# Polynomial Kernels for Dominating Set in $K_{i,j}$-free and $d$-degenerate Graphs

Geevarghese Philip, Venkatesh Raman, and Somnath Sikdar

The Institute of Mathematical Sciences, Chennai, India.
`{gphilip|vraman|somnath}@imsc.res.in`

**Abstract.** We show that for any fixed $i, j \geq 1$, the $k$-Dominating Set problem restricted to graphs that do not have $K_{i,j}$ as a subgraph is fixed parameter tractable (FPT) and has a polynomial kernel. This result implies that this problem restricted to bounded-degenerate graphs has a polynomial kernel, solving an open problem posed by Alon and Gutner in [3]. Our result extends the class of graphs for which the $k$-Dominating Set problem is known to have (1) FPT algorithms and (2) polynomial kernels, to the class of $K_{i,j}$-free graphs.

## 1 Introduction

The $k$-Dominating Set problem asks, for a graph $G = (V, E)$ and a positive integer $k$ given as inputs, whether there is a vertex-subset $S \subseteq V$ of size at most $k$ such that every vertex in $V \setminus S$ is adjacent to some vertex in $S$. Such a vertex-subset is called a *dominating set* of $G$. For general graphs this problem is known to be NP-hard [15], and the problem parameterized by $k$ is $W[2]$-complete [9] even if the input has bounded average degree [16]. The latter fact implies that it is unlikely that the problem has a fixed-parameter-tractable (FPT) algorithm on graphs with a bounded average degree, that is, an algorithm that runs in time $f(k) \cdot n^c$ for *some* computable function $f(k)$ independent of $n$ and some constant $c$ independent of $k$. However, the problem has an FPT algorithm on certain restricted families of graphs. See Table 1 for some known FPT algorithms for this problem. A graph $G$ is said to be *$d$-degenerate* if every subgraph of $G$ has a vertex of degree at most $d$. As far as we know, $d$-degenerate graphs are the most general class of graphs for which the $k$-Dominating Set problem has been previously shown to have an FPT algorithm [2].

Closely related to the notion of an FPT algorithm is the concept of a *kernel* for a parameterized problem. For the $k$-Dominating Set problem parameterized by $k$, a kernelization algorithm is a polynomial-time algorithm that takes $(G, k)$ as input and outputs a graph $G'$ and a nonnegative integer $k'$ such that the size of $G'$ is bounded by some function $g(k)$ of $k$ alone, $k' \leq h(k)$ for some function $h(k)$, and $G$ has a dominating set of size at most $k$ if and only if $G'$ has a dominating set of size at most $k'$. The resulting instance $G'$ is called a kernel for the problem. A parameterized problem has a kernelization algorithm if and only if it has an FPT algorithm [9], and so it is unlikely that the

$k$-DOMINATING SET problem on general graphs or on graphs having a bounded average degree has a kernelization algorithm. For the same reason, this problem has a kernelization algorithm when restricted to those graph classes for which it has an FPT algorithm. But the size of the kernel obtained from such an algorithm could be exponential in $k$, and so it is interesting to ask if the kernel size can be made smaller—in particular, whether it can be made polynomial in $k$.

Proving polynomial bounds on the size of the kernel for different parameterized problems has been an important practical aspect in the study of the parameterized complexity of NP-hard problems, and many positive results are known. See [17] for a survey of kernelization results. Recently Bodlaender et al. [4] building on the work of Fortnow and Santhanam [13] developed a lower-bound technique that allows one to prove that a number of parameterized problems do not admit polynomial kernels unless PH (the polynomial hierarchy) collapses to the third level. Dom et al. [8] have recently extended the techniques to show many more parameterized problems do not admit polynomial kernels under the same complexity-theoretic assumptions.

To the best of our knowledge, the class of $K_h$-topological-minor-free graphs is the most general class of graphs for which the $k$-DOMINATING SET problem has been previously shown to have a polynomial kernel. See Table 1 for some known upper bound results on kernel size for the $k$-DOMINATING SET problem on various classes of graphs.

