Tight Kernel Bounds for Problems on Graphs with Small Degeneracy* (Extended Abstract)**

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

[1] Institute of Informatics, University of Warsaw.
Email: cygan@mimuw.edu.pl
[2] Dalle Molle Institute (IDSIA) Galleria 1, 6928, Manno - Switzerland.
Email: fabrizio@idsia.ch
[3] Department of Industrial Engineering and Management
Ben-Gurion University, Beer-Sheva, Israel.
Email: 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 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.

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, 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 NP$\subseteq$ coNP/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 $O(k^d)$.

We also give simple kernels for Connected Vertex Cover and Capacitated Vertex Cover of size $O(k^d)$ and $O(k^{d+1})$ respectively. Both these problems do not have kernels of size $O(k^{d-1-\varepsilon})$ unless coNP/poly.

In this extended abstract we will focus on the lower bound for Dominating Set, which we feel is the central result of our study. The proofs of the other results can be found in the full version of the paper.

* Partially supported by ERC Starting Grant NEWNET 279352.
** A full version of the paper can be found in [8].

1 Introduction

Parameterized complexity is a two-dimensional refinement of classical complexity theory introduced by Downey and Fellows [14] 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 way, 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* (or *kernel*) 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') 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.

Kernelization is a notion that was developed in parameterized complexity, but it is also useful in other areas of computer science such as cryptography [21] and approximation algorithms [27]. In parameterized complexity, not only is it one of the most successful techniques for showing positive results, 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 iff it has a kernel [6]. From a practical point of view, compression algorithms often lead to efficient preprocessing rules which can significantly simplify real life instances [16, 19]. For these reasons, the study of kernelization is one of the leading research frontiers in parameterized complexity. This research endeavor has been fueled by recent tools for showing lower bounds on kernel sizes [2, 4, 5, 7, 10, 11, 13, 22, 26] which rely on the standard complexity-theoretic assumption of coNP$\not\subseteq$ NP/poly.

Since a parameterized problem is fixed-parameter tractable iff it is kernelizable, it is natural to ask which fixed-parameter problems admit kernels of reasonably small size. In recent years there have been significant advances in this area. One particularly prominent line of research in this context is the development of *meta-kernelization* algorithms for problems on sparse graphs. Such algorithms typically provide compressions of either linear or quadratic size to 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 [20], which extended the ideas used in the classical linear kernel for Dominating Set 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.* [17]

	Lower Bound	Upper Bound
Dominating Set	$(d-3)(d-1)-\varepsilon$	$(d+1)^2$ [28]
Independent Dominating Set	$d-4-\varepsilon$	$d+1$ [28]
Induced Matching	$d-3-\varepsilon$	d [15, 23]
Connected Vertex Cover	$d-1-\varepsilon$ [9]	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.

provided a meta-kernel for problems on H-minor free graphs which include all bounded genus graphs. Finally, a recent manuscript by Langer *et al.* [25] provides a meta-kernelization algorithm for problems on H-topological-minor free graphs. All meta-kernelizations above have either linear or quadratic size.

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 to its right 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 (see *e.g.* [12]). Note that the Independent Set problem has a trivial linear kernel in d-degenerate graphs, which 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.* [28] 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.* [23] independently gave a $O(k^d)$ kernel for the Induced Matching problem, while Cygan *et al.* [9] showed a $O(k^{d+1})$ 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 ConVC in [9] using the standard reduction from d-Set Cover. It is also shown in [9] 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. Furthermore, the results in [10, 22] can be easily used to exclude a $O(k^{d-\varepsilon})$-size kernel for Dominating Set, for some small positive constant ε.

In the full version of the paper we show that all remaining kernelization upper bounds for d-degenerate graphs mentioned above have matching lower bounds up to some small additive constant; see Table 1 for details. 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 of coNP $\not\subseteq$ NP/poly. This result is obtained by an intricate application of *weak compositions* which were introduced by [11], and further applied in [10, 22]. What makes this result surprising is that it implies that Independent Dominating Set is fundamentally easier than Dominating Set in d-degenerate graphs.

