Algorithmic Aspects of Dominator Colorings in Graphs

S. Arumugam[1,2], K. Raja Chandrasekar[1], Neeldhara Misra[3], Geevarghese Philip[3], and Saket Saurabh[3]

[1] National Centre for Advanced Research in Discrete Mathematics (*n*-CARDMATH), Kalasalingam University, Krishnankoil, India.
`{s.arumugam.klu|rajmath84}@gmail.com`
[2] Conjoint Professor, School of Electrical Engineering and Computer Science The University of Newcastle, NSW 2308, Australia.
[3] The Institute of Mathematical Sciences, Chennai, India.
`{neeldhara|gphilip|saket}@imsc.res.in`

Abstract. In this paper we initiate a systematic study of a problem that has the flavor of two classical problems, namely Coloring and Domination, from the perspective of algorithms and complexity. A *dominator coloring* of a graph G is an assignment of colors to the vertices of G such that it is a proper coloring and every vertex dominates all the vertices of at least one color class. The minimum number of colors required for a dominator coloring of G is called the *dominator chromatic number* of G and is denoted by $\chi_d(G)$. In the Dominator Coloring (DC) problem, a graph G and a positive integer k are given as input and the objective is to check whether $\chi_d(G) \leq k$. We first show that unless P=NP, DC cannot be solved in polynomial time on bipartite, planar, or split graphs. This resolves an open problem posed by Chellali and Maffray [*Dominator Colorings in Some Classes of Graphs, Graphs and Combinatorics, 2011*] about the polynomial time solvability of DC on chordal graphs. We then complement these hardness results by showing that the problem is fixed parameter tractable (FPT) on chordal graphs and in graphs which exclude a fixed apex graph as a minor.

1 Introduction

Dominating Set and Coloring are among the most fundamental problems in graph theory, algorithms and combinatorial optimization. Dominating Set asks for the minimum set of vertices such that every vertex of the graph not in this set has a neighbor in it. In Coloring we are asked to color the vertices with as few colors as possible, so that no edge is monochromatic, that is, both the endpoints of each edge receive different colors. These are classical NP-hard problems [17] and are well-studied from the point of view of approximation algorithms [12, 23, 25–27] and parameterized complexity [10, 14, 16]. Dominating Set and Coloring are "hard" problems from these perspectives. Thus, Dominating Set and Coloring are known to be W[2]-complete and para-NP complete, respectively, in parameterized complexity [10]. Further, $(1 - o(1)) \ln n$ and $n^{\epsilon}; \epsilon > 0$

are respective thresholds below which these problems cannot be approximated efficiently (unless NP has slightly super-polynomial time algorithm [12] or unless P=NP [27]).

DOMINATING SET and COLORING have a number of applications and this has led to the algorithmic study of numerous variants of these problems. Among the most well known ones are CONNECTED DOMINATING SET, INDEPENDENT DOMINATING SET, PERFECT CODE, LIST COLORING, EDGE COLORING, ACYCLIC EDGE COLORING and CHOOSABILITY. Since both the problem and its variants are computationally hard problems, most of the research centers around algorithms in special classes of graphs like interval graphs, chordal graphs, planar graphs and H-minor free graphs. In this paper we initiate a systematic algorithmic study on the DOMINATOR COLORING (DC) problem that has a flavor of both these classical problems. A *dominator coloring* of a graph G is an assignment of colors to the vertices of G such that it is a proper coloring (no edge is monochromatic) and every vertex dominates all vertices of at least one color class. The minimum number of colors required for a dominator coloring of G is called the *dominator chromatic number* of G and is denoted by $\chi_d(G)$. The problem we study is formally defined as follows.

DOMINATOR COLORING (DC)
Input: A graph G and an integer $k \geq 1$.
Parameter: k.
Question: Does there exist a dominator coloring of G using at most k colors?

Gera et al. [22] introduced the concept of dominator chromatic number, and a number of basic combinatorial and algorithmic results on DC have been obtained [20–22, 24]. For example, it was observed by Gera [20] that DC is NP-complete on general graphs by a simple reduction from 3-COLORING. More precisely, for any fixed $k \geq 4$, it is NP-complete to decide if a graph admits a dominator coloring with at most k colors [20]. In a recent paper Chellali and Maffray [6] show that unlike 3-COLORING, one can decide in polynomial time if a graph has dominator chromatic number 3. Furthermore, they show that the problem is polynomial time solvable on P_4 free graphs, and leave as a "challenging open problem" whether the problem can be solved in polynomial time on chordal graphs.