| *Graph Class* | *FPT Algorithm Running Time* | *Kernel Size* |
|---|---|---|
| Planar | $O(k^4 + 2^{15.13\sqrt{k}}k + n^3)$ [12] | $O(k)$ [1, 5] |
| Genus-$g$ | $O((24g^2 + 24g + 1)^k n^2)$ [10] | $O(k + g)$ [11] |
| $K_h$-minor-free | $2^{O(\sqrt{k})}n^c$ [6], $O((\log h))^{hk/2} \cdot n$ [2] | $O(k^c)$ [3] |
| $K_h$-topological-minor-free | $(O(h))^{hk} \cdot n$ [2] | $O(k^c)$ [3] |
| $d$-degenerate | $k^{O(dk)}n$ [2] | $k^{O(dk)}$ [2] |

**Table 1.** Some known FPT and kernelization results for $k$-DOMINATING SET

*Our Results.* We show that for any fixed $i, j \geq 1$, the $k$-DOMINATING SET problem has a polynomial kernel on graphs that do not have $K_{i,j}$ (a complete bipartite graph with the two parts having $i$ and $j$ vertices) as a subgraph. For input graph $G$ and parameter $k$, the size of the kernel is bounded by $k^c$ where $c$ is a constant that depends only on $i$ and $j$. Since a $d$-degenerate graph does not have $K_{d+1,d+1}$ as a subgraph, it follows that the $k$-DOMINATING SET problem has a polynomial kernel on graphs of bounded degeneracy. This settles a question posed by Alon and Gutner in [3]. We also provide a slightly simpler and a smaller kernel for the version where we want the $k$-DOMINATING SET to be independent as well.

Note that except for $d$-degenerate graphs, all the other graph classes in Table 1 are minor-closed. This seems to be indicative of the state of the art — the only other previous FPT or kernelization result for the $k$-DOMINATING SET problem on a non-minor-closed class of graphs that we know of is the $O(k^3)$ kernel and the resulting FPT algorithm for graphs that exclude triangles and 4-cycles [18]. In fact, this result can be modified to obtain similar bounds on graphs with just no 4-cycles (allowing triangles). Since a 4-cycle is just $K_{2,2}$, this result follows from the main result of this paper by setting $i = j = 2$.

It is immediate from our result that for any fixed $i, j \geq 1$, there is an FPT algorithm for the $k$-DOMINATING SET problem on graphs that do not have $K_{i,j}$ as a subgraph. Since a $K_h$-topological-minor-free graph has bounded degeneracy [3, Proposition 3.1] (for a constant $h$), the class of $K_{i,j}$-free graphs is more general than the class of $K_h$-topological-minor-free graphs. Thus we extend the class of graphs for which the $k$-DOMINATING SET problem is known to have (1) FPT algorithms and (2) polynomial kernels to the class of $K_{i,j}$-free graphs.

*Organization of the rest of the paper.* In Section 2, we develop our main algorithm that runs in $O(n^{i+O(1)})$ time and constructs a kernel of size $O((j+1)^{2i}k^{2i^2})$ for $k$-DOMINATING SET on $K_{i,j}$-free graphs, for fixed $j \geq i \geq 2$. As a corollary we obtain, in Section 3, a polynomial kernel for $d$-degenerate graphs, with running time $O(n^{O(d)})$ and kernel size $O((d+2)^{2(d+1)}k^{2(d+1)^2})$. In Section 3.1 we describe an improvement to the above algorithm that applies to $d$-degenerate input graphs which yields a kernel of the same size as above and runs in time $O(2^d dn^2)$. In Section 4 we describe a modification of the algorithm in Section 2 that constructs a polynomial kernel for the $k$-INDEPENDENT DOMINATING SET problem on $K_{i,j}$-free graphs with the kernel having $O(jk^i)$ vertices, resulting in a kernel of size $O((d+1)k^{d+1})$ for $d$-degenerate graphs. In Section 5 we state our conclusions and list some open problems.

*Notation.* All the graphs in this paper are finite, undirected and simple. In general we follow the graph terminology from [7]. We let $V(G)$ and $E(G)$ denote, respectively, the vertex and edge sets of a graph $G$. The *open-neighborhood* of a vertex $v$ in a graph $G$, denoted $N(v)$, is the set of all vertices that are adjacent to $v$ in $G$. A *$k$-dominating set* of graph $G$ is a vertex-subset $S$ of size at most $k$ such that for each $u \in V(G) \setminus S$ there exists $v \in S$ such that $\{u, v\} \in E(G)$. Given a graph $G$ and $A, B \subseteq V(G)$, we say that $A$ dominates $B$ if every vertex in $B \setminus A$ is adjacent in $G$ to some vertex in $A$.

