Parameterized and Exact Algorithms for Class Domination Coloring

R. Krithika[1], Ashutosh Rai[1], Saket Saurabh[1,2] ⋆, and Prafullkumar Tale[1]

[1] The Institute of Mathematical Sciences, HBNI, Chennai, India.
{rkrithika|ashutosh|saket|pptale}@imsc.res.in
[2] University of Bergen, Norway.

Abstract. A class domination coloring (also called as cd-coloring) of a graph is a proper coloring such that for every color class, there is a vertex that dominates it. The minimum number of colors required for a cd-coloring of the graph G, denoted by $\chi_{cd}(G)$, is called the class domination chromatic number (cd-chromatic number) of G. In this work, we consider two problems associated with the cd-coloring of a graph in the context of exact exponential-time algorithms and parameterized complexity. (1) Given a graph G on n vertices, find its cd-chromatic number. (2) Given a graph G and integers k and q, can we delete at most k vertices such that the cd-chromatic number of the resulting graph is at most q? For the first problem, we give an exact algorithm with running time $\mathcal{O}(2^n n^4 \log n)$. Also, we show that the problem is FPT with respect to the number of colors q as the parameter on chordal graphs. On graphs of girth at least 5, we show that the problem also admits a kernel with $\mathcal{O}(q^3)$ vertices. For the second (deletion) problem, we show NP-hardness for each $q \geq 2$. Further, on split graphs, we show that the problem is NP-hard if q is a part of the input and FPT with respect to k and q. As recognizing graphs with cd-chromatic number at most q is NP-hard in general for $q \geq 4$, the deletion problem is unlikely to be FPT when parameterized by the size of deletion set on general graphs. We show fixed parameter tractability for $q \in \{2, 3\}$ using the known algorithms for finding a vertex cover and an odd cycle transversal as subroutines.

1 Introduction

A *dominating set* is a set of vertices whose closed neighbourhood contains the vertex set of the graph. A *proper coloring* of a graph is a partition of its vertex set into independent sets. That is, the graph induced by all the vertices of any partition is an independent set. The minimum number of colors in any proper coloring is called as the *chromatic number* of the graph. Given a graph G, the Dominating Set problem is to find a minimum dominating set of G and the Graph Coloring or Chromatic Number problem is to compute a coloring

⋆ The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013) / ERC grant agreement no. 306992

that uses the minimum number of colors. DOMINATING SET and GRAPH COLORING are two classical problems in the field of combinatorics and combinatorial algorithms. Being notoriously hard problems, they have been studied extensively in algorithmic realms like exact algorithms [3,17,18,24,25,33], approximation algorithms [4,22,23,26], and parameterized algorithms [1,2,5,14]. Further, the complexity of variants of these two problems like EDGE-CHROMATIC NUMBER, ACHROMATIC NUMBER, b-CHROMATIC NUMBER, INDEPENDENT DOMINATING SET and CONNECTED DOMINATING SET have also been widely investigated in the literature.

DOMINATOR COLORING and CLASS DOMINATION COLORING are two problems that have the flavour of both DOMINATING SET and CHROMATIC NUMBER. The former problem, introduced in [20], is the task of determining a minimum proper coloring of the graph such that every vertex contains at least one color class in its neighbourhood. The latter problem, also known as CD-COLORING, is to obtain a minimum proper coloring such that every color class is contained in the neighbourhood of some vertex. The decision versions of both the problems are known to be NP-complete when the number of colors is at least four and polynomial-time solvable otherwise [19,35]. Characterization of graphs that admit such colorings using at most 3 colors are also known [6,35]. In this paper, we study CLASS DOMINATION COLORING which is formally defined as follows.

> CD-COLORING
> **Input:** A graph G, an integer q.
> **Question:** Can G be properly colored using at most q colors such that every color class is contained in the neighbourhood of some vertex?

The minimum number of colors needed in any cd-coloring of G is called the *class domination chromatic number (cd-chromatic number)* and is denoted by $\chi_{cd}(G)$. Also, G is said to be q-cd-colorable if $\chi_{cd}(G) \leq q$. The problem has also been studied on many restricted graph classes like split graphs, P_4-free graphs [35] and middle and central graphs of $K_{1,n}$, C_n and P_n [37]. We study this problem in the context of exact exponential-time algorithms and parameterized complexity. The field of exact algorithms typically deal with designing algorithms for NP-hard problems that are faster than brute-force search while the goal in parameterized complexity is to provide efficient algorithms for NP-complete problems by switching from the classical view of single-variate measure of the running time to a multi-variate one. In parameterized complexity, we consider instances (I, k) of parameterized a problem $\Pi \subseteq \Sigma^* \times \mathbb{N}$, where Σ is a finite alphabet. Algorithms in this area have running times of the form $f(k)|I|^{\mathcal{O}(1)}$, where k is an integer measuring some part of the problem. This integer k is called the *parameter*, and a problem that admits such an algorithm is said to be *fixed-parameter tractable* (FPT). In most of the cases, the solution size is taken to be the parameter, which means that this approach results in efficient (polynomial-time) algorithms when the solution is of small size. A *kernelization* algorithm for a parameterized problem Π is a polynomial time procedure which takes as input an instance (x, k) of Π and returns an instance (x', k') such that

$(x,k) \in \Pi$ if and only if $(x',k') \in \Pi$ and $|x'| \leq h(k)$ and $k' \leq g(k)$, for some computable functions h, g. The returned instance is called as the *kernel* for Π and $h(k)+g(k)$ is said to be the *size of the kernel*. We say that Π admits a *polynomial kernel* if h and g are polynomial. For more background on parameterized complexity, we refer the reader to the monographs [11,13,15,30].

We first observe that parameterizing cd-Coloring by the solution size (which is the number of colors) does not help in designing efficient algorithms as the problem is para-NP-hard (NP-hard even for constant values of the parameter). Hence, this problem is unlikely to be FPT when parameterized by the solution size. Then, we describe an $\mathcal{O}(2^n n^4 \log n)$-time algorithm for finding the cd-chromatic number of a graph using polynomial method. Next, we show that cd-Coloring is FPT when parameterized by the number of colors and the treewidth of the input graph. Further, we show that the problem is FPT when parameterized by the number of colors on chordal graphs. Kaminski and Lozin [29] showed that determining if a graph of girth at least g admits a proper coloring with at most q colors or not is NP-complete for any fixed $q \geq 3$ and $g \geq 3$. In particular, Chromatic Number is para-NP-hard for graphs of girth at least 5. In contrast, we show that cd-Coloring is FPT on this graph class and admits a kernel with $\mathcal{O}(q^3)$ vertices.

On a graph G that is not q-cd-colorable, a natural optimization question is to check if we can delete at most k vertices from G such that the cd-chromatic number of the resultant graph is at most q? We define this problem as follows.

cd-Partization
Input: Graph G, integers k and q
Question: Does there exist $S \subseteq V(G)$, $\lvert S \rvert \leq k$, such that $\chi_{cd}(G-S) \leq q$?

If q is fixed, then we refer to the problem as q-cd-Partization. Once again, from parameterized complexity point of view, this question is not interesting on general graphs for values of q greater than three, as in those cases, an FPT algorithm with deletion set (solution) size as the parameter is a polynomial-time recognition algorithm for q-cd-colorable graphs. Hence, the deletion question is interesting only on graphs where the recognition problem is polynomial-time solvable. We show that q-cd-Partization is NP-complete for each $q \geq 2$, and that for $q \in \{2,3\}$, the problem is FPT with respect to the solution size as the parameter. Our algorithms use the known parameterized algorithms for finding a vertex cover and an odd cycle transversal as subroutines. We also show that cd-Partization remains NP-complete on split graphs and is FPT when parameterized by the number of colors and solution size.

2 Preliminaries

The set of integers $\{1, 2, \ldots, k\}$ is denoted by $[k]$. All graphs considered in this paper are finite, undirected and simple. For the terms which are not explicitly defined here, we use standard notations from [12]. For a graph G, its vertex set is denoted by $V(G)$ and its edge set is denoted by $E(G)$. For a vertex $v \in V(G)$, its (open) *neighbourhood* $N_G(v)$ is the set of all vertices adjacent to it and its *closed neighborhood* is the set $N_G(v) \cup \{v\}$. We omit the subscript in the notation for neighbourhood if the graph under consideration is implicitly clear.The degree of a vertex v is the size of its open neighborhood.

For a set $S \subseteq V(G)$, the *subgraph of G induced by S*, denoted by $G[S]$, is defined as the subgraph of G with vertex set S and edge set $\{(u,v) \in E(G) : u,v \in S\}$. The subgraph of G obtained after deleting S (and the edges incident on it) is denoted as $G - S$. The *girth* of a graph is the length of a smallest cycle. A set $D \subseteq V(G)$ is said to be a *dominating set* of G if every vertex in $V(G) \setminus D$ is adjacent to some vertex in D. A dominating set is called *total dominating set* if every vertex in $V(G)$ is adjacent to some vertex in it.

