, y, z\} \in \mathcal{P}$, there is an edge in H if and only if $\{x, y, z, u\}$ is a 4-cycle in D.
5. One can now apply Lemma 5 with $q = 1$ and obtain a stronger 1-expansion with sets $\hat{A} \subseteq A$ and $\hat{B} \subseteq B$. It ensures that $|B \setminus \hat{B}| \leq |A \setminus \hat{A}|$. Let $\hat{M} \subseteq E(\hat{A}, \hat{B})$ be the set of edges of H such that every vertex of $\hat{A}$ is incident to exactly one vertex of $\hat{B}$ and every vertex of $\hat{B}$ is incident to exactly one vertex of $\hat{A}$.
6. Finally, the authors apply a reduction rule that deletes the set of vertices from $\hat{B}$, which is not saturated by $\hat{M}$. The 4-cycles that these vertices are part of can be covered by picking one of the vertices from the corresponding 3-paths in $\hat{A}$. The matching M ensures that there exists a 4-cycle for each 3-path in $\hat{A}$. Thus, the reduction rule is safe.
7. The reduction rule is no longer applicable when the set of vertices from $\hat{B}$ that is not saturated by $\hat{M}$ is empty. In this case, we have $|B| = |B \setminus \hat{B}| + |\hat{B}| \leq |A \setminus \hat{A}| + |\hat{A}| = |A|$, which is $\mathcal{O}(k^2)$. Thus, we obtain a kernel with $\mathcal{O}(k^2)$ vertices.

Box 28. Arc Disjoint r-Cycle Packing Problem.

ARC DISJOINT r-CYCLE PACKING
Input: *A directed graph $D = (V, A)$ and an integer k.*
Parameter: k
Question: *Are there at least k pairwise arc disjoint directed cycles of length r in D?*

Fomin et al. [37] provided a kernel for the above problem (as defined in Box 29) with $\mathcal{O}(k^{3/2})$ vertices that use both Lemma 3 (expansion lemma) and Lemma 5 (stronger expansion lemma).

The authors also went on to prove Lemma 6 stated below that simultaneously provides expansion in multiple graphs.

Box 29. Vertex-Disjoint Triangle Packing in Tournament.

TRIANGLE PACKING IN TOURNAMENTS
Input: *A tournament $D = (V, A)$ and an integer k.*
Parameter: k.
Question: *Are there at least k pairwise vertex disjoint directed triangles in D?*

Lemma 6 (Double Expansion Lemma [37])**.** *Let q be a positive integer and $G, H_1, H_2, \ldots, H_d$ be bipartite graphs with bipartitions $A \uplus B, A_1 \uplus R_1, \ldots, A_d \uplus R_d$, respectively, such that for every $i \neq j$, $A_i \cap A_j = \emptyset, R_i \cap R_j = \emptyset$, $\cup_{i=1}^{d} A_i = A$, and $\cup_{i=1}^{d} R_i \subseteq B$. We can in polynomial time compute $\hat{A} \subseteq A, \hat{B} \subseteq B, \hat{A}_i \subseteq A_i, \hat{R}_i \subseteq R_i$ for every $i \in [d]$ such that*

1. $\hat{B} = \cup_{i=1}^{d} \hat{R}_i$,
2. $|B \setminus \hat{B}| \leq q(|A| + | \cup_{i=1}^{d} \hat{A}_i|)$,
3. *$\hat{A}$ has a stronger q-expansion into $\hat{B}$ and for every $i \in [d]$, $\hat{A}_i$ has a stronger q-expansion into $\hat{R}_i$ in H_i,*
4. *$N_G(\hat{B}) \subseteq \hat{A}$, and*
5. *for all $i \in [d]$, $N_{H_i}(R_i) \subseteq A_i$.*

A double expansion lemma was used in Feedback Vertex Set in Tournaments to give an improved kernel. We formally define the problem below.

Feedback Vertex Set in Tournaments can be cast as a 3-hitting set problem (see Box 6), and hence its kernel can be adapted to obtain a kernel with $\mathcal{O}(k^2)$ vertices. Fomin et al. [37] used Lemma 6 (double expansion lemma) to obtain a kernel with $\mathcal{O}(k^{3/2})$ vertices. We give a sketch of the algorithm below.

As hitting directed cycles in a tournament is equivalent to hitting directed triangles in a tournament, a 3-approximation algorithm follows for Feedback Vertex Set in Tournaments. However, a set S has a size of, at most, $7k/3$ can also be obtained due to Minch et al. [59] such that $D - S$ is acyclic. We call an arc $(x, y) \in A$ as *strong arc* if (x, y) is part of at least $k + 2$ triangles with the other endpoints of the triangles from S. In this case, one of x and y has to be in every solution.

Since $D - S$ is acyclic, its vertices have a topological ordering. We can extend this to an ordering of vertices of D by adding vertices in S such that it minimizes certain “conflicts". In this ordering, we can then argue that the vertices of every triangle in D that do not contain a strong arc are within $7k$ distance from each other in the ordering. Hence, these triangles are, in a sense *local*.