## 2 A Polynomial Kernel for $K_{i,j}$-free Graphs

In this section we consider the parameterized $k$-DOMINATING SET problem on graphs that do not have $K_{i,j}$ as a subgraph, for fixed $j \geq i \geq 1$. It is easy to see that the problem has a linear kernel when $i = 1, j \geq i$, so we consider the cases

$j \geq i \geq 2$. We solve a more general problem, namely the rwb-Dominating Set problem, defined as follows: Given a graph $G$ whose vertex set $V$ is partitioned into $R_G, W_G$, and $B_G$ (colored red, white, and black, respectively) and a non-negative integer parameter $k$, is there a subset $S \subseteq V$ of size at most $k$ such that $R_G \subseteq S$ and $S$ dominates $B_G$? We call such an $S$ an *rwb-dominating* set of $G$, and such a graph an *rwb-graph*.

Intuitively, the vertices colored red are those that will be picked up by the reduction rules in the $k$-dominating set $D$ that we are trying to construct. In particular, if there is a $k$-dominating set in the graph, there will be one that contains all the red vertices. White vertices are those that have been already dominated. Clearly all neighbors of red vertices are white, but our reduction rules color some vertices white even if they have no red neighbors (at that point). These are vertices that will be dominated by one of some constant number of vertices identified by the reduction rules. See reduction rule 2 for more details. Black vertices are those that are yet to be dominated. It is easy to see that if we start with a general graph $G$ and color all its vertices black to obtain an rwb-graph $G'$, then $G$ has a dominating set of size at most $k$ if and only if $G'$ has an rwb-dominating set of size at most $k$.

We first describe an algorithm that takes as input an rwb-graph $G$ on $n$ vertices and a positive number $k$, and runs in $O(n^{i+O(1)})$ time. The algorithm either finds that $G$ does not have any rwb-dominating set of size at most $k$, or it constructs an instance $(G', k')$ on $O((j+1)^{i+1}k^{i^2})$ vertices such that $G$ has an rwb-dominating set of size at most $k$ if and only if $G'$ has an rwb-dominating set of size at most $k'$.

The algorithm applies a sequence of reduction rules in a specified order. The input and output of each reduction rule are rwb-graphs. Each reduction rule satisfies the following correctness condition:

**Definition 1.** (Correctness) *A reduction rule $R$ is said to be* ***correct*** *if the following condition holds: if $(G', k')$ is the instance obtained from $(G, k)$ by one application of rule $R$ then $G'$ has an rwb-dominating set $D'$ of size $k'$ if and only if $G$ has an rwb-dominating set $D$ of size $k$.*

**Definition 2.** *We say that graph $G$ is* ***reduced*** *with respect to a reduction rule if an application of the rule to $G$ does not change $G$.*

### 2.1 The reduction rules and the kernelization algorithm

The kernelization algorithm assumes that the input graph is an rwb-graph. It applies the following rules exhaustively in the *given order*. Each rule is repeatedly applied till it causes no changes to the graph and then the next rule is applied.

For each rule below, let $G$ denote the graph on which the rule is applied, and $G'$ the resulting graph. Let $D$ and $D'$ be as in Definition 1.

**Rule 1.** Color all isolated black vertices of $G$ red.

Rule 1 is correct as the only way to dominate the isolated black vertices is by picking them in the proposed rwb-dominating set.

**Rule 2.** For $p = 1, 2, \ldots, i-2$, in this order, apply Rule 2.$p$ repeatedly till it no longer causes any changes in the graph.

**Rule 2.p** Let $b = jk$ if $p = 1$, and $b = jk^p + k^{p-1} + k^{p-2} \cdots + k$ if $2 \le p \le i-2$. If a set of $(i-p)$ vertices $U = \{u_1, u_2, \ldots, u_{i-p}\}$, none of which is red, has more than $b$ common black neighbors, then let $B$ be this set of neighbors.

1. Color all the vertices in $B$ white.
2. Add to the graph $(i-p)$ new (gadget) vertices $X = \{x_1, x_2, \ldots, x_{i-p}\}$ and all the edges $\{u, x\}; u \in U, x \in X$, as in Figure 1.
3. Color all the vertices in $X$ black.