A *proper coloring* of G with q colors is a function $f : V(G) \to [q]$ such that for all $(u,v) \in E(G)$, $f(u) \neq f(v)$. For a proper coloring f of G with q colors and $i \in [q]$, $f^{-1}(i) \subseteq V(G)$ is called a *color class* in the coloring f. The *chromatic number* $\chi(G)$ of G is the minimum number of colors required in a proper coloring of G. A *clique* is a graph which has an edge between every pair of vertices. The *clique number* $\omega(G)$ of G is the size of a largest clique which is a subgraph of G. A *vertex cover* is a set of vertices that contains at least one endpoint of every edge in the graph. An *independent set* is a set of pairwise nonadjacent vertices. A graph is said to be a *bipartite graph* if its vertex set can be partitioned into 2 independent sets. An *odd cycle transversal* is a set of vertices whose deletion results in a bipartite graph. A *tree-decomposition* of a graph G is a pair $(\mathbb{T}, \mathcal{X} = \{X_t\}_{t \in V(\mathbb{T})})$ such that

- $\bigcup_{t \in V(\mathbb{T})} X_t = V(G)$,
- for every edge $(x,y) \in E(G)$ there is a $t \in V(\mathbb{T})$ such that $\{x,y\} \subseteq X_t$, and
- for every vertex $v \in V(G)$ the subgraph of $\mathbb{T}$ induced by the set $\{t \mid v \in X_t\}$ is connected.

The *width* of a tree decomposition is $\max_{t \in V(\mathbb{T})} |X_t| - 1$ and the *treewidth* of G, denoted by $\mathbf{tw}(G)$, is the minimum width over all tree decompositions of G. The syntax of *Monadic Second Order Logic (MSO)* of graphs includes the logical connectives $\vee$, $\wedge$, $\neg$, $\Rightarrow$, $\Leftrightarrow$, variables for vertices, edges, sets of vertices, sets of edges, the quantifiers $\forall$, $\exists$ that can be applied to these variables and the following five binary relations.

- $u \in U$ where u is a vertex variable and U is a vertex set variable;
- $e \in F$ where e is an edge variable and F is an edge set variable;
- $\mathbf{inc}(e,u)$, where e is an edge variable, u is a vertex variable, and the interpretation is that the edge e is incident with the vertex u;
- $\mathbf{adj}(u,v)$, where u and v are vertex variables and the interpretation is that u and v are adjacent;
- equality of variables representing vertices, edges, sets of vertices, and sets of edges.

For a MSO formula ϕ, $||\phi||$ denotes the length of its encoding as a string.

Theorem 1 (Courcelle's theorem, [9,10]). *Let ϕ be a graph property that is expressible in MSO. Suppose G is a graph on n vertices with treewidth tw equipped with the evaluation of all the free variables of ϕ. Then, there is an algorithm that verifies whether ϕ is satisfied in G in $f(||\phi||, tw) \cdot n$ time for some computable function f.*

We end the preliminaries section with following simple observations.

Observation 1 *If $G_1, \ldots, G_l$ are the connected components of G, then $\chi_{cd}(G) = \sum_{i=1}^{l} \chi_{cd}(G_i)$.*

Observation 2 *If G is q-cd-colorable, then G has a dominating set of size at most q.*

3 Exact Algorithm for cd-Chromatic Number

Let G denote the input graph on n vertices. Given a coloring of $V(G)$, we can check in polynomial time whether it is a cd-coloring or not. Therefore, to compute $\chi_{cd}(G)$, we can iterate over all possible colorings of $V(G)$ with at most n colors and return the valid cd-coloring that uses the minimum number of colors. This brute force algorithm runs in $2^{\mathcal{O}(n \log n)}$ time. In this section we present an algorithm which runs in $\mathcal{O}(2^n n^4 \log(n))$ time. The idea for this algorithm is inspired by an exact algorithm for b-Chromatic Number presented in [31]. We first list some preliminaries on polynomials and Fast Fourier Transform following the framework of [31].

A binary vector ϕ is a finite sequence of bits and $val(\phi)$ denotes the integer d of which ϕ is the binary representation. All vectors considered here are binary vectors and are synonymous to binary numbers. Further, they are the binary representations of integers less than 2^n and are assumed to consist of n bits. $\phi_1+\phi_2$ denotes the vector obtained by the bitwise addition of the binary numbers (vectors) ϕ_1 and ϕ_2. Let $U = \{u_1, u_2, \ldots, u_n\}$ denote a universe with a fixed ordering on its elements. The *characteristic vector* of a set $S \subseteq U$, denoted by $\psi(S)$, is the vector of length $|U|$ whose j^{th} bit is 1 if $u_j \in S$ and 0 otherwise. The *Hamming weight* of a vector ϕ is the number of 1s in ϕ and it is denoted by $\mathcal{H}(\phi)$. Observe that $\mathcal{H}(\psi(S)) = |S|$. The Hamming weight of an integer is define as hamming weight of its binary representation. To obtain the claimed running time bound for our exponential-time algorithm, we make use of the algorithm for multiplying polynomials based on the Fast Fourier Transform.

Lemma 1 ([34]). *Two polynomials of degree at most d over any commutative ring $\mathcal{R}$ can be multiplied using $\mathcal{O}(d \cdot \log d \cdot \log\log d)$ additions and multiplications in $\mathcal{R}$.*

Let z denote an indeterminate variable. We use the monomial $z^{val(\psi(S))}$ to represent the set $S \subseteq U$ and as a natural extension, we use univariate polynomials to represent a family of sets.

Definition 1 (Characteristic Polynomial of a Family of Sets). *For a family $\mathcal{F} = \{S_1, S_2, \ldots, S_q\}$ of subsets of U, the characteristic polynomial of $\mathcal{F}$ is defined as $p_\psi(\mathcal{F}) = \sum_{i=1}^{q} z^{val(\psi(S_i))}$.*

Definition 2 (Representative Polynomial). *For a polynomial $p(z) = \sum_{i=1}^{q} a_i \cdot z^i$, we define its representative polynomial as $\sum_{i=1}^{q} b_i \cdot z^i$ where $b_i = 1$ if $a_i \neq 0$ and $b_i = 0$ if $a_i = 0$.*

Definition 3 (Hamming Projection). *The Hamming projection of the polynomial $p(z) = \sum_{i=1}^{q} a_i \cdot z^i$ to the integer h is defined as $\mathcal{H}_h(p(z)) := \sum_{i=1}^{q} b_i \cdot z^i$ where $b_i = a_i$ if $\mathcal{H}(i) = h$ and $b_i = 0$ otherwise.*

Next, for two sets $S_1, S_2 \subseteq U$, we define a modified multiplication operation ($\star$) of the monomials $z^{\psi(S_1)}$ and $z^{\psi(S_2)}$ in the following way.

$$z^{val(\psi(S_1))} \star z^{val(\psi(S_2))} = \begin{cases} z^{val(\psi(S_1))+val(\psi(S_2))} & \text{if } S_1 \cap S_2 = \emptyset \\ 0 & \text{otherwise} \end{cases}$$

For a polynomial function $p(z)$ of z and a positive integer $\ell \geq 2$, we inductively define the polynomial $p(z)^\ell$ as $p(z)^\ell := p(z)^{\ell-1} \star p(z)$. Here, coefficients of monomials follow addition and multiplications defined over underlying field. We now describe an algorithm for implementing the $\star$ operation using the standard multiplication operation and the notion of Hamming weights of bit strings associated with exponents.

Algorithm 3.1: Compute ($\star$) product of two polynomials

Input: Two polynomials $q(z), r(z)$ of degree at most 2^n
Output: $q(z) \star r(z)$

Initialize polynomials $t(z)$ and $t'(z)$ to 0
for *each ordered pair (i, j) such that $i + j \leq n$* **do**
  Compute $s_i(z) = \mathcal{H}_i(q(z))$ and $s_j(z) = \mathcal{H}_j(r(z))$
  Compute $s_{ij}(z) = s_i(z) * s_j(z)$ using Lemma 1
  $t'(z) = t(z) + \mathcal{H}_{i+j}(s_{ij}(z))$
  Set $t(z)$ as the representative polynomial of $t'(z)$
return $t(z)$

Lemma 2. *Let $\mathcal{F}_1$ and $\mathcal{F}_2$ be two families of subsets of U. Let $\mathcal{F}$ denote the collection $\{S_1 \cup S_2 | \ S_1 \in \mathcal{F}_1, S_2 \in \mathcal{F}_2$ and $S_1 \cap S_2 = \emptyset\}$. Then, $p_\psi(\mathcal{F}_1) \star p_\psi(\mathcal{F}_2)$ computed by Algorithm 3.1 is $p_\psi(\mathcal{F})$.*

Proof. Define $q(z) = p_\psi(\mathcal{F}_1)$, $r(z) = p_\psi(\mathcal{F}_2)$ and $t(z) = q(z) \star r(z)$. Let $S_1 \in \mathcal{F}_1$ and $S_2 \in \mathcal{F}_2$ be sets such that $S_1 \cap S_2 = \emptyset$. Define $S = S_1 \cup S_2$ and let ϕ_1, ϕ_2 and ϕ be the characteristic vectors of S_1, S_2 and S respectively. We claim that the term $z^{val(\phi)}$ is present in $t(z)$. For a vector ϕ and an integer $i \in [n]$, let $\phi[i]$ denote the i^{th} bit in ϕ. As $\phi[i]$ is 1 if and only if exactly one of the two bits $\phi_1[i]$, $\phi_2[i]$ is 1, it follows that there is no carry at any position (and hence no overflow) while adding ϕ_1 and ϕ_2. Therefore, $\phi = \phi_1 + \phi_2$ is a binary string of n bits and $\mathcal{H}(\phi) = \mathcal{H}(\phi_1) + \mathcal{H}(\phi_2)$. Now, as $q(z)$ contains $z^{val(\phi_1)}$ and $r(z)$ contains $z^{val(\phi_2)}$, in the execution of Algorithm 3.1, for $i = |S_1|$ and $j = |S_2|$, polynomials $s_i(z)$ and $s_j(z)$ contain $z^{val(\phi_1)}$ and $z^{val(\phi_2)}$ respectively. Step 4 multiplies $s_i(z)$ and $s_j(z)$ using Fast Fourier Transformation to obtain $s_{ij}(z)$. As $\mathcal{H}(\phi_1) = i$, $\mathcal{H}(\phi_2) = j$ and $\mathcal{H}(\phi_1) + \mathcal{H}(\phi_2) = i + j$, $s_{ij}(z)$ contains the term $z^{val(\phi)} = z^{val(\phi_1)+val(\phi_2)}$. Moreover, $z^{val(\phi)}$ is present in $\mathcal{H}_{i+j}(s_{ij}(z))$ and hence it is a monomial in $t(z)$ as Step 6 ensures that every monomial in $t(z)$ is of the form z^d for some integer d.