The problem boils down to simultaneously hitting strong arcs and the local triangles. This is achieved by using Lemma 6 by constructing a *global* graph corresponding to the strong arcs and *local* graphs for the local triangles.

First, the vertices of $D - S$ are broken into groups of size $100k$ according to the topological ordering. Then, the authors [37] look at the first $p = 4\sqrt{k}$ of such groups $Y_1, Y_2, \ldots, Y_p$, and discard the first and last $7k + 1$ vertices of each to form intervals $Y'_1, Y'_2, \ldots, Y'_p$. For $i \in [p]$, let S_i be the set of vertices in S that are placed within Y'_i in the ordering of $V(D)$. Note that the vertices of any local triangle containing a vertex in Y'_i are contained in the set $S_i \cup Y_i$. Then each interval $Y'_i, i \in [p]$ can be divided into q subintervals $Y_{i,1}, Y_{i,2}, \ldots, Y_{i,q}$, each of size $3\sqrt{k}$.

Next, they define a global bipartite graph G with bipartition $S \uplus \{Y_{i,j} : i \in [p], j \in [q]\}$ and strong arcs. Additionally, the authors define local bipartite graphs H_i with bipartition $S_i \uplus \{Y_{i,j} : i \in [p], j \in [q]\}$ that contains the local triangles. Finally, they appropriately apply a double expansion lemma (Lemma 6) to obtain expansion and neighborhood containment properties in all of the graphs simultaneously. This allows them to find a vertex in $V(D) \setminus S$, which can be safely deleted. When the lemma is no longer applicable, the number of vertices is $|S| + |\cup_{i\in[p]} Y_i| \leq 2k + 100k \cdot 4\sqrt{k}$. This gives us a summary of the kernel for Feedback Vertex Set in Tournaments (defined in Box 30) with $\mathcal{O}(k^{3/2})$ vertices (See [37] for more details).

Box 30. Feedback Vertex Set Problem in Tournament.

Feedback Vertex Set in Tournaments
Input: *A tournament $D = (V, A)$ and an integer k.*
Parameter: *k.*
Question: *Is there a set $S \subseteq V(D)$ of, at most, k vertices, such that $D - S$ is acyclic?*

5.3. Balanced Expansion and Balanced Crown Decomposition

Now, we introduce the notion of *balanced crown decomposition*, *balanced expansion*, and some related polynomial-time algorithms developed by Casel et al. [60].

Definition 9 (Balanced Expansion)**.** *Let $G = (V = A \uplus B, E, w)$ be a vertex-weighted bipartite graph, where $w^B_{\max} = \max_{b\in B} w(b)$. For $q \in \mathbb{N}_0$, a partition $A_1 \cup A_2$ of A, and $f : B \to A$, the tuple (A_1, A_2, f, q) is called a* balanced expansion *if:*

1. $w(a) + w(f^{-1}(a)) \geq q - w^B_{\max} + 1$ *if* $a \in A_1$,
2. $w(a) + w(f^{-1}(a)) \leq q + w^B_{\max} - 1$ *if* $a \in A_2$,

3. *for all $b \in B$, $f(b) \in N(b)$, and*
4. $N(f^{-1}(A_1)) \subseteq A_1$.

(See Figure 3 for an illustration).

Based on the above definition, the weight of a vertex of A combined with the weight of its pre-image is in the range of $q - w^B_{\max} + 1$ and $q + w^B_{\max} - 1$. The function f is similar to the condition of expansion in Section 4. Using the above definition, Casel et al. [60] proved the following result.

Lemma 7 (Balanced Expansion Lemma)**.** *Consider a vertex-weighted bipartite graph $G = (V = A \uplus B, E, w)$ with no isolated vertices in B and $q \geq \max_{b \in B} w(b) = w^B_{\max}$. A balanced expansion (A_1, A_2, f, q) for G can be computed in the $\mathcal{O}(mn)$-time. Furthermore, if $w(A) + w(B) \geq q|A|$, then $A_1 \neq \emptyset$.*

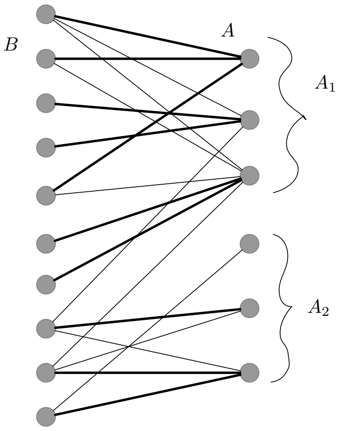

Figure 3. An illustration of a balanced expansion for $q = 3$ and with $w(v) = 1$ for every vertex v. The assignment f is depicted with bold edges.

The following is a more fine-grained version of the balanced expansion defined in [60].