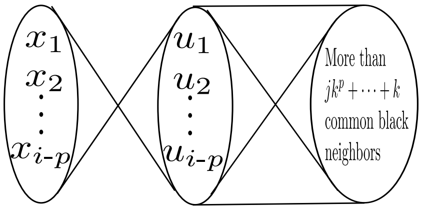

**Fig. 1.** Rule 2

The following claim is not difficult to prove:

*Claim 1.* Consider the application of Rule 2.$p$, $1 \le p \le i-2$. If $U$ is a set of vertices of $G$ that satisfies the condition in Rule 2.$p$, then at least one vertex in $U$ must be in any subset of $V(G)$ of size at most $k$ that dominates $B$.

*Proof.* We give a proof when $p = 1$. The proof for larger values of $p$ is along similar lines by reducing it to the case for smaller values of $p$ as in the proof of Claim 2 below.

When $p = 1$, suppose that there is a rwb-dominating set $D$ of $G$ of size at most $k$ that does not contain any vertex of $U$. Since $U$ has more than $b = jk$ common black neighbors, there is a vertex in $D$ that dominates at least $j + 1$ common black neighbors of $U$ (possibly including itself). That vertex along with $U$ form a $K_{i,j}$ in $G$ which is a contradiction to the property of the input graph. $\square$

**Lemma 1.** *Rule* 2.$p$ *is correct for* $1 \le p \le i-2$.

*Proof.* If $D$ exists, then $D \cap U \ne \emptyset$ by Claim 1, and so we can set $D' := D$ as $D \cap U$ dominates $X$. For the other direction, assume that $D'$ exists. If $D' \cap U = \emptyset$ then since $D'$ dominates $X$ and $X$ is independent, $X \subseteq D'$, and so set $D := D' \setminus X \cup U$. If $D' \cap X = \emptyset$ then since $D'$ dominates $X$, $D' \cap U \ne \emptyset$, and so set $D := D'$. If $D' \cap U \ne \emptyset$ and $D' \cap X \ne \emptyset$ then pick an arbitrary vertex $b \in B$ and set $D := D' \setminus X \cup \{b\}$. $\square$

**Rule 3.** If a black or white vertex $u$ has more than $jk^{i-1} + k^{i-2} + \cdots + k^2 + k$ black neighbors, then color $u$ red and color all the black neighbors of $u$ white.

*Claim 2.* Let $G$ be reduced with respect to Rules 1 and 2.1 to 2.$(i-2)$. If a black or white vertex $u$ of $G$ has more than $h = jk^{i-1} + k^{i-2} + \cdots + k^2 + k$ black neighbors (let this set of neighbors be $B$), then $u$ must be in any subset of $V(G)$ of size at most $k$ that dominates $B$.

*Proof.* Let $S \subseteq V(G)$ be a set of size at most $k$ that dominates $B$. If $S$ does not contain $u$, then there is a $v \in S$ that dominates at least $(h/k) + 1$ of the vertices in $B$. The vertex $v$ is not red, and $u, v$ have $h/k > jk^{i-2} + k^{i-3} + \cdots + 1$ common black neighbors, a contradiction to the fact that $G$ is reduced with respect to Rule 2.$(i-2)$. □

This proves the correctness of Rule 3 on graphs reduced with respect to rules 1 and 2.1 to 2.$(i-1)$.

**Rule 4.** If a white vertex $u$ is adjacent to at most one black vertex, then delete $u$ and apply Rule 1.

It is easy to see that Rule 4 is correct, since if $u$ has no black neighbor in $G$ then $u$ has no role in dominating $B_G$; if $u$ has a single black neighbor $v$ then we can replace $u$ with $v$ in $D'$.

**Rule 5.** If there is a white vertex $u$ and a white or black vertex $v$ such that $N(u) \cap B_G \subseteq N(v) \cap B_G$, then delete $u$ and apply Rule 1.

The correctness of this rule follows from the fact that if $D$ chooses $u$, then we can choose $v$ in $D'$.

**Rule 6.** If $|R_G| > k$ or $|B_G| > jk^i + k^{i-1} + k^{i-2} + \cdots + k^2$ then output "NO".

The correctness of the rule when $|R_G| > k$ is obvious as the proposed dominating set we construct should contain all of $R_G$. Note that for a graph $G$ reduced with respect to Rules 1 and 2.1 to 2.$(i-1)$ and 3, no white or black vertex has more than $jk^{i-1} + k^{i-2} + \cdots + k$ black neighbors, or else Rule 3 would have applied. Hence $k$ of these vertices can dominate at most $jk^i + k^{i-1} + k^{i-2} + \cdots + k^2$ black vertices and hence if $|B_G| > jk^i + k^{i-1} + k^{i-2} + \cdots + k^2$, the algorithm is correct in saying "NO".

