Weak Compositions and Their Applications to Polynomial Lower Bounds for Kernelization

Danny Hermelin
Max Plank Institute for Informatics, Germany
hermelin@mpi-inf.mpg.de

Xi Wu†
University of Wisconsin-Madison, USA
xiwu@cs.wisc.edu

Abstract. We introduce a new form of composition called *weak composition* that allows us to obtain polynomial kernelization lower-bounds for several natural parameterized problems. Let $d \geq 2$ be some constant and let $L_1, L_2 \subseteq \{0,1\}^* \times \mathbb{N}$ be two parameterized problems where the unparameterized version of L_1 is NP-hard. Assuming coNP $\not\subseteq$ NP/poly, our framework essentially states that composing t L_1-instances each with parameter k, to an L_2-instance with parameter $k' \leq t^{1/d}k^{O(1)}$, implies that L_2 does not have a kernel of size $O(k^{d-\varepsilon})$ for any $\varepsilon > 0$. We show two examples of weak composition and derive polynomial kernelization lower bounds for d-Bipartite Regular Perfect Code and d-Dimensional Matching, parameterized by the solution size k. By reduction, using linear parameter transformations, we then derive the following lower-bounds for kernel sizes when the parameter is the solution size k (assuming coNP $\not\subseteq$ NP/poly):

- d-Set Packing, d-Set Cover, d-Exact Set Cover, Hitting Set with d-Bounded Occurrences, and Exact Hitting Set with d-Bounded Occurrences have no kernels of size $O(k^{d-3-\varepsilon})$ for any $\varepsilon > 0$.
- K_d Packing and Induced $K_{1,d}$ Packing have no kernels of size $O(k^{d-4-\varepsilon})$ for any $\varepsilon > 0$.
- d-Red-Blue Dominating Set and d-Steiner Tree have no kernels of sizes $O(k^{d-3-\varepsilon})$ and $O(k^{d-4-\varepsilon})$, respectively, for any $\varepsilon > 0$.

Our results give a negative answer to an open question raised by Dom, Lokshtanov, and Saurabh [ICALP2009] regarding the existence of *uniform polynomial kernels* for the problems above. All our lower bounds transfer automatically to compression lower bounds, a notion defined by Harnik and Naor [SICOMP2010] to study the compressibility of NP instances with cryptographic applications. We believe weak composition can be used to obtain polynomial kernelization lower bounds for other interesting parameterized problems.

In the last part of the paper we strengthen previously known super-polynomial kernelization lower bounds to super-quasi-polynomial lower bounds, by showing that quasi-polynomial kernels for compositional NP-hard parameterized problems implies the collapse of the exponential hierarchy. These bounds hold even the kernelization algorithms are allowed to run in quasi-polynomial time.

†Work done during the research visit at MPII.

1 Introduction

In parameterized complexity [12], a kernelization algorithm for a parameterized problem $L \subseteq \{0,1\}^* \times \mathbb{N}$ is a polynomial time algorithm that transforms a given instance $(x,k) \in \{0,1\}^* \times \mathbb{N}$ to an instance $(x',k') \in \{0,1\}^* \times \mathbb{N}$ such that:

- $(x,k) \in L \iff (x',k') \in L$, and
- $|x'| + k' \leq f(k)$ for some arbitrary function f.

In other words, a kernelization algorithm (or kernel) is a polynomial-time reduction from a problem onto itself that compresses the problem instance to a size depending only on the parameter. Appropriately, the function f above is called the *size* of the kernel. It is customary in many cases to not insist on the kernelization to be a reduction from a problem onto itself, but rather to allow the reduction to be between two different problems. This has been referred to as bikernelization in [2]. In this present paper, we will not distinguish between the two notions.

Kernelization is *the* central technique in parameterized complexity. Not only is it one of the most successful techniques for showing that a problem is fixed-parameter tractable, it also provides an equivalent way of defining fixed-parameter tractability: a parameterized problem is solvable in $f(k) \cdot n^{O(1)}$ time iff it has a kernel [8]. Furthermore, kernelization gives the only known mathematical framework for studying and analyzing the ancient and ubiquitous technique of preprocessing (data reduction). For these reasons, kernelization has become a research topic in its own right, with many papers on the topic appearing each year, and an annual international workshop devoted entirely to it. Notable success stories include the linear kernels for Vertex Cover [23] and Planar Dominating Set [1], a quadratic kernel for Feedback Vertex Set [25], and the meta-theorems for kernelization on bounded genus graphs [5] (see also the surveys in [3,18]).

Recently, there has been an effort in developing tools that allow showing lower-bounds for kernel sizes. This started with the work of Bodlaender *et al.* [4] which developed a machinery for showing evidence for the non-existence of polynomial size kernels. The key component of this machinery is the notion of a *composition algorithm* for parameterized problems. Roughly speaking, a composition algorithm for a parameterized problem L takes as input a sequence of instances of L, each with the same parameter value k, and outputs an instance of L with parameter bounded by $k^{O(1)}$ such that the output is a yes-instance of L iff one of the inputs is also a yes-instance. Using a lemma by Fortnow and Santhanam [15], this machinery was used to show that problems such as Path and Clique parameterized by treewidth do not have a polynomial-size kernels unless coNP $\subseteq$ NP/poly [4].

Extensions of the framework in [4] were not late to appear. Chen, Flum, and Müller [9] extended this framework to allow exclusion of kernelizations with sizes that are sublinear in the original input size, *i.e.* kernelizations of size $k^{O(1)} \cdot |x|^{1-\varepsilon}$. Following this, several new lower-bounds for kernel sizes were obtained using appropriately defined reductions called *polynomial parameter transformations*. These reductions were used to show that problems such as Leaf Out Branching [14] and Disjoint Cycles [7] do not have polynomial size kerels. Polynomial parameter transformations have since been used extensively, *e.g.* in [17,21]. Recently, Bodlaender *et al.* [6] extended the kernelization lower bounds machinery in a new direction by introducing the notion of so-called *cross composition*.

Dom *et al.* [11] took the notion of polynomial parameter transformations a step further and developed a general schema for combining these with compositions. Their schema first transforms the given problem to a colored variant, and then uses this color variant for composition by assigning IDs to the different problem instances. Using their schema, Dom *et al.* [11] were able to show that important problems such as Connected Vertex Cover and Subset Sum are unlikely to have polynomial kernels. Later their technique was used for showing several important results, including dichotomy theorems for CSP kernelization [20,22].

A common aspect of all the lower bound techniques mentioned above is that they only allow super-polynomial lower-bounds for kernel sizes. This feature has been superseded by a recent breakthrough result of Dell and van Melkebeek [10]. Dell and van Melkebeek extended the framework of [4] to a communication model, and showed using their scheme that the Vertex Cover problem does not have a kernel with $O(k^{2-\varepsilon})$ edges unless coNP $\subseteq$ NP/poly. They also showed several other kernelization lower-bounds, including an extension of the above result to a $\Omega(k^{d-\varepsilon})$ lower-bound for the d-Hitting Set problem (the Hitting Set problem restricted to families of sets of size d). However, their key technical tool for deriving these lower bounds, namely the Packing Lemma, requires sharing vertices and does not seem to extend directly to natural problems like packing and covering problems.

1.1 Our results

In this paper, we introduce a new form of composition called *weak composition*. In weak compositions, the output parameter is allowed to depend also on the length of the input sequence, and not only on the parameter. Building on the framework of of Dell and van Melkebeek [10], we show that weak compositions yield polynomial kernelization lower-bounds, as opposed to the super-polynomial lower-bounds given by the previously used compositions. We note that the composition algorithm of Dell and van Melkebeek [10] for Vertex Cover can be viewed as weak-composition where the parameter is the number of vertices in the graph.

We proceed by showing two examples of weak-composition. Specifically, we show that d-Bipartite Regular Perfect Code (d-BRPC) and d-Dimensional-Matching have weak composition, and prove both problems have no kernel of size $O(k^{d-3-\varepsilon})$ for any $\varepsilon > 0$ unless coNP $\subseteq$ NP/poly. Our construction is inspired by the composition algorithm of Dom *et al.* [11], but also differs from it quite substantially, requiring several novel ideas to make it work.

By reduction from d-BRPC, we use a variant of polynomial parameter transformations called *linear parameter transformations* to obtain new lower-bounds for several other problems, including d-Set Packing, d-Set Cover, Hitting Set with d-Bounded Occurrences, K_d Packing, and d-Steiner Tree among several others. These new lower-bounds are very close to being tight, and give a negative answer to the main open question posed in Dom *et al.* [11] regarding what they referred to as *uniform polynomial kernelizations* for the problems listed above. Furthermore, up to a polylogarithmic factor, all our lower bounds transfer automatically to compression lower bounds, an notion defined by Harnik and Naor [19] that has important cryptographic applications.

Finally, in the last part of the paper, we show that all current super-polynomial kernelization lower bounds can be extended to super-quasi-polynomial lower bounds under the assumption that the exponential hierarchy does not collapse.

1.2 Organization

The remainder of this paper is organized as follows. In Section 2 we introduce our modified notion, namely *weak composition*, and prove that it allows obtaining polynomial lower-bounds for kernelization. Section 3 then presents the main composition algorithm for d-BRPC, Section 4 presents a weak composition for d-Dimensional-Matching, and Section 5 presents our remaining kernelization lower-bound results. In Section 6 we discuss quasi-polynomial kernelization lower bounds, and in Section 7 we conclude the paper.

2 Kernelization Lower Bounds Framework

In this section we present our extended framework for proving our kernelization lower bounds. In particular, we introduce the notions of *weak compositions* and *linear parametric transformations*.

2.1 The Dell and van Melkebeek framework

We begin by first discussing the communication framework presented by Dell and van Melkebeek. All definitions and results in this section are taken from [10].

