Tight Kernel Bounds for Problems on Graphs with Small Degeneracy

Marek Cygan[1], Fabrizio Grandoni[⋆2], and Danny Hermelin[⋆⋆3]

[1] Institute for Informatics
Banacha 2, 02-097, Warsaw - Poland
cygan@mimuw@edu.pl
[2] Dalle Molle Institute (IDSIA)
Galleria 1, 6928, Manno - Switzerland
fabrizio@idsia.ch
[3] Ben-Gurion University,
Marcus Family Campus, 8410501, Beer-Sheva - Israel
hermelin@bgu.ac.il

Abstract. Kernelization is a strong and widely-applied technique in parameterized complexity. In a nutshell, a kernelization algorithm for a parameterized problem transforms in polynomial time a given instance of the problem into an equivalent instance whose size depends solely on the parameter. Recent years have seen major advances in the study of both upper and lower bound techniques for kernelization, and by now this area has become one of the major research threads in parameterized complexity.

In this paper we consider kernelization for problems on d-degenerate graphs, i.e. graphs such that any subgraph contains a vertex of degree at most d. This graph class generalizes many classes of graphs for which effective kernelization is known to exist, e.g. planar graphs, H-minor free graphs, and H-topological-minor free graphs. We show that for several natural problems on d-degenerate graphs the best known kernelization upper bounds are essentially tight. In particular, using intricate constructions of weak compositions, we prove that unless coNP$\subseteq$ NP/poly:

– Dominating Set has no kernels of size $O(k^{(d-1)(d-3)-\varepsilon})$ for any $\varepsilon > 0$. The current best upper bound is $O(k^{(d+1)^2})$.
– Independent Dominating Set has no kernels of size $O(k^{d-4-\varepsilon})$ for any $\varepsilon > 0$. The current best upper bound is $O(k^{d+1})$.
– Induced Matching has no kernels of size $O(k^{d-3-\varepsilon})$ for any $\varepsilon > 0$. The current best upper bound is $O(k^d)$.

To the best of our knowledge, Dominating Set is the the first problem where a lower bound with superlinear dependence on d (in the exponent) can be proved.

In the last section of the paper, we also give simple kernels for Connected Vertex Cover and Capacitated Vertex Cover of size $O(k^d)$ and $O(k^{d+1})$ respectively. We show that the latter problem has no kernels of size $O(k^{d-\varepsilon})$ unless coNP$\subseteq$ NP/poly by a simple reduction from d-Exact Set Cover (the same lower bound for Connected Vertex Cover on d-degenerate graphs is already known).

⋆ Partially supported by ERC Starting Grant NEWNET 279352 and SNSF Grant 200021_159697/1.
⋆⋆ The research leading to these results has received funding from the People Programme (Marie Curie Actions) of the European Union's Seventh Framework Programme (FP7/2007-2013) under REA grant agreement number 631163.11

1 Introduction

Parameterized complexity is a two-dimensional refinement of classical complexity theory introduced by Downey and Fellows [13] where one takes into account not only the total input length n, but also other aspects of the problem quantified in a numerical parameter $k \in \mathbb{N}$. The main goal of the field is to determine which problems have algorithms whose exponential running time is confined strictly to the parameter. In this framework, algorithms running in $f(k) \cdot n^{O(1)}$ time for some computable function $f()$ are considered feasible, and parameterized problems that admit feasible algorithms are said to be *fixed-parameter tractable*. This notion has proven extremely useful in identifying tractable instances for generally hard problems, and in explaining why some theoretically hard problems are solved routinely in practice.

A closely related notion to fixed-parameter tractability is that of kernelization. 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) to an instance (x',k') (the *kernel*) such that: (i) $(x,k) \in L \iff (x',k') \in L$, and (ii) $|x'| + k' \leq f(k)$ for some computable function f. In other words, a kernelization algorithm is a polynomial-time reduction from a problem to itself that shrinks the problem instance to an instance with size depending only on the parameter. Appropriately, the function f above is called the *size* of the kernel, and for the sake of shortness we say that the considered problem admits a kernel of that size.

Kernelization is a notion that was developed in parameterized complexity, but it is also useful in other areas of computer science such as cryptography [25] and approximation algorithms [32]. In parameterized complexity, this is not only one of the most successful techniques for showing positive results, but it also provides an equivalent way of defining fixed-parameter tractability: A decidable parameterized problem is solvable in $f(k) \cdot n^{O(1)}$ time if and only if it admits a kernelization algorithm [6]. From a practical point of view, kernelization algorithms often lead to efficient preprocessing rules which can significantly simplify real life instances [17, 23]. For these reasons, the study of kernelization is one of the leading research frontiers in parameterized complexity. This research endeavor has been fuelled by recent tools for showing lower bounds on kernel sizes [2, 4, 5, 7, 9, 10, 12, 27, 30] which rely on the standard complexity-theoretic assumption that coNP$\not\subseteq$ NP/poly.

Since a parameterized problem is fixed-parameter tractable if and only if it is kernelizable, it is natural to ask which fixed-parameter problems admit kernels of reasonably small size. In recent years there has been significant advances in this area. One particulary prominent line of research in this context is the development of *meta-kernelization* algorithms for problems on sparse graphs. Such algorithms typically provide small kernels for a wide range of problems at once, by identifying certain generic problem properties that allow for good compressions. The first work in this line of research is due to Guo and Niedermeier [24] who extended the ideas used in the classical linear kernel for Dominating Set[1] in planar graphs [1] to linear kernels for several other planar graph problems. This result was subsumed by the seminal paper of Bodlaender *et al.* [3], which provided meta-kernelization algorithms for problems on graphs of bounded genus, a generalization of planar graphs. Later Fomin *et al.* [19] provided a meta-kernelization for problems on H-minor free graphs which include all bounded genus graphs. Finally, a recent manuscript by Langer *et al.* [31] provides a meta-kernelization algorithm for problems on H-topological-minor free graphs.

All meta-kernelizations above have either linear or quadratic size and work for a range of problems satisfying two properties. First, the problem should be expressible in Counting Monadic Second Order logic, or should have finite Integer Index. Second, the problem has to satisfy a certain covering property (for precise definitions, see [3]). For example the technique in [3] can be applied to

[1] For problem definitions, please see Section 2.

Connected Vertex Cover, Dominating Set and Feedback Vertex Set to get linear kernels in bounded genus graphs. We recall that Feedback Vertex Set is the problem of removing at most k vertices from a graph to obtain a forest.

How far can these meta-kernelization results be extended? A natural class of sparse graphs which generalizes all graph classes handled by the meta-kernelizations discussed above is the class of d-degenerate graphs. A graph is called *d-degenerate* if each of its subgraphs has a vertex of degree at most d. This is equivalent to requiring that the vertices of the graph can be linearly ordered such that each vertex has at most d neighbors following it in this ordering. For example, any planar graph is 5-degenerate, and for any H-minor (resp. H-topological-minor) free graph class there exists a constant $d(H)$ such that all graphs in this class are $d(H)$-degenerate. (A nice clean proof of this fact can be found in Diestel's classical text [11]). Note that the Independent Set problem has a trivial linear kernel in d-degenerate graphs, since a d-degenerate graph with n vertices always contains an independent set of size at least $n/(d+1)$. This gives some hope that a meta-kernelization result yielding small degree polynomial kernels might be attainable for this graph class.

Arguably the most important kernelization result in d-degenerate graphs is due to Philip *et al.* [33] who showed a $O(k^{(d+1)^2})$ size kernel for Dominating Set, and an $O(k^{d+1})$ size kernel for Independent Dominating Set. Erman *et al.* [15] and Kanj *et al.* [28] independently gave an $O(k^d)$ kernel for the Induced Matching problem, while Cygan *et al.* [8] showed an $O(k^{d+1})$ kernel for Connected Vertex Cover. While all these results give polynomial kernels, the exponent of the polynomial depends on d, leaving open the question of kernels of polynomial size with a fixed constant degree. This question was answered negatively for Connected Vertex Cover in [8] using the standard reduction from d-Exact Set Cover. It is also shown in [8] that other problems such as Connected Dominating Set and Connected Feedback Vertex Set do not admit a kernel of any polynomial size unless coNP$\subseteq$ NP/poly.

Our results. The kernelization lower bounds of [8] may suggest that it could be hard to obtain meta-kernelization in d-degenerate graphs. However, they deal mainly with "connectivity" type problems that typically cannot be handled by the meta-kernelization algorithms for H-minor free and H-topological-minor free graphs. For instance, the only known linear kernel for Connected Dominating Set in H-minor free graphs [20] requires new problem specific ideas which are not used in the generic meta-kernelization for this graph class. In this paper we give further, and perhaps more convincing, indication that truly general meta-kernelization results with output size bound by small degree polynomials are unlikely to exist for d-degenerate graphs. In particular, we show that all kernelization upper bounds for d-degenerate graphs mentioned above have matching lower bounds up to some small additive constant in the exponent.

Perhaps the most surprising result we obtain is the exclusion of $O(k^{(d-3)(d-1)-\varepsilon})$ size kernels for Dominating Set for any $\varepsilon > 0$, under the assumption that coNP $\nsubseteq$ NP/poly. This result is obtained by an intricate application of *weak compositions* which were introduced by [10], and further applied in [9, 27]. What makes this result surprising is that it implies that Independent Dominating Set is fundamentally easier than Dominating Set in d-degenerate graphs. We also show an $O(k^{d-4-\varepsilon})$ lower bound for Independent Dominating Set, and an $O(k^{d-3-\varepsilon})$ lower bound for Induced Matching. The latter result is also somewhat surprising when one considers the trivial linear kernel for the closely related Independent Set problem. Finally, we slightly improve the $O(k^{d+1})$ kernel for Connected Vertex Cover of [8] to $O(k^d)$, and show that the related Capacitated Vertex Cover problem has a kernel of size $O(k^{d+1})$, but no kernel of size $O(k^{d-\varepsilon})$ unless coNP $\subseteq$ NP/poly. Table 1 summarizes the currently known state of the art of kernel sizes for the problems considered in this paper.

	Lower Bound	Upper Bound
Dominating Set	$(d-3)(d-1)-\varepsilon$	$(d+1)^2$ [33]
Independent Dominating Set	$d-4-\varepsilon$	$d+1$ [33]
Induced Matching	$d-3-\varepsilon$	d [15, 28]
Connected Vertex Cover	$d-1-\varepsilon$ [8]	d
Capacitated Vertex Cover	$d-\varepsilon$	$d+1$

Table 1. Lower and upper bounds for kernel sizes for problems in d-degenerate graphs. Only the exponent of the polynomial in k is given. Results without a citation are obtained in this paper.

2 Preliminaries

We begin by reviewing some basic definitions and terminology that will be used throughout the paper. First, we discuss basic graph-theoretic notation and give formal definitions for all the problems we study. We then describe the general framework that we will use for excluding specific-size kernels. The reader interested in a more in-depth introduction to parameterized complexity is referred to [13].

2.1 Basic Graph Theoretic Concepts

In general, we follow standard graph-theoretical terminology and notation, as can be found in *e.g.* [11]. All graphs throughout the paper are simple and without self loops. Typically, we will use the letters G and H to denote two different graphs. We write $G = (V, E)$ to indicate that the graph G has vertex set V and edge set E, and we use $V(G)$ and $E(G)$ when V and E have not been specified in advance. For a vertex $v \in V$, the set of *neighbors* $N(v) = N_G(v)$ of v is $\{u : \{u, v\} \in E\}$, and the degree of v is $|N(v)|$. For a set of vertices $S \subseteq V$, we define $N(S) = (\bigcup_{v \in S} N(v)) \setminus S$. A graph $H = (V, E)$ is a *subgraph* of a graph $G = (V', E')$, denoted $H \subseteq G$, if $V \subseteq V'$ and $E \subseteq E'$. The subgraph $G[W]$ induced by a subset of vertices $W \subseteq V$ is the graph with vertex set W and edge set $\{e = \{u, v\} \in E : u, v \in W\}$.