Next, we show that for every monomial z^d in in $t(z)$, there is a set $S \in \mathcal{F}$ such that $d = val(\psi(S))$. Let i and j be integers such that $\mathcal{H}_{i+j}(s_{ij}(z))$ contains the term z^d. As $t(z)$ was initialized to 0, z^d was obtained as the product of two terms z^{d_1}, z^{d_2} in $s_i(z)$ and $s_j(z)$ respectively such that $d_1 + d_2 = d$. Let $S_1 \in \mathcal{F}_1$ be the set such that $\psi(S_1)$ is the binary representation of d_1. Similarly, let $S_2 \in \mathcal{F}_2$ be the set such that $\psi(S_2)$ is the binary representation of d_2. Let ϕ_1 and ϕ_2 be the characteristic vectors of S_1 and S_2 respectively. Then, $|S_1| = i$, $|S_2| = j$ and there is no integer k between 1 and n such that $\phi_1[k] = \phi_2[k] = 1$. Therefore, $S_1 \cap S_2 = \emptyset$ and $z^d = z^{val(S_1 \cup S_2)}$. Hence, the claimed set S is $S_1 \cup S_2$ which is in $\mathcal{F}$ as $S_1 \cap S_2 = \emptyset$. $\square$

Corollary 1. *Given a polynomial $p(z)$ of degree at most 2^n, there is an algorithm that computes $p(z)^\ell$ in $\mathcal{O}(2^n n^3 \log n \cdot l)$ time.*

Proof. By Lemma 1, an execution of the Fast Fourier multiplication algorithm takes $\mathcal{O}(2^n n \log n)$ time. As the **for** loop of Algorithm 3.1 is executed n^2 times, the total time to compute $p(z)^\ell$ is $\mathcal{O}(2^n n^3 \log n)$. $\square$

We now prove a result which correlates the existence of a partition of a set with the presence of a monomial in a polynomial associated with it.

Lemma 3. *Consider a universe U and a family $\mathcal{F}$ of its subsets with characteristic polynomial $p(z)$. For any $W \subseteq U$, W is the disjoint union of ℓ sets from $\mathcal{F}$ if and only if there exists a monomial $z^{val(\psi(W))}$ in $p(z)^\ell$.*

Proof. Let W be the disjoint union of $S_1, S_2, \ldots, S_\ell$ such that $S_i \in \mathcal{F}$ for all $i \in [\ell]$. For any $j \in [n]$, the j^{th} bit of $\psi(W)$ is 1 if and only if there is exactly one S_i such that j^{th} bit of $\psi(S_i)$ is 1. Thus, $val(\psi(W)) = val(\psi(S_1)) + val(\psi(S_2)) + \cdots + val(\psi(S_\ell))$. Now, for every S_i there is a term $z^{val(\psi(S_i))}$ in $p(z)$. Further, as the S_i's are pairwise disjoint, the monomial $z^{val(\psi(S_1))} \star z^{val(\psi(S_2))} \star \cdots \star z^{val(\psi(S_\ell))}$ which is equal to $z^{val(\psi(W))}$ is present in $p(z)^\ell$. We prove the converse by induction on ℓ. For $\ell = 1$, the statement is vacuously true and for $\ell = 2$, the claim holds from the proof of Lemma 2. Assume that the claim holds for all the integers which are smaller than ℓ, i.e. if there exists a monomial $z^{val(\psi(W))}$ in $p(z)^{\ell-1}$ then W can be partitioned into $\ell - 1$ disjoint sets from $\mathcal{F}$.

If there exists a monomial $z^{val(\psi(W))}$ in $p(z)^\ell = p(z)^{\ell-1} \star p(z)$ then it is the product of two monomials, say $z^{val(\psi(W_1))}$ in $p(z)^{\ell-1}$ and $z^{val(\psi(W_2))}$ in $p(z)$ respectively with $W_1 \cap W_2 = \emptyset$. By induction hypothesis, W_1 is the disjoint union of $S_1, S_2, \ldots, S_{\ell-1}$ such that $S_i \in \mathcal{F}$ for all $i \in [\ell - 1]$. Also, W_2 is in $\mathcal{F}$ and since

$W_1 \cap W_2 = \emptyset$, $S_i \cap W_2 = \emptyset$ for each i. Therefore, W can be partitioned into sets $S_1, S_2, \ldots, S_{\ell-1}, W_2$ each of which belong to $\mathcal{F}$. $\square$

We now are in a position to prove the main theorem of this section.

Theorem 2. *Given a graph G on n vertices, there is an algorithm which finds its cd-chromatic number in $\mathcal{O}(2^n n^4 \log n)$ time.*

Proof. Fix an arbitrary ordering on $V(G)$. With $V(G)$ as the universe, we define the family $\mathcal{F}$ of its subsets as follows.

$$\mathcal{F} := \{X \subseteq V(G) \mid X \text{ is an independent set and } \exists\, y \in V(G) \text{ s.t. } X \subseteq N(y)\}$$

Note that every set in $\mathcal{F}$ is an independent set and there exists a vertex which dominates it. That is, $\mathcal{F}$ is the collection of the possible color classes in any cd-coloring of G. Let $p(z)$ be the characteristic polynomial of $\mathcal{F}$. By Lemma 3, if there exists a monomial $z^{val(\psi(V(G)))}$ in $p(z)^\ell$ then $V(G)$ can be partitioned into ℓ sets each belonging to $\mathcal{F}$. Hence the smallest integer ℓ for which there exists a monomial $z^{val(\psi(V(G)))}$ in $p(z)^\ell$ is $\chi_{cd}(G)$. By Corollary 1, $p(z)^\ell$ can be computed in $\mathcal{O}(2^n n^3 \log n \cdot l)$ time. As the cd-chromatic number of a graph is upper bounded by n, the claimed running time bound follows. $\square$

4 FPT Algorithms for cd-Chromatic Number

Determining whether a graph G has cd-chromatic number at most q is NP-hard on general graphs for $q \geq 4$. This implies that the cd-Coloring problem parameterized by the number of colors is para-NP-hard on general graphs. Thus this necessitates the search for special classes of graphs where cd-Coloring is FPT. In this section we give FPT algorithms for cd-Coloring on chordal graphs and graphs of girth at least 5.

We start by proving that cd-Coloring parameterized by the number of colors and treewidth of graph is FPT. Towards this, we will use Courcelle's powerful theorem which interlinks the fixed parameter tractability of a certain graph property with its expressibility as an MSO formula. We can write many graph theoretical properties as an MSO formula. Following are three examples which we will use in writing an MSO formula to check whether a graph has cd-chromatic number at most q.

- To check whether $V_1, V_2, \ldots, V_q$ is a partition of $V(G)$.

  $$\mathsf{Part}(V_1, V_2, \ldots, V_q) \equiv \forall u \in V(G)[\exists i \in [q][(u \in V_i) \wedge (\forall j \in [q][i \neq j \Rightarrow u \notin V_j)]]]$$

- To check whether a given vertex set V_i is an independent set or not.

  $$\mathsf{IndSet}(V_i) \equiv \forall u \in V_i[\forall v \in V_i[\neg adj(u, v)]]$$

- To check whether given vertex set V_i is dominated by some vertex or not.

  $$\mathsf{Dom}(V_i) \equiv \exists u \in V(G)[\forall v \in V_i[adj(u, v)]]$$

We use $\phi(G, q)$ to denote the MSO formula which states that G has cd-chromatic number at most q. We use the formulas defined above as macros in $\phi(G, q)$.

$$\begin{aligned}\phi(G,q) \equiv \exists V_1, V_2, \ldots, V_q \subseteq V(G)[&\mathsf{Part}(V_1, V_2, \ldots, V_q)\wedge \\ &\mathsf{IndSet}(V_1) \wedge \cdots \wedge \mathsf{IndSet}(V_q) \wedge \mathsf{Dom}(V_1) \wedge \cdots \wedge \mathsf{Dom}(V_q)]\end{aligned}$$

It is easy to see that the length of $\phi(G, q)$ is upper bounded by a linear function of q. By applying Theorem 1 we obtain the following result.

Theorem 3. CD-COLORING *parameterized by the number of colors and the treewidth of the input graph is* FPT.

4.1 Chordal Graphs

As the graph gets more structured, we expect many NP-hard problems to get *easier* in some sense on the restricted class of graphs having that structure. For example, CHROMATIC-COLORING is NP-hard on general graphs but it is polynomial time solvable on chordal graphs. However, CD-COLORING is NP-hard even on the chordal graphs [36] and we show that it is FPT when parameterized by the number of colors.