In this paper we do a thorough algorithmic study of this problem, analyzing both the classical complexity and the parameterized complexity. We begin by showing that unless P=NP, DC cannot be solved in polynomial time on bipartite, planar, or split graphs. The first two arguments are simple but make use of an unusual sequence of observations. The NP-completeness reduction on split graphs is quite involved. Since split graphs form a subclass of chordal graphs, this answers, in the negative, the open problem posed by Chellali and Maffray.

We complement our hardness results by showing that the problem is "fixed parameter tractable" on several of the graph classes mentioned above. Informally, a *parameterization* of a problem assigns an integer k to each input instance and a parameterized problem is *fixed-parameter tractable* (FPT) if there is an algorithm that solves the problem in time $f(k) \cdot |I|^{O(1)}$, where $|I|$ is the size of the input and f is an arbitrary computable function that depends only on the parameter k. We refer the interested reader to standard texts [10, 14] on parameterized complexity. We show that DC is FPT on planar graphs, apex minor free graphs, split graphs and chordal graphs.

2 Preliminaries

All graphs in this article are finite and undirected, with neither loops nor multiple edges. n denotes the number of vertices in a graph, and m the number of edges. A subset $D \subseteq V$ of the vertex set V of a graph G is said to be a *dominating set* of G if every vertex in $V \setminus D$ is adjacent to some vertex in D. The *domination number* $\gamma(G)$ of G is the size of a smallest dominating set of G. A *proper coloring* of graph G is an assignment of colors to the vertices of G such that the two end vertices of any edge have different colors. 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 in which there is 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. We make use of the following known results.

Theorem 1. [20] *Let G be a connected graph. Then* $max\{\chi(G), \gamma(G)\} \leq \chi_d(G) \leq \chi(G) + \gamma(G)$.

Definition 1. *A* tree decomposition *of a (undirected) graph $G = (V, E)$ is a pair (X, U) where $U = (W, F)$ is a tree, and $X = (\{X_i \mid i \in W\})$ is a collection of subsets of V such that*

1. $\bigcup_{i \in W} X_i = V$,
2. *for each edge $vw \in E$, there is an $i \in W$ such that $v, w \in X_i$, and*
3. *for each $v \in V$, the set of vertices $\{i \mid v \in X_i\}$ forms a subtree of U.*

The width *of (X, U) is $\max_{i \in W}\{|X_i| - 1\}$. The* treewidth *$tw(G)$ of G is the minimum width over all the tree decompositions of G.*

Both our FPT algorithms make use of the fact that the DC problem can be expressed in Monadic Second Order Logic (MSOL) on graphs. The syntax of MSOL on graphs includes the logical connectives $\vee$, $\wedge$, $\neg$, $\Leftrightarrow$, $\Rightarrow$, variables for vertices, edges, sets of vertices and sets of edges, the quantifiers $\forall$, $\exists$ that can be applied to these variables, and the following five binary relations: (1) $u \in U$ where u is a vertex variable and U is a vertex set variable; (2) $d \in D$ where d is an edge variable and D is an edge set variable; (3) $\mathbf{inc}(d, u)$, where d is an edge variable, u is a vertex variable, and the interpretation is that the edge d is incident on the vertex u; (4) $\mathbf{adj}(u, v)$, where u and v are vertex variables u, and the interpretation is that u and v are adjacent; (5) equality of variables representing vertices, edges, sets of vertices and sets of edges.

Many common graph and set-theoretic notions can be expressed in MSOL [5, 8]. In particular, let $V_1, V_2, \ldots, V_k$ be a set of subsets of the vertex set $V(G)$ of a graph G. Then the following notions can be expressed in MSOL:

– $V_1, V_2, \ldots, V_k$ is a partition of $V(G)$:

$$\begin{aligned} Part(V(G); V_1, V_2, \ldots, V_k) \equiv \forall v \in V(G)[(v \in V_1 \vee v \in V_2 \vee \cdots \vee v \in V_k) \quad \wedge \\ (\neg(v \in V_1 \cap V_2)) \wedge (\neg(v \in V_1 \cap V_3)) \wedge \cdots \wedge (\neg(v \in V_{k-1} \cap V_k))] \quad \wedge \\ (\exists v \in V(G)[v \in V_1]) \wedge (\exists v \in V(G)[v \in V_2]) \wedge \cdots \wedge (\exists v \in V(G)[v \in V_k]) \end{aligned}$$

– V_i is an independent set in G:

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

– Vertex v dominates all vertices in the set V_i:

$$Dom(v, V_i) \equiv \forall w \in V_i[\neg(w = v) \implies \mathbf{adj}(v, w)]$$