### 2.2 Algorithm correctness and kernel size

We begin by noting the following.

*Remark 1.*

1.1 None of the reduction rules in Section 2 introduces a $K_{i,j}$ into a graph.
1.2 In the rwb-graphs constructed by the algorithm, red vertices have all white neighbors.

1.3 Let $R$ be any reduction rule, and let $R'$ be a rule that precedes $R$ in the given order. If $G$ is a graph that is reduced with respect to $R'$ and $G'$ is a graph obtained by applying $R$ to $G$, then $G'$ is reduced with respect to $R'$.

The following claim giving the correctness of the kernelization algorithm follows from the correctness of the reduction rules and the above remarks.

*Claim 3.* Let $G$ be the input rwb-graph and $H$ the rwb-graph constructed by the algorithm after applying all the reduction rules. Then $G$ has an rwb-dominating set of size at most $k$ if and only if there is an rwb-dominating set of size at most $k$ in $H$.

Now we move on to prove a polynomial bound on the size of the reduced instance.

**Lemma 2.** *Starting with a $K_{i,j}$-free rwb-graph $G$ as input, if the kernelization algorithm says "NO" then $G$ does not have an rwb-dominating set of size at most $k$. Otherwise, if the algorithm outputs the rwb-graph $H$, then $|V(H)| = O((j+1)^{i+1}k^{i^2})$.*

*Proof.* The correctness of the Rule 6 establishes the claim if the algorithm says "NO". Now suppose the algorithm outputs $H$ without saying "NO". The same rule establishes that $|R_H| \leq k$ and $b = |B_H| \leq jk^i + k^{i-1} + \cdots + k \leq (j+1)k^i$. Now we bound $|W_H|$. Note that no two white vertices have identical black neighborhoods, or else Rule 5 would have applied. Also each white vertex has at least two black neighbors, or else Rule 4 would have applied. Hence the number of white vertices that have less than $i$ black neighbors is at most $\binom{b}{2} + \binom{b}{3} + \cdots + \binom{b}{i-1} \leq 2b^{i-1}$. No set of $i$ black vertices has more than $(j-1)$ common white neighbors, or else these form a $K_{i,j}$. Hence the number of white vertices that have $i$ or more black neighbors is at most $\binom{b}{i}(j-1) \leq (j-1)b^i$. The bound in the lemma follows. □

The algorithm can be implemented in $O(n^{i+O(1)})$ time, as the main Rule 2 can be applied by running through various subsets of $V(G)$ of size $p$ for $p$ ranging from 1 to $i-2$. Thus, we have

**Lemma 3.** *For any fixed $j \geq i \geq 1$, the* rwb-Dominating Set *problem (with parameter $k$) on $K_{i,j}$-free graphs has a polynomial kernel with $O((j+1)^{i+1}k^{i^2})$ vertices.*

To obtain a polynomial kernel for the $k$-Dominating Set problem on $K_{i,j}$-free graphs, we first color all the vertices black and use Lemma 3 on this rwb-Dominating Set problem instance. To transform the reduced colored instance $H$ to an instance of (the uncolored) $k$-dominating Set, we can delete all red vertices. But we need to capture the fact that the white vertices need not be dominated. This can be done by, for example, adding a new vertex $v_x$ for every vertex $x$ in $W_H$ of the reduced graph $H$, and attaching $k + |W_H| + 1$ separate pendant vertices to each of the vertices $v_x$. It is easy to see that the new graph does not have a $K_{i,j}$, $j \geq i \geq 2$, if $H$ does not have one and that $H$ has at most $k$ black or white vertices dominating $B_H$ if and only if the resulting (uncolored) graph has a dominating set of size at most $|W_H| + k$. Thus after reducing to the uncolored version, $k$ becomes $k + |W_H|$ and the number of vertices increases by $(k + |W_H| + 2) \cdot |W_H|$. Hence by Lemma 3, we have

**Theorem 1.** *For any fixed $j \geq i \geq 1$, the $k$-*DOMINATING SET *problem on $K_{i,j}$-free graphs has a polynomial kernel with $O((j+1)^{2(i+1)} k^{2i^2})$ vertices.*