Theorem 4. CD-COLORING *parameterized by the number of colors is* FPT *on chordal graphs.*

Proof. For a chordal graph G, $\mathbf{tw}(G) = \omega(G) - 1$ where $\omega(G)$ is the size of a maximum clique in G [21]. Since, a cd-coloring is also a proper coloring, no two vertices in a clique can be in the same color class. Thus, if $\omega(G) \geq k$ then we can conclude that (G, k) is NO instance of CD-COLORING. Otherwise, $\omega(G) \leq k$ which implies that $\mathbf{tw}(G) \leq k$. This bound and Theorem 3 imply that CD-COLORING parameterized by the number of colors is FPT on chordal graphs. $\square$

4.2 Graphs with girth at least 5

In this section, we show that CD-COLORING on graphs of girth at least 5 is FPT with respect to the solution size as the parameter. By Observation 1, we can assume that the input graph G is connected. We can define cd-coloring of a connected graph as a proper coloring such that every color class is contained in the open neighbourhood of some vertex. In other words, we do not allow a vertex to dominate itself. One can verify that the two definitions of cd-coloring are identical on connected graphs. We now define the notion of a *total-dominating set* of a graph G. A set $S \subseteq V(G)$ is called a *total-dominating set* if $V(G) = \bigcup_{v \in S} N(v)$. That is, for every vertex $v \in V(G)$, there exists a vertex $u \in S$, $u \neq v$, such that $v \in N(u)$. Our interest in total-dominating set is because of its relation to cd-coloring in graphs that do not contain triangles, that is, graphs of girth at least 4. In particular, we show the following lemma.

Lemma 4. *If* $g(G) \geq 4$*, then the size of a minimum total dominating set is equal* $\chi_{cd}(G)$.

Proof. Let ϕ be a cd-coloring of G that uses $\chi_{cd}(G)$ colors and let $V_1, \ldots, V_q$ be the color classes in this coloring. Then, for every color class V_i, there is a vertex v_i such that $V_i \subseteq N(v_i)$. Let X denote the set of these vertices. Then, X has at most q vertices and by definition, it is a total dominating set of G. Hence, the size of a minimum total dominating set of a graph is at most the cd-chromatic number of the graph.

Suppose $X = \{v_1, v_2, \ldots, v_k\}$ is a minimum total dominating set of G. We construct a cd-coloring of G using at most k colors. We define the color classes in the following way. Let $V_1 = N(v_1)$ and for $i = 2, \ldots, k$, define $V_i = N(v_i) \setminus (V_1 \cup V_2 \cup \cdots \cup V_{i-1})$. Note that $V_1, \ldots, V_q$ forms a partition of $V(G)$. Since, $g(G) \geq 4$, it follows that each V_i is an independent set. Furthermore, since X is a total dominating set, for each $i \in [k]$, we have a vertex $v_i \in X$ such that $V_i \subseteq N(v_i)$. Hence, this gives a cd-coloring of G. Therefore, the cd-chromatic number of a graph is at most the cardinality of a minimum total dominating set. Now the lemma follows by combining the above two inequalities. $\square$

Lemma 4 shows that to prove that CD-COLORING is FPT on graphs of $g(G) \geq 4$ it suffices to show that finding a total dominating set of size at most k is FPT on these graphs. This leads to the TOTAL DOMINATING SET problem. Given a graph G and an integer k, the TOTAL DOMINATING SET problem asks whether there exists a total dominating set of size at most k. Observe that we can test whether G has a total dominating set of size at most k by enumerating all subsets S of $V(G)$ of size at most k and checking whether it forms a total-dominating set in polynomial time. This immediately gives an algorithm with running time $n^{\mathcal{O}(k)}$ for CD-COLORING on graphs with girth at least 4. It is not hard to modify the reduction given in [32] to show that TOTAL DOMINATING SET is $W[2]$ hard on bipartite graphs. Thus, Lemma 4 implies that even CD-COLORING is $W[2]$ hard on bipartite graphs. Hence, if we need to show that CD-COLORING is FPT, we must assume that the girth of the input graph is at least 4. In the rest of the section, we show that CD-COLORING is FPT on graphs with girth at least 5 by showing that TOTAL DOMINATING SET is FPT on those graphs. Before proceeding further, we note some simple properties of graphs with girth at least 5.

Observation 3 *For a graph G, if $g(G) \geq 5$ then for any v in $V(G)$, $N(v)$ is an independent set and for any u, v in $V(G)$, $|N(v) \cap N(u)| \leq 1$.*

Raman and Saurabh [32] defined a variation of SET COVER problem, namely, BOUNDED INTERSECTION SET COVER. An input to the problem consists of a universe $\mathcal{U}$, a collection $\mathcal{F}$ of subsets of $\mathcal{U}$ and a positive integer k with the property that for any two S_i, S_j in $\mathcal{F}$, $|S_i \cap S_j| \leq c$ for some constant c and the objective is to check whether there exists a sub-collection $\mathcal{F}_0$ of $\mathcal{F}$ of size at most k such that $\bigcup_{S \in \mathcal{F}_0} = \mathcal{U}$. In the same paper, the authors proved that the BOUNDED INTERSECTION SET COVER is FPT when parameterized by the solution size. TOTAL DOMINATING SET on (G, k) where G has girth at least 5 can be reduced to BOUNDED INTERSECTION SET COVER with $\mathcal{U} = V(G)$ and

$\mathcal{F} = \{N(v) | \; \forall v \in V(G)\}$. By Observation 3, we can fix the constant c to be 1. Hence we have the following lemma.

Lemma 5. *On graphs with girth at least* 5*,* Total Dominating Set *is* FPT *when parameterized by the solution size.*

We now prove that the problem has a polynomial kernel and use it to design another FPT algorithm.

Lemma 6. Total Dominating Set *admits a kernel of* $\mathcal{O}(k^3)$ *vertices on the class of graphs with girth at least 5.*

Proof. We start the proof with the following claim which says that every high degree vertex should be included in every total dominating set of size at most k.

Claim. In a graph G with $g(G) \geq 5$, if there is a vertex u with degree at least $k+1$, then any total dominating set of size at most k contains u.

Proof. Suppose there exists a total dominating set X of G of size at most k which does not contain u. Since $N(u)$ (having size at least $k+1$) is dominated by X and no vertex can dominate itself, by the Pigeon Hole Principle, there exists a vertex, say w, in X which is adjacent to at least two vertices, say, v_1, v_2 in $N(u)$. This implies that w, v_1, v_2, u forms a cycle of length 4, contradicting the fact that girth of G is at least 5. ◻

Suppose G has a total dominating set of size at most k. Construct a tri-partition of $V(G)$ as follows:

$$\begin{aligned}
H &= \{u \in V(G) \mid |N(u)| \geq k+1\};\\
J &= \{v \in V(G) \mid v \notin H, \; \exists u \in H \text{ such that } (u,v) \in E(G)\};\\
R &= V(G) \setminus (H \cup J)
\end{aligned}$$

By the above claim, H is contained in every total dominating set of size at most k. Hence, the size of H is upper bounded by k. Note that there is no edge between a vertex in H and a vertex in R. Thus, R has to be dominated by at most k vertices from $J \cup R$. However, the degree of vertices in $J \cup R$ is at most k and hence $|R| \leq \mathcal{O}(k^2)$ and $|J \cap N(R)|$ is upper bounded by $\mathcal{O}(k^3)$. We will now bound the size of $J^\star = J \setminus N(R)$. For that, we first apply the following reduction rule on the vertices in $J^\star$.

Reduction Rule 1 *For $u, v \in J^\star$, if $N(u) \cap H \subseteq N(v) \cap H$ then delete u.*

The correctness of this reduction follows from the observation that all the vertices in J have been dominated by the vertices in H. The only reason any vertex in $J^\star$ is part of a total dominating set is because that vertex is used to dominate some vertex in H. If this is the case then the vertex u in the solution can be replaced by the vertex v. In the reverse direction, if X is a total dominating set of $G - \{u\}$ and $|X| \leq k$, then $H \subseteq X$. Hence u is dominated by $x \in X \cap H$ in G too. That is, X is a total dominating set of G.

All that remains is to bound the size of $J^\star$. We partition $J^\star$ into two sets namely J_1 and J_2. The set J_1 is the set of vertices which are adjacent to exactly one vertex in H whereas each vertex in J_2 is adjacent to at least two vertices in H. After exhaustive application of Reduction Rule 1, no two vertices in J_1 can be adjacent to one vertex in H and hence $|J_1| \leq |H| \leq k$. Any vertex in J_2 is adjacent to at least two vertices in H. For every vertex u in J_2, we assign a pair of vertices in H to which u is adjacent. By Observation 3, no two vertices in J_2 can be assigned to the same pair and hence the size of J_2 is upper bounded by $\binom{k}{2} \leq k^2$. Combining all the bounds, we get a kernel with $\mathcal{O}(k^3)$ number of vertices. □

Combining Lemmas 4 and 6 we obtain the following theorem.

Theorem 5. *On graphs with girth at least* 5*,* CD-COLORING *admits an algorithm running in* $\mathcal{O}(2^{\mathcal{O}(q^3)} q^{12} \log q^3)$ *time and an* $\mathcal{O}(q^3)$ *sized vertex kernel, where* q *is number of colors.*

5 Complexity of CD-Partization

In this section, we study the complexity of CD-PARTIZATION. As recognizing graphs with cd-chromatic number at most q is NP-hard on general graphs for $q \geq 4$, the deletion problem is also NP-hard on general graphs for such values of q. For $q = 1$, the problem is trivial as $\chi_{cd}(G) = 1$ if and only if G is the graph on one vertex. In this section, we show NP-hardness for $q \in \{2, 3\}$. We remark that $\mathcal{G} = \{G \mid \chi_{cd}(G) \leq q\}$ is not a hereditary graph class and so the generic result of Lewis and Yannakakis [27] does not imply the claimed NP-hardness.