Definition 10 (Fractional Balanced Expansion)**.** *Let $G = (V = A \uplus B, E, w)$ be a vertex-weighted bipartite graph, where $w^B_{\max} = \max_{b \in B} w(b)$. For $q \in \mathbb{N}_0$, a partition $A_1 \cup A_2$ of A, and $g : E(G) \to \mathbb{N}_0$, the tuple (A_1, A_2, g, q) is called a* fractional balanced expansion *if:*

1. $w(a) + \sum_{b \in B} g(ab) \geq q$ *if* $a \in A_1$,
2. $w(a) + \sum_{b \in B} g(ab) \leq q$ *if* $a \in A_2$,
3. *for all $b \in B$, $\sum_{a \in A} g(ab) \leq w(b)$, and*
4. $N(B_U \cup B_{A_1}) \subseteq A_1$, *where $B_a = \{b \in B \mid g(ab) > 0\}$ for $a \in A$, $B_{A'} = \cup_{a \in A'} B_a$ for $A' \subseteq A$ and $B_U = \{b \in B \mid \sum_{a \in A} g(ab) < w(b)\}$.*

Such a tuple (A_1, A_2, g, q) can be interpreted as a fractional balanced expansion as follows. The weights of the vertices of the f^{-1} function are distributed to edges through the g function, i.e., g can be interpreted as a fractional assignment from $v \in B$ to $u \in N(b)$. For the non-fractional version (in Definition 9, we aim that for all $v \in B$, $g(uv) = w(u)$ for a unique $u \in A$ while $g(u'v) = 0$ when $u' \in A \setminus \{u\}$.

The authors [60] prove the following fractional version of Lemma 7.

Lemma 8 (Fractional Balanced Expansion Lemma)**.** *Let $G = (V = A \uplus B, E, w)$ be a vertex-weighted bipartite graph, where $w^B_{\max} = \max_{b \in B} w(b)$. For $q \in \mathbb{N}_0$, a partition $A_1 \cup A_2$ of A and $g : E(G) \to \mathbb{N}_0$, a fractional balanced expansion (A_1, A_2, g, q) can be computed $\mathcal{O}(mn)$-time. Furthermore, if $w(A) + w(B) \geq q|A|$, then $A_1 \neq \emptyset$.*

The authors [60] use Lemma 7 to prove the construction of a structure called λ-*balanced crown decomposition* for a fixed integer λ described below.

Definition 11 (λ-balanced crown decomposition [60])**.** *A λ-balanced crown decomposition for a vertex-weighted graph $G = (V, E, w)$ is a tuple $(C, H, \mathfrak{R}, f)$ where (H, C, R) is a partition of $V(G)$ and $\mathfrak{R}$ is a partition of R and $f : \mathbb{CC} \to H$ where $\mathbb{CC}(C)$ is the set of connected components of $G[C]$ such that:*

1. *there are no edges from C to R,*
2. *$w(Q) < \lambda$ for each $Q \in \mathbb{CC}(C)$,*
3. *$f(Q) \in N(Q)$ for every $Q \in \mathbb{C}(C)$,*
4. *$w(h) + w(f^{-1}(h)) \geq \lambda$ for each $h \in H$, and*
5. *$G[R']$ is connected and $\lambda \leq w(R') \leq 3\lambda - 3$ for each $R' \in \mathfrak{R}$.*

The first four properties are those of crown property with some added weight specifications (see [57]), and the last property ensures that all parts of the partition are fairly balanced. Moreover, the fourth condition implies that every $u \in H$ should be connected to component(s) of $G[C]$ with weight at least $\lambda - 1$. The notion of the λ-balanced crown decomposition is something that Casel et al. [60] contributed toward when obtaining new results. However, a crucial contribution of Definition 11 by [60] is the fifth condition. The fifth condition states that every part of the partition of R is of weight within the range $[\lambda, 3\lambda - 3]$.

Lemma 7 (balanced expansion lemma) was not directly used to prove any kernel upper bound result, but was used by Casel et al. [60] to prove the "balanced crown lemma" that we state below.

Lemma 9 (Balanced Crown Lemma [60])**.** *Let $G = (V, E, w)$ be a vertex-weighted graph and $\lambda \in \mathbb{N}$, such that each connected component in G has a weight of at least λ. A λ-balanced crown decomposition $(C, R, \mathfrak{R}, f)$ of G can be computed in $\mathcal{O}(k^2|V||E|)$-time where $k = |H| + |\mathfrak{R}| \leq \min\{w(G)/\lambda, |V|\}$.*

Casel et al. [60] used Lemma 9 to obtain a kernel for the weighted version of a separation and packing problem that we define below. Note that W-Weight Separator can be viewed as a weighted variant of the p-component order connectivity problem, and W-Weight Packing is simply its packing analog (see Box 31 and Box 32 below).

Box 31. W-Weight Separator Problem.

W-Weight Separator
Input: *A vertex-weighted undirected graph $G = (V, E, w)$ and integer k.*
Question: *Is there a set $S \subseteq V(G)$ of, at most, k vertices, such that every connected component of $G - S$ has a total weight of less than W?*

Box 32. W-Weighted Packing Problem.