Definition 1 (oracle communication protocol). *An oracle communication protocol for a (unparameterized) language $L \subseteq \{0,1\}^*$ is a communication protocol between two players. The first player is given the input $x \in \{0,1\}^*$ and is allowed to run polynomial-time with respect to $|x|$; the second player is computationally unbounded but is not given any part of x. At the end of the protocol the first player should be able to decide whether $x \in L$. The* cost of the protocol *is* the number of bits *of communication* from first player to the second player*.*

For a language $L \subseteq \{0,1\}^*$, we let $\mathsf{OR}_{n,t}(L)$ denote the language

$$\mathsf{OR}_{n,t}(L) := \{\langle x_1, x_2, \ldots, x_t\rangle \ : \ |x_i| = n \text{ for all } i, \text{ and } x_i \in L \text{ for some } i\}.$$

We next introduce the so-called Complementary Witness Lemma that forms the basis of the framework of Dell and van Melkebeek. The proof of the lemma closely follows the arguments given by Fortnow and Santhanam in [15].

Lemma 1 (Complementary Witness Lemma). *Let $L \subseteq \{0,1\}^*$ be a language and $t : \mathbb{N} \to \mathbb{N}\backslash\{0\}$ be polynomially bounded. If there is an oracle communication protocol that decides $\mathsf{OR}_{n,t(n)}(L)$ with cost $O(t(n)\log t(n))$, then $L \in \text{coNP/poly}$. This holds even when the first player runs in* conondeterministic polynomial time*.*

The following lemma gives the connection between oracle communication protocols for classical problems and kernels for parameterized problems. For a parameterized problem $L \subseteq \{0,1\}^* \times \mathbb{N}$, we let $\widetilde{L} := \{x\#1^k : (x,k) \in L\}$ denote the unparameterized version of L.

Lemma 2. *If $L \subseteq \{0,1\}^* \times \mathbb{N}$ has a kernel of size $f(k)$, then $\widetilde{L}$ has an oracle communication protocol of cost $f(k)$.*

2.2 Our modified framework

One of the main components of the kernelization lower bounds engine of Bodlaender *et al.* [4] is the notion of a composition algorithm for a parameterized problem. This notion has been extended to the notion of a cross-composition in [6]. However, both compositions and cross compositions are suitable for showing super-polynomial lower-bounds. Below we introduce a new variant of compositions that allow showing polynomial lower-bounds.

Definition 2 (weak d-composition). *Let $d \geq 2$ be a constant, and let $L_1, L_2 \subseteq \{0,1\}^* \times \mathbb{N}$ be two parameterized problems. A* weak d-composition *from L_1 to L_2 is an algorithm $\mathbb{A}$ that on input $(x_1,k), \ldots, (x_t,k) \in \{0,1\}^* \times \mathbb{N}$, outputs an instance $(y,k') \in \{0,1\}^* \times \mathbb{N}$ such that:*

- *$\mathbb{A}$ runs in conondeterministic polynomial time with respect to $\sum_i(|x_i| + k)$.*
- *$(y,k') \in L_2 \iff (x_i,k) \in L_1$ for some i, and*

– $k' \leq t^{1/d} k^{O(1)}$.

Note that in the regular compositions the output parameter is required to be polynomially bounded by the input parameter, while in d-compositions it is also allowed to depend on the number of inputs t.

Lemma 3. *Let $d \geq 2$ be a constant, and let $L_1, L_2 \subseteq \{0,1\}^* \times \mathbb{N}$ be two parameterized problems such that $\widetilde{L_1}$ is* NP*-hard. Also assume* NP $\not\subseteq$ coNP/poly*. A weak-d-composition from L_1 to L_2 implies that L_2 has no kernel of size $O(k^{d-\varepsilon})$ for all $\varepsilon > 0$.*

Proof. Assume for the sake of contradiction that L_2 has a kernel of size $O(k^{d-\varepsilon})$ for some $\varepsilon > 0$. By Lemma 2 this implies that $\widetilde{L_2}$ has a communication protocol of cost $O(k^{d-\epsilon})$. We show that this yields a low cost oracle communication protocol for $\mathsf{OR}_{n,t(n)}(\widetilde{L_1})$ for some polynomial t. Because $\widetilde{L_1}$ is assumed to be NP-hard, this results in a contradiction to the assumption that NP $\not\subseteq$ coNP/poly by applying the Complementary Witness Lemma.

Consider a sequence of $\widetilde{L_1}$ instances $(\tilde{x}_1, \dots, \tilde{x}_t)$ with $|\tilde{x}_i| = n$ and $t := t(n)$, where t is some sufficiently large polynomial. Let the corresponding parameterized problem sequence be $((x_1, k_1), \dots, (x_t, k_t))$. The low cost protocol proceeds as follows:

1. Divide the parameterized problem sequence into *subsequences*, where each subsequence consists of instances with equal parameter values. Clearly there are at most $k := \max_i k_i \leq n$ subsequences.
2. For each subsequence, apply the d-composition from L_1 to L_2. This results in at most n instances of L_2, each with parameter bounded by $k' \leq t^{1/d} k^{O(1)}$.
3. For each instance of L_2, apply the assumed $O(k'^{(d-\epsilon)})$ protocol to decide it. If one of the composed instances is a YES instance, then accept, otherwise reject.

It is clear that the protocol has cost $O(n \cdot k'^{(d-\epsilon)})$, plug in that $k' \leq t^{1/d} k^c$ for some $c > 0$, and write $t = t(n)$. We have:

$$
\begin{aligned}
O(n \cdot k'^{(d-\epsilon)}) &= O(n \cdot (t^{1/d} k^c)^{d-\epsilon}) \\
&= O(n \cdot t^{(1-\epsilon/d)} k^{c(d-\epsilon)}) \\
&= O(n \cdot t^{(1-\epsilon/d)} n^{c(d-\epsilon)}) \quad \text{(as } k \leq n\text{)} \\
&= O(n^{1+cd-c\epsilon} \cdot t^{(1-\epsilon/d)}) \\
&= O(t) \quad \text{(since } t \text{ is sufficiently large)} \\
&= O(t \log t).
\end{aligned}
$$

By the Complementary Witness Lemma it follows that $\widetilde{L_1} \in$ coNP/poly, causing the desired contradiction. □

2.3 Linear parametric transformations

Bodlaender *et al.* [7] introduced the notion of *polynomial parametric transformations* to obtain new kernelization lower-bound results from existing ones. However these type of reductions are suitable for super-polynomial lower-bounds. Here we introduce the notion of *linear parametric transformations* that facilitate polynomial lower-bounds.

Definition 3 (linear parametric transformation). *Let L_1 and L_2 be two parameterized problems. We say that L_1 is* linear parameter reducible *to L_2, written $L_1 \leq_{ltp} L_2$, if there exists a polynomial time computable function $f : \{0,1\}^* \times \mathbb{N} \to \{0,1\}^* \times \mathbb{N}$, such that for all $(x, k) \in \Sigma^* \times \mathbb{N}$, if $(x', k') = f(x, k)$ then:*

- *$(x, k) \in L_1 \iff (x', k') \in L_2$, and*
- *$k' = O(k)$.*

The function f is called linear parameter transformation*.*

Lemma 4. *Let L_1 and L_2 be two parameterized problems, and let $d \in \mathbb{N}$ be some constant. If $L_1 \leq_{lpt} L_2$ and L_2 has a kernel of size $O(k^d)$, then L_2 also has a kernel of size $O(k^d)$.*

Proof. Composing the linear parametric transformation from L_1 to L_2 with the kernel of size $O(k^d)$ of L_2, gives an $O(k^d)$-size kernel for L_1. □

The application of Lemma 4 above is to obtain a polynomial lower-bound for any kernelization of L_2, assuming we already know a similar lower-bound for L_1. In Section 5 we will see several applications of this lemma. There we will use implicitly the easily seen fact that $\leq_{lpt}$ is transitive.

3 Main Composition Algorithm

In this section we present our main weak d-composition algorithm from which we will derive all of our kernelization lower-bound results. Throughout this section, we let d be some fixed integer with $d \geq 3$.

Our weak d-composition algorithm will be for the d-Bounded Regular Perfect Code (d-BRPC) problem. In this problem, we are given a bipartite graph $G := (N \uplus T, E)$ along with a parameter k, such that the degree of each vertex in N is exactly d. The set N is called the set of *non-terminal vertices* and the set T is referred to as the set of *terminal vertices*. The goal is to find a subset of non-terminal vertices $N' \subseteq N$ of size k such that each terminal vertex in T has exactly one neighbor in N'. For a solution set $N' \subseteq N$, we say that $v \in N'$ *dominates* $u \in T$ if $\{u, v\} \in E(G)$. The main result of this section is stated in the following theorem.

Theorem 1. *Unless* NP $\subseteq$ coNP/poly*, the d-*BRPC *problem has no kernel of size $O(k^{d-3-\varepsilon})$ for any $\varepsilon > 0$.*

We mention that the d-BRPC problem is one of the central problems used by Dom *et al.* in [11] for obtaining their super-polynomial kernelization lower-bound results. Indeed, the construction we present in this section is very much inspired by the construction in [11], but it also differs from it quite substantially in order to confirm with all requirements of a d-composition (Definition 2).

To prove Theorem 1, we will be working with a colored variant of d-BRPC called Colored d-Bipartite Regular Perfect Code (Col-d-BRPC), where the input is appended by a surjective color function $col : N \to \{1, \ldots, k\}$, and the goal is to find a solution $N' \subseteq N$ that consists of exactly one vertex of each color. Our d-composition will be from Col-3-BRPC to $(d+3)$-BRPC. Overall, our construction proceeds in two stages:

- In the first step we will compose to an instance of Bipartite Perfect Code (BPC); that is, to an instance where the vertices of N do not all have degree $d+3$, but a few of them have high degree (actually degree k).
- In the second step, we will split the vertices of high degree into many vertices of degree $d+3$, using an *equality gadget* that preserves the correctness of our construction.