For a graph G and a positive integer k, we use $\varphi(G, k)$ to denote an MSOL formula which states that G has a dominator coloring with at most k colors:

$$\begin{aligned} \varphi(G, k) \equiv \exists V_1, V_2, \ldots, V_k \subseteq V(G)[Part(V(G); V_1, V_2, \ldots, V_k) \quad \wedge \\ IndSet(V_1) \wedge IndSet(V_2) \wedge \cdots \wedge IndSet(V_k) \quad \wedge \\ \forall v \in V(G)[Dom(v, V_1) \vee Dom(v, V_2) \vee \cdots \vee Dom(v, V_k)]] \end{aligned} \tag{1}$$

The following well known result states that every optimization problem expressible in MSOL has a linear time algorithm on graphs of bounded treewidth.

Proposition 1. [1, 3, 5, 7, 9] *Let φ be a property that is expressible in Monadic Second Order Logic. For any fixed positive integer t, there is an algorithm that, given a graph G of treewidth at most t as input, finds a largest (alternatively, smallest) set S of vertices of G that satisfies φ in time $f(t, |\varphi|)|V(G)|$ for a computable function $f()$.*

Since the size $|\varphi(G, k)|$ of the MSOL expression 1 is a function of k, we have

Theorem 2. *Given a graph G of treewidth t and a positive integer k as inputs, the* Dominator Coloring *problem can be solved in $f(t, k)|V(G)|$ time for a computable function $f()$.*

The operation of *contracting* an edge $\{u, v\}$ of a graph consists of replacing the two vertices u, v with a single vertex which is adjacent to all the former neighbours of u and v. A graph H is said to be a *contraction* of a graph G if H can be obtained from G by contracting zero or more edges of G. H is said to be a *minor* of G if H is a contraction of some subgraph of G. A graph G is said to be *apex* graph if there exists a vertex in G whose removal from G yields a planar graph. A family $\mathcal{F}$ of graphs is said to be *apex minor free* if there is a specific apex graph H such that no graph in $\mathcal{F}$ has H as a minor. For instance, planar graphs are apex minor free since they exclude the apex graph K_5 as a minor. The treewidth of an apex minor free graph can be approximated to within a constant factor in polynomial time:

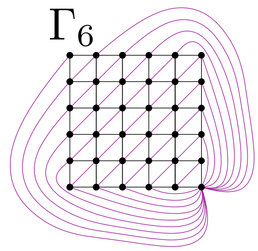

Fig. 1. The graph Γ_6.

Proposition 2. [**13, Theorem 6.4**] *For any graph H, there is a constant w_H and a polynomial time algorithm which finds a tree decomposition of width at most $w_H t$ for any H-minor-free graph G of treewidth t.*

For $\ell \in \mathbb{N}$, Γ_ℓ is defined [15] to be the graph obtained from the $\ell \times \ell$-grid by (1) triangulating the internal faces such that all the internal vertices become of degree 6 and all non-corner external vertices are of degree 4, and (2) adding edges from one corner of degree two to all vertices of the external face. Figure 1 depicts Γ_6. Fomin et al. showed that any apex minor free graph of large treewidth contains a proportionately large Γ_ℓ as a contraction. More precisely:

Proposition 3. [**15, Theorem 1**] *For any apex graph H, there is a constant c_H such that every connected graph G which excludes H as a minor and has treewidth at least $c_H \ell$ contains Γ_ℓ as a contraction.*

3 Hardness Results

In this section we show that DC is NP-hard on very restricted classes of graphs. The only known hardness result for this problem is that it is NP-complete on general graphs [20]. In fact even determining whether there exists a dominator coloring of G using at most 4 colors is NP-complete. The proof is obtained by a reduction from 3-Coloring – checking whether an input graph is 3-colorable or not – to DC. Given an instance G to 3-Coloring, an instance G' for DC is obtained by adding a new vertex (universal vertex) and making it adjacent to every vertex of G. Now one can easily argue that G is 3 colorable if and only if G' has dominator coloring of size at most 4. Notice, however, that this simple reduction cannot be used to show that DC is NP-complete on restricted graph classes like planar graphs or split graphs or chordal graphs. We start with a few simple claims that we will make use of later.