In the current extend abstract we focus on the lower bound obtained for Dominating Set, since we feel it is the most interesting and most technically challenging result we obtain. Following a brief overview of the lower bound machinery that we will use for our result, including the definition of

2 Kernelization Lower Bounds

In the following section we quickly review the main tool that we will be using for showing our kernelization lower bounds, namely compositions. A composition algorithm is typically a transformation from a classical NP-hard 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 iff 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 guarantee bound on the parameter of the output. More formally, for an integer $d \geq 1$, a weak d-composition is defined as follows:

Definition 1 (weak d-composition). *Let $d \geq 1$ 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 d-composition *from L_1 to L_2 is a polynomial time algorithm that on input $x_1, \ldots, x_{t^d} \in \{0,1\}^n$ outputs an instance $(y, k') \in \{0,1\}^* \times \mathbb{N}$ such that:*

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

The connection between compositions and kernelization lower bounds was discovered by [2] using ideas from [21] and a complexity theoretic lemma of [18]. The following particular connection was first observed in [11].

Lemma 1 ([11]). *Let $d \geq 1$ be an integer, let $L_1 \subseteq \{0,1\}^*$ be a classical* NP-*hard 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 some arbitrary set.

3 Construction Overview

We next briefly sketch the main ideas behind our lower bound construction for Dominating Set. The general idea and framework that we use was introduced in [13], and further developed in [22]. For convenience purposes, we will show a lower bound for essentially equivalent Red Blue Dominating Set problem (d-RBDS). In this problem we are given a parameter k and a n-vertex d-degenerate graph which is properly colored by two colors, red and blue, and the question is whether there are k red vertices which dominate all the blue vertices in the graph. Our goal is show how to compose any $T \approx t^d$ instances $I_1, \ldots, I_T$ of some NP-hard problem – each with the same size n – into a single instance I of RBDS which is (i) a YES-instance iff one of the T input instances is a YES-instance, and (ii) has parameter at most $t \cdot n^{O(1)}$.

As our starting problem we use the NP-hard Multicolored Perfect Matching problem: Given an undirected graph G with an even number of vertices n, together with a color function that assigns one of $n/2$ colors to the edges of the graph, determine whether G has a perfect matching where all edges have distinct colors. This problem easily reduces to RBDS by constructing a red vertex for each edge in G, and a blue vertex for each vertex and each edge-color, and then connecting each red vertex representing an edge to the blue vertices representing its endpoints and color. Clearly, the graph obtained by this construction is 3-degenerate.

So we want to compose a sequence $I_1, \ldots, I_T$ of MPM instances into a single instance of RBDS. As the first step, we transform each instance I_i to an instance I'_i of RBDS almost as described above. The difference is that we do not create blue vertices for edge colors, but we store the color of an edge that a given red vertex represents to use that information later on. Now the "bipartiteness" of RBDS allows for an easy start in our construction. To obtain a single RBDS instance from $I'_1, \ldots, I'_T$, we identify the T sets of blue vertices in each instance into a single set. This can easily be done as all instances I'_i have the same number of blue vertices (since all instances I_i had the same number of edges and edge-colors). We then take the disjoint union of the T sets of red vertices, and connect each set to the identified set of blue vertices in the natural way. In this way, we obtain a single 2-degenerate bipartite graph H_{inst} for our instance I of RBDS. It is easy to see that if one of the instances I'_i has a solution of at most k red vertices these same red vertices form a solution in H_{inst}. The problem is that I can have a solution of size k even if all instances $I_1, \ldots I_T$ are NO-instances, since a solution in I can be composed of red vertices from more than one instance in $I_1, \ldots I_T$ and moreover we do not control the colorfulness aspect of the initial MPM instances.

Observe that we still have a lot of leeway in the parameter of our output, as we can afford a solution size of $t \cdot n^{O(1)}$, and also some leeway in the degeneracy of the output graph. Thus, we circumvent the problem above by adding an enforcement graph H_{enf} to our construction which is essentially another instance of RBDS which ensures that vertices corresponding to edges in different instances I_i will not be selected in any solution of size k', for some carefully chosen $k' = tn^{O(1)}$.

This is done by connecting red vertices in H_{inst} via the edge set E_{conn} to the blue vertices of H_{enf} in an intricate manner that ensures that the resulting graph is only $(d+2)$-degenerate.

4 Construction Details

Let us begin recalling the definition of the Dominating Set (DS) problem. In this problem, we are given an undirected graph $G = (V, E)$ together with an integer k, and we are asked whether there exists a set S of at most k vertices such that each vertex of G either belongs to S or has a neighbor in S (i.e. $N[S] = V$). The main result of this section is stated in Theorem 1 below.