W-Weight Packing
Input: *A vertex-weighted undirected graph $G = (V, E, w)$ and integers k.*
Question: *Are there k pairwise disjoint sets $V_1, \ldots, V_k$, such that for every $1 \leq i \leq k$, $V_i \subseteq V(G), w(V_i) \geq k$ and $G[V_i]$ is connected?*

For both these above mentioned problems, Casel et al. [60] provided kernels with $3k(W - 1)$ vertices, which can be computed in $\mathcal{O}(k^2|V||E|)$-time.

Finally, they studied the balanced connected partition problem stated below.

In the same paper, they use Definition 11 and Lemma 9 to give a polynomial time for Max-Min (Min–Max) Balanced Connected Partition, which outputs a partition

$V_1^*, \ldots, V_k^*$ of $V(G)$ such that for each $i \in [k]$, $G[V_i]$ is connected and $w(V_i^*) \geq W/3$ to give an approximation algorithm for MAX-MIN (MIN–MAX) BALANCED CONNECTED PARTITION as defined in Box 33.

Box 33. Balanced Connected Partition Problem.

MAX-MIN (MIN–MAX) BALANCED CONNECTED PARTITION
Input: *A vertex-weighted undirected graph $G = (V, E, w)$ and integer k.*
Question: *Is there a partition $V_1, \ldots, V_k$ of $V(G)$ such that for each $i \in [k]$, $w(V_i) \geq W$ and $G[V_i]$ is connected?*

5.4. Additive Expansion Lemma

Koana et al. [61] introduced the following variant of the expansion and the expansion lemma.

Definition 12 (q-Additive Expansion, [61])**.** *Let $G = (V = A \uplus B, E)$ be a bipartite graph and q be a positive integer. Then, G is said to have a q-*additive expansion *from A into B if for every $B' \subseteq B$ of size q, there is a matching saturating A in $G[A, B \setminus B']$.*

For completeness, we give proof of the following proposition that is stated in [61].

Proposition 3. *Let $G = (V = A \uplus B, E)$ be a bipartite graph. There is a q-additive expansion from A into B if and only if for every nonempty $X \subseteq A$, $|N(X)| \geq |X| + q$.*

Proof. We will prove this using Hall's theorem.

Consider the forward direction ($\Rightarrow$) first. Suppose that G has a q-additive expansion from A to B. Then, for every $B' \subseteq B$ of size q, there is a matching saturating A in $G' = G[A, B \setminus B']$. It means that for every non-empty $X \subseteq A$, $|N(X) \setminus B'| \geq |X|$. Consider the value of $|N_G(X)|$. If $|N_G(X)| < |X| + q$, then consider a set of an arbitrary set $\hat{B}$ of q vertices from $N_G(X)$. Clearly, $\hat{B} \subseteq B$ has q vertices. However, then in the graph $\hat{G} = G[A, B \setminus \hat{B}]$, for the same set $X \subseteq A$, it holds that $|N_{\hat{G}}(X)| < |X|$. Then it leads to a contradiction that there is a matching saturating A in $\hat{G}$.

Let us now prove the backward direction ($\Leftarrow$). Suppose that for every nonempty $X \subseteq A$, $|N(X)| \geq |X| + q$. We have to justify that for every $B' \subseteq B$ of size q, there is a matching saturating A in $G[A, B \setminus B']$. Suppose that for some $B^* \subseteq B$ of size q, there is no matching saturating A in $G^* = G[A, B \setminus B^*]$. Then, by Proposition 1 (Hall's theorem), there is $X \subseteq A$, such that $|N_{G^*}(X)| < |X|$. As B^* is of size q, then in G also, it holds that $|N_G(X)| < |X| + q$. This contradicts the premise, which completes the proof of the observation. □

Now, we state the *additive expansion lemma* and give a short explanation of how it has been used to provide an improved kernel for PARTIAL VERTEX COVER.

Lemma 10 (Additive Expansion Lemma [61])**.** *Let $q \geq 1$ and $G = (V = A \uplus B, E)$ be a bipartite graph. If $|B| > q|A|$ and there is no isolated vertex in B, then there exists nonempty $\hat{A} \subseteq A$ and $\hat{B} \subseteq B$ such that*

- *There is a q-additive expansion of $\hat{A}$ into $\hat{B}$; and*
- *No vertex of $\hat{B}$ has the neighbor outside $\hat{A}$, i.e., $N(\hat{B}) \subseteq \hat{A}$.*

First, the authors [61] provide a kernel with $(\ell + 2)(k + \ell)$ vertices that use the expansion lemma (Lemma 3). It works as follows.

- First, they consider the LP formulation of VERTEX COVER. By Nemhauser and Trotter [62], there is a fractional optimal solution for the LP formulation of VERTEX COVER such that all vertices are assigned values of either 0, 1, or 1/2.