5.1 Para-NP-hardness in General Graphs

Consider the following problem.

PARTIZATION
Input: Graph G, integers k and q
Question: Does there exist $S \subseteq V(G)$, $|S| \leq k$, such that $\chi(G - S) \leq q$?

Once again if q is fixed, we refer to the problem as q-PARTIZATION. Observe that the classical NP-complete problems VERTEX COVER [16] and ODD CYCLE TRANSVERSAL [16] are 1-PARTIZATION and 2-PARTIZATION, respectively. Now, we proceed to show the claimed hardness.

Theorem 6. q-CD-PARTIZATION *is* NP*-complete for* $q \in \{2, 3\}$*.*

Proof. The problem is in NP as determining if the cd-chromatic number of a graph is at most $q \in \{1, 2, 3\}$ is polynomial-time solvable. Given an instance (G, k) of q-PARTIZATION where $q \in \{1, 2\}$, we construct the instance (G', k) of $(q + 1)$-CD-PARTIZATION as follows: G' is obtained from G by adding a new vertex v adjacent to every vertex in $V(G)$ and adding $k + q + 2$ new vertices

$v_1, \cdots, v_{k+q+2}$ adjacent to v. We claim that G has a set of k vertices whose deletion results in a q-colorable graph if and only if G' has a set of k vertices whose deletion results in a $(q+1)$-cd-colorable graph.

Consider a set S of k vertices such that $\chi(G - S) \leq q$. Then, $G' - S$ is $(q+1)$-cd-colorable as a new color can be assigned to v and any of the q colors of $G - S$ can be assigned to $v_1, \cdots, v_{k+q+2}$. The color class containing v is a singleton set. This class is dominated by all vertices in $G' - (S \setminus \{v\})$. Further, v dominates each of the other q color classes as v is a universal vertex in G'.

Conversely, let $S' \subseteq V(G')$ be a minimal set of at most k vertices such that $\chi_{cd}(G' - S') \leq q + 1$. Now, if $v \in S'$, then vertices $v_1, \cdots, v_{k+q+2}$ are isolated in $G - \{v\}$ implying that either $|\{v_1, \cdots, v_{k+q+2}\} \cap S'| \geq k+1$ or $\chi_{cd}(G' - S') > q + 1$. So, we can assume that $v \notin S'$. Further, as S' is minimal, it follows that $\{v_1, \cdots, v_{k+q+2}\} \cap S' = \emptyset$. Also, as v is a universal vertex in G', we have that $\chi(G - (S' \setminus \{v\})) \leq q$. So, S' is a subset of $V(G)$ of size at most k such that $G - S'$ is q-colorable. □

5.2 NP-hardness and Fixed-Parameter Tractability in Split Graphs

A graph is a *split graph* if its vertex set can be partitioned into a clique and an independent set. As split graphs are perfect (clique number is equal to the chromatic number for every induced subgraph), we have the following observation.

Observation 7 *A split graph G is r-colorable if and only if $\omega(G) \leq r$.*

The following result is known for the corresponding deletion problem.

Theorem 8 ([8,38]). Partization on Split Graphs *is* NP*-complete.*

This hardness was shown by a reduction from Set Cover [16]. We modify this reduction to show that cd-Partization is NP-complete on split graphs. The problem is in NP as the cd-chromatic coloring of a split graph can be verified in polynomial time due to the following result.

Theorem 9 ([35]). *If G is a connected split graph G, then $\omega(G) = \chi_{cd}(G)$. Furthermore, there is an $\mathcal{O}(|V(G)|^2)$ time algorithm that returns a minimum cd-coloring of G.*

Theorem 10. cd-Partization *on split graphs is* NP*-hard.*

Proof. Consider a Set Cover instance $(U, \mathcal{F}, k)$ where $U = \{x_1, \cdots, x_n\}$ is a finite set and $\mathcal{F}$ is a family $\{S_1, \cdots, S_m\}$ of subsets of U. The problem is to determine if there is a collection of at most k sets in $\mathcal{F}$ such that each element of U is in at least one set of the collection. The corresponding instance of *cd*-Partization is $(G, k' = m - k, q = k + 1)$ where G is a split graph on the vertex set $C \cup I \cup \{w_0, w_1, \cdots, w_{k+k'+2}\}$ where $C = \{u_i \mid S_i \in \mathcal{F}\}$ and $I = \{v_i \mid x_i \in U\}$. Also, $(v_i, u_j) \in E(G)$ if and only if $x_i \notin S_j$ and w_0 is adjacent to every vertex in $C \cup I \cup \{w_1, \cdots, w_{k+k'+2}\}$. Further, $I \cup \{w_1, \cdots, w_{k+k'+2}\}$ and

C induce an independent set and a clique, respectively, in G. We claim that a set $\mathcal{F}' \subseteq \mathcal{F}$ of size k is a set cover if and only if $G - S'$ is q-cd-colorable where $S' = \{u_i \in C \mid S_i \in \mathcal{F} \setminus \mathcal{F}'\}$ and $|S'| = k'$.

Consider a set cover $\mathcal{F}' \subset \mathcal{F}$ of size k. If there is a clique Q (without loss of generality assume $w_0 \in Q$) of size $k+2$ in $G - S'$, then Q must contain an element $v_i \in I$ that is adjacent to k vertices in $C \setminus S'$. However, since $\mathcal{F}'$ is a set cover, v_i is non-adjacent to at least one u_j in $C \setminus S'$ leading to a contradiction. Thus, S' has a non-empty intersection with every $(k+2)$-clique in G. As G is a split graph, it is $(k+1)$-colorable due to Observation 7. Further, $G - S'$ is $(k+1)$-cd-colorable as the color class containing $\{w_0\}$ is a singleton set (since it is an universal vertex) which is dominated by itself and the other color classes are dominated by w_0.

Conversely, consider a minimal subset S' of k' vertices such that $G - S'$ is $(k+1)$-cd-colorable. Now, if $w_0 \in S'$, then vertices $w_1, \cdots, u_{k+k'+2}$ are isolated in $G - \{w_0\}$ implying that either $|\{w_1, \cdots, w_{k+k'+2}\} \cap S'| \geq k'+1$ or $\chi_{cd}(G - S') > k+1$. So, we can assume that $w_0 \notin S'$. Further, as S' is minimal, it follows that $\{w_1, \cdots, w_{k+k'+2}\} \cap S' = \emptyset$. Now, all vertices in S' must belong to C. If there exists $v_i \in S' \cap I$, there is a clique of size $k+2$ in $G - S'$ as C is a clique. Also, no vertex in I is adjacent to all nodes in $C \setminus S'$ as if there is such a vertex v_i then there is a $(k+2)$-clique in $G - S'$. Thus, every vertex in I is nonadjacent to at least one element in $C \setminus S'$ implying that $\{s_i \in \mathcal{F} \mid u_i \in C \setminus S'\}$ is a set cover of $(U, \mathcal{F})$ of size at most k. □

As Set Cover parameterized by solution size is W[2]-hard [11], we have the following result.

Corollary 2. cd-Partization *on split graphs parameterized by q is* W[2]*-hard.*

Now, we show that the problem is FPT with respect to q and k.

Theorem 11. cd-Partization *on split graphs is* FPT *with respect to parameters q and k. Furthermore, the problem does not admit a polynomial kernel unless* NP $\subseteq$ coNP*/poly.*

Proof. Compute a maximum clique Q of G in polynomial time. If $|Q| \leq q$, then the input instance is an YES instance as $\chi_{cd}(G) \leq q$ from Theorem 9. Otherwise, choose an arbitrary subset of size $q+1$ from Q. Since any solution contains at least one of the $q+1$ vertices, a straightforward branching algorithm runs in $\mathcal{O}^*((q+1)^k)$ time. Now, we move on to the kernelization hardness. Set Cover is known not to admit a polynomial kernel when parameterized by the solution size k' and family size m as combined parameters unless NP $\subseteq$ coNP/poly [11]. The reduction in Theorem 10 produces instances of cd-Partization where solution size k is $m - k'$ and q is $k'+1$ implying that $q+k$ is $m+1$. Therefore, an $(q+k)^{\mathcal{O}(1)}$ kernel for cd-Partization implies an $m^{\mathcal{O}(1)}$ kernel for Set Cover which is unlikely. □

6 Deletion to 3-cd-Colorable Graphs

In a graph G, an edge $e = (u, v)$ is said to be a dominating edge if $N(u) \cup N(v) = V(G)$. Let $\overline{N[v]}$ denote the set $V(G) \setminus N[v]$. The following characterization of 3-cd-colorable graphs is known from [35].

Theorem 12 ([35]). *A connected graph G satisfies $\chi_{cd}(G) \leq 3$ if and only if G is one of the following types.*
(Type 0) G is a graph on at most 3 vertices.
(Type 1) G is a bipartite graph with a dominating edge.
(Type 2) G has a vertex v such that $G - v$ is a bipartite graph with a dominating edge.
(Type 3) G has an ordered pair (x, y) of adjacent vertices such that,

- $V(G) = \{x, y\} \uplus X \uplus Y$,
- $G[X \cup \{y\}]$ *is a bipartite graph with at least one edge,*
- $Y \cup \{x\}$ *is an independent set,* $Y \cup \{x\} \subseteq N(y)$ *and* $X \cup \{y\} \subseteq N(x)$.