Theorem 1. *Let $d \geq 4$. The* Dominating Set *problem in d-degenerate graphs has no kernel of size $O(k^{(d-1)(d-3)-\varepsilon})$ for any constant $\varepsilon > 0$ unless* NP $\subseteq$ coNP/poly*.*

In order to prove Theorem 1, we show a lower bound for a similar problem called the Red Blue Dominating Set problem (RBDS): Given a bipartite graph $G = (R \cup B, E)$ and an integer k, where R is the set of *red* vertices and B is the set of *blue* vertices, determine whether there exists a set $D \subseteq R$ of at most k red vertices which dominate all the blue vertices (i.e. $N(D) = B$). According to Remark 1, the lemma below shows that focusing on RBDS is sufficient for proving Theorem 1 above.

Lemma 2. *There is a polynomial time algorithm, which given a d-degenerate instance $I = (G = (R \cup B, E), k)$ of* RBDS *creates a $(d+1)$-degenerate instance $I' = (G', k')$ of* DS*, such that $k' = k + 1$ and I is YES-instance iff I' is a YES-instance.*

Proof. As the graph G' we initially take $G = (R \cup B, E)$ and then we add two vertices r, r' and make r adjacent to all the vertices in $R \cup \{r'\}$. Clearly G' is $(d+1)$-degenerate. Note that if $S \subseteq R$ is a solution in I, then $S \cup \{r\}$ is a dominating set in I'. In the reverse direction, observe that w.l.o.g. we may assume that a solution S' for I' contains r and moreover contains no vertex of B. Therefore I is a YES-instance iff I' is a YES-instance. $\square$

We next describe a weak $d(d+2)$-composition from Multicolored Perfect Matching to RBDS in $(d+2)$-degenerate graphs. The Multicolored Perfect Matching problem (MPM) is as follows: Given an undirected graph $G = (V, E)$ with even number n of vertices, together with a color function $\mathrm{col} : E \to \{0, \ldots, n/2 - 1\}$, determine whether there exists a perfect matching in G with all the edges having distinct colors. A simple reduction from 3-Dimensional Perfect Matching, which is NP-complete due to Karp [24], where we encode one coordinate using colors, shows that MPM is NP-complete when we consider multigraphs. In the full version of the paper we show that MPM is NP-complete even for simple graphs.

The construction of the weak composition is rather involved (see Fig. 1). We construct an instance graph H_{inst} which maps feasible solutions of each MPM instance into feasible solutions of the RBDS instance. Then we add an enforcement gadget (H_{enf}, E_{conn}) which prevents partial solutions of two or more MPM instances to form altogether a solution for the RBDS instance. The overall RBDS instance will be denoted by (H, k), where H is the union of H_{inst} and H_{enf} along with the edges E_{conn} that connect between these graphs. The construction of the instance graph is relatively simple, while the enforcement gadget is rather complex. In the next subsection we describe H_{enf} and its crucial properties. In the following subsection we describe the rest of the construction, and prove the claimed lower bound on RBDS (and hence DS). Both H_{enf} and H_{inst} contain red and blue nodes. We will use the convention that R and B denote sets of red and blue nodes, respectively. We will use r and b to indicate red and blue nodes, respectively. A color is indicated by ℓ.

4.1 The Enforcement Graph

The enforcement graph $H_{enf} = (R_{enf} \cup B_{enf}, E_{enf})$ is a combination of 3 different gadgets: the *encoding gadget*, the *choice gadget*, and the *fillin gadget* (see also Fig. 1), i.e. $R_{enf} = R_{code} \cup R_{fill}$ and $B_{enf} = B_{code} \cup B_{choice} \cup B_{fill}$ (R_{choice} is empty).

Encoding gadget: The role of this gadget is to encode the indices of all the instances by different partial solutions. It consists of nodes $R_{code} \cup B_{code}$, plus the edges among them. The set R_{code} contains one node $r_{\delta,\lambda,\gamma}$ for all integers $0 \le \delta < d+2$, $0 \le \lambda < d$, and $0 \le \gamma < t$. In particular, $|R_{code}| = (d+2)dt$. The set B_{code} is the union of sets B^{ℓ}_{code} for each color $0 \le \ell < n/2$. In turn, B^{ℓ}_{code} contains a node b^{ℓ}_a for each integer $0 \le a < (dt)^{d+2}$. We connect nodes $r_{\delta,\lambda,\gamma}$ and b^{ℓ}_a iff $a_\delta = \lambda \cdot t + \gamma$, where $(a_0, \ldots, a_{d+1})$ is the expansion of a in base dt, i.e. $a = \sum_{0 \le \delta < d+2} a_\delta (dt)^\delta$. There is a subtle reason behind this connection scheme, which hopefully will be clearer soon. Note that since $0 \le \gamma < t$, pairs (λ, γ) are in one to one correspondence with possible values of digits a_δ.