Lemma 1. *Let $G = (V, E)$ be a graph. Given a proper a-coloring $\mathcal{C}$ of G and a dominating set D of G with $|D| = b$, we can find, in $O(|V| + |E|)$ time, a dominator coloring of G with at most $a + b$ colors.*

Proof. Let $\mathcal{C} = \{V_1, V_2, \ldots, V_a\}$ be a proper coloring of G and let D be a dominating set with $|D| = b$. Then $\mathcal{C}' = \{\{v\} : v \in D\} \cup \{V_i \cap (V - D) : V_i \in \mathcal{C}\}$ is a dominator coloring of G with at most $a + b$ colors. □

Corollary 1. [⋆][4] *If there exists an α-approximation algorithm for the chromatic number problem and a β-approximation algorithm for the domination number problem, then there exists an $(\alpha + \beta)$-approximation algorithm for the dominator chromatic number problem.*

Lemma 2. [⋆] *Let $\mathcal{F}$ be a class of graphs on which the Dominating Set problem is NP-complete. If the disjoint union of any two graphs in $\mathcal{F}$ is also in $\mathcal{F}$, then there is no polynomial time algorithm that finds a constant additive approximation for the Dominating Set problem on $\mathcal{F}$, unless $P = NP$.*

Corollary 2. [⋆] DOMINATOR COLORING *on planar graphs cannot be solved in polynomial time, unless $P = NP$.*

Corollary 3. [⋆] DOMINATOR COLORING *on bipartite graphs cannot be solved in polynomial time, unless $P = NP$.*

3.1 NP-hardness of DC on Split Graphs

We now proceed to prove that the DC problem is NP-complete for split graphs. Our starting point is the following known characterization:

Theorem 3. [2] *Let G be a split graph with split partition (K, I) and $|K| = \omega(G)$, where K is a clique and I an independent set. Then $\chi_d(G) = \omega$ or $\omega + 1$. Further $\chi_d(G) = \omega$ if and only if there exists a dominating set D of G such that $D \subseteq K$ and every vertex v in I is nonadjacent to at least one vertex in $K \setminus D$.*

We exploit this characterization, and prove NP-completeness on split graphs by demonstrating the NP-completeness of the problem of checking if there exists a dominating set D of G such that $D \subseteq K$ and every vertex v in I is nonadjacent to at least one vertex in $K \setminus D$. We call this problem SPLIT GRAPH DOMINATION.

For showing SPLIT GRAPH DOMINATION NP-complete, we will need to define an intermediate problem called PARTITION SATISFIABILITY, and demonstrate that it is NP-complete. We will then show that DC is NP-hard on split graphs by establishing a reduction from PARTITION SATISFIABILITY.

Let ϕ be a CNF formula. Then we use $\mathcal{C}(\phi)$ to denote the set of clauses of ϕ. If C is a clause of ϕ, then we use $\nu(C)$ to denote the set of variables that appear in C. A clause is said to be all-positive (negative) if all the literals that appear in it are positive (negative).

[4] Due to space constraints, proofs of results marked with a [⋆] have been deferred to a longer version of the paper.

Definition 2 (Partition Normal Form). *A CNF formula ϕ over the variable set V is said to be in partition normal form if $\mathcal{C}(\phi)$ admits a partition into two parts $C_P(\phi)$ and $C_N(\phi)$ and there exists a bijection $f : C_P(\phi) \to C_N(\phi)$ such that for every $C \in C_P(\phi)$ the following conditions are satisfied:* (1) $\nu(C) \cup \nu(f(C)) = V$ *and* (2) $\nu(C) \cap \nu(f(C)) = \emptyset$. *Any clause in $C_P(\phi)$ is required to be an all-positive clause and any clause in $C_N(\phi)$ is required to be an all-negative clause.*

We are now ready to describe the problem Partition Satisfiability.

Partition Satisfiability
Input: A formula ϕ in CNF, over variables in V, given in partition normal form.
Question: Is ϕ satisfiable?

We establish the NP-completeness of Partition Satisfiability by a reduction from Disjoint Factors:

Disjoint Factors
Input: A word w over an alphabet Σ.
Question: For every $a \in \Sigma$, does there exist a substring w_a of w that begins and ends in a, such that for every $a, b \in \Sigma$, w_a and w_b do not overlap in w?

The problem of Disjoint Factors is known to be NP-complete [4]. Substrings that begin and end with the same letter a are referred to as *a-factors*.

Lemma 3. Partition Satisfiability *is* NP-complete.

Proof. Let $w = w_1 w_2 \dots w_n$ be an instance of Disjoint Factors over the alphabet

$$\Sigma = \{a_1, \dots, a_k\}.$$

For $1 \le i < j \le n$ and $1 \le l \le k$, we call the triplet (i, j, l) *valid* if the substring $w_i \dots w_j$ is an a_l-factor. Let $\mathcal{F}$ denote the set of valid triplets. We construct an instance of Partition Satisfiability as follows:

For every valid triplet (i, j, l), introduce the variable $P_l(i, j)$. For every $1 \le l \le k$, introduce the clause:

$$C_l := \left(\bigvee_{\{i,j \,:\, (i,j,l) \in \mathcal{F}\}} P_l(i, j) \right).$$

Let ϕ_{FACTOR} be the conjunction of the clauses thus formed: $\phi_{\text{FACTOR}} := C_1 \wedge C_2 \wedge \ldots \wedge C_k$.