The main protagonists of this paper are d-degenerate graphs: For a fixed positive integer $d \geq 1$, a graph G is called *d-degenerate* if in each of its subgraphs there is a vertex of degree at most d. That is, $\max_{H \subseteq G} \min_{v \in V'} |N_H(v)| \leq d$. For example, forests are 1-degenerate, and planar graphs are 5-degenerate. It is known that G is d-degenerate if and only if the vertices of G can be ordered in such a way so that each vertex has at most d neighbors that follow it in the order (see *e.g.* [11]). We will extensively use this characterization in this paper. Another property that we will extensively use it the fact that d-degeneracy is closed under vertex deletions (which follows directly from the definition). We are interested in classical graph-theoretic problems restricted to d-degenerate graphs for some fixed constant $d \geq 1$.

Let $G = (V, E)$ be a graph (which in our context will usually be d-degenerate). A subset of vertices $D \subseteq V$ *dominates* the set of vertices $D \cup N(D)$. A *dominating set* of G is a subset $D \subseteq V$ which dominates V. An *independent set* of G is a subset of vertices $I \subseteq V$ which are pairwise not adjacent, i.e. $\{u, v\} \notin E$ for all $u, v \in I$. An *independent dominating set* of G is a subset of vertices that form an independent set and a dominating set at the same time. A *matching* in G is a subset of edges $M \subseteq E$ that are pairwise disjoint, i.e. given $\{u_1, v_1\}, \{u_2, v_2\} \in M$ one has $\{u_1, v_1\} \cap \{u_2, v_2\} = \emptyset$. A matching is *perfect* if all vertices are matched, i.e. every vertex belongs to some edge of the matching. An *induced matching* in G is a matching M such that the graph induced by the vertex set of M contains no further edges besides M. In the Dominating Set (*resp.* Independent Set, Independent Dominating Set, Induced Matching) problem we wish to determine whether a given graph G has a dominating set (*resp.* independent set, independent dominating set, induced matching) of size k.

In the last part of the paper we will deal with variants of the Vertex Cover problem. A *vertex cover* of a graph $G = (V, E)$ is a subset of vertices $A \subseteq V$ such that for each edge $\{u, v\} \in G$ we have $\{u, v\} \cap A \neq \emptyset$. A *connected vertex cover* of G is a vertex cover A which induces a connected subgraph in G; *i.e.* the graph $(A, \{\{u, v\} \in E : u, v \in A\})$ is required to be connected. In the Connected Vertex Cover problem, we are asked to determine whether a given graph G has a connected vertex cover of size k. In the Capacitated Vertex Cover problem, we are given a graph $G = (V, E)$, an integer k, and a vertex capacity function $cap : V \to \mathbb{N}$. The goal is to determine whether there exists a vertex cover A of size k and an injective mapping α mapping each edge of E to one of its endpoints such that $|\alpha^{-1}(v)| \leq cap(v)$ for every $v \in V(G)$. Intuitively, each vertex can cover a limited number of edges specified by its capacity.

2.2 Kernelization Lower Bounds Framework

We next review the main tool – namely, *compositions* – that we will be using for showing our kernelization lower bounds. A composition algorithm is a transformation from a classical NP-complete problem L_1 to a parameterized problem L_2. It takes as input a sequence of T instances of L_1, each of size n, and outputs in polynomial time an instance of L_2 such that (i) the output is a YES-instance if and only if at least one of the inputs is a YES-instance, and (ii) the parameter of the output is polynomially bounded by n and has only "small" dependency on T. Thus, a composition may intuitively be thought of as an "OR-gate" with a guaranteed bound on the parameter of the output.

There are quite a few variations of composition algorithms that have been used for showing kernelization lower bounds. For our purposes, we chose a variant called *weak compositions*, a notion coined in [27] but first used in [10]. In this variant, the small dependency on T is defined as the q-th root of T, for some constant q.

Definition 1 (weak q-composition). *Let $q \geq 1$ be an integer constant, let $L_1 \subseteq \{0,1\}^*$ be a classical (non-parameterized) problem, and let $L_2 \subseteq \{0,1\}^* \times \mathbb{N}$ be a parameterized problem. A* weak q-composition *from L_1 to L_2 is a polynomial time algorithm that on input $x_1, \ldots, x_{t^q} \in \{0,1\}^n$ outputs an instance $(y, k') \in \{0,1\}^* \times \mathbb{N}$ such that:*

- $(y, k') \in L_2 \Longleftrightarrow x_i \in L_1$ *for some* i, *and*
- $k' \leq t \cdot n^{O(1)}$.

The connection between compositions and kernelization lower bounds was first discovered by [2] using ideas from [25] and a complexity theoretic lemma of [21]. The following particular connection was first observed in [10], although the statement in its current form is taken from [27].

Lemma 1 ([27]). *Let $d \geq 1$ be an integer, let $L_1 \subseteq \{0,1\}^*$ be a classical* NP*-complete problem, and let $L_2 \subseteq \{0,1\}^* \times \mathbb{N}$ be a parameterized problem. A weak-d-composition from L_1 to L_2 implies that L_2 has no kernel of size $O(k^{d-\varepsilon})$ for any $\varepsilon > 0$, unless* coNP $\subseteq$ NP/poly.

Remark 1. Lemma 1 also holds for *compressions*, a stronger notion of kernelization, in which the reduction is not necessarily from the problem to itself, but rather from the problem to any other problem.

3 Dominating Set

Recall that in the Dominating Set (DS) problem, we are asked whether there exists a set D of at most k vertices in a given graph H such that each vertex of H either belongs to D or is adjacent to a vertex in D. The main result of this section is stated in Theorem 1 below.

Theorem 1. *Let $d \geq 3$. The* Dominating Set *problem in d-degenerate graphs has no kernel of size $O(k^{(d-3)(d-1)-\varepsilon})$ for any constant $\varepsilon > 0$ unless* NP $\subseteq$ coNP/poly.

For proving this theorem, we present a rather elaborate weak $d(d+2)$-composition from a problem called Multicolored Perfect Matching (described below) to DS in $(d+3)$-degenerate graphs. We first give an overview of our construction, and then discuss in further details each of its components.

3.1 Construction Overview

Let us begin by introducing the non-parameterized NP-complete problem that we will exploit in our weak composition, Multicolored Perfect Matching (MPM): Given an undirected graph $G = (V, E)$ with an even number n of vertices, and an edge color function col : $E \to \{0, \ldots, n/2-1\}$, determine whether there exists a perfect matching in G in which all the edges have distinct colors. Let us show that MPM is NP-complete via a reduction from the following 3-Dimensional Perfect Matching problem that Karp [29] proved to be NP-complete: We are given three equal size disjoint sets $X = \{x_1, \ldots, x_n\}$, $Y = \{y_1, \ldots, y_n\}$, and $Z = \{z_1, \ldots, z_n\}$, and a family of triplets $\mathcal{S} \subseteq X \times Y \times Z$. Our goal is to determine whether there exists a subset $\mathcal{S}' \subseteq \mathcal{S}$ such that triplets in $\mathcal{S}'$ are pairwise disjoint and each element of $X \cup Y \cup Z$ is contained in some triplet in $\mathcal{S}'$. We start by showing the NP-completeness of MPM in its natural generalization to multigraphs[2], and then extend it to simple graphs.

Lemma 2. *The* Multicolored Perfect Matching *problem in multigraphs is* NP*-complete.*

Proof. Consider an instance $(X, Y, Z, \mathcal{S})$ of 3-Dimensional Perfect Matching as defined above. We encode it into an instance of MPM by constructing a (bipartite) multigraph G over the vertex set $X \cup Y$, where there is an edge $e = \{x_i, y_j\}$ in G with $\text{col}(e) = k$ if and only if $(x_i, y_j, z_k) \in \mathcal{S}$. It is not hard to verify that $(X, Y, Z, \mathcal{S})$ has a feasible solution if and only if (G, col) also has one. □

Lemma 3. *The* Multicolored Perfect Matching *problem in simple graphs is* NP*-complete.*

Proof. We show a polynomial time reduction from MPM in multigraphs, where several parallel edges are allowed. Recall that the latter problem is NP-complete by Lemma 2.

Let $(G = (V, E), \text{col} : E \to \{0, \ldots, |V|/2-1\})$ be an instance of MPM where G is a multigraph and $|V|$ is even. Let $\pi : E \to \{0, \ldots, |E|-1\}$ be an arbitrary bijection between the multiset of edges of G and integers from 0 to $|E|-1$. Moreover we assume that there is some fixed linear order on V, and we write $u < v$ to denote that u appears before v in this order and vice versa for $u > v$. We create a graph G' as follows. The set of vertices of G' is $V' = \{x_v : v \in V\} \cup \{y_{v,e} : e \in E, v \in V, v \in e\}$, that is we create a vertex in G' for each vertex of G and for each endpoint of an edge of G. The set of edges of G' is $E' = \{y_{u,e}y_{v,e} : e = uv \in E\} \cup \{x_v y_{v,e} : v \in V, e \in E, v \in e\}$. Finally, we define the color function $\text{col}' : E' \to \{0, \ldots, n'/2-1\}$, where $n' = 2|E| + |V|$, as follows. For $y_{u,e}y_{v,e} \in E'$

[2] We recall that a multigraph $G = (V, E)$ is defined similarly to a graph, with the difference that E is a multiset of edges rather than a set of edges.

we set $\mathrm{col}'(y_{u,e}y_{v,e}) = \pi(e)$; for $x_v y_{v,e} \in E'$, where $e = \{u,v\}$ and $u < v$, we set $\mathrm{col}'(x_v y_{v,e}) = \pi(e)$; finally, for $x_v y_{v,e} \in E'$, where $e = \{u,v\}$ and $u > v$, we set $\mathrm{col}'(x_v y_{v,e}) = |E| + \mathrm{col}(e)$.

Clearly the construction can be performed in polynomial time, and the graph G' is simple. Hence it suffices to show that the instance (G', col') is a YES-instance if and only if (G, col) is a YES-instance. Let $M \subseteq E$ be a solution for (G, col). Observe that the set $M' = \{y_{u,e}y_{v,e} : e = uv \in E \setminus M\} \cup \{x_v y_{v,e} : e \in M, v \in e\}$ is a solution for (G', col'). In the opposite direction, assume that $M' \subseteq E'$ is a solution for (G', col'). Note that for each $e = uv \in E$ either we have $y_{u,e}y_{v,e} \in M'$ or both $x_u y_{u,e}$ and $x_v y_{v,e}$ belong to M'. Consequently we define $M \subseteq E$ to be the set of edges $e = uv$ of E such that $y_{u,e}y_{v,e} \notin M'$. It is not hard to verify that M is a solution for (G, col). □

Instead of directly using DS as our parameterized problem in the weak composition, we will consider the following Red Blue Dominating Set problem (RBDS): Given a bipartite graph $H = (R \cup B, E)$, where R is the set of *red* vertices and B is the set of *blue* vertices, and an integer k, determine whether there exists a set $D \subseteq R$ of at most k red vertices that dominates all the blue vertices (i.e., each blue vertex is adjacent to some vertex in D). We remark that RBDS is also prominent in other kernelization lower-bound results [12, 27]. The lemma below shows that it is sufficient for our purposes to focus on RBDS as the target problem in our composition.