- Let $V_0, V_1, V_{1/2}$ denote the vertices that are assigned values 0, 1, 1/2, respectively. If $|V_1| + |V_{1/2}| \geq k + \ell + 1$, then the instance is a no-instance.
- Otherwise, they construct an auxiliary bipartite graph, and apply the expansion lemma to obtain a kernel with $(\ell + 2)(k + \ell)$ vertices.

After that, they give a proof using the additive expansion lemma (Lemma 10), a kernel with $(\ell' + 1)(k + \ell)$ vertices can be obtained, such that $\ell' = \max\{\ell, 1\}$. The rest of the proof for the problem defined in Box 34 has similar structures, but a crucial difference is that they use Lemma 10 instead of Lemma 3.

Box 34. Partial Vertex Cover Problem.

PARTIAL VERTEX COVER
Input: *An undirected graph $G = (V, E)$ and two integers k, ℓ.*
Parameter: $k + \ell$
Question: *Is there a set S of, at most, k vertices, such that $G - S$ has (at most) $|E(G)| - \ell$ edges?*

6. Conclusions and Future Research

Our aim in this survey is to bring variations and parameterized complexity applications of an important tool called expansion lemma into one place. Different authors have obtained variations to suit the specific problems they deal with, and so some of the applications tend to apply to specific kinds of problems. The survey is non-exhaustive simply because new variations and applications are exploding at a fast pace. We hope that our survey will trigger further applications of these variations in parameterized complexity. It will be interesting to see more applications of recent variants, such as Lemma 4, Lemma 6, and Lemma 10, and to find tighter bounds for the kernel results to which they are applied. In particular, the following research directions would be interesting to investigate.

- There are some other well-studied vertex deletion problems in the kernelization perspective, e.g., an $\mathcal{O}(k^2)$ vertex kernel for SPLIT VERTEX DELETION (see [63]) but it has a kernel lower bound of $\mathcal{O}(k^{2-\varepsilon})$ edges [4]. It would be interesting to see if one of these variants of the expansion lemma can be useful to obtain a sub-quadratic vertex kernel for these problems.
- Recently, Jacob et al. [64,65] introduced the study of vertex deletion problems to scattered graph classes. While they proved parameterized complexity of this problem, kernelization complexity is open and unexplored. It would be nice to explore if the expansion lemma or some of its variants can be useful for this.

Author Contributions: Conceptualization, A.J., D.M. and V.R.; Writing—original draft preparation, A.J., D.M. and V.R.; Writing—review and editing, A.J., D.M. and V.R. All authors have read and agreed to the published version of the manuscript.

Funding: Research of Ashwin Jacob was funded by European Research Council (ERC) grant titled PARAPATH.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Cygan, M.; Fomin, F.V.; Kowalik, L.; Lokshtanov, D.; Marx, D.; Pilipczuk, M.; Pilipczuk, M.; Saurabh, S. *Parameterized Algorithms*; Springer: Berlin/Heidelberg, Germany, 2015.
2. Bodlaender, H.L.; Downey, R.G.; Fellows, M.R.; Hermelin, D. On problems without polynomial kernels. *J. Comput. Syst. Sci.* **2009**, *75*, 423–434. [CrossRef]
3. Dom, M.; Lokshtanov, D.; Saurabh, S. Kernelization lower bounds through colors and IDs. *ACM Trans. Algorithms (TALG)* **2014**, *11*, 1–20. [CrossRef]