For ease of notation, we will assume that our composition algorithm is given a sequence of $m = t^d/d!$ instances with parameter k, and the goal is to output a single instance with parameter bounded by $t \cdot k^{O(1)}$. We can assume that $k > d$, since otherwise all instances can be solved in polynomial-time, and a trivial instance of size $O(1)$ can be used as output. We will also assume that $k \equiv 0 \pmod{d+3}$ (and justify this assumption later on).

3.1 First step of the composition

Let $(G_1, col_1, k), \ldots, (G_m, col_m, k)$ be the input sequence of Col-3-BRPC instances, where $m = t^d/d!$ and $G_i = (N_i \uplus T_i, E_i)$. Observe that if $|T_i| \neq 3k$ for some i, then $(G_i, k) \notin$ Col-3-BRPC, and so we can assume that $|T_i| = 3k$ for all i. For $i \in \{1, \ldots, t\}$, we let $T_i = \{u^i_1, \ldots, u^i_{3k}\}$ and $N_i = \{v^i_1, \ldots, v^i_{n_i}\}$. We will use $G = (N \uplus T, E)$ and k' to denote the instance of BPC which is the output of our composition. The set of terminal vertices will consist of $k+1$ terminal components $T = T' \cup W_1 \cup \cdots \cup W_k$ and the set of non-terminals will consist of all sets of non-terminals N_i, in addition to another set X; that is, $N = (\bigcup_i N_i) \cup X$. We proceed in describing each of these terminal and non-terminal components in detail.

- The set T' consists of $3k$ vertices $\{u_1, \ldots, u_{3k}\}$. These are connected to the nonterminals in N_i, $1 \leq i \leq m$, in a way that matches the adjacency between the terminals and non-terminals in G_i. That is, $\{u_\alpha, v^i_\beta\} \in E(G) \iff \{u^i_\alpha, v^i_\beta\} \in E(G_i)$.
- For each $i \in \{1, \ldots, m\}$, we assign to N_i a *unique identifier* $\mathsf{ID}_i \subseteq \{1, \ldots, t+d\}$ with $|\mathsf{ID}_i| = d$. This is possible since $\binom{t+d}{d} > t^d/d! = m$.
- The set X of non-terminals consists of $t+d$ vertices, and we write $X = \{x_1, \ldots, x_{t+d}\}$.
- For each $j \in \{1, \ldots, k\}$, the set W_j consists of $t+d$ vertices, and we write $W_j = \{w^j_1, \ldots, w^j_{t+d}\}$.
- For each $i \in \{1, \ldots, m\}$ and $j \in \{1, \ldots, k\}$, we add edges between the nonterminal component N_i and the terminal component W_j as follows: For each vertex $v \in N_i$ with $col_i(v) = j$, we connect v to all vertices in W_j that have indices belonging to ID_i; that is, we add the edge $\{v, w^j_\ell\}$ to $E(G)$ for all $\ell \in \mathsf{ID}_i$.
- For each $\ell \in \{1, \ldots, t+d\}$ and $j \in \{1, \ldots, k\}$, add the edge $\{x_\ell, w^j_\ell\}$ to $E(G)$.
- Set $k' = k + t$.

This completes the construction of the first stage (see Fig. 1). It is clear that it can be carried out in polynomial time. The general idea is that the selection of t vertices from X encodes the selection of an ID which uniquely identifies some non-terminal component N_i. The terminal sets $W_1, \ldots, W_k$ then enforce that the remaining k vertices of the solution will be selected only from a single N_i. The next lemma makes this more precise, and proves the correctness of the first step of our construction.

Lemma 5. $(G, k') \in$ BPC $\iff$ $(G_i, k) \in$ Col-3-BRPC *for some* $i \in \{1, \ldots, m\}$.

Proof. $(\Leftarrow)$ This is the easy direction. Suppose $(G_i, k) \in$ Col-3-BRPC for some $i \in \{1, \ldots, m\}$, and let $N'_i \subseteq N_i$ be a solution of size k. We take $N' = \{v_j \in N : v^i_j \in N'_i\}$ and $X' = \{x_j \in X : j \in \overline{\mathsf{ID}_i}\}$ to be our solution for (G, k'), where $\overline{\mathsf{ID}_i} = \{1, \ldots, t+d\} \setminus \mathsf{ID}_i$. Observe that $|N' \cup X'| = k + t = k'$. Furthermore, each vertex in T' is dominated by exactly one vertex in N', by definition of N'_i and by our construction. Also, for each $j \in \{1, \ldots, k\}$, a vertex w^j_ℓ is dominated by exactly one vertex in N' in case $\ell \in \mathsf{ID}_i$ (the vertex corresponding to the vertex in N'_i with color j), and dominated by exactly one vertex in X' if $\ell \notin \mathsf{ID}_i$.

$(\Rightarrow)$ This is the more interesting direction. Let S denote a solution for (G, k') with $|S| = k' = k+t$. The first observation is that, because the terminal component T' is only connected to $N_1, \ldots, N_m$

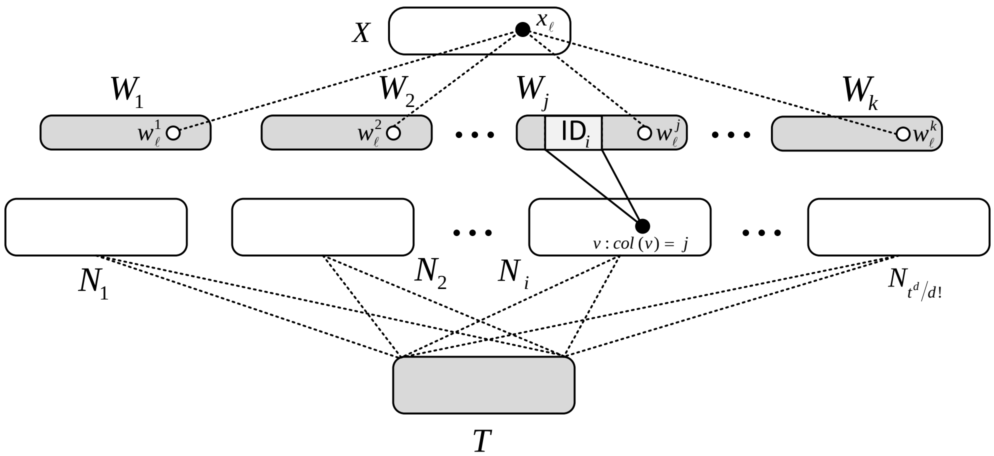

Fig. 1. A graphical description of the construction in the first step. The white boxes represent components of terminal vertices, the gray boxes represent components of non-terminal vertices.

but not to X, and has size exactly $3k$, any solution for (G, k') has to pick exactly k vertices from $N_1, \ldots, N_m$. This implies that S contains precisely t vertices from X, since $k' = k+t$. Let $X' \subseteq S \cap X$ denote this set of t vertices, and let $N' = S \setminus X'$. Since $|X'| = t$, we know that N' includes vertices from k different colors (in their Col-3-BRPC instances), because if color $j \in \{1, \ldots, k\}$ is not present, some vertices in W_j will not be dominated. Write $\overline{\mathsf{ID}} = \{\ell \in \{1, \ldots, t+d\} : x_\ell \in X'\}$, and let $\mathsf{ID} = \{1, \ldots, t+d\} \setminus \overline{\mathsf{ID}}$. Observe that $|\overline{\mathsf{ID}}| = t$ and $|\mathsf{ID}| = d$.

We argue that ID must equal some ID_i for some $i \in \{1, \ldots, m\}$. To see this, assume for contradiction that $\mathsf{ID} \neq \mathsf{ID}_i$ for all $i \in \{1, \ldots, m\}$. Consider a vertex $v \in N'$, and suppose $v \in N_i$. Let $j = col_i(v)$. Recall that the set of neighbors of v in W_j is precisely $\{w^j_\ell \in W_j : \ell \in \mathsf{ID}_i\}$. Now as $\mathsf{ID} \neq \mathsf{ID}_i$, it must be that $\overline{\mathsf{ID}} \cup \mathsf{ID}_i \neq \{1, \ldots, t+d\}$; that is, there is some $\ell^* \in \{1, \ldots, t+d\} \setminus (\overline{\mathsf{ID}} \cup \mathsf{ID}_i)$. But then, by our construction, S does not dominate $w^j_{\ell^*}$, a contradiction.

Thus $\mathsf{ID} = \mathsf{ID}_i$ for some $i \in \{1, \ldots, m\}$. We argue next that $N' \subseteq N_i$. Assume for contradiction that this is not the case; that is, there is some $v \in N' \cap N_{i^*}$ for $i^* \neq i$. Let $j = col_{i^*}(v)$. The set of neighbors of v in W_j is $\{w^j_\ell \in W_j : \ell \in \mathsf{ID}_{i^*}\}$. Since $\mathsf{ID} = \mathsf{ID}_i \neq \mathsf{ID}_{i^*}$, there is some $\ell^* \in \{1, \ldots, t+d\} \setminus (\overline{\mathsf{ID}} \cup \mathsf{ID}_{i^*})$, and S does not dominate $w^j_{\ell^*}$. We have therefore established that $N' \subseteq N_i$. Since N' dominates all vertices in T', and $|N'| = k$, it follows that N' is also a solution for (G_i, k). Thus, $(G_i, k) \in$ Col-3-BRPC, and the lemma follows. □

3.2 Second step of the composition

We next alter the output instance $(G, k') = ((N \uplus T, E), k')$ of the composition algorithm in the previous section so that it becomes an instance of $(d+3)$-BRPC. That is, we create an instance $(G^*, k^*) = ((N^* \uplus T^*, E^*), k^*)$ where all non-terminal vertices in N^* have degree $d+3$, and $(G^*, k^*) \in (d+3)$-BRPC $\iff$ $(G, k') \in$ BPC. Initially we will start with $G^* = G$, and then we modify G^* so that it fits our requirements. Note that we require all non-terminals in N^* to have degree exactly $d+3$, and not merely a degree bounded by $d+3$. This actually introduces some complications, but will prove useful in showing our other kernelization lower-bounds in Section 5.