Lemma 4. *There is a polynomial time algorithm that, given a d-degenerate instance $I = (H = (R \cup B, E), k)$ of* RBDS*, creates a $(d+1)$-degenerate instance $I' = (H', k')$ of* DS*, such that $k' = k + 1$ and I is a YES-instance if and only if I' is a YES-instance.*

Proof. As the graph H' we initially take $H = (R \cup B, E)$ and then we add two vertices r, r' and make r adjacent to all the vertices in $R \cup \{r'\}$. Clearly H' is $(d+1)$-degenerate. Note that if $D \subseteq R$ is a solution in I, then $D \cup \{r\}$ is a dominating set in I'. In the reverse direction, observe that without loss of generality we may assume that a solution D' for I' contains r. Indeed, if $r \notin D'$, then $r' \in D'$ and consequently $(D' \setminus \{r'\}) \cup \{r\}$ is also a feasible solution. Moreover we may assume that D' contains no vertex of B, as r dominates all the vertices of R and consequently we may replace each vertex of $D' \cap B$ by any one of its neighbors in R. Therefore I is a YES-instance if and only if I' is a YES-instance. □

Let $(G_i = (V, E_i), \mathrm{col}_i)$ be the input MPM instances of our weak composition, with $0 \le i < T = t^{d(d+2)}$. By standard renaming and padding arguments (*e.g.* adding isolated edges to equalize the sizes of vertex sets), we may assume that all the graphs G_i are defined over the same set V of even size n, *i.e.* $G_i = (V, E_i)$ for all i. The instance of RBDS constructed from these T instances of MPM will have two main ingredients: an *instance graph* H_{inst}, and an *enforcement gadget* (H_{enf}, E_{conn}), where H_{enf} is an (*enforcement*) graph and E_{conn} is a set of edges between H_{inst} and H_{enf}. Intuitively, the instance graph encodes feasible solutions of each of the MPM instances, while the enforcement gadget prevents partial solutions of two or more MPM instances to form together a solution for the RBDS instance. The overall RBDS instance will be denoted by (H, k), for a proper choice of k, where H is the union of H_{inst}, H_{enf}, and E_{conn}.

In the following subsection we describe in detail our construction. For purposes that will hopefully be clearer later on, we associate with each MPM instance $(G_i = (V, E_i), \mathrm{col}_i)$ a distinct $d \times (d+2)$ matrix M_i with entries $M_i[\alpha, \beta] \in \{0, \ldots, t-1\}$, for all possible values of α and β. We will refer to the matrix M_i as the *identifier* of the instance $(G_i = (V, E_i), \mathrm{col}_i)$. Note that there are $t^{d(d+2)} = T$ such distinct identifiers, and so there is a one-to-one bijection between identifers and the input instances of MPM.

Notational conventions. We will use the convention that R and B denote sets of red and blue vertices, respectively. We will use r and b to indicate red and blue vertices, respectively. We will also use a few indices that vary in range. The following list summarizes these:

- A color (of an edge in an MPM instance) in the range $\{0, \ldots, n/2-1\}$ will be indicated by the letter ℓ.
- A row index in the range $\{0, \ldots, d-1\}$ and a column index in the range $\{0, \ldots, d+1\}$ of some matrix M_i will be denoted by letters α and β, respectively.
- The index of an MPM instance in the range $\{0, \ldots, T-1\}$ will be denoted by i or j.
- We use x to denote an integer in the range $\{0, \ldots, (dt)^{d+2}\}$, while y denotes an integer in the range $\{1, \ldots, d^{d+2}-d\}$.

3.2 Construction Details

The Instance Graph: The construction of the instance graph $H_{inst} = (R_{inst} \cup B_{inst}, E_{inst})$ is done as follows (see also Figure 1): Recall that we assume that all the graphs G_i are defined over the same set V of even size n. For each $v \in V$, we create a blue vertex $b_v \in B_{inst}$. For each $i \in \{0, \ldots, T-1\}$, we create a set of red vertices $R^i_{inst} = \{r_{e,i} : e \in E_i\}$ with a vertex for each edge in G_i, and we let $R_{inst} := \bigcup_{0 \le i < T} R^i_{inst}$. Finally, the set of edges E_{inst} is constructed by adding an edge between any $r_{e,i} \in R_{inst}$ and any $b_v \in B_{inst}$ with $v \in e$.

Observe that if there exists a perfect matching $E' \subseteq E_i$ in one of the G_i's, then the set $\{r_{e,i} : e \in E'\} \subseteq R^i_{inst}$ dominates all blue vertices in H_{inst}. Thus, if one of the G_i's is a YES-instance then H_{inst} has a solution of size $n/2$. However, the converse is not necessarily true: For example, if none of the G_i's is a YES-instance, then there still might be $n/2$ red vertices in H_{inst}, say derived from different MPM instances, that dominate all blue vertices of H_{inst}. The rest of our construction is meant to circumvent this issue. Furthermore, it ensures that only matchings which are multicolored will translate to feasible solutions of the RBDS instance.

The Enforcement Graph: The graph $H_{enf} = (R_{enf} \cup B_{enf}, E_{enf})$ is the union of the following three gadgets:

- *Encoding gadget.* The role of this gadget is encode selection of one MPM instance index. It consists of vertices $R_{code} \cup B_{code}$, plus some edges among them. The set R_{code} contains one vertex $r_{\alpha,\beta,m}$ for all integers $0 \le \alpha < d$, $0 \le \beta < d+2$, and $0 \le m < t$. In particular, $|R_{code}| = d(d+2)t$. The set B_{code} contains a vertex b^ℓ_x for each $0 \le \ell < n/2$ and $0 \le x < (dt)^{d+2}$. Here it will be useful to think of the subscripts of vertices in $B^\ell_{code} = \{b^\ell_x : 0 \le x < (dt)^{d+2}\}$ as integers in base dt with $d+2$ digits. We will write $x = (x_0, \ldots, x_{d+1})$ to indicate that $(x_0, \ldots, x_{d+1})$ is the expansion of x in base dt, *i.e.* $x = \sum_{0 \le \beta < d+2} x_\beta (dt)^\beta$. We add an edge between $r_{\alpha,\beta,m} \in R_{code}$ and $b^\ell_x \in B_{code}$ whenever $x_\beta = \alpha \cdot t + m$, where $x = (x_0, \ldots, x_{d+1})$. This way, each vertex b^ℓ_x is adjacent to exactly $d+2$ vertices of R_{code}, one for each digit in the expansion of x. Note that since $0 \le \alpha < d$ and $0 \le m < t$, pairs (α, m) are in one to one correspondence with possible values of the digits x_β. Indeed, each possible value of such digit can be expressed uniquely as $\alpha t + m$.

- *Choice gadget.* The choice gadget will help us to control the number of vertices from R_{code} that are selected by any feasible solution. It consists of a single set of vertices B_{choice}, containing a vertex b_{α,β,m_1,m_2} for every pair (α, β) and for every $0 \le m_1 < m_2 < t$. We add edges $\{b_{\alpha,\beta,m_1,m_2}, r_{\alpha,\beta,m_1}\}$ and $\{b_{\alpha,\beta,m_1,m_2}, r_{\alpha,\beta,m_2}\}$.

– *Fill-in gadget.* The role of the fill-in gadget is to dominate a specific number of vertices of B_{code} not dominated by R_{code}. The fill-in gadget consists of vertices $R_{fill} \cup B_{fill}$ and some edges between them. The set R_{fill} contains one vertex $r^{\ell}_{x,y}$ for each $0 \leq \ell < n/2$, $0 \leq x < (dt)^{d+2}$, and $1 \leq y \leq d^{d+2} - d$. We add one edge between $r^{\ell}_{x,y}$ and $b^{\ell}_x \in B_{code}$, for all possible values of ℓ, x, and y. The set B_{fill} contains one vertex b^{ℓ}_y, for each color ℓ and for all $1 \leq y \leq d^{d+2} - d$. We add one edge between b^{ℓ}_y and all vertices $\{r^{\ell}_{x,y} : 0 \leq x < (dt)^{d+2}\}$.

In the above construction, the red and blue vertices are $R_{enf} = R_{code} \cup R_{fill}$ and $B_{enf} = B_{code} \cup B_{choice} \cup B_{fill}$, respectively.

The Overall Instance: We next describe the edges E_{conn} between H_{enf} and H_{inst}. Recall that each MPM instance is associated with a distinct $d \times (d+2)$ matrix M_i with entries $M_i[\alpha, \beta] \in \{0, \dots, t-1\}$. Consider an MPM instance $(G_i = (V, E_i), \mathrm{col}_i)$. Recall that vertices in B_{code} are indexed according to colors ℓ and integers x which are viewed as numbers in base dt with $d+2$ digits. We add an edge between $r_{e,i} \in R_{inst}$ and $b^{\ell}_x \in B_{code}$ if and only if $\ell = \mathrm{col}_i(e)$ and there exists $0 \leq \alpha < d$ such that the expansion $(x_0, \dots, x_{d+1})$ of x in base dt satisfies $x_\beta = \alpha \cdot t + M_i[\alpha, \beta]$ on each digit $0 \leq \beta < d+2$. This way, each vertex $r_{e,i} \in R_{inst}$ is adjacent to precisely d vertices of B_{code}, one for each value $\alpha = 0, 1, \dots, d-1$. The final graph $H := (R \cup B, E)$ is then given by $R = R_{inst} \cup R_{enf}$, $B = B_{inst} \cup B_{enf}$, and $E = E_{inst} \cup E_{enf} \cup E_{conn}$. See also Figure 1.

To complete the description of the weak composition, we set $k = k' + n/2$, where $k' = (d+2)d(t-1) + n/2(d^{d+2} - d)$. Note that indeed $k \leq t \cdot n^{O(1)}$ as required by Definition 1.

Lemma 5. *H is $(d+2)$-degenerate.*

Proof. We show that the vertices in H can be ordered in such a way that each vertex has at most $d+2$ neighbors following it in this order. Observe that each vertex of $R_{inst} = \bigcup_{0 \leq i < T} R^i_{inst}$ is of degree exactly $d+2$ in H, since it is adjacent to exactly two vertices of B_{inst} and exactly d vertices of the enforcement graph. We place these vertices first in our ordering. Next, we add vertices of B_{inst}, as all neighbors of these vertices are already in the ordering. Therefore it is enough to argue about the $(d+2)$-degeneracy of the enforcement graph. We place next in the order vertices of $R_{fill} \cup B_{choice}$, since those are of degree exactly two in H. In $H \setminus R_{fill}$ the vertices of B_{fill} become isolated, so we put them next in our ordering. We are left with the vertices of the encoding gadget. Observe that each blue vertex b^{ℓ}_x of the encoding gadget has exactly $d+2$ neighbors in R_{code}, one for each position in the dt-ary expansion of x. We place the vertices of B_{code} next, and finally the vertices of R_{code}. □

3.3 Correctness

To complete the proof of Theorem 1, we argue that (H, k) is a YES-instance of RBDS if and only if there exists some $i \in \{0, \dots, T-1\}$ such that (G_i, col_i) is a YES-instance of MPM. For this, it will be helpful to distinguish two sets of blue vertices in B_{code} which we associate with each given identifier $M \in \{0, \dots, t-1\}^{d \times (d+2)}$.