(Type 4) G has an ordered set (x, y, z) of vertices inducing a triangle such that,

- $V(G) = \{x, y, z\} \uplus X \uplus Y \uplus Z$,
- $X \subseteq N(x)$, $Y \subseteq N(y)$ *and* $Z \subseteq N(z)$,
- $X \cup \{y\}$, $Y \cup \{z\}$ *and* $Z \cup \{x\}$ *are independent sets.*

(Type 5) G has an ordered triple (x, y, z) of vertices such that,

- $V(G) = \{x, y\} \uplus X \uplus Y \uplus Z$,
- $z \in X \cup Y$, $(x, y) \notin E(G)$ *and* $(x, z), (y, z) \in E(G)$,
- $X \subseteq N(x)$, $Y \subseteq N(y)$ *and* $Z \subseteq N(z)$,
- X, Y, $Z \cup \{x\}$ *and* $Z \cup \{y\}$ *are independent sets.*

We refer to the ordered sets in Types 3, 4 and 5 as dominators. In [35], they are called as d-pair, cd-triangle and NB-triplet respectively. Now, we proceed to solve 3-cd-Partization. Let G be the input graph on n vertices, m edges and k be a positive integer. Consider a set $S \subseteq V(G)$ such that $H = G - S$ is 3-cd-colorable. Then, H is of one of the types listed in Theorem 12. Before we proceed to describe algorithms for each of these types, we list the following well-known results on Vertex Cover and Odd Cycle Transversal that we use in our algorithms.

Theorem 13 ([7]). *Given a graph G and a positive integer k, there is an algorithm running in $\mathcal{O}^*(1.2738^k)$ time that determines if G has a vertex cover of size at most k or not.*

Theorem 14 ([28]). *Given a graph G and a positive integer k, there is an algorithm running in $\mathcal{O}^*(2.3146^k)$ time that determines if G has an odd cycle transversal of size at most k or not.*

$\mathcal{O}^*$ supresses functions which are polynomial in size of input

As we would subsequently show, our algorithms reduce the problem of finding an optimum deletion set into finding appropriate vertex covers and constrained odd cycle transversals. The current best parameterized algorithm for finding a vertex cover can straightaway be used as a subroutine in our algorithm while the one for finding an odd cycle transversal requires the following results. Consider a graph G and let v be a vertex in G. Define the graph G' to be the graph obtained from G by deleting v and adding a new vertex v_{ij} for each pair v_i, v_j of neighbors of v; further v_{ij} is adjacent to v_i and v_j.

Lemma 7. *G has a minimal odd cycle transversal of size at most k that excludes vertex v if and only if G' has a minimal odd cycle transversal of size at most k.*

Proof. Consider an odd cycle transversal O of G excluding v and let (X, Y) be a bipartition of $G - O$. Without loss of generality, let $v \in X$. Then, every vertex in $N(v)$ is either in O or in Y. Thus, $X' = (X \setminus \{v\}) \cup (V(G') \setminus V(G))$ is an independent set in G'. Consequently, (X', Y) is a bipartition of $G' - O$ implying that O is an odd cycle transversal of G'. Conversely, any odd cycle transversal O' of G' can be modified to one that excludes each vertex in $\{v_{ij} \in V(G') \mid v_i, v_j \in N(v)\}$ without increasing the size since any induced odd cycle through v_{ij} is also an induced odd cycle through v_i and v_j. Then, it follows that O' is an odd cycle transversal of G that excludes v. $\square$

Let $P, Q \subseteq V(G)$ be two disjoint sets. Let G'' be the graph constructed from G by adding an independent set I_P of $k+1$ new vertices each of which is adjacent to every vertex in P and an independent set I_Q of $k+1$ new vertices each of which is adjacent to every vertex in Q. Further, every vertex in I_P is adjacent to every vertex in I_Q.

Lemma 8. *G has a minimal odd cycle transversal O of size at most k such that $G - O$ has a bipartition (X, Y) with $P \subseteq X$ and $Q \subseteq Y$ if and only if G'' has a minimal odd cycle transversal of size at most k.*

Proof. Suppose $G - O$ has a bipartition (X, Y) such that $P \subseteq X$ and $Q \subseteq Y$. Then, $G'' - O$ has a bipartition (X', Y') where $X' = X \cup I_Q$ and $Y' = Y \cup I_P$. Thus, O is an odd cycle transversal of G'' too. Conversely, consider a minimal odd cycle transversal O' of size k of G''. Clearly, O' excludes at least one vertex a from I_P and at least one vertex b from I_Q. Consider an arbitrary bipartition (A, B) of $G'' - O'$ and let $a \in A$ and $b \in B$. Then, as O' is minimal $I_P \subseteq A$ and $I_Q \subseteq B$. That is, $O' \cap (I_P \cup I_Q) = \emptyset$. Further, as any two vertices $p \in I_P$ and $q \in I_Q$ are adjacent, $I_P \cap V(G'' - O') \subseteq A$ and $I_Q \cap V(G'' - O') \subseteq B$. Thus, $P \subseteq B$ and $Q \subseteq A$. $\square$

6.1 Deletion to Types 0, 1 and 2

It is trivial to check if G has a solution whose deletion results in a graph H with at most 3 vertices. So, deletion to Type 0 is easy. Now, suppose H is of Type

1. Then, we need to identify an edge of G that is a dominating edge for H. We describe an algorithm based on this observation.

Algorithm 6.1: Deletion-to-Type1(G, k)

Input : A graph G and a positive integer k.
Output: $S \subseteq V(G)$, $|S| \leq k$ such that $G - S$ is of Type 1 (if one exists).

2 **for** *each edge* (x, y) *in* G **do**
3   Let $X' = N(x) \cap \overline{N[y]}$ and $Y' = N(y) \cap \overline{N[x]}$.
4   Let S' be $V(G) \setminus (X' \cup Y')$ and decrease k by $|S'|$.
6   **for** *each* k_1 *and* k_2 *such that* $k_1 + k_2 \leq k$ **do**
8   Compute a vertex cover S_1 of $G[X']$ with $|S_1| \leq k_1$ (if one exists).
9   /* $(X' \setminus S_1) \cup \{y\}$ is an independent set */
11   Compute a vertex cover S_2 of $G[Y']$ with $|S_2| \leq k_2$ (if one exists).
12   /* $(Y' \setminus S_2) \cup \{x\}$ is an independent set */
13   **if** S_1 *and* S_2 *are non-empty sets* **then**
14   **return** $S' \cup S_1 \cup S_2$

Lemma 9. *Algorithm 6.1 runs in* $\mathcal{O}^*(1.2738^k)$ *time.*

Proof. The outer loop (step 1) is executed at most m times (once for each edge) and the inner loop (step 2) is executed at most k^2 times. Let (x, y) be an edge in G. We need to extend $\{x\}$ and $\{y\}$ into independent sets Y and X respectively, such that X is dominated by x and Y is dominated by y. Clearly, neighbors of x and non-neighbors of y cannot be in Y. Similarly, neighbors of y and non-neighbors of x cannot be in X. No common neighbor of x and y can be in either X or Y. Thus, the candidates for X and Y are $X' = N(x) \cap \overline{N[y]}$ and $Y' = N(y) \cap \overline{N[x]}$ respectively. All vertices in $V(G) \setminus (X' \cup Y')$ are in any solution. Let $k' = k - |V(G) \setminus (X' \cup Y')|$. Then, G has a 3-cd-partization solution S of size at most k such that $G - S$ is of Type 1 with (u, v) as a dominating edge if and only if there exists integers k_1, k_2 with $k_1 + k_2 \leq k'$ such that $G[X']$ has a vertex cover of size at most k_1 and $G[Y']$ has a vertex cover of size at most k_2. Now, Steps 3 and 4 take $\mathcal{O}^*(1.2738^k)$ time from Theorem 13. Thus, the overall running time is $\mathcal{O}^*(1.2738^k)$. ☐

Suppose H is of Type 2. Then, for each vertex v of G, we simply run Algorithm 6.1 on $G - \{v\}$ with parameter k.

Algorithm 6.2: Deletion-to-Type2(G, k)

Input : A graph G and a positive integer k.
Output: $S \subseteq V(G)$, $|S| \leq k$ such that $G - S$ is of Type 2 (if one exists).

2 **for** *each vertex* x *in* G **do**
3   Deletion-to-Type1$(G - \{x\}, k)$.

Lemma 10. *Algorithm 6.2 runs in* $\mathcal{O}^*(1.2738^k)$ *time.*

Proof. As Algorithm 6.2 calls Algorithm 6.1 at most n times, its running time is $\mathcal{O}^*(1.2738^k)$. ☐

6.2 Deletion to Type 3

Suppose H is of Type 3 with dominator (x, y). Then, the following holds.

Observation 15 ([35]) *$\overline{N_H[x]}$ is an independent set and $\overline{N_H[x]} \subseteq N_H(y)$. Further, $N_H(x)$ induces a bipartite graph with at least one edge.*

This observation leads to the following algorithm.

Algorithm 6.3: Deletion-to-Type3(G, k)

Input : A graph G and a positive integer k.
Output: $S \subseteq V(G)$, $|S| \le k$ such that $G - S$ is of Type 3 (if one exists).