4. Dell, H.; Van Melkebeek, D. Satisfiability allows no nontrivial sparsification unless the polynomial-time hierarchy collapses. *J. ACM (JACM)* **2014**, *61*, 1–27. [CrossRef]
5. Chor, B.; Fellows, M.; Juedes, D. Linear kernels in linear time, or how to save k colors in $O(n^2)$ steps. In *Proceedings of the International Workshop on Graph-Theoretic Concepts in Computer Science*; Springer: Berlin/Heidelberg, Germany, 2004; pp. 257–269.
6. Thomassé, S. A $4k^2$ kernel for feedback vertex set. *ACM Trans. Algorithms* **2010**, *6*, 1–8. [CrossRef]
7. Niedermeier, R. *Invitation to Fixed-Parameter Algorithms*; Oxford University Press: Oxford, UK, 2006.
8. Guo, J.; Niedermeier, R. Invitation to data reduction and problem kernelization. *SIGACT News* **2007**, *38*, 31–45. [CrossRef]
9. Downey, R.G.; Fellows, M.R. *Fundamentals of Parameterized Complexity*; Texts in Computer Science; Springer: Berlin/Heidelberg, Germany, 2013.
10. Flum, J.; Grohe, M. *Parameterized Complexity Theory*; Texts in Theoretical Computer Science. An EATCS Series; Springer: Berlin/Heidelberg, Germany, 2006.
11. Fomin, F.V.; Lokshtanov, D.; Saurabh, S.; Zehavi, M. *Kernelization—Theory of Parameterized Preprocessing*; Cambridge University Press: Cambridge, UK, 2019.
12. Diestel, R. *Graph Theory*, 4th ed.; Graduate Texts in Mathematics; Springer: Berlin/Heidelberg, Germany, 2012; Volume 173.
13. Hall, P. On Representatives of Subsets. *J. Lond. Math. Soc.* **1987**, *S1–S10*, 26–30.
14. Edmonds, J. Paths, Trees, and Flowers. *Can. J. Math.* **1965**, *17*, 449–467. [CrossRef]
15. Lovasz, L.; Plumber, M.D. *Matching Theory*; AMS Chelsea Publishing, American Mathematical Society: Providence, RI, USA, 2000.
16. Robertson, N.; Seymour, P.D. Graph minors. III. Planar tree-width. *J. Comb. Theory Ser. B* **1984**, *36*, 49–64. [CrossRef]
17. Robertson, N.; Seymour, P.D. Graph Minors. II. Algorithmic Aspects of Tree-Width. *J. Algorithms* **1986**, *7*, 309–322. [CrossRef]
18. Robertson, N.; Seymour, P.D. Graph minors. IV. Tree-width and well-quasi-ordering. *J. Comb. Theory Ser. B* **1990**, *48*, 227–254. [CrossRef]
19. Chen, J.; Fernau, H.; Shaw, P.; Wang, J.; Yang, Z. Kernels for packing and covering problems. *Theor. Comput. Sci.* **2019**, *790*, 152–166. [CrossRef]
20. Fellows, M.R. Blow-ups, win/win's, and crown rules: Some new directions in FPT. In *Proceedings of the International Workshop on Graph-Theoretic Concepts in Computer Science*; Springer: Berlin/Heidelberg, Germany, 2003; pp. 1–12.
21. Hopcroft, J.E.; Karp, R.M. An n^5/2 algorithm for maximum matchings in bipartite graphs. *SIAM J. Comput.* **1973**, *2*, 225–231. [CrossRef]
22. Lampis, M. A kernel of order $2k - c \log k$ for vertex cover. *Inf. Process. Lett.* **2011**, *111*, 1089–1091. [CrossRef]
23. Li, W.; Zhu, B. A 2k-kernelization algorithm for Vertex Cover based on Crown Decomposition. *Theor. Comput. Sci.* **2018**, *739*, 80–85. [CrossRef]
24. Li, W.; Ding, Y.; Yang, Y.; Rong, G. A (2 + ϵ)k-vertex kernel for the dual coloring problem. *Theor. Comput. Sci.* **2021**, *868*, 6–11. [CrossRef]
25. Lokshtanov, D. *New Methods in Parameterized Algorithms and Complexity*; University of Bergen: Bergen, Norway, 2009.
26. Banik, A.; Jacob, A.; Paliwal, V.K.; Raman, V. Fixed-parameter tractability of $(n - k)$ list coloring. *Theory Comput. Syst.* **2020**, *64*, 1307–1316. [CrossRef]
27. Bodlaender, H.L.; Jansen, B.M.P.; Kratsch, S. Kernel bounds for path and cycle problems. *Theor. Comput. Sci.* **2013**, *511*, 117–136. [CrossRef]
28. Fellows, M.; Heggernes, P.; Rosamond, F.; Sloper, C.; Telle, J.A. Finding k disjoint triangles in an arbitrary graph. In *Proceedings of the International Workshop on Graph-Theoretic Concepts in Computer Science*; Springer: Berlin/Heidelberg, Germany, 2004; pp. 235–244.
29. Prieto, E.; Sloper, C. Reducing to independent set structure: The case of k-internal spanning tree. *Nord. J. Comput.* **2005**, *12*, 308–318.
30. Chlebík, M.; Chlebíková, J. Crown reductions for the minimum weighted vertex cover problem. *Discret. Appl. Math.* **2008**, *156*, 292–312. [CrossRef]
31. Abu-Khzam, F.N. A kernelization algorithm for d-hitting set. *J. Comput. Syst. Sci.* **2010**, *76*, 524–531. [CrossRef]
32. Abu-Khzam, F.N. An improved kernelization algorithm for r-Set Packing. *Inf. Process. Lett.* **2010**, *110*, 621–624. [CrossRef]
33. Wang, J.; Ning, D.; Feng, Q.; Chen, J. An improved kernelization for P2-packing. *Inf. Process. Lett.* **2010**, *110*, 188–192. [CrossRef]
34. Xiao, M.; Kou, S. Kernelization and Parameterized Algorithms for 3-Path Vertex Cover. In Proceedings of the Theory and Applications of Models of Computation—14th Annual Conference, TAMC 2017, Bern, Switzerland, 20–22 April 2017; Lecture Notes in Computer Science; Gopal, T.V., Jäger, G., Steila, S., Eds.; Volume 10185, pp. 654–668.
35. Gallai, T. Maximum-minimum Sätze und verallgemeinerte Faktoren von Graphen. *Acta Math. Hung.* **1961**, *12*, 131–173. [CrossRef]
36. Iwata, Y. Linear-Time Kernelization for Feedback Vertex Set. In Proceedings of the 44th International Colloquium on Automata, Languages, and Programming, ICALP 2017, Warsaw, Poland, 10–14 July 2017; LIPIcs; Chatzigiannakis, I., Indyk, P., Kuhn, F., Muscholl, A., Eds.; Schloss Dagstuhl—Leibniz-Zentrum für Informatik: Wadern, Germany; Volume 80, pp. 68:1–68:14.
37. Fomin, F.V.; Le, T.; Lokshtanov, D.; Saurabh, S.; Thomassé, S.; Zehavi, M. Subquadratic Kernels for Implicit 3-Hitting Set and 3-Set Packing Problems. *ACM Trans. Algorithms* **2019**, *15*, 13:1–13:44. [CrossRef]