– The first of these sets, $\tilde{B}_M \subseteq B_{code}$, is a set of size $d^{d+2} \cdot n/2$, and is defined as follows:

$$\tilde{B}_M = \left\{ b^{\ell}_x \in B_{code} \;\; : \;\; x = \sum_{0 \leq \beta < d+2} (\alpha_\beta \cdot t + M[\alpha_\beta, \beta])(dt)^\beta \text{ for some } \alpha_0, \dots, \alpha_{d+1} \in \{0, \dots, d-1\} \right\}.$$

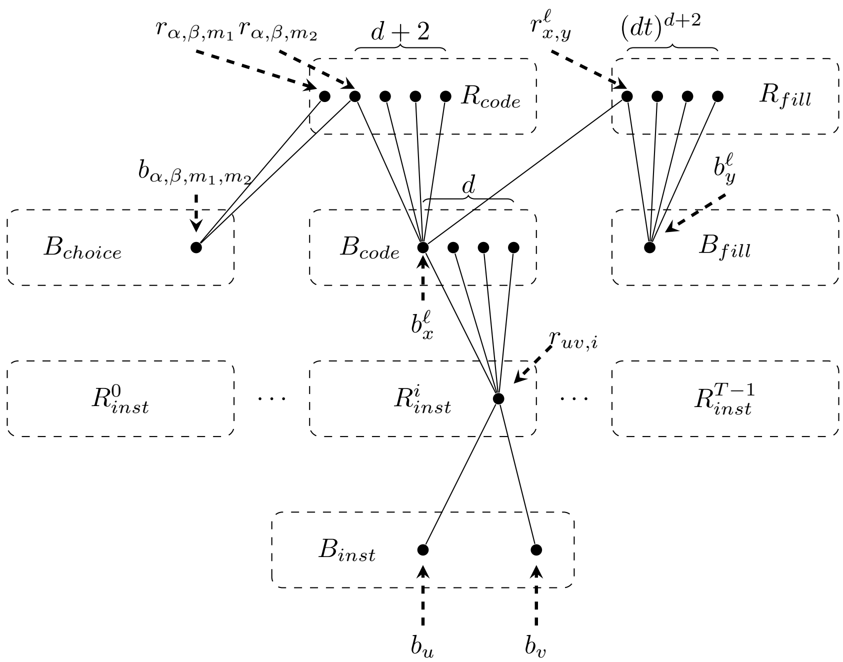

Fig. 1. A depiction of the construction of the graph H in the RBDS instance. The upper two layers correspond to H_{enf} and the bottom two to H_{inst}. A vertex $b^\ell_x \in B_{code}$ is adjacent to $r_{\alpha,\beta,m_1} \in R_{code}$ if the β'th digit in the base dt expansion of x equals $\alpha \cdot t + m_1$. There are $d+2$ vertices in B_{code} that fulfill this criterion. A vertex $b_{\alpha,\beta,m_1,m_2} \in B_{choice}$ is adjacent to two vertices $r_{\alpha,\beta,m_1}, r_{\alpha,\beta,m_2} \in R_{code}$, while a vertex $b^\ell_y \in B_{fill}$ is adjacent to $(dt)^{d+2}$ vertices $r^\ell_{x,y} \in R_{fill}$. In the instance graph, a vertex $r_{e,i} \in R_{inst}$, $e = \{u,v\}$, is adjacent to the pair of blue vertices $b_u, b_v \in B_{inst}$. Finally, there are some edges E_{conn} between H_{enf} and H_{inst}: Each $r_{e,i} \in R_{inst}$ is adjacent to d vertices $b^\ell_x \in B_{code}$ satisfying (a) $\mathrm{col}_i(e) = \ell$ and (b) there exists an $\alpha \in \{0,\ldots,d-1\}$ such that for every β the β'th digit in the base dt expansion of x equals $\alpha \cdot t + M_i[\alpha,\beta]$.

– The second set, $B_M \subseteq \tilde{B}_M$, is of size $d \cdot n/2$, and is defined by:

$$B_M = \Big\{ b_x^\ell \in B_{code} \ : \ x = \sum_{0 \le \beta < d+2} (\alpha \cdot t + M[\alpha,\beta])(dt)^\beta \text{ for some } \alpha \in \{0,\dots,d-1\} \Big\}.$$

For a color ℓ, we let $\tilde{B}_M^\ell$ and B_M^ℓ respectively denote the subsets of vertices in $\tilde{B}_M$ and B_M which are indexed by ℓ.

Lemma 6. *For any identifier $M \in \{0,\dots,t-1\}^{d\times(d+2)}$ there exists a set $D_{enf} \subseteq R_{enf}$ of size k' that dominates all vertices in $B_{enf} \setminus B_M$.*

Proof. For each $0 \le \beta < d+2$ and $0 \le \alpha < d$, add to D_{enf} the set $\{r_{\alpha,\beta,m} \in R_{code} : 0 \le m < t, m \ne M[\alpha,\beta]\}$ containing $t-1$ vertices of R_{code}. By construction, D_{enf} dominates all the vertices in B_{choice}. Consider a vertex $b_x^\ell \in B_{code}^\ell$ not dominated by D_{enf}, and let $x = (x_0,\dots,x_{d+1})$. Observe that for each digit $0 \le \beta < d+2$, there are at most d values that x_β can have. Indeed, for any $0 \le \beta < d+2$, we have $x_\beta \in X_\beta = \{\alpha t + M[\alpha,\beta] : 0 \le \alpha < d\}$, since otherwise b_x^ℓ would be dominated by D_{enf} due to the digit x_β. Moreover, if we consider any $b_{x'}^\ell \in B_{code}^\ell$, $x' = (x'_0,\dots,x'_{d+1})$, such that $x'_\beta \in X_\beta$ for $0 \le \beta < d+2$, then $b_{x'}^\ell$ is not dominated by the vertices added to D_{enf} so far. Thus, the vertices currently in D_{enf} dominate all vertices of B_{code} except for vertices in $\tilde{B}_M$.

Note that $B_M \subseteq \tilde{B}_M$, and that $|\tilde{B}_M \setminus B_M| = (d^{d+2} - d) \cdot n/2$. We next select, for each $1 \le y \le d^{d+2} - d$ and $0 \le \ell < n/2$, exactly one distinct vertex $b_x^\ell \in \tilde{B}_M \setminus B_M$, and add to D_{enf} the vertex $r_{x,y}^\ell \in R_{fill}$. Observe that after this operation D_{enf} dominates all vertices in B_{fill}, and moreover the only vertices of B_{code} not dominated by D_{enf} are the vertices of B_M. Since the total size of D_{enf} equals $d(d+2)(t-1) + (d^{d+2} - d)n/2 = k'$, the lemma follows. □

Lemma 7. *If there exists some $i \in \{0,\dots,T-1\}$ such that (G_i, col_i) is a YES-instance of* MPM *then (H,k) is a YES-instance of* RBDS.

Proof. Suppose (G_i, col_i) is a YES-instance of MPM, and let $E' \subseteq E_i$ be the corresponding solution of this instance. We apply Lemma 6 to the identifer M_i of (G_i, col_i) to obtain a set D_{enf} of size k' promised by the lemma. We construct a solution for (H,k) by taking the set $D = D_{enf} \cup D_{inst}$, where D_{inst} is defined by $D_{inst} = \{r_{e,i} : e \in E'\}$. Clearly $|D| = k$. To see that D dominates all blue vertices of H, note that as E' is a perfect matching, D_{inst} dominates B_{inst} by construction. Furthermore, by Lemma 6, D_{enf} dominates $B_{enf} \setminus B_M$. Finally, the set of neighbors of $r_{e,i}$ in B_{code} is exactly $B_M^{\mathrm{col}(e)}$. Thus, since E' is multicolored, D_{inst} dominates $\bigcup_{0 \le \ell < n/2} B_M^\ell = B_M$, and so D dominates $B_{inst} \cup B_{enf}$. □

We have shown one direction in the correctness of our composition: If one of the source instances is a YES-instance, then the target instance is a YES-instance as well. We next show the opposite direction: The only way to dominate all blue vertices of H with k red vertices is to select a subset of $n/2$ vertices in some R_{inst}^i, $0 \le i < T$, that correspond to a multicolored perfect matching in (G_i, col_i). This is established in the following four lemmas.

Lemma 8. *Any feasible solution D for the* RBDS *instance (H,k) can be partitioned into three sets $D = D_{code} \cup D_{fill} \cup D_{inst}$, where*

- *$D_{code} = D \cap R_{code}$ and $|D_{code}| = d(d+2)(t-1)$.*
- *$D_{fill} = D \cap R_{fill}$ and $|D_{fill}| = (d^{d+2} - d) \cdot n/2$.*
- *$D_{inst} = D \cap R_{inst}$ and $|D_{inst}| = n/2$.*

Proof. Let D be any feasible solution to (H,k). Observe that by construction, in order for D to dominate B_{choice}, it must be that $|D \cap \{r_{\alpha,\beta,m} : 0 \le m < t\}| \ge t-1$ for each $0 \le \alpha < d$ and $0 \le \beta < d+2$. Thus, in total at least $d(d+2)(t-1)$ vertices of R_{code} are included in D. Moreover, in order to dominate all vertices of B_{fill}, the solution D has to contain at least $(d^{d+2}-d) \cdot n/2$ vertices of R_{fill}. Finally, at least $n/2$ vertices of R_{inst} are needed to dominate B_{inst}, since each vertex in R_{inst} is adjacent to only two vertices from B_{inst}. Since $|D| \le k = d(d+2)(t-1) + (d^{d+2}-d) \cdot n/2 + n/2$, the set D contains no more vertices, and the lemma follows. □

Lemma 9. *Let D be a feasible solution for (H,k) with $D_{code}, D_{fill} \subseteq D$ as specified in Lemma 8. For each $0 \le \ell < n/2$, there exists an identifier $M \in \{0,\dots,t-1\}^{d(d+2)}$, such that there are at least d vertices $U^\ell \subseteq \tilde{B}^\ell_M$ which are not dominated by $D_{code} \cup D_{fill}$.*

Proof. By Lemma 8, the set D_{code} contains exactly $d(d+2)(t-1)$ vertices of R_{code}. More specifically, as argued in the proof of the lemma, D_{code} contains exactly $t-1$ vertices $r_{\alpha,\beta,m}$ for each fixed α and β. Consequently, for each pair (α,β) there is exactly one $m(\alpha,\beta) \in \{0,\dots,t-1\}$ such that $r_{\alpha,\beta,m(\alpha,\beta)} \notin D_{code}$. Let M be the identifer defined by $M[\alpha,\beta] = m(\alpha,\beta)$ for each α and β. By construction, we infer that for each ℓ, the set $B^\ell_{code} \setminus N(D_{code})$ contains exactly d^{d+2} vertices, namely the d^{d+2} vertices of $\tilde{B}^\ell_M$. Since D_{fill} dominates at most $d^{d+2}-d$ vertices of $\tilde{B}^\ell_M$, for each $0 \le \ell < n/2$, the lemma holds. □

Lemma 10. *Let D be a feasible solution for (H,k) with $D_{inst} \subseteq D$ as specified in Lemma 8. Then D_{inst} is* multicolored*; that is, for each color ℓ there exists some $r_{e,i} \in D_{inst}$ with $col_i(e) = \ell$.*

Proof. Fix ℓ to be some specific color. By Lemma 9, there is a non-empty subset of vertices $U^\ell \subseteq \tilde{B}^\ell_M$ which are not dominated by $D \setminus D_{inst}$. The lemma follows by observing that the only way for D_{inst} to dominate vertices in U^ℓ is if there is some $r_{e,i} \in D_{inst}$ with $\text{col}_i(e) = \ell$. □

Lemma 11. *If (H,k) is a YES-instance of* RBDS *then (G_i, col_i) is a YES-instance of* MPM *for some $i \in \{0,\dots,T-1\}$.*

Proof. Let D be a feasible solution to (H,k), and let $D_{enf} = D \cap R_{enf}$ and $D_{inst} = D \cap R_{inst}$. Then by Lemma 8, we have $|D_{enf}| = k'$ and $D_{inst} = n/2$. Let M be the identifier given by Lemma 9, and let (G_i, col_i) denote the RBDS instance associated with M; that is, $M = M_i$. Our goal is to show that $D_{inst} \subseteq R^i_{inst}$. If this is in fact the case, then the set of edges $E' = \{e \in E_i : r_{e,i} \in D_{inst}\}$ is a solution to (G_i, col_i). Indeed, we have $|E'| = |D_{inst}| = n/2$, and E' is multicolored since D_{inst} is multicolored according to Lemma 10. Furthermore, E' is a perfect matching in G_i since we must have $\{v \in V : v \in e \text{ for some } r_{e,i} \in D_{inst}\} = V$; for otherwise, D_{inst} would not dominate all vertices in B_{inst}.