2 **for** *each ordered pair (x, y) of adjacent vertices in G* **do**
3   Let $Y' = N(y) \cap \overline{N[x]}$ and $X' = N(x)$.
4   Let S' be $V(G) \setminus (X' \cup Y')$ and decrease k by $|S'|$.
6   **for** *each k_1 and k_2 such that $k_1 + k_2 \le k$* **do**
8   Compute a vertex cover S_1 of $G[Y']$ with $|S_1| \le k_1$ (if one exists).
9   /* $(Y' \setminus S_1) \cup \{x\}$ is an independent set */
11   Compute a minimal odd cycle transversal S_2 of at most k_2 vertices (if one exists) in $G[X']$ such that $y \notin S_2$.
12   **if** *S_1 and S_2 are non-empty sets* **then**
13   **return** $S' \cup S_1 \cup S_2$

Lemma 11. *Algorithm 6.3 runs in $\mathcal{O}^*(2.3146^k)$ time.*

Proof. The outer loop (step 1) is executed at most $2m$ times (as there are two ordered pairs for each edge) and the inner loop (step 2) is executed at most k^2 times. Consider an edge (x, y) in G. If (x, y) is a dominator in H, then we need to extend $\{x\}$ into an independent set Y that is dominated by y and extend $\{y\}$ into an induced bipartite graph B (with at least one edge) such that $V(B)$ is dominated by x. Observe that Y contains only neighbors of y and $V(B)$ contains only neighbors of x. Further, a neighbor of y that is not adjacent to x cannot be in $V(B)$ and a neighbor of y that is adjacent to x cannot be in Y. Thus, the candidates for $V(B)$ and Y are $X' = N(x)$ and $Y' = N(y) \cap \overline{N[x]}$ respectively. All vertices in $V(G) \setminus (X' \cup Y')$ are in any solution. Let $k' = k - |V(G) \setminus (X' \cup Y')|$. Now, G has a 3-cd-partization solution S of size at most k such that $G - S$ is of Type 3 with (x, y) as a dominator if and only if there exists integers k_1 and k_2 with $k_1 + k_2 \le k'$ such that $G[Y']$ has a vertex cover of size at most k_1 and $G[X']$ has an odd cycle transversal of size k_2 not containing y such that the resultant bipartite graph is non-edgeless. Clearly step 3 takes $\mathcal{O}^*(1.2738^k)$ time. For step 4, we need to find a minimal odd cycle transversal that excludes vertex y. We construct a graph G' obtained from $G[X']$ by deleting y and adding a new vertex

y_{ij} for each pair y_i, y_j of neighbors of y; further y_{ij} is adjacent to y_i and y_j. From Lemma 7, we have that $G[X']$ has a minimal odd cycle transversal of size at most k_2 not containing y if and only if G' has a minimal odd cycle transversal of size at most k_2. Now, by using Theorem 14, it follows that step 4 takes $\mathcal{O}^*(2.3146^k)$ time and this gives us the claimed running time of the algorithm. □

6.3 Deletion to Type 4

Suppose H is of Type 4 and has (x, y, z) as a dominator. Then, we have the following observation.

Observation 16 ([35]) *$N_H(x) \cap N_H(y) \cap N_H(z) = \emptyset$ and $\overline{N_H[x]} \cap \overline{N_H[y]} \cap \overline{N_H[z]} = \emptyset$. Further, $X = N_H(x) \cap \overline{N_H[y]}$, $Y = N_H(y) \cap \overline{N_H[z]}$ and $Z = N_H(z) \cap \overline{N_H[x]}$.*

Now, we have the following algorithm.

Algorithm 6.4: Deletion-to-Type4(G, k)

Input : A graph G and a positive integer k
Output: $S \subseteq V(G)$, $|S| \leq k$ such that $G - S$ is of Type 4 (if one exists)

2 **for** *each ordered triple (x, y, z) of pairwise adjacent vertices of G* **do**
3   Let $X' = N(x) \cap \overline{N[y]}$, $Y' = N(y) \cap \overline{N[z]}$ and $Z' = N(z) \cap \overline{N[x]}$.
4   Let S' be $V(G) \setminus (X' \cup Y' \cup Z')$ and decrease k by $|S'|$.
6   **for** *each k_1, k_2 and k_3 such that $k_1 + k_2 + k_3 \leq k$* **do**
8   Compute a vertex cover S_1 of $G[X']$ with $|S_1| \leq k_1$ (if one exists).
9   /* $(X' \setminus S_1) \cup \{y\}$ is an independent set */
11   Compute a vertex cover S_2 of $G[Y']$ with $|S_2| \leq k_2$ (if one exists).
12   /* $(Y' \setminus S_2) \cup \{z\}$ is an independent set */
14   Compute a vertex cover S_3 of $G[Z']$ with $|S_3| \leq k_3$ (if one exists).
15   /* $(Z' \setminus S_3) \cup \{x\}$ is an independent set */
16   **if** *S_1, S_2 and S_3 are non-empty sets* **then**
17   **return** $S' \cup S_1 \cup S_2 \cup S_3$

Lemma 12. *Algorithm 6.4 runs in $\mathcal{O}^*(1.2738^k)$ time.*

Proof. The outer loop (step 1) is executed at most n^3 times and the inner loop (step 2) is executed at most k^3 times. Consider a triangle $\{x, y, z\}$ in G. If (x, y, z) is a dominator in H, then we need to extend $\{x\}$, $\{y\}$, $\{z\}$ into independent sets Y, Z, X dominated by y, z and x respectively. Thus, the candidates for X, Y and Z are sets $X' = N(x) \cap \overline{N[y]}$, $Y' = N(y) \cap \overline{N[z]}$ and $Z' = N(z) \cap \overline{N[x]}$. All vertices in $S' = V(G) \setminus (X' \cup Y' \cup Z')$ are in any solution. Let $k' = k - |V(G) \setminus S'|$. Then, G has a 3-cd-partization solution S of size at most k such that $G - S$ is of Type 4 with (x, y, z) as a dominator if and only if there exists integers k_1, k_2 and k_3 with $k_1 + k_2 + k_3 \leq k'$ such that $G[Y']$ has a vertex cover of size at most k_1, $G[Z']$ has a vertex cover of size at most k_2 and $G[Z']$ has a vertex cover of size at most k_3. Steps 3, 4 and 5 take $\mathcal{O}^*(1.2738^k)$ time from Theorem 13 and the overall running time is $\mathcal{O}^*(1.2738^k)$. □

6.4 Deletion to Type 5

Suppose H is of Type 5 and has (x, y, z) as a dominator. Then, we have the following observation.

Observation 17 ([35]) *$\overline{N_H[x]} \cap \overline{N_H[y]}$ is an independent set. Further, $z \in N_H(x) \cup N_H(y)$ and $\overline{N_H[x]} \cap \overline{N_H[y]} \subseteq N_H(z)$. Moreover, in $G - Z$, $N[x] \cup N[y] = V(G - Z)$, $N(x) \setminus N(y) \subseteq X$ and $N(y) \setminus N(x) \subseteq Y$.*

Now, we have the following algorithm.

Algorithm 6.5: Deletion-to-Type5(G, k)

Input : A graph G and a positive integer k
Output: $S \subseteq V(G)$, $|S| \leq k$ such that $G - S$ is of Type 5 (if one exists)

```
2  for each ordered triple (x, y, z) of vertices of G such that (x, y) ∉ E(G)
     and (x, z), (y, z) ∈ E(G) do
3      Let Z′ be the set N(z) ∩ (N̄[y] ∩ N̄[x]).
4      Let S′ be (N(y) ∩ N(z)) \ N(x).
5      Let B′ be the set {z} ∪ (((N(x) ∪ N(y)) \ S′).
6      Let S″ be V(G) \ (Z′ ∪ B′) and decrease k by |S″|.
8      for each k1 and k2 such that k1 + k2 ≤ k do
10         Compute a vertex cover S1 of G[Z′] with |S1| ≤ k1 (if one exists).
11         /* (Z′ \ S1) ∪ {x, y} is an independent set */
13         Compute a minimal odd cycle transversal S2 of G[B′] with
             |S2| ≤ k2 not containing z (if one exists) such that the resultant
             bipartite graph has a bipartition (X, Y) such that X ⊆ N(x),
             Y ⊆ N(y) and z ∈ Y.
14         if S1 and S2 are non-empty sets then
15             return S″ ∪ S1 ∪ S2
```

Lemma 13. *Algorithm 6.5 runs in $\mathcal{O}^*(2.3146^k)$ time.*

Proof. Consider an ordered triple (x, y, z) of vertices in G. If (x, y, z) is a dominator in H, then we need to extend $\{x, y\}$ into an independent set Z that is dominated by z and extend $\{z\}$ into a bipartite graph B with bipartition (X, Y) such that X is dominated by x and Y is dominated by y. Thus, the candidates for Z and $V(B)$ are $Z' = N(z) \cap (\overline{N[y]} \cap \overline{N[x]})$ and $B' = \{z\} \cup ((N(x) \cup N(y)) \setminus (N(y) \cap N(z)) \setminus N(x))$ respectively. All vertices in $V(G) \setminus (Z' \cup B')$ are in any solution. Let $k' = k - |V(G) \setminus (Z' \cup B')|$. Then, G has a 3-cd-partization solution S of size at most k such that $H = G - S$ is of Type 5 with (x, y, z) as a dominator if and only if there exists integers k_1 and k_2 with $k_1 + k_2 \leq k'$ such that $G[Z']$ has a vertex cover of size at most k_1 and $G[B']$ has an odd cycle transversal of size k_2 not containing z such that the resultant bipartite graph has a bipartition (X, Y) such that $X \subseteq N(x)$, $Y \subseteq N(y)$ and $z \in Y$. Step 3 takes $\mathcal{O}^*(1.2738^k)$ time. For step 4, we use Lemmas 7 and 8. Let G' be the graph obtained from $G[B']$ by deleting z and adding a new vertex z_{ij} for each pair z_i, z_j of neighbors of z, adjacent to z_i and z_j. Now, a minimal odd cycle transversal of G' corresponds to a minimal odd cycle transversal of $G[B']$ not containing z. However, we also need the additional constraint that such an odd cycle transversal results in a bipartite graph B which has a bipartition (X, Y) such that $X \subseteq N(x)$ and $Y \subseteq N(y)$. The possible vertices in B are from the set $\{z\} \cup (((N(x) \cup N(y)) \setminus (N(y) \cap N(z)) \setminus N(x))$. The following observations on vertices from this set are easy to verify.