## 3 A Polynomial Kernel for *d*-degenerate Graphs

A $d$-degenerate graph does not contain $K_{d+1,d+1}$ as a subgraph, and so the kernelization algorithm of the previous section can be applied to a $d$-degenerate graph, setting $i = j = d+1$. The algorithm runs in time $O((d+1)^2 n^{d+O(1)})$ and constructs a kernel with $O((d+2)^{2(d+2)} \cdot k^{2(d+1)^2})$ vertices. Since a $d$-degenerate graph on $v$ vertices has at most $dv$ edges, we have:

**Corollary 1.** *The $k$-*DOMINATING SET *problem on $d$-degenerate graphs has a kernel on $O((d+2)^{2(d+2)} \cdot k^{2(d+1)^2})$ vertices and edges.*

Corollary 1 settles an open problem posed by Alon and Gutner in [3].

### 3.1 Improving the running time

We describe a modification of our algorithm to $d$-degenerate graphs that makes use of the following well known property of $d$-degenerate graphs, to reduce the running time to $O(2^d \cdot n^3)$; the bound on the kernel size remains the same.

**Fact 1.** [14, Theorem 2.10] Let $G$ be a $d$-degenerate graph on $n$ vertices. Then one can compute, in $O(dn)$ time, an ordering $v_1, v_2, \ldots, v_n$ of the vertices of $G$ such that for $1 \leq i \leq n$, $v_i$ has at most $d$ neighbors in the subgraph of $G$ induced on $\{v_{i+1}, \ldots, v_n\}$.

The modification to the algorithm pertains to the way rules 2.1 to 2.$(d-1)$ are implemented: the rest of the algorithm remains the same.

In implementing Rule 2.$p$, $1 \leq p \leq (d-1)$, instead of checking each $(d-p+1)$-subset of vertices in the graph to see if it satisfies the condition in the rule, we make use of Fact 1 to quickly find such a set of vertices, if it exists. Let $G$ be the graph instance on $n$ vertices on which Rule 2.$p$ is to be applied. First we delete, temporarily, all the red vertices in $G$. We then find an ordering $v_1, v_2, \ldots, v_n$ of the kind described in the above fact, of all the remaining vertices in $G$. Let $U$ and $B$ be as defined in the rule. The first vertex $v_l$ in $U \cup B$ that appears in the ordering has to be from $B$, since each vertex in $U$ has degree greater than $d$. The vertex $v_l$ will then have a neighborhood of size $d-p+1$ that in turn has $B$ as its common neighborhood. We use this fact to look for such a pair $(U, B)$ and exhaustively apply Rule 2.$p$ to $G$. See Algorithm 1 for a pseudocode of the

**Algorithm 1** Faster implementation of Rule 2.$p$ in $d$-degenerate graphs.

```
for $l := 1$ to $n$
do
  if $v_l$ is black and its degree in $G[v_{l+1}, \ldots, v_n]$ is at least $d - p + 1$
  then
    Find the neighborhood $N$ of $v_l$ in $G[v_{l+1}, \ldots, v_n]$
    for each $(d - p + 1)$-subset $S$ of $N$
    do
      if $S$ has more than $(d + 1)k^p + k^{p-1} + \cdots + k$
      common black neighbors in $G$
      then
        Apply the three steps of Rule $2.p$, taking $S$ as $U$
      endif
    done
  endif
done
```

algorithm. We then add back the red vertices that we deleted prior to this step, along with all their edges to the rest of the graph.

As $|N| \leq d$, the inner *for* loop is executed at most $\binom{d}{p-1}$ times for each iteration of the outer loop. Each of the individual steps in the algorithm can be done in $O(dn)$ time, and so Rule 2.$p$ can be applied in $O(dn \sum_{l=1}^{n} \binom{d}{p-1})$ time. All the rules 2.$p$ can therefore be applied in $O(dn \sum_{l=1}^{n} \sum_{p=1}^{d-1} \binom{d}{p-1}) = O(2^d \cdot dn^2)$ time. Thus we have:

**Theorem 2.** *For any fixed $d \geq 1$, the $k$-*DOMINATING SET *problem on $d$-degenerate graphs has a kernel on $O((d+2)^{2(d+2)} \cdot k^{(2(d+1))^2})$ vertices and edges, and this kernel can be found in $O(2^d \cdot dn^2)$ time for an input graph on $n$ vertices.*