Suppose then that $D_{inst} \cap R^j_{inst} \neq \emptyset$ for some $j \neq i$, and let α_0 and β_0 be such that $M_i[\alpha_0,\beta_0] \neq M_j[\alpha_0,\beta_0]$. Choose a color ℓ such that there exists some $r_{e,j} \in D_{inst} \cap R^j_{inst}$ with $\text{col}_j(e) = \ell$. By Lemma 9, there is a set $U^\ell \subseteq \tilde{B}^\ell_{M_i}$ of at least d vertices not dominated by D_{enf}, hence $U^\ell \subseteq N(D_{inst})$. Moreover, by Lemma 10, the set D_{inst} is multicolored, and consequently $r_{e,j}$ is the only vertex of D_{inst} satisfying $\text{col}_j(e) = \ell$. As $U^\ell \subseteq N(D_{inst})$, we infer that in fact $U^\ell \subseteq N(r_{e,j})$. However, by construction, each vertex of R_{inst} has exactly d neighbors in B_{code}. Since $|U^\ell| \ge d$, we have $|N(r_{e,j}) \cap B_{code}| = d = |U^\ell|$, and consequently $N(r_{e,j}) \cap B_{code} = U^\ell$. Let

$$x = \sum_{0 \le \beta < d+2} (\alpha_0 \cdot t + M_j[\alpha_0,\beta]) = (x_0,\dots,x_{d+1})\,.$$

Note that for each $0 \leq \alpha < d$ we have $x_{\beta_0} \neq \alpha \cdot t + M_i[\alpha, \beta_0]$. Consequently $b_x^\ell \notin \tilde{B}_{M_i}$ and $b_x^\ell \notin U^\ell$. This contradicts $N(r_{e,j}) \cap B_{code} = U^\ell$, as by the definition of x the vertex b_x^ℓ is a neighbor of $r_{e,j}$ in $B_{code} \setminus U^\ell$. □

We now have all the ingredients to prove Theorem 1.

Proof. (of Theorem 1) Consider the above construction. It is clear that it can be carried out in polynomial time in T and n. The resulting graph H is $(d+2)$-degenerate by Lemma 5, and k is bounded by $t \cdot n^{O(1)}$. Together with Lemmas 7 and 11 this gives a weak $d(d+2)$-composition from MPM to RBDS in $(d+2)$-degenerate graphs. Using Lemma 4, we infer that there exists a weak $d(d+2)$-composition from MPM to DS in $(d+3)$-degenerate graphs. The claim then follows directly from Lemma 1. □

4 Independent Dominating Set

Recall that the Independent Dominating Set (IDS) problem is the variant of DS where we require the dominating set D to induce an independent set (i.e. vertices in D have to be pairwise non-adjacent). In this section we present a weak d-composition from 3-Exact Set Cover to IDS in $(d+4)$-degenerate graphs, which will show that the IDS in d-degenerate graphs problem is unlikely to have kernels of size $O(k^{d-4-\varepsilon})$ for any constant $\varepsilon > 0$.

The input of 3-Exact Set Cover (3-ESC) is a set system $(U, \mathcal{F})$, where each set in $\mathcal{F}$ contains exactly three elements from U. The goal is to determine whether there is a collection $\mathcal{S} \subseteq \mathcal{F}$ of disjoint sets that partition U, i.e. $\bigcup_{S \in \mathcal{S}} S = U$. The 3-ESC problem is NP-complete as it generalizes the already mentioned NP-complete 3-Dimensional Perfect Matching problem. Consider a fixed value of $d \geq 1$ and let $(U_0, \mathcal{F}_0), \ldots, (U_{T-1}, \mathcal{F}_{T-1})$ be $T := t^d$ instances of 3-ESC. Without loss of generality, we can assume that the same universe U of size $n \equiv 0 \pmod 3$ is used in each instance. Indeed, if for some i we have $|U_i| \not\equiv 0 \pmod 3$, then $(U_i, \mathcal{F}_i)$ is clearly a NO-instance. Moreover, we can add additional triples to make sure that each U_i has the same cardinality n. Finally, we can rename vertices so that $U_i = U$ for all i.

We construct an instance $(H = (V, E), k)$ of IDS for a properly chosen parameter k. Similarly to the construction in Section 3, the graph H will consist of an *instance graph* $H_{inst} = (V_{inst}, E_{inst})$, and an *enforcement gadget* (H_{enf}, E_{conn}), where $H_{enf} = (V_{enf}, E_{enf})$ is an (*enforcement*) graph and E_{conn} is a set of edges between H_{inst} and H_{enf} (see also Figure 2). In more detail:

- The graph H_{inst} contains the vertex set $V_{univ} := \{v_u : u \in U\}$ (i.e., one vertex per element of the universe). Furthermore, it contains a vertex set $V_{triple} = \bigcup_{0 \leq i < T} V_{triple}^i$, where V_{triple}^i contains one vertex $v_{i,S}$ for every triple of elements $S \in \binom{U}{3}$ of the universe U. We add an edge between vertices v_u and $v_{i,S}$ if and only if $S \in \mathcal{F}_i$ and $u \in S$. Observe that there might be some isolated vertices in V_{triple}.

- The graph H_{enf} consists of two induced matchings. It contains an *encoding matching* X_{code} with edges $\{x_{\alpha,\beta}, x'_{\alpha,\beta}\}$ for all integers $0 \leq \alpha < t$ and $0 \leq \beta < d$. Furthermore, it contains a *choice matching* Y_{choice} with edges $\{y_S, y'_S\}$ for all triples $S \in \binom{U}{3}$.

- The edge set E_{conn} is constructed as follows: First of all, for each $S \in \binom{U}{3}$, we add an edge between y_S and every $v_{i,S}$. Second, we add an edge between $x_{\alpha,\beta}$ and $v_{i,S}$ if and only if $i_\beta = \alpha$, where $(i_0, \ldots, i_{d-1})$ is the t-ary expansion of index i (*i.e.* $i = \sum_{\beta=0}^{d-1} i_\beta \cdot t^\beta$ with $0 \leq i_\beta < t$).

To conclude the construction, we set $k := dt + \binom{n}{3} + n/3$.

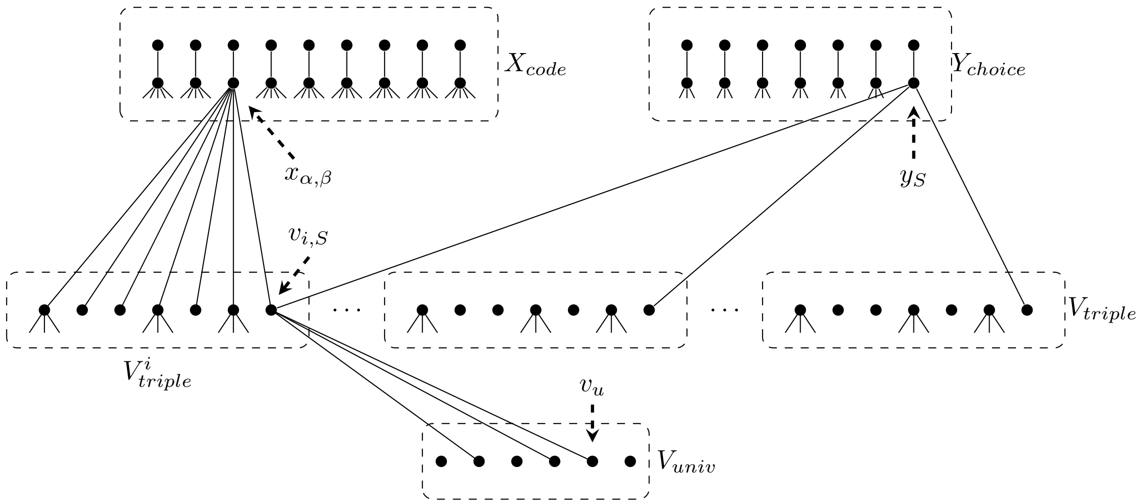

Fig. 2. A graphical depiction of the construction of the graph H in the composition for Independent Dominating Set. A vertex v_u corresponding to an element $u \in U$ of the universe common to all 3-ESC instances is adjacent to a vertex $v_{i,S}$ if and only if S is part of the i'th 3-ESC instance and $u \in S$. The vertex $v_{i,S}$ is in turn adjacent to $y_S \in Y_{choice}$, and to each $x_{\alpha,\beta} \in X_{code}$ where the β'th digit of the t-ary expansion of i equals α.

Lemma 12. *The graph H is $d+4$ degenerate.*

Proof. Consider any ordering of vertices of H where we put vertices of V_{triple} first, and then the remaining vertices. Observe, that each vertex of V_{triple} is of degree at most $d+4$ in H, since it has exactly d neighbors in X_{code}, exactly one neighbor in Y_{choice} and zero or three neighbors in V_{univ}. After removing V_{triple}, all vertices have degree at most 1. The claim follows. $\square$

Lemma 13. *(H, k) is a YES-instance of* IDS *if and only if there exists $0 \leq j < T$, such that $(U, \mathcal{F}_j)$ is a YES-instance of* 3-ESC.

Proof. For the if part, let us assume that the instance $(U, \mathcal{F}_j)$ of 3-ESC is a YES-instance for some $0 \leq j < T$, and let $\mathcal{S} \subseteq \mathcal{F}_j$ be a solution (*i.e.* a collection of $n/3$ disjoint sets). Construct an independent dominating set D in H of exactly k vertices as follows. Let $(j_0, \ldots, j_{d-1})$ be the t-ary expansion of j. For $0 \leq \alpha < t$ and $0 \leq \beta < d$, add $x'_{\alpha,\beta}$ to D if $j_\beta = \alpha$. Otherwise, add to D the vertex $x_{\alpha,\beta}$. For $S \in \binom{U}{3}$, add y'_S to D if $S \in \mathcal{S}$ and add y_S to D otherwise. Finally, add to D the $n/3$ vertices $v_{j,S}$ with $S \in \mathcal{S}$.

Clearly $|D| = k$. Moreover D is an independent set. In fact, $H[X_{code} \cup Y_{choice}]$ is a matching and we have taken exactly one endpoint of each one of its edges. Moreover for each $v_{j,S} \in D$ by construction there is no edge between $v_{j,S}$ and the remaining vertices of D. To prove that D is a dominating set observe that all the vertices of V_{univ} are dominated because $\mathcal{S}$ is a solution for $(U, \mathcal{F}_j)$. Furthermore, all the vertices of $X_{code} \cup Y_{choice}$ are dominated because D contains exactly one endpoint of each edge of the matching $H[X_{code} \cup Y_{choice}]$. Finally, each vertex $v_{i,S}$ is dominated. Indeed, if $i \neq j$ then $v_{i,S}$ is dominated by $X_{code} \cap D$ due to the coordinate at which i and j differ. Otherwise, $S \in \mathcal{S}$ and then $v_{j,S} \in D$, or $S \notin \mathcal{S}$ and then $y_S \in D$.

For the only if part, let D be an independent dominating set in H of size at most k. Observe that since each vertex $x'_{\alpha,\beta}$ and y'_S is of degree exactly one in H, we have $|D \cap X_{code}| \geq dt$ and $|D \cap Y_{choice}| \geq \binom{n}{3}$. Moreover, since no vertex has more than three neighbors in V_{univ}, we have $|D \cap (V_{univ} \cup V_{triple})| \geq n/3$. Therefore, all the three mentioned inequalities are tight. Moreover D

contains exactly $n/3$ vertices of V_{triple}. Indeed, a vertex of V_{univ} dominates only a single vertex of V_{univ}, namely itself, therefore if D would contain a vertex of V_{univ} then $|D \cap (V_{univ} \cup V_{triple})|$ would be strictly greater than $n/3$.

Define $\mathcal{S} = \{S \in \binom{U}{3} \mid \exists 0 \leq i < T : v_{i,S} \in D\}$. Observe, that for $i_1 \neq i_2$ vertices $v_{i_1,S}$ and $v_{i_2,S}$ dominate exactly same vertices in V_{univ}. Since $|D \cap V_{triple}| = n/3$, we infer that $|\mathcal{S}| = n/3$. We want to show that there exists $0 \leq j < T$ such that $\mathcal{S} \subseteq \mathcal{F}_j$, which is enough to prove that $(U, \mathcal{F}_j)$ is a YES-instance. Assume the contrary. Then there exist two indices $0 \leq i_1 < i_2 < T$, such that there exist $S_1, S_2 \in \mathcal{S}$ with $v_{i_1,S_1} \in D$ and $v_{i_2,S_2} \in D$. Since D is an independent set, and since a vertex $v_{i,S}$ is adjacent to y_S, no vertex of $V' := \{v_{i,S} : i \in \{i_1, i_2\}, S \in \mathcal{S}\}$ has a neighbor in $D \cap Y_{choice}$. Also, since D is an independent set by definition of connections between X_{code} and V_{triple} we know that V' has no neighbors in $D \cap X_{code}$. Since V' is not dominated by $D \cap Y_{choice}$ nor by $D \cap X_{code}$ we know that V' is dominated by itself, i.e. $V' \subseteq D$. However, $|V'| = 2|\mathcal{S}| = 2n/3 > n/3$, a contradiction. □

Theorem 2. *Let $d \geq 5$. The* Independent Dominating Set *problem in d-degenerate graphs has no kernel of size $O(k^{d-4-\varepsilon})$ for any constant $\varepsilon > 0$ unless* coNP $\subseteq$ NP/poly.