- $N_x = N(x) \setminus (N(y) \cup N(z))$ cannot be dominated by y and $N_y = N(y) \setminus (N(x) \cup N(z))$ cannot be dominated by x.
- $N_{zx} = (N(x) \cap N(z)) \setminus N(y)$ and $N_{xyz} = N(x) \cap N(y) \cap N(z)$ cannot be in a part of the bipartition that contains z.

It follows that we need an odd cycle transversal (of size at most k_2) of $G[B']$ after deleting which the resultant bipartite graph has a 2-coloring in which any vertex from $P = \{z\} \cup N_y$ receives color 1 and any vertex from $Q = N_x \cup N_{zx} \cup N_{xyz}$ receives color 2. This is achieved by constructing graph G'' from G' by adding an independent set I_P of $k_2 + 1$ new vertices each of which is adjacent to every vertex in P and an independent set I_Q of $k_2 + 1$ new vertices each of which is adjacent to every vertex in Q. Further, every vertex in I_P is adjacent to every vertex in I_Q. Now, $G[B']$ has a minimal odd cycle transversal of size at most k_2 not containing z such that the resultant bipartite graph has a bipartition (X, Y) such that $X \subseteq N(x)$, $Y \subseteq N(y)$ and $z \in Y$ if and only if G'' has a minimal odd cycle transversal of size at most k_2. Now, using Theorem 14, it follows that step 4 takes $\mathcal{O}^*(2.3146^k)$ time and the overall running time is dominated (upto polynomial factors) by this computation. □

From Lemmata 9,10,11,12 and 13, we have the following result.

Theorem 18. *Given a graph G and an integer k, there is an algorithm that determines if there is a set S of size k whose deletion results in a graph H with $\chi_{cd}(H) \leq 3$ in $\mathcal{O}^*(2.3146^k)$ time.*

7 Concluding Remarks

In this work, we described exact and FPT algorithms for problems associated with cd-coloring. We also explored the complexity of finding the cd-chromatic number in graphs of girth at least 5 and chordal graphs. On the former graph class, we described a polynomial kernel. The kernelization complexity on other graph classes and whether the problem is FPT parameterized by only treewidth are natural directions for further study.

References

1. Alber, J., Bodlaender, H.L., Fernau, H., Kloks, T., Niedermeier, R.: Fixed parameter algorithms for DOMINATING SET and related problems on planar graphs. Algorithmica 33(4), 461–493 (2002)

2. Alon, N., Gutner, S.: Linear time algorithms for finding a dominating set of fixed size in degenerated graphs. Algorithmica 54(4), 544–556 (2009)
3. Björklund, A., Husfeldt, T., Koivisto, M.: Set partitioning via inclusion-exclusion. SIAM J. Comput. 39(2), 546–563 (2009)
4. Blum, A., Karger, D.R.: An õ(nˆ{3/14})-coloring algorithm for 3-colorable graphs. Inf. Process. Lett. 61(1), 49–53 (1997)
5. Cai, L.: Parameterized complexity of vertex colouring. Discrete Applied Mathematics 127(3), 415–429 (2003)
6. Chellali, M., Maffray, F.: Dominator colorings in some classes of graphs. Graphs and Combinatorics 28(1), 97–107 (2012)
7. Chen, J., Kanj, I., Xia, G.: Improved Upper Bounds for Vertex Cover. Theoretical Computer Science 411(40-42), 3736–3756 (2010)
8. Corneil, D.G., Fonlupt, J.: The Complexity of Generalized Clique Covering. Discrete Applied Mathematics 22(2), 109–118 (1989)
9. Courcelle, B.: The monadic second-order logic of graphs. I. recognizable sets of finite graphs. Inf. Comput. 85(1), 12–75 (1990)
10. Courcelle, B.: The monadic second-order logic of graphs III: tree-decompositions, minor and complexity issues. ITA 26, 257–286 (1992)
11. Cygan, M., Fomin, F.V., Kowalik, L., Lokshtanov, D., Marx, D., Pilipczuk, M., Pilipczuk, M., Saurabh, S.: Parameterized Algorithms. Springer-Verlag (2015)
12. Diestel, R.: Graph theory {graduate texts in mathematics; 173}. Springer-Verlag Berlin and Heidelberg GmbH & amp (2000)
13. Downey, R.G., Fellows, M.R.: Fundamentals of Parameterized Complexity. Texts in Computer Science, Springer (2013)
14. Downey, R.G., Fellows, M.R., McCartin, C., Rosamond, F.A.: Parameterized approximation of dominating set problems. Inf. Process. Lett. 109(1), 68–70 (2008)
15. Flum, J., Grohe, M.: Parameterized Complexity Theory (Texts in Theoretical Computer Science. An EATCS Series). Springer-Verlag New York, Inc., Secaucus, NJ, USA (2006)
16. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W.H.Freeman and Company (1979)
17. Gaspers, S., Kratsch, D., Liedloff, M., Todinca, I.: Exponential time algorithms for the minimum dominating set problem on some graph classes. ACM Trans. Algorithms 6(1) (2009)
18. Gaspers, S., Liedloff, M.: A branch-and-reduce algorithm for finding a minimum independent dominating set. Discrete Mathematics & Theoretical Computer Science 14(1), 29–42 (2012)
19. Gera, R.: On dominator colorings in graphs. Graph Theory Notes of New York LII, 25–30 (2007)
20. Gera, R., Rasmussen, C., Horton, S.: Dominator colorings and safe clique partitions. Congressus numerantium 181, 19 (2006)
21. Golumbic, M.C.: Algorithmic Graph Theory and Perfect Graphs, Second Edition. Elsevier Science B.V. (2004)
22. Guha, S., Khuller, S.: Improved methods for approximating node weighted steiner trees and connected dominating sets. Inf. Comput. 150(1), 57–74 (1999)
23. Kim, D., Zhang, Z., Li, X., Wang, W., Wu, W., Du, D.Z.: A better approximation algorithm for computing connected dominating sets in unit ball graphs. IEEE Transactions on Mobile Computing 9(8), 1108–1118 (2010)
24. Kratsch, D., Fomin, F.V., Grandoni, F.: Exact algorithms for dominating set. In: Encyclopedia of Algorithms (2008)

25. Lawler, E.: A note on the complexity of the chromatic number problem. Information Processing Letters 5(3), 66 – 67 (1976)
26. Lenzen, C., Wattenhofer, R.: Minimum dominating set approximation in graphs of bounded arboricity. In: Distributed Computing, 24th International Symposium, DISC 2010, Cambridge, MA, USA, September 13-15, 2010. Proceedings. pp. 510–524 (2010)
27. Lewis, J.M., Yannakakis, M.: The Node-Deletion Problem for Hereditary Properties is NP-Complete. Journal of Computer and System Sciences 20(2), 219–230 (1980)
28. Lokshtanov, D., Narayanaswamy, N.S., Raman, V., Ramanujan, M.S., Saurabh, S.: Faster Parameterized Algorithms Using Linear Programming. ACM Trans. Algorithms 11(2), 15:1–15:31 (2014)
29. Lozin, V.V., Kaminski, M.: Coloring edges and vertices of graphs without short or long cycles. Contributions to Discrete Mathematics 2(1) (2007)
30. Niedermeier, R.: Invitation to Fixed Parameter Algorithms (Oxford Lecture Series in Mathematics and Its Applications). Oxford University Press, USA (March 2006)
31. Panolan, F., Philip, G., Saurabh, S.: B-chromatic number: Beyond NP-hardness. In: LIPIcs-Leibniz International Proceedings in Informatics. vol. 43. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik (2015)
32. Raman, V., Saurabh, S.: Short cycles make W-hard problems hard: FPT algorithms for W-hard problems in graphs with no short cycles. Algorithmica 52(2), 203–225 (2008)
33. van Rooij, J.M.M., Bodlaender, H.L.: Exact algorithms for dominating set. Discrete Applied Mathematics 159(17), 2147–2164 (2011)
34. Schönhage, D.D.A., Strassen, V.: Schnelle multiplikation grosser zahlen. Computing 7(3-4), 281–292 (1971)
35. Shalu, M.A., Sandhya, T.P.: The cd-coloring of graphs. In: Algorithms and Discrete Applied Mathematics - Second International Conference, CALDAM 2016, Thiruvananthapuram, India, February 18-20, 2016, Proceedings. pp. 337–348 (2016)
36. Shalu, M.A., Sandhya, T.P.: Personal communication (2016)
37. Venkatakrishnan, Y.B., Swaminathan, V.: Color class domination number of middle graph and center graph of $\mathrm{K}_{1,n}$, C_n, P_n. Advanced Modeling and Optimization 12, 233–237 (2010)
38. Yannakakis, M., Gavril, F.: The Maximum k-colorable Subgraph Problem for Chordal Graphs. Information Processing Letters 24(2), 133–137 (1987)