Choice gadget: The role of the choice gadget is to guarantee the following *choice property*: Any feasible solution to the overall RBDS instance (H, k) contains all nodes R_{code} except possibly one node $r_{\delta,\lambda,\gamma_{\delta,\lambda}}$ for each pair (δ, λ) (hence at least $(d+2)d(t-1)$ nodes of R_{code} altogether are taken). Intuitively, the $\gamma_{\delta,\lambda}$'s will be used to identify the index of one MPM input instance. In order to do that, we introduce a set of nodes B_{choice}, containing a node $b_{\delta,\lambda,\gamma_1,\gamma_2}$ for every pair (δ, λ) and for every $0 \le \gamma_1 < \gamma_2 < t$. We connect $b_{\delta,\lambda,\gamma_1,\gamma_2}$ to both $r_{\delta,\lambda,\gamma_1}$ and $r_{\delta,\lambda,\gamma_2}$. It is not hard to see that, in order to dominate B_{choice}, it is necessary and sufficient to select from R_{code} a subset of nodes with the choice property.

Fillin gadget: We will guarantee that, in any feasible solution, precisely $(d+2)d(t-1)$ nodes from R_{code} are selected. Given that, for each pair (δ, λ), there will be precisely one node $r_{\delta,\lambda,\gamma_{\delta,\lambda}}$ which is not included in the solution. Consequently, as we will prove, for each $0 \leq \ell < n/2$ in B^{ℓ}_{code} there will be exactly d^{d+2} uncovered nodes, namely the nodes $b^{\ell}_{a} = b^{\ell}_{(a_0,\ldots,a_{d+1})}$ such that for each $0 \leq \delta < d+2$ and $\lambda t \leq a_\delta < (\lambda+1)t$ one has $a_\delta = \lambda t + \gamma_{\delta,\lambda}$. Ideally, we would like to cover such nodes by means of red nodes in the instance graph H_{inst} (to be defined later), which encode a feasible solution to some MPM instance. However, the degeneracy of the overall graph would be too large. The role of the fillin gadget is to circumvent this problem, by leaving at most d uncovered nodes in each B^{ℓ}_{code}. The fillin gadget consists of nodes $R_{fill} \cup B_{fill}$, with some edges incident to them. The set R_{fill} is the union of sets R^{ℓ}_{fill} for each color ℓ. In turn R^{ℓ}_{fill} contains one node $r^{\ell}_{a,j}$ for each $1 \leq j \leq d^{d+2} - d$ and $0 \leq a < (dt)^{d+2}$. We connect each $r^{\ell}_{a,j}$ to b^{ℓ}_{a}. The set B_{fill} contains one node b^{ℓ}_{j}, for each color ℓ and for all $1 \leq j \leq d^{d+2} - d$. We connect b^{ℓ}_{j} to all nodes $\{r^{\ell}_{a,j} : 0 \leq a < (dt)^{d+2}\}$. Observe that, in order to cover B_{fill}, it is necessary and sufficient to select one node $r^{\ell}_{a,j}$ for each ℓ and j. Furthermore, there is a way to do that such that each selected $r^{\ell}_{a,j}$ covers one extra node in B^{ℓ}_{code} w.r.t. selected nodes in R_{code}. Note that we somewhat abuse notation as we denote by b^{ℓ}_{j} vertices of B_{fill}, while we use b^{ℓ}_{a} for vertices of B_{code}, hence the only distinction is by the variable name.

Lemma 3. *For any matrix $(\gamma_{\delta,\lambda})_{0\leq\delta<d+2,0\leq\lambda<d}$ of size $(d+2)\times d$ with entries from $\{0,\ldots,t-1\}$, there exists a set $\tilde{R}_{enf} \subseteq R_{enf}$ of size $k' := \frac{n}{2}(d^{d+2} - d) + (d+2)d(t-1)$, such that:*

- *each vertex in $B_{choice} \cup B_{fill}$ has a neighbor in $\tilde{R}_{enf}$, and*
- *for every $0 \leq \ell < n/2$ we have $B^{\ell}_{code} \setminus N(\tilde{R}_{enf}) = \{b^{\ell}_{a} : 0 \leq \lambda < d, a = \sum_{0\leq\delta<d+2}(\lambda t + \gamma_{\delta,\lambda})(dt)^{\delta}\}$.*