Recall that the set of non-terminals in the BPC instance of the previous section is composed of several components, *i.e.* $N = (\bigcup_{i \in \{1,\ldots,m\}} N_i) \cup X$. Observe that the degree of each non-terminal vertex $v \in \bigcup_i N_i$ is precisely $d+3$, and that the degree of each non-terminal vertex $x \in X$ is precisely k. Thus, we only need to fix the degree of vertices in $X = \{x_1, \ldots, x_{t+d}\}$. The goal of these vertices is to encode the selection of an ID which identifies some non-terminal component N_i. This ID is then verified in the k different terminal components $W_1, \ldots, W_k$. For this reason, the naive approach of splitting the vertices in X to vertices of bounded degree might result in the selection of k different ID's. In the following we introduce an *equality* gadget that enforces the selection k ID's which are actually the same.

Let $\ell \in \{1, \ldots, t+d\}$, and consider $x_\ell \in X$. Recall that we assume that $k \equiv 0 \pmod{d+3}$. We replace x_ℓ with k vertices $x_1^\ell, \ldots, x_k^\ell$ in N^*, and we add the edges $\{x_j^\ell, w_\ell^j\}$ to E^*. We then add to N^* a set of additional non-terminals $\{y_1^\ell, \ldots, y_{k-1}^\ell\}$. Each one of these new non-terminal vertices will be connected to a distinct set of $d+2$ new terminal vertices. This gives us $k-1$ disjoint sets of new terminals, $Z_1^\ell, \ldots, Z_{k-1}^\ell$, with $|Z_j^\ell| = d+2$. Now we connect x_j^ℓ to the first 2 vertices of Z_j^ℓ, and the last d vertices of Z_{j-1}^ℓ, for all $j \in \{2, \ldots, k-1\}$. We also connect x_1^ℓ to the first 2 vertices of Z_1^ℓ, and x_k^ℓ to the last d vertices of Z_{k-1}^ℓ. (See Fig. 2 for a graphical depiction of this construction.)

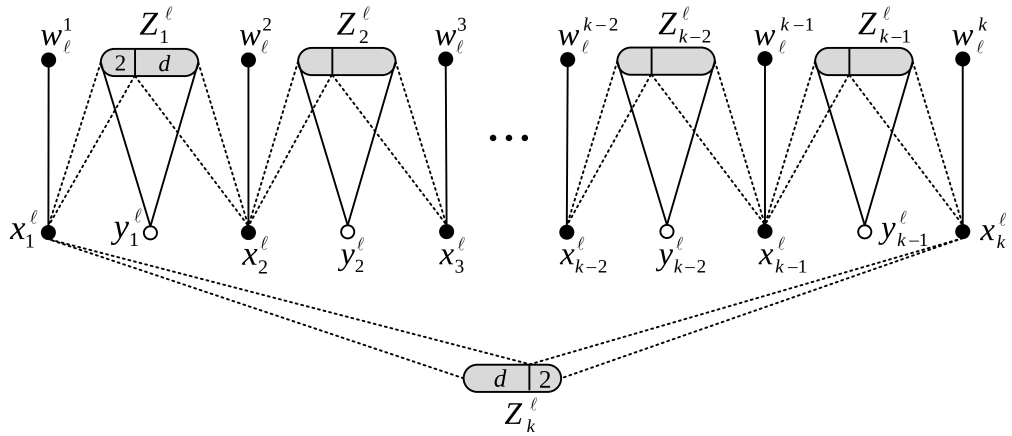

Fig. 2. A graphical description of the main part of the equality gadget used to replace x_ℓ.

Note that all for each $\ell \in \{1, \ldots, t+d\}$, the non-terminal vertices $\{x_2^\ell, \ldots, x_{k-1}^\ell\}$ have degree $d+3$ as required. Vertex x_1^ℓ has degree 3, x_k^ℓ has degree $d+1$, and all non-terminals $\{y_1^\ell, \ldots, y_{k-1}^\ell\}$ have degree $d+2$. We next add some additional terminals so that all non-terminals have degree $d+3$. First we add a new set of terminals Z_k^ℓ of size $d+2$. We connect x_1^ℓ to the first d terminals of this set, and x_k^ℓ to the last 2 terminals. We also connect the non-terminals $y_1^\ell, \ldots, y_{d+2}^\ell$ to Z_k^ℓ by a perfect matching. This fixes the degree of x_1^ℓ, x_k^ℓ, and $\{y_1^\ell, \ldots, y_{d+2}^\ell\}$. To fix the remaining non-terminals, we add $p = (k-d-3)/(d+3)$ new disjoint sets of terminals, $Z_{k+1}^\ell, \ldots, Z_{k+p}^\ell$, each of size $d+3$. Note that p is in fact an integer since we assume $k > d$ and $k \equiv 0 \pmod{d+3}$. We then add p new non-terminal vertices, $x_{k+1}^\ell, \ldots, x_{k+p}^\ell$, and connect x_{k+i}^ℓ to all vertices in Z_{k+i}^ℓ, for $i \in \{1, \ldots, p\}$. Finally, we group the the non-terminals $\{y_{d+3}^\ell, \ldots, y_{k-1}^\ell\}$ into p groups of size $d+3$ each, and connect group i, $1 \le i \le p$, to Z_{k+i}^ℓ by a perfect matching.

We do the above for each $\ell \in \{1, \ldots, t+d\}$. This gives us our graph $G^* = (N^* \uplus T^*, E^*)$. It is easy to see that all non-terminals in G^* have degree $d+3$, and that constructing G^* can be done in polynomial-time. Observe that the size of $\bigcup_{\ell\in\{1,\dots,t+d\}}(\{w_\ell^1,\dots,w_\ell^k\}\cup\bigcup_{i\in\{1,\dots,k+p\}}Z_i^\ell)$ or equivalently the total number of terminal vertices except those in T', is:

$$\begin{aligned}(t+d)(k+(d+2)k+(d+3)p) &= (t+d)((d+3)k+(d+3)p)\\ &= (t+d)(d+3)(k+p).\end{aligned}$$

To conclude our construction we set $k^*=k+t(k+p)+d(k-1)$. The next two lemmas prove the correctness of our construction.

Lemma 6. *Let $S\subseteq N^*$ be any solution for (G^*,k^*). For each $\ell\in\{1,\dots,t+d\}$, exactly one of the following cases occur:*

- $\{x_1^\ell,\dots,x_{k+p}^\ell\}\subseteq S$ *and* $\{y_1^\ell,\dots,y_{k-1}^\ell\}\cap S=\emptyset$.
- $\{y_1^\ell,\dots,y_{k-1}^\ell\}\subseteq S$ *and* $\{x_1^\ell,\dots,x_{k+p}^\ell\}\cap S=\emptyset$.

Proof. Let $\ell\in\{1,\dots,t+d\}$, and consider some $i\in\{1,\dots,k-1\}$. Clearly, either $x_i^\ell,x_{i+1}^\ell\in S$ or $y_i^\ell\in S$, since otherwise S would not dominate the terminals in Z_i^ℓ. Furthermore, if $\{x_i^\ell,y_i^\ell,x_{i+1}^\ell\}\subseteq S$ then vertices of Z_i^ℓ would have two neighbors in S, contradicting the fact that S is indeed a solution. From this it follows that either

- $\{x_1^\ell,\dots,x_k^\ell\}\subseteq S$ and $\{y_1^\ell,\dots,y_{k-1}^\ell\}\cap S=\emptyset$.
- $\{y_1^\ell,\dots,y_{k-1}^\ell\}\subseteq S$ and $\{x_1^\ell,\dots,x_k^\ell\}\cap S=\emptyset$.

Furthermore, in the latter case, we must have that $\{x_{k+1}^\ell,\dots,x_{k+p}^\ell\}\cap S=\emptyset$ since otherwise some of the terminals in $Z_k^\ell,\dots,Z_{k+p}^\ell$ would have more than one neighbor in S. In the former case, we it must be that $\{x_{k+1}^\ell,\dots,x_{k+p}^\ell\}\subseteq S$ since otherwise some of the terminals in $Z_k^\ell,\dots,Z_{k+p}^\ell$ would not be dominated by any vertex in S. The lemma follows. $\square$

Lemma 7. $(G,k')\in$ BPC $\iff$ $(G^*,k^*)\in(d+3)$-BRPC.

Proof. ($\Rightarrow$) Suppose S is a solution for (G,k'). Then as argued in Lemma 5, S consists of a subset k vertices $N'\subseteq N_i$, for some $i\in\{1,\dots,m\}$, and a subset of t vertices $X'\subseteq X$. It is not difficult to verify that

$$S^*=N'\cup\{x_1^\ell,\dots,x_{k+p}^\ell : x_\ell\in X'\}\cup\{y_1^\ell,\dots,y_{k-1}^\ell : x_\ell\notin X'\}$$

is a solution for (G^*,k^*).

($\Leftarrow$) Assume that S^* is a solution for (G^*,k^*), and let $N'=S^*\cap(\bigcup_{i\in\{1,\dots,m\}}N_i)$ and $S'=S^*\setminus N'$. Since $|T'|=kd$ and the degree of each non-terminal vertex is d, we must have $|N'|=k$, which implies that $|S'|=k^*-k=t(k+p)+d(k-1)$. Observe that for any vertex $v\in\bigcup_{i\in\{1,\dots,m\}}N_i$, its number of neighbors in $\bigcup_{j\in\{1,\dots k\}}W_j$ is precisely d, hence N' can dominate at most kd vertices in $\bigcup_{j\in\{1,\dots k\}}W_j$. Therefore the number of terminal vertices S' dominates is at least $(t+d)(d+3)(k+p)-kd$.