Proof. From Lemmas 12 and 13, there exists a weak d-composition from 3-ESC to IDS in $(d+4)$-degenerate graphs for any $d \geq 1$. The claim thus follows from Lemma 1. □

5 Induced Matching

In this section we show a kernelization lower bound for the Induced Matching (IM) problem in d-degenerate graphs. Recall that in IM, the input is a graph G and an integer k, and the goal is to determine whether there exists a matching $M = \{e_1, \ldots, e_k\}$ of size k in G such that there is no edge in G with one endpoint in e_i and the other in e_j for any two distinct $i, j \in \{1, \ldots, k\}$. The main result of this section is given by the following theorem.

Theorem 3. *Let $d \geq 3$. The* Induced Matching *problem in d-degenerate graphs has no kernel of size $O(k^{d-3-\varepsilon})$ for any constant $\varepsilon > 0$ unless* coNP $\subseteq$ NP/poly.

For our kernelization lower bound, we present a weak d-composition from the following Multicolored Clique problem: The input is a graph $G := (V, E)$ and a vertex-coloring $\text{col} : V \to \{1, \ldots, \ell\}$. Our goal is to determine whether there exists a multicolored clique of size ℓ in G. We recall that a clique is a subset of pairwise adjacent nodes. We say that a clique $v_1, \ldots, v_\ell$ is *multicolored* if $\text{col}(v_i) \neq \text{col}(v_j)$ for all distinct $i, j \in \{1, \ldots, \ell\}$. It is well known that Multicolored Clique is NP-complete [18].

Let $(G_i = (V_i, E_i), \text{col}_i)$, $0 \leq i < T := t^d$, be the input instances of Multicolored Clique. By standard padding and vertex-renaming arguments (similarly to previous sections), we can assume that all graphs G_i are defined over the same vertex set V of size n, and that each vertex $v \in V$ is assigned the same color $\text{col}(v) \in \{1, \ldots, \ell\}$ by all coloring functions col_i's. We can further assume that for each $\{u, v\} \in E_i$ we have $\text{col}(u) \neq \text{col}(v)$, since edges between vertices of the same color can never appear in any multicolored clique. Finally, we also assume that $\binom{\ell}{2} - \ell > d$, since otherwise this would imply that ℓ is constant (as d is constant), and each instance can be solved separately in polynomial time. Once a solution to each instance is obtained, a weak d-composition can trivially be constructed.

We next construct an instance (H, k) of IM for a properly chosen parameter k (see also Figure 3). As in previous sections, H consists of an *instance graph* $H_{inst} = (A \cup B, E_{inst})$ which is bipartite, along with an *enforcement gadget* (H_{enf}, E_{conn}) which consists of a bipartite graph $H_{enf} = (X \cup Y, E_{enf})$ and a set E_{conn} of properly chosen edges between H_{inst} and H_{enf}. The role of the instance graph is to guarantee that feasible solutions to any MULTICOLORED CLIQUE instance induce feasible solutions to the IM instance. The enforcement gadget ensures that we cannot combine partial solutions of different MULTICOLORED CLIQUE instances to obtain a feasible solution for the IM instance.

The vertex set of H_{inst} is defined by $A = A_{vertices} \cup A_{edges}$ and $B = B_{vertices} \cup B_{col-pairs}$. The sets $A_{vertices}$ and $B_{vertices}$ contain one vertex a_v and b_v, respectively, for each $v \in V$. We have $A_{edges} = \bigcup_{0 \le i < T} A^i_{edges}$, where the set A^i_{edges} contains a vertex $a_{i,e}$ for each instance i and $e \in E_i$. The set $B_{col-pairs}$ contains a vertex $b_{\alpha,\beta} \in B$ for every pair of colors $1 \le \alpha < \beta \le \ell$. The set E_{inst} contains all the edges of type $\{a_v, b_v\}$, plus edges between each $a_{i,e} = a_{i,uv}$ and vertices b_u, b_v and $b_{\mathrm{col}(u),\mathrm{col}(v)}$. The bipartite graph $H_{enf} = (X \cup Y, E_{enf})$ is defined as follows: The sets X and Y contain a vertex $x_{\alpha,\beta}$ and $y_{\alpha,\beta}$ respectively, for all integers $0 \le \alpha < t$ and $0 \le \beta < d$, and the edge set E_{enf} contains all edges between each pair $\{x_{\alpha,\beta}, y_{\alpha,\beta}\}$. The set of edges E_{conn} that connects H_{inst} and H_{enf} is constructed as follows: We add an edge between $x_{\alpha,\beta}$ and $a_{i,e}$ if and only if $i_\beta = \alpha$, where $(i_0, i_1, \ldots, i_{d-1})$ is the t-ary expansion of index i. Observe that each $a_{i,e}$ is adjacent to exactly d distinct vertices $x_{\alpha,\beta}$ in X. We conclude the construction by setting $k := (t-1)d + \binom{\ell}{2} + n - \ell$.

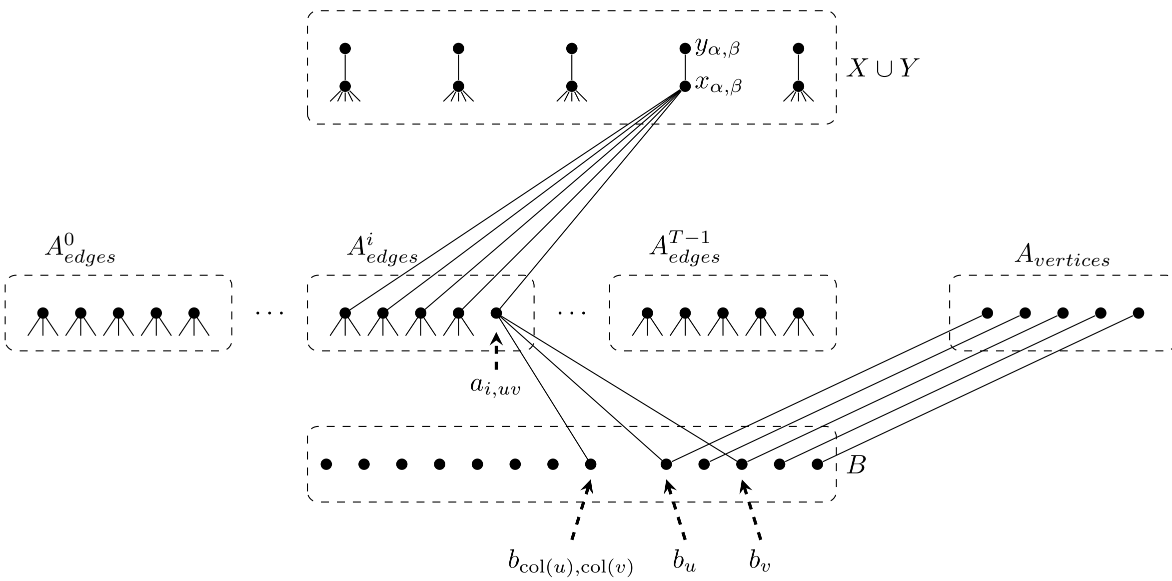

Fig. 3. The construction of the graph H in the composition for INDUCED MATCHING. In H_{inst}, the vertex $a_{i,uv}$ is adjacent to b_u, b_v, and $b_{\mathrm{col}(u),\mathrm{col}(v)}$. The subgraph H_{enf} consists of a matching $\{\{x_{\alpha,\beta}, x_{\alpha,\beta}\} : 0 \le \alpha < t, 0 \le \beta < d\}$. The vertex $x_{\alpha,\beta}$ is adjacent to $a_{i,uv}$ if and only if the β'th digit of the t-ary expansion of i equals α.

Lemma 14. *If for some $0 \le i < T$ there exists a multicolored clique of size ℓ in (V, E_i), then there is an induced matching M' in H_{inst} of size $\binom{\ell}{2} + n - \ell$, such that $V(M') \cap A_{edges} \subseteq A^i_{edges}$.*

Proof. Let V' be a multicolored clique of size ℓ in (V, E_i). As M' we take

$$M' = \{\{a_v, b_v\} : v \in V \setminus V'\} \cup \{\{a_{i,uv}, b_{\mathrm{col}(u),\mathrm{col}(v)}\} : u, v \in V' \wedge u \ne v\} .$$

A direct check shows that M' is indeed an induced matching in H_{inst}, while the size of M' equals $\binom{\ell}{2} + n - \ell$ by construction.

Lemma 15. *Let M' be an induced matching in H_{inst} which does not contain any edge between A_{edges} and $B_{vertices}$. Then $|M'| \leq \binom{\ell}{2} + n - \ell$. Moreover, if equality holds, then the graph with vertex set $V' = \{v \in V : b_v \notin V(M')\}$ and edge set $E' := \{e : a_{i,e} \in V(M') \text{ for some } i \in \{0, \ldots, T-1\}\}$ is a multicolored clique of the graph $(V, \cup_i E_i)$.*

Proof. Let M' be a maximum induced matching in H_{inst}. Clearly $|M'| \leq n + \binom{\ell}{2}$, as $|B| = n + \binom{\ell}{2}$. If $|M'| = n$ then we are done, so assume $|M'| > n$. Thus, there are vertices of A_{edges} that are matched in M'. Recall, that in the statement of the lemma we assumed that M' does not contain edges between A_{edges} and $V_{vertices}$, hence there is a vertex in $B_{col-pairs}$ that is matched in M'. Consider some edge $m \in M'$ which includes some vertex $b_{\ell_1,\ell_2} \in B_{col-pairs}$. Then by construction, $m = \{b_{\ell_1,\ell_2}, a_{i,e}\}$, where $a_{i,e}$ corresponds to an edge $e = \{u, v\} \in E_i$ with $\mathrm{col}(u) = \ell_1$ and $\mathrm{col}(v) = \ell_2$. As $\{b_u, a_{i,e}\}, \{b_v, a_{i,e}\} \in E_{inst}$, the vertices b_u and b_v cannot be matched in M'. Thus, for each vertex b_{ℓ_1,ℓ_2} that is matched in M', a vertex of $\{b_u : u \in V, \mathrm{col}(v) = \ell_1\}$ is not matched in M', and a vertex of $\{b_v : v \in V, \mathrm{col}(v) = \ell_2\}$ is not matched in M'. In other words, if $c \in \{1, \ldots, \ell\}$ is the number of colors appearing in color-pair indices of vertices in $B_{col-pairs}$ that are matched in M', then there are at least c vertices in $B_{vertices}$ that are not matched in M'. Since the maximum of $\binom{c}{2} - c$ over the range $c \in \{1, \ldots, \ell\}$ is obtained when $c = \ell$, we obtain the bound $|M'| \leq n + \binom{\ell}{2} - \ell$ which is stated in the lemma.