Proof. For each $0 \leq \delta < d+2$ and $0 \leq \lambda < d$, add to $\tilde{R}_{enf}$ the set $\{r_{\delta,\lambda,\gamma} : 0 \leq \gamma < t, \gamma \neq \gamma_{\delta,\lambda}\}$ containing $t-1$ vertices. Note that by construction $\tilde{R}_{enf}$ dominates the whole set B_{choice}. Consider a vertex $b^{\ell}_{a} \in B^{\ell}_{code} \setminus N(\tilde{R}_{enf})$ and observe that for each coordinate $0 \leq \delta < d+2$, there are exactly d values that a_δ can have, where $(a_0,\ldots,a_{d+1})$ is the (dt)-ary representation of a. Indeed, for any $0 \leq \delta < d+2$, we have $a_\delta \in X_\delta = \{\lambda t + \gamma_{\delta,\lambda} : 0 \leq \lambda < d\}$, since otherwise b^{ℓ}_{a} would be covered by $\tilde{R}_{enf}$ due to the δ-th coordinate. Moreover if we consider any $b^{\ell}_{(a_0,\ldots,a_{d+1})} \in B^{\ell}_{code}$ such that $a_\delta \in X_\delta$ for $0 \leq \delta < d+2$, then $b^{\ell}_{(a_0,\ldots,a_{d+1})}$ is not dominated by the vertices added to $\tilde{R}_{enf}$ so far.

Next, for each ℓ define $M^{\ell} := \{b^{\ell}_{a} : 0 \leq \lambda < d, a = \sum_{0\leq\delta<d+2}(\lambda t + \gamma_{\delta,\lambda})(dt)^{\delta}\}$ and observe that M^{ℓ} are not dominated $\tilde{R}_{enf}$. For each $0 \leq \ell < n/2$, let Z^{ℓ} be the vertices of B^{ℓ}_{code} not yet covered by $\tilde{R}_{enf}$ and for each $1 \leq j \leq d^{d+2} - d$ select exactly one distinct vertex $v_j \in Z^{\ell} \setminus M^{\ell}$, where $v_j = b^{\ell}_{a}$, and add to $\tilde{R}_{enf}$ the vertex $r^{\ell}_{a,j}$. Observe that after this operation $\tilde{R}_{enf}$ covers B_{fill} and moreover the only vertices of B_{code} not covered by $\tilde{R}_{enf}$ are the vertices of $\bigcup_{0\leq\ell<n/2} M^{\ell}$. Since the total size of $\tilde{R}_{enf}$ equals $d(d+2)(t-1) + \frac{n}{2}(d^{d+2} - d)$, the lemma follows. $\square$

Lemma 4. *Consider an* RBDS *instance* $(H = (R \cup B, E), k)$ *containing* $G_{enf} = (R_{enf} \cup B_{enf}, E_{enf})$ *as an induced subgraph, with* $R_{enf} \subseteq R$ *and* $B_{enf} \subseteq B$*, such that no vertex of* $B_{choice} \cup B_{fill}$ *has a neighbor outside of* R_{enf}*. Then any feasible solution* $\tilde{R}$ *to* (H, k) *contains at least* $k' := \frac{n}{2}(d^{d+2} - d) + (d+2)d(t-1)$ *nodes* $\tilde{R}_{enf}$ *of* R_{enf}*. Furthermore, for any feasible solution* $\tilde{R}$ *to* (H, k) *containing exactly* k' *vertices of* R_{enf}*, there exist a matrix* $(\gamma_{\delta,\lambda})_{0 \le \delta < d+2, 0 \le \lambda < d}$ *of size* $(d+2) \times d$ *with entries from* $\{0, \ldots, t-1\}$*, such that for each* $0 \le \ell < n/2$*:*

(a) there are at least d *vertices in* $U^\ell = B^\ell_{code} \setminus N(\tilde{R} \cap R_{enf})$*, and*
(b) U^ℓ *is a subset of the* d^{d+2} *nodes* $b^\ell_a = b^\ell_{(a_0,\ldots,a_{d+1})}$ *such that for each* $\delta \in \{0, \ldots, d+1\}$ *there exists* $\lambda \in \{0, \ldots d-1\}$ *with* $a_\delta = \lambda t + \gamma_{\delta,\lambda}$*.*