By Lemma 6, we get that for each $\ell\in\{1,\dots,t+d\}$, either $\{x_1^\ell,\dots,x_{k+p}^\ell\}\subseteq S'$ or $\{y_1^\ell,\dots,y_{k-1}^\ell\}\subseteq S'$, and if one set is contained in S', the other must be completely disjoint from S'. Let $\overline{\mathsf{ID}}=\{\ell : \{x_1^\ell,\dots,x_{k+p}^\ell\}\subseteq S'\}$. Observe that if $\ell\in\overline{\mathsf{ID}}$, then all the terminals in $\{w_\ell^1,\dots,w_\ell^k\}$ are dominated, and otherwise none of them are dominated.

Let $k_1=|\{w_\ell^1,\dots,w_\ell^k\}\cup(\bigcup_{i\in\{1,\dots,k+p\}}Z_i^\ell)|=k+(d+2)k+(d+3)p$ and $k_2=|\bigcup_{i\in\{1,\dots,k+p\}}Z_i^\ell|=(d+2)k+(d+3)p$.

We have:

$$\begin{aligned} k_1|\overline{\mathsf{ID}}| + k_2(t+d-|\overline{\mathsf{ID}}|) &= k|\overline{\mathsf{ID}}| + ((d+2)k + (d+3)p)(t+d) \\ &= k|\overline{\mathsf{ID}}| + ((d+3)(k+p) - k)(t+d) \\ &= k|\overline{\mathsf{ID}}| + (t+d)(d+3)(k+p) - (t+d)k. \end{aligned}$$

This number must be at least $(t+d)(d+3)(k+p) - kd$, which means that $|\overline{\mathsf{ID}}| \geq t$.

We next argue that $|\overline{\mathsf{ID}}| \leq t$. Assume for the sake of contradiction that this is not the case, then by construction, for some subset $H \subseteq \{1,\dots,t+d\}$ of size at least $(t+1)$, we dominate $\{w^j_\ell : \ell \in H\}$ for $j = 1,\dots,k$. Consider any vertex $v \in N'$, and suppose it connects to W_j for some $j \in \{1,\dots,k\}$. The number of neighbors v has in $\{w^j_1,\dots,w^j_{t+d}\}$ is d, and so some terminal in $\{w^q_1,\dots,w^q_{t+d}\}$ must be dominated twice, a contradiction. It follows that $|\overline{\mathsf{ID}}| = t$, and so $S' = N \cup \{x_\ell : \ell \in \overline{\mathsf{ID}}\}$ is a solution for (G,k'). □

3.3 Proof of Theorem 1

We are now in position to complete the proof of Theorem 1. We begin with the following lemma.

Lemma 8. *Let d be a fixed positive integer. The* Col-3-BRPC *problem restricted to the case where the solution size k satisfies $k \equiv 0 \pmod d$ is* NP*-hard.*

Proof. We first show that 3-BRPC is NP-hard even when restricted to the case with $k \equiv 0 \pmod d$. This is done by a reduction from the 3-Dimensional Matching problem which is well known to be NP-complete [16]. In 3-Dimensional Matching, we are given 3 disjoint sets A, B, and C, each of size k, and a set $M \subseteq A \times B \times C$. The question is whether there exists a subset $M' \subseteq M$ of size k which is pairwise disjoint. By padding k until $k \equiv 0 \pmod d$ and padding M, we have that 3-Dimensional Matching restricted to the case that $k \equiv 0 \pmod d$ is NP-complete. 3-Dimensional Matching can easily be reduced to 3-BRPC problem by letting $A \cup B \cup C$ be the set of terminals, and each set $S \in M$ be the neighborhood of a nonterminal vertex. Using next the reduction in Dom *et al.* [11] from 3-BRPC to Col-3-BRPC that preserves the solution size completes the proof of the lemma. □

Proof (of Theorem 1). Let $d' = d+3$, and let $t' = m = t^d/d! = t^{d'-3}/(d'-3)!$. The composition algorithm presented above composes t' Col-3-BRPC instances with parameter k such that $k \equiv 0 \pmod{d'}$ to a d'-BRPC instance with parameter $k^* = O(kt) = O(k(t'd!)^{1/d}) = O(t'^{1/(d'-3)}k)$. Thus, our composition is in fact a weak $(d'-3)$-composition from Col-3-BRPC to d'-BRPC. Since Col-3-BRPC is NP-hard even when $k \equiv 0 \pmod{d'}$ (Lemma 8), applying Lemma 3 shows that d-BRPC has no kernel of size $O(k^{d-3-\varepsilon})$, for any $\varepsilon > 0$, unless coNP $\subseteq$ NP/poly. □

4 Composition for *d*-Dimensional Matching

In this section we present another example of weak d-composition. Specifically, we derive polynomial kernelization lower bound for d-Dimensional Matching (d-DM) using weak composition. In d-DM, we are given a set $S \subseteq \mathcal{A} = A_1 \times \cdots \times A_d$ for some collection $A_1,\dots,A_d$ of pair-wise disjoint sets. The parameter is a positive integer k. The question is whether there is a subset $P \subseteq S$ of size k that are pairwise disjoint. The d-DM problem is a natural generalization of maximum matching in bipartite graphs to high dimensions, and is known to be NP-hard for every $d \geq 3$ [16].

Theorem 2. *Unless* NP $\subseteq$ coNP/poly*,* d-DM *has no kernel of size* $O(k^{d-3-\varepsilon})$ *for any* $\varepsilon > 0$*.*

To prove this theorem we will be working with a variant of d-DM called Colored d-Dimensional Perfect Matching (Col-d-PDM), where the input is appended by a surjective color function $col : S \mapsto \{1, \ldots, k\}$, and each dimension A_i, $i \in \{1, \ldots, d\}$, has exactly k elements. The goal is to find a solution $P \subseteq S$ that sets in P has distinct colors. Note each such solution corresponds to a perfect matching of elements across different dimensions. The Col-d-PDMproblem is also known to be NP-hard for every $d \geq 3$ [16]. Our d-composition will be from Col-3-PDM to $(d+3)$-DM. Similar to the construction for BRPC, the construction for $(d+3)$-DM has two steps. In the first step we will compose to an instance of Set Packing, where the sets can have arbitrary size and are allowed to include more than one element from the same A_i. In the second step, we will transform the Set Packing instance to a $(d+3)$-DM instance. First, we will split and pad the sets of high cardinality into many sets of cardinality exactly $(d+3)$ which include exactly one set of each dimension. Again, we use an equality gadget to preserve the correctness of our construction.

Our composition algorithm will compose a sequence of t^d instances with parameter k, and its output will be a single instance with parameter bounded by $t \cdot k^{O(1)}$. Again, we assume that $k > d$, since otherwise all instances can be solved in polynomial-time by brute force, and a trivial instance of size $O(1)$ can be used as output.

4.1 First step

In this step we will compose to the Set Packing problem. Consider input sequence $\{I_\ell = (S_\ell \subseteq X_\ell \times Y_\ell \times Z_\ell, col_\ell, k) : 1 \leq \ell \leq t^d\}$ of Col-3-PDM instances, where $X_\ell = \{x^\ell_1, \ldots, x^\ell_k\}$, $Y_\ell = \{y^\ell_1, \ldots, y^\ell_k\}$, and $Z_\ell = \{z^\ell_1, \ldots, z^\ell_k\}$ are pairwise disjoint sets. We will use $I = (S, k')$ to denote the instance of Set Packing, which is the output of our composition. We proceed by describing dimensions and sets of S in detail.

- We first create pairwise-disjoint dimensions X, Y, Z of k elements each. Let $X = \{x_1, \ldots, x_k\}$, $Y = \{y_1, \ldots, y_k\}$, and $Z = \{z_1, \ldots, z_k\}$. Then for each set $R \in S_\ell$, we create the same set using elements in X, Y, Z in a way that matches R. That is $(x_a, y_b, z_c) \in S \Leftrightarrow (x^\ell_a, y^\ell_b, z^\ell_c) \in S_\ell$ for all $a, b, c \in \{1, \ldots, k\}$. In the following when we mention $R \in S_\ell$ in I, we mean the set created in S in this manner.
- Create d new dimensions, $P_1, \ldots, P_d$ where each dimension has kt new elements. Every dimension is organized as k layers, with each layer t elements. That is, for $i \in \{1, \ldots, d\}$:

$$P_i = \bigcup_{j \in \{1,\ldots,k\}} \{c^i_{j,1}, \cdots, c^i_{j,t}\}.$$

- For $\ell \in \{1, \ldots, t^d\}$, assign instance I_ℓ a unique d-tuple as its identifier ID_ℓ. This is done by picking an element from $\{1, \ldots, t\}^d$. Let $\mathsf{ID}_\ell(i)$ be the value at index $i \in \{1, \ldots, d\}$.
- For $\ell \in \{1, \ldots, t^d\}$, consider $R \in S_\ell$ of color $j \in \{1, \ldots, k\}$. We extend R by adding elements from $P_1, \ldots, P_d$, such that for $i \in \{1, \ldots, d\}$, $R(P_i) = c^i_{j,\alpha}$ where $\alpha = \mathsf{ID}_\ell(i)$. After this, every set $R \in S_\ell$ has exactly $(d+3)$ elements.
- Now we construct *gadget sets*. For $r \in \{1, \ldots, t\}$, and $i \in \{1, \ldots, d\}$, construct $W^i_r = \{c^i_{j,r} : 1 \leq j \leq k\}$. Basically W^i_r occupies every position r of each of the k layers in dimension i. It is straightforward to check that each of the td gadget sets has unbounded size k. Further the elements of W^i_r are from the same dimension P_i.
- To conclude the construction, we set $k' = k + (t-1)d$.