Suppose now that $|M'| = n + \binom{\ell}{2} - \ell$, and let V' and E' be defined as in the statement of the lemma. Then M' is a maximum size induced matching in H_{inst}, and $|V'| = \ell$ by the above discussion. Moreover, we also know that $|E'| = \binom{\ell}{2}$, as there are $\binom{\ell}{2}$ vertices of $B_{col-pairs}$ that are matched in M', and these can only be matched to vertices of $\cup_i A^i_{edges}$ by construction. Finally by the definition of V' and E' and by the fact that M' is an induced matching, we infer that each edge of E' has both endpoints in V'. By the cardinality of E' this implies that (V', E') is a multicolored clique of size ℓ in $(V, \cup_i E_i)$. ◻

Lemma 16. *H is $(d+3)$-degenerate.*

Proof. Consider any ordering of the vertices which places first vertices of $A_{vertices}$ and Y, then vertices of A_{edges}, then vertices of B, and finally vertices X. It is not difficult to check that each vertex is adjacent to at most $d+3$ vertices appearing to its right in this ordering. ◻

Lemma 17. *(H, k) is a YES-instance of* IM *if and only if (G_i, col_i) is a YES-instance of* Multicolored Clique *for some index $i \in \{0, \ldots, T-1\}$.*

Proof. Suppose some G_i has a multicolored clique of size ℓ. Then by Lemma 14 we can find an induced matching M' of size $\binom{\ell}{2} + n - \ell$ in H_{inst} such that $V(M') \cap A_{edges} \subseteq A^i_{edges}$, i.e. such that among A_{edges} the matching M' matches only vertices of type $a_{i,e}$. Let us add all the edges $\{x_{\alpha,\beta}, y_{\alpha,\beta}\}$ such that $i_\beta \neq \alpha$. There are precisely $d(t-1)$ such edges, which together with the edges of M', form an induced matching of size k in H.

For the converse direction, suppose M is an induced matching of size k in H. First, observe that we can assume that M does not contain any edge of E_{conn}, since any such edge $\{a_{i,e}, x_{\alpha,\beta}\}$ can be safely replaced in M with the edge $\{x_{\alpha,\beta}, y_{\alpha,\beta}\}$. Similarly, we may assume that M does not contain any edge between A_{edges} and $B_{vertices}$, as any such edge $\{a_{i,uv}, b_u\}$ can be replaced with $\{a_u, b_u\}$. Now, as $\binom{\ell}{2} - \ell > d$, the matching M must include some edge between B and A_{edges}, since there are $n + td < k$ edges altogether in $H_{inst}[A_{vertices} \cup B]$ and H_{enf}. So let $a_{i,e}$ be a vertex of

A_{edge} which is matched in M. Then as $a_{i,e}$ is matched in M, this means that d vertices of X cannot be matched in M, precisely those vertices $x_{\alpha,\beta}$ with $i_\beta = \alpha$. Thus, M includes at most $d(t-1)$ edges from E_{enf}. Since the remaining edges of M are edges in H_{inst}, and since a maximum induced matching in H_{inst} has size at most $\binom{\ell}{2} + n - \ell$ by Lemma 15, this implies that M contains exactly $\binom{\ell}{2} + n - \ell$ edges of H_{inst} and $d(t-1)$ edges of H_{enf}. By construction of E_{conn}, the latter assertion implies that there exists some i such that each vertex of A_{edges} that is matched by M is of the form $a_{i,e}$ for some $e \in E_i$. By Lemma 15, this together with the the former assertion implies that $\{v \in V : b_v \notin V(M)\}$ is a multicolored clique of size ℓ in G_i. □

Proof. (of Theorem 3) Consider the above construction. It can be trivially performed in polynomial time. The claim then follows from Lemmas 16, 17, and 1.

6 Connected and Capacitated Vertex Cover

In this section we consider the Connected Vertex Cover (ConVC) and Capacitated Vertex Cover (CapVC) problems. Both problems are known not to have polynomial kernels in general graphs [12]. For ConVC an upper bound of $O(k^{d+1})$ and a lower bound of $O(k^{d-1-\varepsilon})$ for the size of a kernel were shown for d-degenerate graphs in [8]. We improve the upper bound of [8] for ConVC to $O(k^d)$, and show an upper bound of $O(k^d)$ and lower bound of $O(k^{d-1-\varepsilon})$ for CapVC. Both upper bounds that we present rely on the following simple lemma.

Lemma 18. *Let $G := (A \cup B, E)$ be a bipartite d-degenerate graph where all vertices in B have degree greater than d. Then $|B| \leq d|A|$.*

Proof. By the d-degeneracy of G, we know that $|E| \leq d(|A| + |B|)$. Since each vertex of B has degree greater than d, we also know that $(d+1)|B| \leq |E|$. Subtracting $d|B|$ from both inequalities gives $|B| \leq d|A|$. □

Let us begin with the kernel for ConVC. Let (G, k) be an instance of ConVC. Recall that our goal is to determine whether G has a set of k vertices A such that $G[V(G) \setminus A]$ is edgeless (A is a *vertex cover*) and $G[A]$ is connected (A is *connected*). Our kernelization algorithm uses two simple reduction rules which are given below, the second of which is a variant of the well-known *crown reduction rule* [16]. We say that a set of vertices $S \subseteq V(G)$ is a set of *twins* in G if $N(v) = N(u)$ for all $u, v \in S$ (note that this implies that S is an independent set in G).

Rule 1. *If G has an isolated vertex remove it.*

Rule 2. *If $S \subseteq V(G)$ is a set of at least two twin vertices with $|N(S)| < |S|$, remove an arbitrary vertex of S from G.*

Rule 1 can be trivially implemented in polynomial time. Rule 2 can be implemented in $O(n^3)$ time as follows: First, one can determine the twins of each vertex $v \in V(G)$ in $O(n^2)$ by a simple scan of the neighborhoods of all other vertices in G. This results in a partitioning of $V(G)$ into twin sets in $O(n^3)$ time. The size of the common neighborhood of each twin class can then be determined in $O(n^2)$ time. While clearly there are more efficient implementations of Rule 2, it suffices for us that it can be implemented in polynomial-time. In particular, we can apply both reduction rules exhaustively, until none can be applied, in polynomial time.

Lemma 19. *Let G be a graph, and let G' be a graph resulting from applying either Rule 1 or Rule 2 to G. Then for any integer k, the graph G has a connected vertex cover of size at most k if and only if G' has a connected vertex cover of size at most k.*

Proof. The claim is obvious for Rule 1, so let us focus on Rule 2. Let G, G', and S be as in the statement of the rule, and let $s \in S$ be such that $V(G) = V(G') \cup \{s\}$. Also, let $S' = S \setminus \{s\}$, and let $s_0 \in S'$ be a distinct twin of s (which is guaranteed to exist as $|S| > 1$). Note that G is connected iff G' is connected, since $s_0 \in V(G')$ is adjacent to all neighbors of s. To prove the lemma, we argue that any connected vertex cover of G can be transformed into a connected vertex cover of G' with equal size, and vice versa.

For the first direction, let A be a connected vertex cover of G. If $s \notin A$, then $A \subseteq V(G')$, and we are done. Assume then that $s \in A$, and consider the set of vertices $A' = (A \setminus \{s\}) \cup \{s_0\}$. Then $A' \subseteq V(G')$, and $|A'| \leq |A|$. Moreover, A' is a vertex cover of G', as already $A \setminus \{s\}$ is a vertex cover of G'. Since A is connected, and since s and s_0 are twins, we infer that A' is connected in G'. Conversely, assume that B' is a connected vertex cover in G'. If $N(S) \subseteq B'$, then B' is also a connected vertex cover of G, as all edges involving s are covered by $N(S)$. If $N(S) \nsubseteq B'$, then B' must include all vertices of S' to cover all edges between S' and $N(S)$. Furthermore, for B' to be connected, it must also include at least one vertex of $N(S)$, as $S' \subseteq B'$ is only adjacent to $N(S)$ in G'. Thus, we have $|B| \leq |B'|$ for $B = (B' \setminus S') \cup N(S) \cup \{s_0\}$. The proof is completed by noting that B is a vertex cover of G as S is an independent set, and it is connected since B' is connected and vertices in S' are only adjacent to vertices in $N(S)$. Thus, B is a connected vertex cover of G with $|B| \leq |B'|$. □

Theorem 4. ConVC *in d-degenerate graphs has a polynomial-time computable kernel of size $O(k^d)$.*

Proof. Our kernelization algorithm for Connected Vertex Cover in d-degenerate graphs exhaustively applies Rule 1 and Rule 2 until they no longer can be applied. Note that exhaustively applying Rules 1 and 2 takes polynomial time. Let G be the original graph, while let G' be the graph obtained after all reductions. Observe that both reduction rules that were used do not increase the degeneracy of the graph, and so G' is d-degenerate as well. Furthermore, due to Lemma 19, we know that G has a connected vertex cover of size k if and only if G' has one as well. We next show that $|V(G')| = O(k^d)$, or otherwise G' has no connected vertex cover of size k.

Suppose that G' has a connected vertex cover A of size k. Then as A is a vertex cover, the set $B := V(G') \setminus A$ is an independent set in G'. For $i := 0, \dots, d$, define $B_i \subseteq B$ to be the set of all vertices in B with degree i in G', and let $B_{>d} \subseteq B$ be the vertices in B with degree greater than d in G'. Then $B := B_0 \cup \dots \cup B_d \cup B_{>d}$, and $|B_0| = 0$ since Rule 1 cannot be applied. Due to Rule 2, for each subset of i vertices $A' \subseteq A$, $1 \leq i \leq d$, there are at most i vertices $b \in B_i$ with $N(b) = A'$. As there are $\binom{k}{i}$ such sets A', we have $|B_i| \leq i\binom{k}{i}$, which implies that $\sum_{i:=0,\dots,d} |B_i| \leq dk^d$. Furthermore, we also have $|B_{>d}| \leq dk$ by applying Lemma 18 to the bipartite graph on A and $B_{>d}$. Accounting also for A, we get

$$|V(G')| = |A| + |B| \leq k + \sum_{i:=0,\dots,d} |B_i| + |B_{>d}| \leq k + dk^d + dk = O(k^d),$$

and the claim follows, since in d-degenerate graphs the number of edges is at most d times larger than the number of vertices. □

Next we consider CapVC. Recall that in this problem we are given a graph G, an integer k, and a vertex capacity function $cap : V(G) \to \mathbb{N}$. The goal is to determine whether there exists a vertex cover of size at most k where each vertex covers a number of edges not larger than its capacity. That is, whether there is a vertex cover C of size k and a function $\alpha : E(G) \to C$ mapping each edge of $E(G)$ to one of its endpoints (belonging to C) such that $|\alpha^{-1}(v)| \leq cap(v)$ for every $v \in V(G)$.

For CapVC we replace Rule 2 with a similar crown-reduction-like rule that is tailored to the problem at hand. We may assume that $k + 2 > d$, since otherwise a kernel is trivially obtained by solving the problem in polynomial time.

Rule 3. *If $S \subseteq V(G)$ is a subset of twin vertices with a common neighborhood $N(S)$ such that $|S| = k + 2 \geq |N(S)|$, remove a vertex with minimum capacity in S from G, and decrease all the capacities of vertices in $N(S)$ by one.*

Lemma 20. *Let $k \geq 1$ be an arbitrary integer, let G be a vertex capacitated graph, and let G' be a vertex capacitated graph resulting from applying either Rule 1 or Rule 3 to G. Then G has a capacitated vertex cover of size k if and only if G' has a capacitated vertex cover of size k.*

Proof. The claim is trivial for Rule 1. Next consider Rule 3. Let A be a capacitated vertex cover of size k in G, and let s be a vertex of minimum capacity in S. As $|S| > k$, there is some $s' \in S \setminus A$, and this implies that $N(S) \subseteq A$ in order for A to cover all edges which include s'. Moreover, each vertex in $v \in N(S)$ can cover at most $cap(v) - 1$ edges that do not include s. This means that A is also a capacitated vertex cover of G' if $s \notin A$. Otherwise, if $s \in A$, we can replace s with s' in A. As $cap(s) \leq cap(s')$, and s and s' are twins, this would result in another capacitated vertex cover for G which is also a capacitated vertex cover for G'. Conversely, any capacitated vertex of size k in G' must also include all vertices of $N(S)$, since it cannot include all $k + 1$ in $S \cap V(G')$. Thus, by increasing the capacities of all these vertices by one, all edges involving s can be covered, and we get a capacitated vertex cover of size k for G. ◻

Theorem 5. CapVC *has a polynomial-time computable kernel of size $O(k^{d+1})$ in d-degenerate graphs.*

Proof. The claim follows along the same line as the proof of Theorem 4. The only difference is that now the size of sets B_i is bounded by $(k+1)\binom{k}{i}$ instead of $d\binom{k}{i}$, which yields a kernel of size $O(k^{d+1})$ instead of $O(k^d)$. ◻

The following complementing lower bound follows from a simple reduction from d-Exact Set Cover (d-ESC). The input of d-ESC is a set system $(U, \mathcal{F})$, where each set in $\mathcal{F}$ contains exactly d elements from U. The goal is to determine whether there is a collection $\mathcal{S} \subseteq \mathcal{F}$ of k disjoint sets that partition U, *i.e.* $\bigcup_{S \in \mathcal{S}} S = U$. It is well known that d-ESC is NP-complete for $d \geq 3$ [22]. In the proof of the theorem below we use the kernelization lower bound for d-ESC of Dell and Marx [9] who showed that, unless coNP $\subseteq$ NP/poly, the d-ESC problem has no kernel of size $O(k^{d-\varepsilon})$ for any $\varepsilon > 0$. In fact, this lower bound holds for a more generalized variant of kernelization called *compression*: In this variant, the output is allowed to be an instance of another problem, so long as it is decidable. Thus, a compression algorithm is a polynomial-time reduction from a parameterized problem to another, which has a guaranteed bound on its output size with respect to parameter of the input.