Proof. Let $\tilde{R}$ be any feasible solution to (H, k). Observe that since $\tilde{R}$ dominates B_{choice}, for each $0 \le \delta < d+2$ and $0 \le \lambda < d$ we have $|\tilde{R} \cap \{r_{\delta,\lambda,\gamma} : 0 \le \gamma < t\}| \ge t-1$. Moreover in order to dominate vertices of B_{fill}, the set $\tilde{R}$ has to contain at least $n/2(d^{d+2} - d)$ vertices of R_{fill}. Consequently, if $\tilde{R}$ contains exactly k' vertices of R_{enf}, then for each $0 \le \delta < d+2$ and $0 \le \lambda < d$, there is exactly one $\gamma_{\delta,\lambda}$ such that $r_{\delta,\lambda,\gamma_{\delta,\lambda}} \notin \tilde{R}$. By the same argument as in the proof of Lemma 3, we infer that for each ℓ, the set $B^\ell_{code} \setminus N(\tilde{R} \cap R_{code})$ contains exactly d^{d+2} vertices, and we denote them as U^ℓ_0. Observe that the set $\tilde{R} \setminus R_{code}$ dominates at most $d^{d+2} - d$ vertices of U^ℓ_0, for each $0 \le \ell < n/2$, which proves properties (a) and (b) of the lemma. □

4.2 The Overall Graph

The construction of $H_{inst} = (R_{inst} \cup B_{inst}, E_{inst})$ is rather simple. Let $(G_i = (V, E_i), \mathrm{col}_i)$ be the input MPM instances, with $0 \le i < T = t^{d(d+2)}$. By standard padding arguments 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 each $v \in V$, we create a blue node $b_v \in B_{inst}$. For each $e_i = \{u, v\} \in E_i$, we create a red node $r_{e,i} \in R^i_{inst}$ and connect it to both b_u and b_v. We let $R_{inst} := \bigcup_{0 \le i < T} R^i_{inst}$. Intuitively, we desire that a RBDS solution, if any, selects exactly $n/2$ nodes from one set R^i_{inst}, corresponding to edges of different colors, which together dominate all nodes B_{inst}: This induces a feasible solution to MPM for the i-th instance.

It remains to describe the edges E_{conn} which connect H_{enf} with H_{inst}. This is the most delicate part of the entire construction. We map each index i, $0 \le i < T$, into a distinct $(d+2) \times d$ matrix M_i with entries $M_i[\delta, \lambda] \in \{0, \ldots, t-1\}$, for all possible values of δ and λ. Consider an instance G_i. We connect $r_{e,i}$ to b^ℓ_a iff $\ell = \mathrm{col}_i(e_i)$ and there exists $0 \le \lambda < d$ such that the expansion $(a_0, \ldots, a_{d+1})$ of a in base dt satisfies $a_\delta = M_i[\delta, \lambda] + \lambda \cdot t$ on each coordinate $0 \le \delta < d+2$. The final graph $H := (R \cup B, E)$ we construct for our instance RBDS is then given by $R := R_{inst} \cup R_{enf}$ and $E := E_{inst} \cup E_{enf} \cup E_{conn}$. See Fig. 1.

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

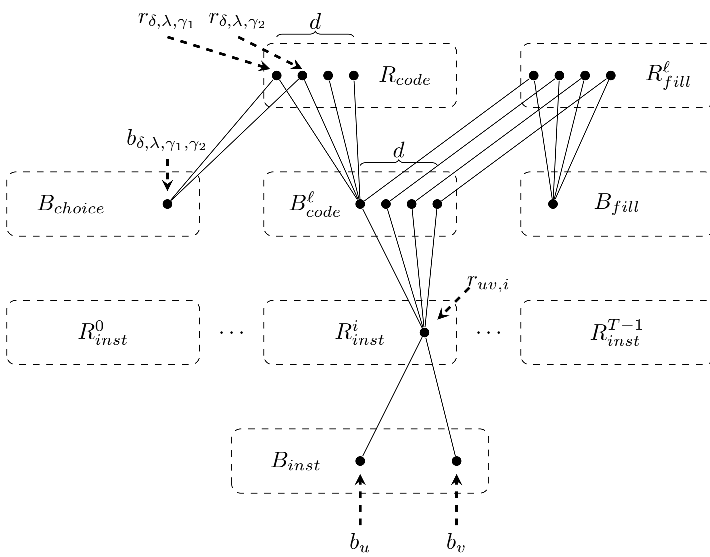

Fig. 1. Construction of the graph H. For simplicity the figure does not include sets $R^{\ell'}_{fill}$ and $B^{\ell'}_{code}$ for $\ell' \neq \ell$.