Lemma 9. *$I \in$ SET PACKING if and only if $I_\ell \in$ COL-3-PDM for some $\ell \in \{1, \ldots, t^d\}$.*

Proof. ($\Leftarrow$) Suppose $(S_\ell, k) \in$ COL-3-PDM for some $\ell \in \{1, \ldots, t^d\}$, and let $P_\ell \subseteq S_\ell$ be a solution of size k for (S_ℓ, k). We take P_ℓ and $X = \{W_r^i : r \neq \mathsf{ID}_\ell(i), r \in \{1, \ldots, t\}, i \in \{1, \ldots, d\}\}$ to be our solution for (S, k'). It is easy to check that all sets in X are pairwise disjoint and that $|P_\ell \cup X| = k + (t-1)d$.

($\Rightarrow$) We show that if I has a pairwise-disjoint solution of size $k + (t-1)d$, then I_ℓ has a perfect matching for some $\ell \in \{1, \ldots, t^d\}$. For this, fix one $i \in \{1, \ldots, d\}$, and consider the number of sets in the solution for I which are from $\{W_r^i : r \in \{1, \ldots, t\}\}$. Clearly, the solution can contain at most t of them. However, if we pick t of them into solution, then we *cannot* pick any set from $S_1, \ldots, S_{t^d}$ into the solution due to the disjointness constraint. This means the solution can have at most $td < k + (t-1)d$ pairwise-disjoint sets, which cannot satisfy the number of disjoint sets required by the solution.

Therefore there can be at most $(t-1)$ sets in the solution which are from $\{W_r^i : r \in \{1, \ldots, t\}\}$. This gives us at most $(t-1)d$ pairwise-disjoint sets from td gadget sets. Hence by requirement of the size of the solution, it indicates that we have to pick at least k sets from $S_1, \ldots, S_{t^d}$. The crucial observation now is that, we can pick at most k pairwise disjoint sets from $S_1, \ldots, S_{t^d}$, because $|X| = |Y| = |Z| = k$. Therefore the only possibility is that we pick k sets from $S_1, \ldots, S_{t^d}$, and pick $(t-1)d$ gadget sets, with $(t-1)$ sets from $\{W_r^i : r \in \{1, \ldots, t\}\}$ for each $i \in \{1, \ldots, d\}$.

Now the observation is that, after we fix $(t-1)$ gadget sets for each $i \in \{1, \ldots, d\}$. It leaves, for each dimension, exactly the same position in each of the k layers not used. That is there exists a set $\mathsf{ID} = \{\alpha_1, \ldots, \alpha_d\} \in \{1, \ldots, t\}^d$ such that c_{j,α_i}^i is not used for $j \in \{1, \ldots, k\}$. By the requirement of disjointness, now the k sets from $S_1, \ldots, S_{t^d}$ must come from S_ℓ where $\mathsf{ID}_\ell = \mathsf{ID}$. Further for any color $j \in \{1, \ldots, k\}$, c_{j,α_i}^i cannot be matched twice, hence these sets must have distinct colors. This gives that $I_\ell \in$COL-3-PDM, which completes the proof. ◻

4.2 Second step

Now we transform the SET PACKING instance $I = (S, k')$ into an equivalent $(d+3)$-DM instance $I^* = (S^*, k^*)$. Note the SET PACKING instance derived in the last section fails the requirement of d-DM in two ways. First, gadget sets W_r^i ($i \in \{1, \ldots, d\}, r \in \{1, \ldots, t\}$) are of size k. Second, the elements of W_r^i come from the same dimension P_i. Initially let S^* be S. Fix arbitrary W_r^i, we show how to modify W_r^i so that it is of dimension $(d+3)$ and each of its element comes from different dimension, while preserving the correctness of the construction. Recall that $W_r^i = \{c_{j,r}^i : 1 \leq j \leq k\}$.

First, we split W_r^i into $W_{1,r}^i, W_{2,r}^i, \ldots, W_{k,r}^i$, where $W_{j,r}^i = \{c_{j,r}^i\}$. In the following, when we mention elements of the form x_*^* ($*$ is the wild-card symbol), it means to extend X with this new element, similar for y_*^* and z_*^*. Now we begin to construct constraints among $\{W_{j,r}^i : j \in \{1, \ldots, k\}\}$. We use two operations, *extend* means extending an existing set with new elements, *add* means adding a new set into S^*. The gadgets are modified as follows, *extend* $W_{1,r}^i$ with $x_{k+1,r}^i$, *add* $U_{1,r}^i = (x_{k+1,r}^i, y_{k+1,r}^i)$, *extend* $W_{2,r}^i$ with $y_{k+1,r}^i, z_{k+2,r}^i$, *add* $U_{2,r}^i = (z_{k+2,r}^i, x_{k+2,r}^i)$, and so on until we extend $W_{k,r}^i$. Now it is clear that the size of all gadget sets is bounded by 3. Further the elements of every gadget set come from different dimensions. Finally to make each gadget set of dimension exactly $(d+3)$, we simply pad each gadget set. To conclude the construction we set $k^* = k + (t-1)kd + (k-1)d$. The following lemma shows that the construction preserves correctness.

Lemma 10. *$(S^*, k^*) \in (d+3)$-DM $\iff$ $(S, k') \in$ SET PACKING.*

Proof. ($\Leftarrow$) Assume we found a solution $P \subseteq S$ of $k' = k + (t-1)d$ pairwise-disjoint sets in S. As we have argued, the solution must consist of k sets from some S_ℓ for some $\ell \in \{1, \ldots, t^d\}$ with

$\mathsf{ID}_\ell = \{\alpha_1, \ldots, \alpha_d\} \in \{1, \ldots, t\}^d$, and $W_r^i \in P$ for $i \in \{1, \ldots, d\}, r \neq \alpha_i$. Hence for (S^*, k^*), we choose $W_{1,r}^i, \ldots, W_{k,r}^i$ for $r \neq \alpha_i$, and choose $U_{1,r}^i, \ldots, U_{k-1,r}^i$ for $r = \alpha_i$. This gives a solution in S^* of size $k + [(t-1)k + (k-1)] \cdot d = k + (t-1)kd + (k-1)d = k^*$.

($\Rightarrow$) Suppose there is a set of k^* pairwise-disjoint sets in S^*. Fix $i \in \{1, \ldots, d\}$ and $r \in \{1, \ldots t\}$ and consider $W_{1,r}^i, U_{1,r}^i, \ldots, U_{k-1,r}^i, W_{k,r}^i$. The first observation is that, by our construction, we can pick at most k pairwise disjoint sets from $W_{1,r}^i, U_{1,r}^i, \ldots, U_{k-1,r}^i, W_{k,r}^i$. And if we pick k of them, the k sets must be $W_{1,r}^i, W_{2,r}^i, \ldots, W_{k,r}^i$.

Now if we pick $W_{1,r}^i, W_{2,r}^i, \ldots, W_{k,r}^i$ for every $r \in \{1, \ldots, t\}$, S^* cannot contain any set from $S_1, \ldots, S_{t^d}$. This indicates that the solution size is at most $tkd < k^*$, contradiction. Therefore for every $i \in \{1, \ldots, d\}$, there are at most $k(t-1) + (k-1)$ sets in S^*. Hence S^* contains at most $[k(t-1) + (k-1)] \cdot d = (t-1)kd + (k-1)d$ from gadget sets.

Therefore we have to pick k sets from $S_1, \ldots, S_{t^d}$. Let Q be this family of k sets. Further, after fixing $W_{1,r}^i, \ldots, W_{k,r}^i$ for $(t-1)$ different r's and for every $i \in \{1, \ldots, d\}$. There exists a set $\mathsf{ID} = \{\alpha_1, \ldots, \alpha_d\} \in \{1, \ldots, t\}^d$ such that c_{j,α_i}^i is not matched for $j \in \{1, \ldots, k\}$. Now all sets of Q must come from S_ℓ with $\mathsf{ID}_\ell = \mathsf{ID}$ and of different colors. Choosing Q and $\{W_r^i : r \neq \alpha_i, i \in \{1, \ldots, d\}\}$ gives a solution in (S, k'). This completes the proof. □

4.3 Proof of Theorem 2

We are now in position to complete the proof of Theorem 2.

Proof (of Theorem 2). Let $d' = d + 3$, and let $t' = t^d = t^{d'-3}$. The composition algorithm presented above composes t' Col-3-PDM instances with parameter k to a d'-DM instance with parameter $k^* = O(kt) = O(t'^{1/d'-3}k)$. Thus, our composition is in fact a weak $(d'-3)$-composition from Col-3-PDM to d'-DM. Since Col-3-PDM is NP-hard, applying Lemma 3 shows that d-DM has no kernel of size $O(k^{d-3-\varepsilon})$, for any $\varepsilon > 0$, unless coNP $\subseteq$ NP/poly. □

5 Applications

In this section we derive polynomial lower bounds for several problems using our lower bound for d-BRPC and linear parameter transformations discussed in Section 2.3. Some of the reductions appearing in this section appeared also in [11].

5.1 Set-theoretic problems

The d-Set Packing takes as input a set system $(U, \mathcal{F})$ with each set in $\mathcal{F}$ having cardinality d, and a parameter k, and the goal is to determine whether there are k pairwise disjoint subsets in $\mathcal{F}$. The d-Set Cover problem takes the same input as d-Set Packing, and the goal is to determine whether there exists a subfamily of $\mathcal{F}$ with at most k sets whose union is U. If these sets are required to be pairwise disjoint, then the problem is known as d-Exact Set Cover. The Hitting Set with d-Bounded Occurrences problem takes as input a set system $(U, \mathcal{F})$ such that each element $u \in U$ appears in d sets of $\mathcal{F}$, and a parameter k, and the goal is to find a subset of U of size k that has non-empty intersection with each set in $\mathcal{F}$. When the size of this intersection is required to be precisely 1, we get the Exact Hitting Set with d-Bounded Occurrences problem. Observe that all these problems have a trivial kernel of size $\binom{kd}{d} = O(k^d)$ by removing identical sets. The following theorem shows that trivial kernelization cannot be substantially improved.