38. Cervený, R.; Suchý, O. Faster FPT Algorithm for 5-Path Vertex Cover. In Proceedings of the 44th International Symposium on Mathematical Foundations of Computer Science, MFCS 2019, Aachen, Germany, 26–30 August 2019; LIPIcs; Rossmanith, P., Heggernes, P., Katoen, J., Eds.; Schloss Dagstuhl—Leibniz-Zentrum für Informatik: Wadern, Germany; Volume 138, pp. 32:1–32:13.
39. Cervený, R.; Choudhary, P.; Suchý, O. On Kernels for d-Path Vertex Cover. *arXiv* **2021**, arXiv:2107.12245.
40. Majumdar, D.; Raman, V.; Saurabh, S. Polynomial Kernels for Vertex Cover Parameterized by Small Degree Modulators. *Theory Comput. Syst.* **2018**, *62*, 1910–1951. [CrossRef]
41. Philip, G.; Raman, V.; Villanger, Y. A Quartic Kernel for Pathwidth-One Vertex Deletion. In Proceedings of the Graph Theoretic Concepts in Computer Science—36th International Workshop, WG 2010, Zarós, Crete, Greece, 28–30 June 2010; Lecture Notes in Computer Science; Thilikos, D.M., Ed.; Volume 6410, pp. 196–207.
42. Cygan, M.; Pilipczuk, M.; Pilipczuk, M.; Wojtaszczyk, J.O. An Improved FPT Algorithm and a Quadratic Kernel for Pathwidth One Vertex Deletion. *Algorithmica* **2012**, *64*, 170–188. [CrossRef]
43. Kim, E.J.; Kwon, O. A Polynomial Kernel for Block Graph Deletion. In Proceedings of the 10th International Symposium on Parameterized and Exact Computation, IPEC 2015, Patras, Greece, 16–18 September 2015; LIPIcs; Husfeldt, T., Kanj, I.A., Eds.; Schloss Dagstuhl—Leibniz-Zentrum für Informatik: Wadern, Germany; Volume 43, pp. 270–281.
44. Agrawal, A.; Kolay, S.; Lokshtanov, D.; Saurabh, S. A Faster FPT Algorithm and a Smaller Kernel for Block Graph Vertex Deletion. In Proceedings of the LATIN 2016: Theoretical Informatics—12th Latin American Symposium, Ensenada, Mexico, 11–15 April 2016; Lecture Notes in Computer Science; Kranakis, E., Navarro, G., Chávez, E., Eds.; Springer: Berlin/Heidelberg, Germany, 2016; Volume 9644, pp. 1–13.
45. Jansen, B.M.P.; Pilipczuk, M. Approximation and Kernelization for Chordal Vertex Deletion. *SIAM J. Discret. Math.* **2018**, *32*, 2258–2301. [CrossRef]
46. Agrawal, A.; Lokshtanov, D.; Misra, P.; Saurabh, S.; Zehavi, M. Feedback vertex set inspired kernel for chordal vertex deletion. *ACM Trans. Algorithms (TALG)* **2018**, *15*, 1–28.
47. Misra, P.; Raman, V.; Ramanujan, M.S.; Saurabh, S. Parameterized Algorithms for Even Cycle Transversal. In Proceedings of the Graph-Theoretic Concepts in Computer Science—38th International Workshop, WG 2012, Jerusalem, Israel, 26–28 June 2012; Revised Selected Papers; Lecture Notes in Computer Science; Golumbic, M.C., Stern, M., Levy, A., Morgenstern, G., Eds.; Springer: Berlin/Heidelberg, Germany, 2012; Volume 7551, pp. 172–183.
48. Mnich, M.; van Leeuwen, E.J. Polynomial kernels for deletion to classes of acyclic digraphs. *Discret. Optim.* **2017**, *25*, 48–76. [CrossRef]
49. Agrawal, A.; Saurabh, S.; Sharma, R.; Zehavi, M. Kernels for deletion to classes of acyclic digraphs. *J. Comput. Syst. Sci.* **2018**, *92*, 9–21. [CrossRef]
50. Misra, N.; Philip, G.; Raman, V.; Saurabh, S. On Parameterized Independent Feedback Vertex Set. *Theor. Comput. Sci.* **2012**, *461*, 65–75. [CrossRef]
51. Bodlaender, H.L.; Thomassé, S.; Yeo, A. Kernel bounds for disjoint cycles and disjoint paths. *Theor. Comput. Sci.* **2011**, *412*, 4570–4578. [CrossRef]
52. Agrawal, A.; Lokshtanov, D.; Majumdar, D.; Mouawad, A.E.; Saurabh, S. Kernelization of Cycle Packing with Relaxed Disjointness Constraints. *SIAM J. Discret. Math.* **2018**, *32*, 1619–1643. [CrossRef]
53. Bessy, S.; Bougeret, M.; Thilikos, D.M.; Wiederrecht, S. Kernelization for Graph Packing Problems via Rainbow Matching. In Proceedings of the 2023 ACM-SIAM Symposium on Discrete Algorithms, SODA 2023, Florence, Italy, 22–25 January 2023; Bansal, N., Nagarajan, V., Eds.; SIAM: Bangkok, Thailand, 2023; pp. 3654–3663.
54. Lokshtanov, D.; Saurabh, S. Even faster algorithm for set splitting! In *Proceedings of the International Workshop on Parameterized and Exact Computation*; Springer: Berlin/Heidelberg, Germany, 2009; pp. 288–299.
55. Fomin, F.V.; Gaspers, S.; Saurabh, S.; Thomassé, S. A linear vertex kernel for maximum internal spanning tree. *J. Comput. Syst. Sci.* **2013**, *79*, 1–6. [CrossRef]
56. Kumar, M.; Lokshtanov, D. A 2lk Kernel for l-Component Order Connectivity. In Proceedings of the 11th International Symposium on Parameterized and Exact Computation, IPEC 2016, Aarhus, Denmark, 24–26 August 2016; pp. 20:1–20:14.
57. Xiao, M. Linear kernels for separating a graph into components of bounded size. *J. Comput. Syst. Sci.* **2017**, *88*, 260–270. [CrossRef]
58. Babu, J.; Krithika, R.; Rajendraprasad, D. Packing Arc-Disjoint 4-Cycles in Oriented Graphs. In Proceedings of the 42nd IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2022, IIT Madras, Chennai, India, 18–20 December 2022; LIPIcs; Dawar, A., Guruswami, V., Eds.; Schloss Dagstuhl—Leibniz-Zentrum für Informatik: Wadern, Germany, 2022; Volume 250, pp. 5:1–5:16.
59. Mnich, M.; Williams, V.V.; Végh, L.A. A 7/3-Approximation for Feedback Vertex Sets in Tournaments. In Proceedings of the 24th Annual European Symposium on Algorithms, ESA 2016, Aarhus, Denmark, 22–24 August 2016; LIPIcs; Sankowski, P., Zaroliagis, C.D., Eds.; Schloss Dagstuhl—Leibniz-Zentrum für Informatik: Wadern, Germany, 2016; Volume 57, pp. 67:1–67:14.
60. Casel, K.; Friedrich, T.; Issac, D.; Niklanovits, A.; Zeif, Z. Balanced Crown Decomposition for Connectivity Constraints. In Proceedings of the 29th Annual European Symposium on Algorithms, ESA 2021, Lisbon, Portugal, 6–8 September 2021; LIPIcs; Mutzel, P., Pagh, R., Herman, G., Eds.; Schloss Dagstuhl—Leibniz-Zentrum für Informatik: Wadern, Germany, 2021; Volume 204, pp. 26:1–26:15.
61. Koana, T.; Nichterlein, A.; Wünsche, N. Kernelization for Partial Vertex Cover via (Additive) Expansion Lemma. *arXiv* **2022**, arXiv:2211.07001.