Proof. Observe that each vertex of $\bigcup_{0 \le i < T} R^i_{inst}$ is of degree exactly $d+2$ in H, so we put all those vertices first to our ordering. Next, we take vertices of B_{inst}, as those have all neighbors already put into the ordering. Therefore it is enough to argue about the $(d+2)$-degeneracy of the enforcement gadget. We 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 to our ordering. We are left with the vertices of the encoding gadget. Observe, that each blue vertex of the encoding gadget has exactly $d+2$ neighbors in R_{code}, one due to each coordinate, hence we put the vertices of B_{code} next and finish the ordering with vertices of R_{code}. □

Lemma 6. *Let* $k := (d+2)d(t-1) + n/2(d^{d+2} - d) + n/2 = k' + n/2$. *Then* (H,k) *is a YES-instance of* RBDS *iff* (G_i, col_i) *is a YES-instance of* MPM *for some* $i \in \{0, \dots, T-1\}$.

Proof. Let us assume that for some i_0 the instance $(G_{i_0}, \mathrm{col}_{i_0})$ is a YES-instance and $E' \subseteq E_{i_0}$ is the corresponding solution. We use Lemma 3 with the matrix M_{i_0} assigned to the instance i_0 to obtain the set $\tilde{R}_{enf}$ of size $(d+2)d(t-1) + \frac{n}{2}(d^{d+2} - d)$. As the set $\tilde{R}$ we take $\tilde{R}_{enf} \cup \{r_{e,i_0} : e \in E'\}$. Clearly $|\tilde{R}| = k$. Since E' is a perfect matching, $\tilde{R}$ dominates B_{inst}. By Lemma 3, $\tilde{R}$ dominates $B_{fill} \cup B_{choice}$ and all but d vertices of each B^{ℓ}_{code}, so denote those d vertices by M^{ℓ}. Consider each $0 \le \ell < n/2$, and observe that since E' is multicolored and by the construction of H, the set of neighbors of r_{e,i_0} in B_{code} is exactly $M^{\mathrm{col}_{i_0}(e)}$; and hence $\tilde{R}$ is a solution for (H,k).

In the other direction, assume that (H,k) is a YES-instance and let $\tilde{R}$ be a solution of size at most k. By Lemma 4, the set $\tilde{R}$ contains at least $k' = \frac{n}{2}(d^{d+2} - d) + (d+2)d(t-1)$ vertices of R_{enf} and since $\tilde{R}$ needs to dominate also B_{inst} it contains at least $\frac{n}{2}$ vertices of $\bigcup_{0 \le i < T} R^i_{inst}$, since no vertex of H dominates more than two vertices of B_{inst}. Consequently $|\bigcup_{0 \le i < T} R^i_{inst} \cap \tilde{R}| = n/2$ and $|R_{enf} \cap \tilde{R}| = k'$. We use Lemma 4 to obtain a matrix $M = (\gamma_{\delta,\lambda})$ of size $(d+2) \times d$. Moreover, by property (a) of Lemma 4, there are at least d vertices in U^ℓ, and consequently for each color ℓ the set $\tilde{R}$ contains exactly one vertex of the set $\{r_{e,i} : 0 \le i < T, \mathrm{col}_i(e) = \ell\}$. Our goal is to show that for each $0 \le i < T$, such that a matrix different than M is assigned to the i-th instance, we have $\tilde{R} \cap R^i_{inf} = \emptyset$, which is enough to finish the proof of the lemma. Consider any such i and assume that the matrices M_i and M differ in the entry $M_i[\delta', \lambda'] \neq \gamma_{\delta',\lambda'}$. Let ℓ be a color such that $r_{e,i} \in \tilde{R}$ and $\mathrm{col}_i(e) = \ell$. By property (b) of Lemma 4, the set of at least d vertices of B^ℓ_{code} not dominated by $\tilde{R} \cap R_{enf}$ is contained in $U^\ell_0 = \{b^\ell_{(a_0,\ldots,a_{d+1})} : \forall_{0 \le \delta < d+2}$ if $\lambda t \le a_\delta < (\lambda+1)t$ then $a_\delta = \lambda t + \gamma_{\delta,\lambda}\}$. However, by our construction of edges of H between R^i_{inst} and B^ℓ_{code}, we have $(N_H(r_{e,i}) \cap B^\ell_{code}) \not\subseteq U^\ell_0$ since the vertex $b^\ell_{(a_0,\ldots,a_{\delta+1})} \in N_H(r_{e,i}) \cap B^\ell_{code}$ with $a_\delta = \lambda' t + M_i[\delta', \lambda']$ does not belong to U^ℓ_0 and consequently does not belong to U^ℓ, which leaves at least one vertex of B^ℓ_{code} not dominated by $\tilde{R}$; a contradiction. $\square$

Acknowledgements

We thank anonymous referees for their helpful remarks and comments.