Theorem 3. *Unless* coNP $\subseteq$ NP/poly*, d-*Set Packing*, d-*Set Cover*, d-*Exact Set Cover*,* Hitting Set with d-Bounded Occurrences*, and* Exact Hitting Set with d-Bounded Occurrences *have no kernels of size $O(k^{d-3-\varepsilon})$ for any $\varepsilon > 0$.*

Proof. We present a linear parametric transformation from d-BRPC to all of the problems mentioned in the theorem. The theorem will then follow from Theorem 1 and Lemma 4.

Given a d-BRPC instance (G, k) with $G = (N \uplus T, E)$ and $|T| = kd$ terminals, we construct a d-Set Packing instance $(U, \mathcal{F}, k)$ as follows. We let our universe U be $U = T$. For each nonterminal $v \in N$, construct set $S_v = N(v)$ in $\mathcal{F}$, where $N(v)$ is the neighbors of v in T. Obviously each set in the family has cardinality d, and every solution for (G, k) one to one corresponds to a solution for $(U, \mathcal{F}, k)$. Thus, d-BRPC $\leq_{lpt}$ d-Set Packing.

Note that any solution for the d-Set Packing instance $(U, \mathcal{F}, k)$ constructed above is also a solution for d-Exact Set Cover with the same instance. This is because each set in $\mathcal{F}$ is of cardinality d and $|U| = kd$. Thus, and k pairwise disjoint sets in $\mathcal{F}$ must cover U. We therefore have d-BRPC $\leq_{lpt}$ d-Exact Set Cover, and since d-Exact Set Cover is special case of d-Set Cover, we also have d-BRPC $\leq_{lpt}$ d-Set Cover. Finally, using the well-known reduction (which can be viewed as linear parametric transformation) from d-Exact Set Cover to Exact Hitting Set with d-Bounded Occurrences, we get that d-BRPC $\leq_{lpt}$ Exact Hitting Set with d-Bounded Occurrences and d-BRPC $\leq_{lpt}$ Hitting Set with d-Bounded Occurrences. □

5.2 Graph-theoretic problems

In the d-Red-Blue Dominating Set problem, the input is a bipartite graph $G = (N \uplus T, E)$ with the degree of every vertex $v \in N$ at most d, and a parameter k. The goal is to determine whether there exists a subset $N' \subseteq N$ of size at most k so that every vertex in T has at least one neighbor in N'. Again, d-Red-Blue Dominating Set has a simple kernel of size $O(k^d)$ by assuring that each vertex in N has a unique set of neighbors in T. The d-Steiner Tree takes the same input but we are asked whether there is a subset $N' \subseteq N$ of size at most k such that $G[T \cup N']$ is connected.

Theorem 4. *Unless* coNP $\subseteq$ NP/poly*, d-*Red-Blue Dominating Set *and d-*Steiner Tree *have no kernels of sizes $O(k^{d-3-\varepsilon})$ and $O(k^{d-4-\varepsilon})$, respectively, for any $\varepsilon > 0$.*

Proof. Observe that d-BRPC is a special case of d-Red-Blue Dominating Set, and so d-BRPC $\leq_{lpt}$ d-Red-Blue Dominating Set. For the transformation from $(d-1)$-BRPC to d-Steiner Tree, we take an instance $((N \uplus T, E), k)$ of $(d-1)$-BRPC and create an instance $((N' \uplus T', E')k)$ of d-Steiner Tree by adding a new vertex $\hat{u}$ and setting $N' = N$, $T' = T \cup \{\hat{u}\}$, and $E' = E \cup \{\{\hat{u}, v\} : v \in N\}$. Clearly this is a d-Steiner Tree instance since all vertices in N have degree d. It is easy to see that any solution for the $(d-1)$-BRPC one-to-one corresponds to a solution for the d-Steiner Tree instance, and so $(d-1)$-BRPC $\leq_{lpt}$ d-Steiner Tree. Applying Theorem 1 and Lemma 4, the proof follows. □

Let us next consider two graph packing problems. In the K_d Packing problem we are given graph G and a parameter k, and the question is whether G contains at least k vertex-disjoint cliques of size d. This problem has a kernel of size $O(k^d)$ due to [13]. The Induced $K_{1,d}$ Packing takes the same input but asks whether there are k pairwise disjoint subset of vertices, each inducing a d-star in G.

Theorem 5. *Unless* coNP $\subseteq$ NP/poly*, K_d* Packing *and* Induced $K_{1,d}$ Packing *have no kernels of size $O(k^{d-4-\varepsilon})$ for any $\varepsilon > 0$.*

Proof. Let (G, k) be an instance of $(d-1)$-BRPC instance with $G = (N \uplus T, E)$ and $|T| = k(d-1)$. We transform (G, k) to an instance of K_d Packing by connecting every pair of vertices in T to make it a clique. Let the resulting graph be G'. Clearly a solution for (G, k) of size k corresponds to a packing of k vertex-disjoint d-cliques. In the other direction, observe that the non-terminal component N is an independent set, therefore at most k non-terminals can appear in the clique packing. Further, k vertex-disjoint d-cliques require kd vertices but $|T| = k(d-1)$, hence we have to pick k vertices from N, with each of them in a different clique. Thus every k vertex-disjoint cliques of size d in G' corresponds to a perfect code in (G, k).

To reduce $(d-1)$-BRPC to Induced $K_{1,d}$ Packing, first we add a set of k new nonterminal vertices, $X = \{x_1, \ldots, x_k\}$. Then we make $N \cup X$ a clique by connecting every pair of nonterminal vertices. Let the resulting instance be G'. Observe that X is only connected to nonterminal vertices. It is now straightforward to check that the k star-centers must come from N. Consider any of these centers, say $v \in N$. We argue that in the d-star centered at v, $(d-1)$ star petals come from T and the remaining one comes from $N \cup X$. Indeed because $G'[N \cup X]$ is clique, we cannot pick $u, w \in N$ into the star because this gives a triangle, contradicted with requirement that every star must be an induced subgraph. Now because the number of neighbors of any vertex of N in T is exactly $(d-1)$, it is clear every k vertex-disjoint $K_{1,d}$ in G' corresponds to a perfect code in (G, k). □

Indeed the above argument for Induced $K_{1,d}$ Packing works for Induced $K_{s,d}$ Packing for any constant $s \geq 1$. To reduce $(d-1)$-BRPC to Induced $K_{s,d}$ Packing, we split each nonterminal vertex in N into s vertices, add a set X of k new nonterminal vertices, and make $N \cup X$ a clique. Then the same argument above shows that the s centers of $K_{s,d}$ must come from N, and indeed correspond to the same nonterminal before splitting. Hence k vertex-disjoint $K_{s,d}$ packing one-to-one corresponds to perfect code in the $(d-1)$-BRPC instance. Hence we have:

Corollary 1. *Let $s \geq 1$ be a constant. Unless* coNP $\subseteq$ NP/poly*,* Induced $K_{s,d}$ Packing *has no kernels of size $O(k^{d-4-\varepsilon})$ for any $\varepsilon > 0$.*

6 Quasi-polynomial Lower Bounds

In this section we extend the state of the art of the kernelization lower bounds mechanically in another direction. We will show that essentially all previously known super-polynomial lower bound results can be strengthen to super-quasi-polynomial lower bounds, assuming that the exponential hierarchy is proper. For this, we will use a recent quasi-polynomial analog of Yap's Theorem due to Pavan *et al.* [24]:

Lemma 11 ([24]). *If* NP $\subseteq$ coNP/qpoly *then the exponential hierarchy collapses to its third level.*

The above result of Pavan *et al.* implies that to obtain quasi-polynomial kernelization lower bounds under the assumption that the exponential hierarchy is proper, a quasi-polynomial analog of the Complementary Witness Lemma of Dell and van Melkebeek is needed. Fortunatlly, Dell and van Melkebeek's arguments, which extend the ideas of Fortnow and Santhanam [15], can easily be adapted to the quasi-polynomial case.

Lemma 12. *Let $L \subseteq \{0,1\}^*$ be a language and $t : \mathbb{N} \to \mathbb{N} \backslash \{0\}$ be quasi-polynomially bounded. If there is a quasi-polynomial time oracle communication protocol that decides $\mathsf{OR}_{n,t(n)}(L)$ with cost $O(t(n) \log t(n))$, then $L \in$* coNP/qpoly*. This holds even when the first player runs in* conondeterministic quasi-polynomial time*.*

Proof. To prove the lemma, we need to make use of the concept of oracle-query Turing machine and communication transcript, both of which are defined in [10]. Essentially a communication transcript logs the history of communications between the first player and the second player: this includes each query he sends to the oracle along with the answers for all query.

We call a communication transcript τ *valid* if the answers to all queries are correct, and call τ *consistent* with input sequence $(x_1, \ldots, x_t)$ of L instances if simulating the first player with this input sequence and τ causes no inconsistency with the answers of the oracle. The key idea of the proof hinges on constructing, as advice, a small set of *valid* transcripts S of those communications where the input sequence $(x_1, \ldots, x_t)$ is in $\mathsf{AND}(\overline{L})$, such that:

- If $x \in \overline{L}$, then there exists a sequence $(x_1, \ldots, x_t)$ where $x = x_i$ for some i and there exists a transcript $\tau \in S$ consistent with $(x_1, \ldots, x_t)$, such that the simulation of the first player *halts* and *rejects*. This indicates that $x_i \in \overline{L}$ for all i, and hence $x \in \overline{L}$.
- If $x \in L$, then for all sequences $(x_1, \ldots, x_t)$ where $x = x_i$ for some i, there is either no transcript $\tau \in S$ consistent with $(x_1, \ldots, x_t)$, or we find a consistent transcript but then the simulation of the first player must end up with *accept* (due to the correctness of the protocol).