62. Nemhauser, G.L.; Trotter, L.E., Jr. Vertex packings: Structural properties and algorithms. *Math. Program.* **1975**, *8*, 232–248. [CrossRef]
63. Agrawal, A.; Gupta, S.; Jain, P.; Krithika, R. Quadratic vertex kernel for split vertex deletion. *Theor. Comput. Sci.* **2020**, *833*, 164–172. [CrossRef]
64. Jacob, A.; Majumdar, D.; Raman, V. Parameterized Complexity of Deletion to Scattered Graph Classes. In Proceedings of the 15th International Symposium on Parameterized and Exact Computation, IPEC 2020, Hong Kong, China, 14–18 December 2020; LIPIcs; Cao, Y., Pilipczuk, M., Eds.; Schloss Dagstuhl—Leibniz-Zentrum für Informatik: Wadern, Germany, 2020; Volume 180, pp. 18:1–18:17.
65. Jacob, A.; Majumdar, D.; Raman, V. Faster FPT Algorithms for Deletion to Pairs of Graph Classes. In Proceedings of the Fundamentals of Computation Theory—23rd International Symposium, FCT 2021, Athens, Greece, 12–15 September 2021; Lecture Notes in Computer Science; Bampis, E., Pagourtzis, A., Eds.; Springer: Berlin/Heidelberg, Germany, 2021; Volume 12867, pp. 314–326.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.