References

1. J. Alber, M. R. Fellows, and R. Niedermeier. Polynomial-time data reduction for dominating set. *Journal of the ACM*, 51(3):363–384, 2004.
2. H. L. Bodlaender, R. G. Downey, M. R. Fellows, and D. Hermelin. On problems without polynomial kernels. *Journal of Computer and System Sciences*, 75(8):423–434, 2009.
3. H. L. Bodlaender, F. V. Fomin, D. Lokshtanov, E. Penninkx, S. Saurabh, and D. M. Thilikos. (Meta) kernelization. In *FOCS*, pages 629–638, 2009.
4. H. L. Bodlaender, B. M. P. Jansen, and S. Kratsch. Cross-composition: A new technique for kernelization lower bounds. In *STACS*, pages 165–176, 2011.
5. H. L. Bodlaender, S. Thomassé, and A. Yeo. Kernel bounds for disjoint cycles and disjoint paths. In *ESA*, pages 635–646, 2009.
6. L. Cai, J. Chen, R. G. Downey, and M. R. Fellows. Advice classes of paramterized tractability. *Annals of Pure and Applied Logic*, 84(1):119–138, 1997.
7. Y. Chen, J. Flum, and M. Müller. Lower bounds for kernelizations and other preprocessing procedures. *Theory of Computing Systems*, 48(4):803–839, 2011.

8. M. Cygan, F. Grandoni, and D. Hermelin. Tight kernel bounds for problems on graphs with small degeneracy. *CoRR*, abs/1305.4914, 2013.
9. M. Cygan, M. Pilipczuk, M. Pilipczuk, and J. O. Wojtaszczyk. Kernelization hardness of connectivity problems in d-degenerate graphs. *Discrete Applied Mathematics*, 160(15):2131–2141, 2012.
10. H. Dell and D. Marx. Kernelization of packing problems. In *SODA*, pages 68–81, 2012.
11. H. Dell and D. van Melkebeek. Satisfiability allows no nontrivial sparsification unless the polynomial-time hierarchy collapses. In *STOC*, pages 251–260, 2010.
12. R. Diestel. *Graph Theory*. Springer-Verlag, 3rd edition, 2005.
13. M. Dom, D. Lokshtanov, and S. Saurabh. Incompressibility through colors and IDs. In *ICALP*, pages 378–389, 2009.
14. R. G. Downey and M. R. Fellows. *Parameterized Complexity*. Springer-Verlag, 1999.
15. R. Erman, L. Kowalik, M. Krnc, and T. Walen. Improved induced matchings in sparse graphs. *Discrete Applied Mathematics*, 158(18):1994–2003, 2010.
16. M. R. Fellows. The lost continent of polynomial time: Preprocessing and kernelization. In *IWPEC*, pages 276–277, 2006.
17. F. V. Fomin, D. Lokshtanov, S. Saurabh, and D. M. Thilikos. Bidimensionality and kernels. In *SODA*, pages 503–510, 2010.
18. L. Fortnow and R. Santhanam. Infeasibility of instance compression and succinct PCPs for NP. In *STOC*, pages 133–142, 2008.
19. J. Guo and R. Niedermeier. Invitation to data reduction and problem kernelization. *SIGACT News*, 38(1):31–45, 2007.
20. J. Guo and R. Niedermeier. Linear problem kernels for NP-hard problems on planar graphs. In *ICALP*, pages 375–386, 2007.
21. D. Harnik and M. Naor. On the compressibility of NP instances and cryptographic applications. *SIAM Journal on Computing*, 39(5):1667–1713, 2010.
22. D. Hermelin and X. Wu. Weak compositions and their applications to polynomial lower-bounds for kernelization. In *SODA*, pages 104–113, 2012.
23. I. A. Kanj, M. J. Pelsmajer, M. Schaefer, and G. Xia. On the induced matching problem. *Journal of Computer and System Sciences*, 77(6):1058–1070, 2011.
24. R. 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.
25. E. J. Kim, A. Langer, C. Paul, F. Reidl, P. Rossmanith, I. Sau, and S. Sikdar. Linear kernels and single-exponential algorithms via protrusion decompositions. In *ICALP*, page to appear, 2013.
26. S. Kratsch. Co-nondeterminism in compositions: a kernelization lower bound for a Ramsey-type problem. In *SODA*, pages 114–122, 2012.
27. G. L. Nemhauser and L. E. T. Jr. Vertex packings: Structural properties and algorithms. *Mathematical Programming*, 8(2):232–248, 1975.
28. G. Philip, V. Raman, and S. Sikdar. Polynomial kernels for dominating set in graphs of bounded degeneracy and beyond. *ACM Transactions on Algorithms*, 9(1):11, 2012.