## 4 A polynomial kernel for Independent Dominating Set on $K_{i,j}$-free graphs

The $k$-INDEPENDENT DOMINATING SET problem asks, for a graph $G$ and a positive integer $k$ given as inputs, whether $G$ has a dominating set $S$ of size at most $k$ such that $S$ is an independent set (i.e. no two vertices in $S$ are adjacent). This problem is known to be NP-hard for general graphs [15], and the problem parameterized by $k$ is $W[2]$-complete [9]. Using a modified version of the set of reduction rules in Section 2 we show that the $k$-INDEPENDENT DOMINATING SET has a polynomial kernel in $K_{ij}$-free graphs for $j \geq i \geq 1$. For $i = 1, j \geq 1$ we can easily obtain trivial kernels as before, and for $i = 2, j \geq 2$ a simplified version of the following algorithm gives a kernel of size $O(j^3 k^4)$.

### 4.1 The reduction rules

Rule 1 is the same as for the DOMINATING SET kernel for $K_{ij}$-free graphs (Section 2.1). Rules 2.1 to 2.$(i-2)$ and Rule 3 are modified to make use of the fact that we are looking for a dominating set that is independent. A vertex $u$ that is made white will never be part of the independent dominating set $D$ that is sought to be constructed by the algorithm, since $u$ is adjacent to some vertex $v \in D$. So a vertex can be deleted as soon as it is made white. Also, rules $1, 2.1 \ldots 2.(i-2)$ and 3 are the only rules. Rules 4 and 5 from that section do not apply, because of the same reason as above. The modified rules ensure that no vertex is colored white, and so they work on *rb-graphs*: graphs whose vertex set is partitioned into red and black vertices. Using these modified rules, the bounds of $|R_H|$ and $|B_H|$ in the proof of Lemma 2, and the fact that there are no white vertices, we have

**Theorem 3.** *For any fixed $j \geq i \geq 1$, the $k$-*Independent Dominating Set *problem on $K_{i,j}$-free graphs has a polynomial kernel with $O(jk^i)$ vertices.*

For $d$-degenerate graphs, we have $i = j = d + 1$, and therefore we have:

**Corollary 2.** *For any fixed $d \geq 1$, the $k$-*Independent Dominating Set *problem on $d$-degenerate graphs has a polynomial kernel with $O((d+1)k^{(d+1)})$ vertices.*

## 5 Conclusions and Future Work

In this paper, we presented a polynomial kernel for the $k$-Dominating Set problem on graphs that do not have $K_{i,j}$ as a subgraph, for any fixed $j \geq i \geq 1$. We used this to show that the $k$-Dominating Set problem has a polynomial kernel of size $O((d+2)^{2(d+2)} \cdot k^{2(d+1)^2})$ on graphs of bounded degeneracy, thereby settling an open problem from [3]. Our algorithm also yielded a slightly simpler and a smaller kernel for the $k$-Independent Dominating Set problem on $K_{i,j}$-free and $d$-degenerate graphs. These algorithms are based on simple reduction rules that look at the common neighborhoods of sets of vertices. It has recently been shown that the $k$-Dominating Set problem on $d$-degenerate graphs does not have a kernel of size polynomial in both $d$ and $k$ unless the Polynomial Hierarchy collapses to the third level [8]. This shows that the kernel size that we obtained for this class of graphs cannot possibly be significantly improved.

Many interesting classes of graphs are of bounded degeneracy. These include all nontrivial minor-closed families of graphs such as planar graphs, graphs of bounded genus, graphs of bounded treewidth, and graphs excluding a fixed minor, and some non-minor-closed families such as graphs of bounded degree. Graphs of degeneracy $d$ are $K_{d+1,d+1}$-free. Since any $K_{i,j}; j \geq i \geq 2$ contains a 4-cycle, every graph of girth 5 is $K_{i,j}$-free. From [19, Theorem 1], there exist graphs of girth 5 and arbitrarily large degeneracy. Hence $K_{i,j}$-free graphs are strictly more general than graphs of bounded degeneracy. To the best of our knowledge, $K_{i,j}$-free graphs form the largest class of graphs for which FPT algorithms and polynomial kernels are known for the dominating set problem variants discussed in this paper.

One interesting direction of future work is to try to improve the running times of the kernelization algorithms: to remove the exponential dependence on

$d$ of the running time for $d$-degenerate graphs, and to get a running time of the form $O(n^c)$ for $K_{i,j}$-free graphs where $c$ is independent of $i$ and $j$.