Theorem 6. *Let $d \geq 3$. Unless* coNP $\subseteq$ NP/poly*,* CapVC *in d-degenerate graphs has no kernel of size $O(k^{d-\varepsilon})$ for any $\varepsilon > 0$.*

Proof. Given an instance $(U, \mathcal{F})$ of d-ESC, we construct a graph $H := (V, E)$ by initially taking the incidence bipartite graph on $U \cup \mathcal{F}$, and then adding one edge between each $u \in U$ and a new leaf-vertex u' which is adjacent only to u. Thus, $V = U \cup \mathcal{F} \cup U'$, where $U' = \{u' : u \in U\}$, and

$E = \{\{u, F\} : u \in U, F \in \mathcal{F},$ and $u \in F\} \cup \{\{u, u'\} : u \in U\}$. To complete our construction, we set the capacity of each vertex in $\mathcal{F}$ to be its degree in H, the capacity of each vertex in $u \in U$ to be its degree minus one, and the capacity of each vertex in U' to zero. It is easy to see that H is d-degenerate by considering any ordering of V where vertices of U' are placed before vertices of $\mathcal{F}$, which in turn are placed before vertices of U.

We argue that $(U, \mathcal{F})$ has a solution of size $k = |U|/d$ if and only if H has a capacitated vertex cover of size $k' = |U| + k = O(k)$. Indeed, if $F_1, \ldots, F_k$ is a solution for $(U, \mathcal{F})$, then $\{F_1, \ldots, F_k\} \cup U$ is a capacitated vertex cover of H, because the vertices of $F_1, \ldots, F_k$ cover all its incident edges, and consequently each vertex $u \in U$ has exactly $\deg(u) - 1$ edges to cover, which is equal to its capacity.

Conversely, let X be a capacitated vertex cover of size k'. Since vertices of U' are of capacity zero we can safely assume that $X \cap U' = \emptyset$ and consequently $U \subseteq X$. Consequently X contains exactly k vertices of $\mathcal{F}$, which we denote as $M = \mathcal{F} \cap X$. Observe that no vertex of U has enough capacity to cover all its incident edges, hence each vertex of U is adjacent to a vertex of M. By construction this implies that M covers the whole universe U and since $|M| = k = |U|/d$ we infer that M is a solution for $(U, \mathcal{F})$.

Suppose now that CapVC in d-degenerate graphs has a kernel of size $O(k^{d-\varepsilon})$ for some $\varepsilon > 0$. Then we can use the above construction to transform any instance $(U, \mathcal{F}, k)$ of d-ESC to an equivalent instance (H, k') of CapVC with $k' = O(k)$, and then apply the kernel to obtain a new CapVC instance (H', k'') of size $O(k'^{d-\varepsilon}) = O(k^{d-\varepsilon})$. Thus, the existence of such a kernel for CapVC in d-degenerate graphs would imply a compression algorithm for d-ESC of size $O(k^{d-\varepsilon})$, contradicting the lower bound of Dell and Marx [9]. □

7 Conclusions and Future Work

This paper studies kernelization bounds for NP-hard problems on d-degenerate graphs, for fixed $d \geq 3$. Such graphs generalize many classes of graphs for which effective meta-kernelization is known to exist, e.g. planar graphs, H-minor free graphs, and H-topological-minor free graphs. We showed that the previously known upper-bounds for Dominating Set, Independent Dominating Set, and Induced Matching are almost tight, and we presented new bounds for Connected Vertex Cover, and Capacitated Vertex Cover (recall Table 1). While we consider d-degenerate graphs to be the most natural model of sparse graphs, it is worthwhile pointing out that there are other notions recently considered in the context of kernelization such as *graphs of bounded expansion* and *nowhere dense graphs* [14]. In those graph classes better bounds are known for Dominating Set, and it is interesting to understand the kernelization complexity of the other problems considered in this paper in that case as well.

Another interesting line of research would be to consider Turing kernelization. Roughly speaking, a Turing kernel of size $f(k)$ for a parameterized problem Π is an algorithm that decides Π in polynomial time with queries to Π of size bounded in $f(k)$ (see *e.g.* [26] for a more formal definition). Note that none of our lower bounds hold for this variant of kernelization, which is arguably almost as useful in practice. Does Dominating Set on d-degenerate graphs admit a polynomial Turing kernel where the degree of the polynomial does not depend on d? What about Independent Dominating Set and Induced Matching?

Finally, as can be seen from Table 1, the kernelization bounds for all problems considered in this paper are note quite tight, in particular for Dominating Set. It would be interesting to find out the exact bounds for any of these problems. The best starting points seem to be Connected Vertex Cover and Capacitated Vertex Cover.

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. 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.
3. 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.
4. Hans L. Bodlaender, Bart M. P. Jansen, and Stefan Kratsch. Kernelization lower bounds by cross-composition. *SIAM Journal of Discrete Mathematics*, 28(1):277–305, 2014.
5. Hans L. Bodlaender, Stéphan Thomassé, and Anders Yeo. Kernel bounds for disjoint cycles and disjoint paths. *Theoretical Computer Science*, 412(35):4570–4578, 2011.
6. 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.
7. Yijia Chen, Jörg Flum, and Moritz Müller. Lower bounds for kernelizations and other preprocessing procedures. *Theory of Computing Systems*, 48(4):803–839, 2011.
8. Marek Cygan, Marcin Pilipczuk, Michal Pilipczuk, and Jakub Onufry Wojtaszczyk. Kernelization hardness of connectivity problems in d-degenerate graphs. *Discrete Applied Mathematics*, 160(15):2131–2141, 2012.
9. Holger Dell and Dániel Marx. Kernelization of packing problems. In *Proc. of the 23rd annual ACM-SIAM Symposium On Discrete Algorithms (SODA)*, pages 68–81, 2012.
10. Holger Dell and Dieter van Melkebeek. Satisfiability allows no nontrivial sparsification unless the polynomial-time hierarchy collapses. *Journal of the ACM*, 61(4):23:1–23:27, 2014.
11. Reinhard Diestel. *Graph Theory*. Springer-Verlag, 3rd edition, 2005.
12. Michael Dom, Daniel Lokshtanov, and Saket Saurabh. Kernelization lower bounds through colors and IDs. volume 11, pages 13:1–13:20, 2014.
13. Rodney G. Downey and Michael R. Fellows. *Parameterized Complexity*. Springer-Verlag, 1999.
14. Pål Grønås Drange, Markus Sortland Dregi, Fedor V. Fomin, Stephan Kreutzer, Daniel Lokshtanov, Marcin Pilipczuk, Michal Pilipczuk, Felix Reidl, Fernando Sánchez Villaamil, Saket Saurabh, Sebastian Siebertz, and Somnath Sikdar. Kernelization and sparseness: the case of dominating set. In *Proc. of the 33rd Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 31:1–31:14, 2016.
15. Rok Erman, Lukasz Kowalik, Matjaz Krnc, and Tomasz Walen. Improved induced matchings in sparse graphs. *Discrete Applied Mathematics*, 158(18):1994–2003, 2010.
16. Michael R. Fellows. Blow-ups, win/win's, and crown rules: Some new directions in FPT. In *Proc. of the 29th International Workshop on Graph theoretic concepts in computer science (WG)*, pages 1–12, 2003.
17. Michael R. Fellows. The lost continent of polynomial time: Preprocessing and kernelization. In *Proc. of the 2nd International Workshop on Parameterized and Exact Computation (IWPEC)*, pages 276–277, 2006.
18. Michael R. Fellows, Danny Hermelin, Frances A. Rosamond, and Stéphane Vialette. On the parameterized complexity of multiple-interval graph problems. *Theoretical Computer Science*, 410(1):53–61, 2009.
19. Fedor V. Fomin, Daniel Lokshtanov, Saket Saurabh, and Dimitrios M. Thilikos. Bidimensionality and kernels. In *Proc. of the 21st annual ACM-SIAM Symposium On Discrete Algorithms (SODA)*, pages 503–510, 2010.
20. Fedor V. Fomin, Daniel Lokshtanov, Saket Saurabh, and Dimitrios M. Thilikos. Linear kernels for (connected) dominating set on H-minor-free graphs. In *Proc. of the 23rd annual ACM-SIAM Symposium On Discrete Algorithms (SODA)*, pages 82–93, 2012.
21. Lance Fortnow and Raul Santhanam. Infeasibility of instance compression and succinct pcps for NP. *Journal of Computer and System Sciences*, 77(1):91–106, 2011.
22. Michael R. Garey and David S. Johnson. *Computers and intractability : A guide to the theory of NP-completeness*. W.H. Freeman, 1979.
23. Jiong Guo and Rolf Niedermeier. Invitation to data reduction and problem kernelization. *SIGACT News*, 38(1):31–45, 2007.
24. Jiong Guo and Rolf Niedermeier. Linear problem kernels for np-hard problems on planar graphs. In *Proc. of 34th International Colloquium on Automata, Languages and Programming (ICALP)*, pages 375–386, 2007.
25. Danny Harnik and Moni Naor. On the compressibility of NP instances and cryptographic applications. *SIAM Journal on Computing*, 39(5):1667–1713, 2010.
26. Danny Hermelin, Stefan Kratsch, Karolina Soltys, Magnus Wahlström, and Xi Wu. A completeness theory for polynomial (turing) kernelization. *Algorithmica*, 71(3):702–730, 2015.
27. Danny Hermelin and Xi Wu. Weak compositions and their applications to polynomial lower-bounds for kernelization. In *Proc. of the 23rd annual ACM-SIAM Symposium On Discrete Algorithms (SODA)*, pages 104–113, 2012.

28. Iyad A. Kanj, Michael J. Pelsmajer, Marcus Schaefer, and Ge Xia. On the induced matching problem. *Journal of Computer and System Sciences*, 77(6):1058–1070, 2011.
29. Richard M. Karp. Reducibility among combinatorial problems. In R. E. Miller and J. W. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press, 1972.
30. Stefan Kratsch. Co-nondeterminism in compositions: A kernelization lower bound for a ramsey-type problem. *ACM Transactions on Algorithms*, 10(4):19:1–19:16, 2014.
31. Alexander Langer, Felix Reidl, Peter Rossmanith, and Somnath Sikdar. Linear kernels on graphs excluding topological minors. *CoRR*, abs/1201.2780, 2012.
32. George L. Nemhauser and Leslie E. Trotter Jr. Vertex packings: Structural properties and algorithms. *Mathematical Programming*, 8(2):232–248, 1975.
33. Geevarghese Philip, Venkatesh Raman, and Somnath Sikdar. Polynomial kernels for dominating set in graphs of bounded degeneracy and beyond. *ACM Transactions on Algorithms*, 9(1):11, 2012.