Following the above discussion, an NP/qpoly machine M can decide whether an input $x \in \{0, 1\}^n$ is in $\overline{L}$ with the help of a quasi-polynomial-sized S, as follows:

1. Nondeterministically guess an input sequence $(x_1, \ldots, x_t)$ where $x = x_i$ for some i and each instance is of length n.
2. Check whether there is a transcript τ consistent with $(x_1, \ldots, x_t)$ in S, and the simulation of the first player on $(x_1, \ldots, x_t)$ with τ halts and rejects. If so, accept, otherwise reject.

It remains to construct the set S of valid transcripts, which is done by a standard averaging argument. Because the communication protocol is of cost $t \log t$, there are at most $2^{t \log t}$ transcripts. Consider all input sequences $(x_1, \ldots, x_t) \in \mathsf{AND}(\overline{L})$, there are at most 2^{tn} of them. We say that τ *covers* x if τ is consistent with $(x_1, \ldots, x_t) \in \mathsf{AND}(\overline{L})$ and $x = x_i$ for some i. We construct S iteratively, at each step consider $X \subseteq \overline{L}$ remained uncovered and pick τ that covers most elements. Initially $X = \overline{L}$ and hence $|X| \leq 2^{tn}$. By averaging, there is one transcript τ covers at least:

$$\sqrt[t]{\frac{|X|^t}{2^{t \log t}}} = \frac{|X|}{2^{\log t}} = \frac{|X|}{t}$$

This indicates that after each step, we have at most $(1 - \frac{1}{t})|X|$ elements uncovered. Now after ℓ steps, there are $(1 - \frac{1}{t})^{\ell}|X| \leq \exp(-\frac{\ell}{t})2^n$ elements uncovered. Hence all elements are covered after $\ell = O(tn)$ steps. Because t is quasi-polynomially bounded by n, the construction ends in at most quasi-polynomial number of steps, which means S is of size quasi-polynomial in n.

This completes the proof for deterministic protocol. For conondeterministic protocol, the same argument as in [10] carries over verbatim so we do not repeat them here. ◻

Theorem 6. *Let $L_1, L_2 \subseteq \{0, 1\}^* \times \mathbb{N}$ be two parameterized problems such that $\widetilde{L_1}$ is* NP*-hard. A composition from L_1 to L_2 and a kernel of quasi-polynomial size for L_2 implies that the exponential hierarchy collapses to its third level.*

Proof. Using a similar argument as in Lemma 3, one can obtain a quasi-polynomial cost communication protocol for L_1, using the assumed quasi-polynomial-size kernel for L_2 along with the composition from L_1 to L_2. Thus, by Lemma 12, we get that NP $\subseteq$ coNP/qpoly, which in turn implies that the exponential hierarchy collapses to its third level due to Lemma 11.

Since all previous super-polynomial lower bounds were obtained via compositions, along with polynomial parametric transformations which also preserves quasi-polynomial kernels, the above theorem implies the strengthening of all previous super-polynomial lower bounds to super-quasi-polynomial lower bounds.

7 Conclusion

In this paper we introduced a new type of composition called *weak composition* that allows proving polynomial kernelization lower-bounds, as opposed to the super-polynomial lower-bounds given by the previously known compositions. Using weak compositions, we showed new kernelization lower-bounds for several natural parameterized problems such as d-Dimensional-Matching, d-Set Packing, d-Set Cover, and K_d Packing. We believe weak compositions could be used to obtain further new lower-bounds.

There are many interesting directions for future research that stem from our work. The most important one is to close the gap between the upper and lower bounds for the kernel sizes of the problems we discussed. Recently we have learned that, independent of our work, Holger Dell and Dániel Marx have made some progress on this issue.

Acknowledgments

We would like to thank Karl Bringmann and Karolina Soltys for fruitful discussions. In particular, Karl provided several insights regarding the main composition algorithm presented in Section 3. We would also like to thank Chandan Saha for referring us to [24].

References

1. Jochen Alber, Michael R. Fellows, and Rolf Niedermeier. Polynomial-time data reduction for dominating set. *Journal of the ACM*, 51(3):363–384, 2004.
2. Noga Alon, Gregory Gutin, Eun Jung Kim, Stefan Szeider, and Anders Yeo. Solving MAX-r-SAT above a tight lower bound. In *Proc. of the 21st annual ACM-SIAM Symposium On Discrete Algorithms (SODA)*, pages 511–517, 2010.
3. Hans L. Bodlaender. Kernelization: New upper and lower bound techniques. In *Proc. of the 4th International Workshop on Parameterized and Exact Computation (IWPEC)*, pages 17–37, 2009.
4. Hans L. Bodlaender, Rodney G. Downey, Michael R. Fellows, and Danny Hermelin. On problems without polynomial kernels. *Journal of Computer and System Sciences*, 75(8):423–434, 2009.
5. Hans L. Bodlaender, Fedor V. Fomin, Daniel Lokshtanov, Eelko Penninkx, Saket Saurabh, and Dimitrios M. Thilikos. (Meta) kernelization. In *Proc. of the 50th annual IEEE symposium on Foundations Of Computer Science (FOCS)*, pages 629–638, 2009.
6. Hans L. Bodlaender, Bart M. P. Jansen, and Stefan Kratsch. Cross-composition: A new technique for kernelization lower bounds. In *Proc. of the 28th international Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 165–176, 2011.
7. Hans L. Bodlaender, Stéphan Thomassé, and Anders Yeo. Kernel bounds for disjoint cycles and disjoint paths. In *Proc. of the 17th annual European Symposium on Algorithms (ESA)*, pages 635–646, 2009.
8. Liming Cai, Jianer Chen, Rodney G. Downey, and Michael R. Fellows. Advice classes of paramterized tractability. *Annals of Pure and Applied Logic*, 84(1):119–138, 1997.
9. Yijia Chen, Jörg Flum, and Moritz Müller. Lower bounds for kernelizations and other preprocessing procedures. *Theory of Computing Systems*, 48(4):803–839, 2011.
10. Holger Dell and Dieter van Melkebeek. Satisfiability allows no nontrivial sparsification unless the polynomial-time hierarchy collapses. In *Proc. of the 42th annual ACM Symposium on Theory Of Computing (STOC)*, pages 251–260, 2010.
11. Michael Dom, Daniel Lokshtanov, and Saket Saurabh. Incompressibility through colors and IDs. In *Proc. of the 36th International Colloquium on Automata, Languages and Programming (ICALP)*, pages 378–389, 2009.

12. Rodney G. Downey and Michael R. Fellows. *Parameterized Complexity.* Springer-Verlag, 1999.
13. Michael R. Fellows, Christian Knauer, Naomi Nishimura, Prabhakar Ragde, Frances A. Rosamond, Ulrike Stege, Dimitrios M. Thilikos, and Sue Whitesides. Faster fixed-parameter tractable algorithms for matching and packing problems. *Algorithmica*, 52(2):167–176, 2008.
14. Henning Fernau, Fedor V. Fomin, Daniel Lokshtanov, Daniel Raible, Saket Saurabh, and Yngve Villanger. Kernel(s) for problems with no kernel: On out-trees with many leaves. In *Proc. of the 26th international Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 421–432, 2009.
15. Lance Fortnow and Raul Santhanam. Infeasibility of instance compression and succinct PCPs for NP. In *Proc. of the 40th annual ACM Symposium on Theory Of Computing (STOC)*, pages 133–142, 2008.
16. Michael R. Garey and David S. Johnson. *Computers and intractability : A guide to the theory of NP-completeness.* W.H. Freeman, 1979.
17. Sylvain Guillemot, Christophe Paul, and Anthony Perez. On the (non-)existence of polynomial kernels for P_l-free edge modification problems. In *Proc. of the 5th International symposium on Parameterized and Exact Computation (IPEC)*, pages 147–157, 2010.
18. Jiong Guo and Rolf Niedermeier. Invitation to data reduction and problem kernelization. *SIGACT News*, 38(1):31–45, 2007.
19. Danny Harnik and Moni Naor. On the compressibility of NP instances and cryptographic applications. *SIAM Journal on Computing*, 39(5):1667–1713, 2010.
20. Stefan Kratsch, Dániel Marx, and Magnus Wahlström. Parameterized complexity and kernelizability of max ones and exact ones problems. In *Proc. of the 35th international symposium on Mathematical Foundations of Computer Science (MFCS)*, pages 489–500, 2010.
21. Stefan Kratsch and Magnus Wahlström. Two edge modification problems without polynomial kernels. In *Proc. of the 4th International Workshop on Parameterized and Exact Computation (IWPEC)*, pages 264–275, 2009.
22. Stefan Kratsch and Magnus Wahlström. Preprocessing of min ones problems: A dichotomy. In *Proc. of the 37th International Colloquium on Automata, Languages and Programming (ICALP)*, pages 653–665, 2010.
23. George L. Nemhauser and Leslie E. Trotter Jr. Vertex packings: Structural properties and algorithms. *Mathematical Programming*, 8(2):232–248, 1975.
24. Aduri Pavan, Alan L. Selman, Samik Sengupta, and Vinodchandran N. Variyam. Polylogarithmic-round interactive proofs for coNP collapse the exponential hierarchy. *Theoretical Computer Science*, 385(3):391–404, 2007.
25. Stéphan Thomassé. A $4k^2$ kernel for feedback vertex set. *ACM Transactions on Algorithms*, 6(2), 2010.