*Acknowledgements.* We thank Aravind Natarajan for pointing out the connection between $K_{i,j}$-free and $d$-degenerate graphs, Saket Saurabh and the other authors of [8] for sharing with us the lower-bound result mentioned in their paper, and Shai Gutner for comments on an earlier draft of the paper.

## References

1. Alber, J., Fellows, M. R., and Niedermeier, R. Polynomial-time data reduction for Dominating Set. *Journal of the ACM 51*, 3 (2004), 363–384.
2. Alon, N., and Gutner, S. Linear time algorithms for finding a dominating set of fixed size in degenerated graphs. In *Computing and Combinatorics, 13th Annual International Conference, COCOON 2007, Banff, Canada, July 16-19, 2007, Proceedings* (2007), vol. 4598 of *Lecture Notes in Computer Science*, Springer, pp. 394–405.
3. Alon, N., and Gutner, S. Kernels for the dominating set problem on graphs with an excluded minor. Technical Report TR08-066, The Electronic Colloquium on Computational Complexity (ECCC), 2008.
4. Bodlaender, H. L., Downey, R. G., Fellows, M. R., and Hermelin, D. On problems without polynomial kernels (extended abstract). In *Automata, Languages and Programming, 35th International Colloquium, ICALP 2008, Reykjavik, Iceland, July 7-11, 2008, Proceedings, Part I: Tack A: Algorithms, Automata, Complexity, and Games* (2008), Lecture Notes in Computer Science, Springer, pp. 563–574.
5. Chen, J., Fernau, H., Kanj, I. A., and Xia, G. Parametric duality and kernelization: Lower bounds and upper bounds on kernel size. *SIAM Journal of Computing 37*, 4 (2007), 1077–1106.
6. Demaine, E. D., Fomin, F. V., Hajiaghayi, M., and Thilikos, D. M. Subexponential parameterized algorithms on bounded-genus graphs and $H$-minor-free graphs. *Journal of the ACM 52*, 6 (2005), 866–893.
7. Diestel, R. *Graph theory*, 2 ed. Springer-Verlag, New York, 2000.
8. Dom, M., Lokshtanov, D., and Saurabh, S. Incompressibility through Colors and IDs. Manuscript, 2008.
9. Downey, R. G., and Fellows, M. R. *Parameterized Complexity.* Springer, 1999.
10. Ellis, J. A., Fan, H., and Fellows, M. R. The Dominating Set problem is fixed parameter tractable for graphs of bounded genus. *Journal of Algorithms 52*, 2 (2004), 152–168.
11. Fomin, F. V., and Thilikos, D. M. Fast parameterized algorithms for graphs on surfaces: Linear kernel and exponential speed-up. In *Automata, Languages and Programming: 31st International Colloquium, ICALP 2004, Turku, Finland, July 12-16, 2004. Proceedings* (2004), vol. 3142 of *Lecture Notes in Computer Science*, Springer, pp. 581–592.
12. Fomin, F. V., and Thilikos, D. M. Dominating sets in planar graphs: Branch-width and exponential speed-up. *SIAM Journal of Computing 36*, 2 (2006), 281–309.

13. Fortnow, L., and Santhanam, R. Infeasibility of instance compression and succinct PCPs for NP. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada, May 17-20, 2008* (2008), ACM, pp. 133–142.
14. Franceschini, G., Luccio, F., and Pagli, L. Dense trees: a new look at degenerate graphs. *Journal of Discrete Algorithms 4* (2006), 455–474.
15. Garey, M. R., and Johnson, D. S. *Computers and Intractability: A Guide to the Theory of NP–Completeness.* Freeman, San Francisco, 1979.
16. Golovach, P. A., and Villanger, Y. Parameterized complexity for domination problems on degenerate graphs. In *Proceedings of the 34th International Workshop on Graph-Theoretic Concepts in Computer Science, WG 2008* (2008), vol. 5344 of *Lecture Notes in Computer Science*, Springer.
17. Guo, J., and Niedermeier, R. Invitation to data reduction and problem kernelization. *SIGACT News 38*, 1 (2007), 31–45.
18. Raman, V., and Saurabh, S. Short cycles make W-hard problems hard: FPT algorithms for W-hard problems in graphs with no short cycles. *Algorithmica 52*, 2 (2008), 203–225.
19. Sachs, H. Regular graphs with given girth and restricted circuits. *Journal of the London Mathematical Society s1-38*, 1 (1963), 423–429.