Further, for every i_1, j_1 and i_2, j_2 such that $1 \le i_1 < j_1 \le n$ and $1 \le i_2 < j_2 \le n$, and $[i_1, j_1] \cap [i_2, j_2] \neq \emptyset$, and there exist l_1, l_2; $1 \le l_1, l_2 \le k$, such that $(i_1, j_1, l_1) \in \mathcal{F}$ and $(i_2, j_2, l_2) \in \mathcal{F}$, we introduce the following clause:

$$C := \left(\overline{P_{l_1}(i_1, j_1)} \vee \overline{P_{l_2}(i_2, j_2)}\right)$$

Let $\mathcal{D}$ denote the set of clauses described above. Further, let ϕ_{DISJOINT} be the conjunction of these clauses: $\phi_{\text{DISJOINT}} := \bigwedge_{C \in \mathcal{D}} C$.

Claim. The formula: $\phi := \phi_{\text{DISJOINT}} \wedge \phi_{\text{FACTOR}}$ is satisfiable if and only if (w, Σ) is a YES-instance of Disjoint Factors.

Proof. ($\Rightarrow$) Let χ be a satisfying assignment of ϕ. For all l, $1 \leq l \leq k$, there exists at least one pair (i, j), $1 \leq i < j \leq n$, such that χ sets $P_l(i, j)$ to 1. Indeed, if not, χ would fail to satisfy the clause C_l. Now, note that $w_i \dots w_j$ is a a_l-factor, since the variable $P_l(i, j)$ corresponds to a valid triplet.

We pick $w_i \dots w_j$ as the factor for a_l (if $P_l(i, j)$ is set to 1 by χ for more than one pair (i, j), then any one of these pairs will serve our purpose). It only remains to be seen that for $r, s \in \Sigma$, if $w_{i_1} \dots w_{j_1}$ is chosen as a a_r-factor, and $w_{i_2} \dots w_{j_2}$ is chosen as a a_s-factor, then $w_{i_1} \dots w_{j_1}$ and $w_{i_2} \dots w_{j_2}$ do not overlap in w. This is indeed the case, for if they did overlap, then it is easily checked that χ would fail to satisfy the clause: $\left(\overline{P_r(i_1, j_1)} \vee \overline{P_s(i_2, j_2)}\right)$.

($\Leftarrow$) If (w, Σ) is a YES-instance of Disjoint Factors, then for every l, $1 \leq l \leq k$, there exist i, j; $1 \leq i < j \leq n$, such that $(i, j, l) \in \mathcal{F}$. We claim that setting all the "corresponding" $P_l(i, j)$ variables to 1 is a satisfying assignment for ϕ.

Indeed, every C_l is satisfied because there exists an a_l-factor for every l. Further, it is routine to verify that all clauses in $\mathcal{D}$ are satisfied because the chosen factors do not overlap in w. □

Now, it remains to construct from ϕ an equivalent formula ψ that is in partition normal form. To this end, we will use two new variables, $\{x, y\}$. Recall that we use V to denote the set of variables that appear in ϕ. For every clause C_l, define the clause $\hat{C}_l$ as: $\hat{C}_l := \left(\overline{x} \vee \overline{y} \vee \bigvee_{z \,\in\, V \setminus \nu(C_l)} \overline{z}\right)$.

Similarly, for every clause $C \in \mathcal{D}$, define $\hat{C}$ as: $\hat{C} := \left(x \vee y \vee \bigvee_{z \,\in\, V \setminus \nu(C)} z\right)$.

Let ψ be obtained by the conjunction of ϕ with the newly described clauses: $\psi := \phi \wedge \left(\bigwedge_{1 \leq l \leq k} \hat{C}_l\right) \wedge \left(\bigwedge_{C \in \mathcal{D}} \hat{C}\right)$.

Clearly, ψ is in partition normal form. The following partition of the clauses of ψ: $C_P = \{C_l \;:\; 1 \leq l \leq k\} \cup \{\hat{C} \;:\; C \in \mathcal{D}\}$ and $C_N = \{\hat{C}_l \;:\; 1 \leq l \leq k\} \cup \{C \;:\; C \in \mathcal{D}\}$ is a partition into all-positive and all-negative clauses. The bijection f defined as: $f(C_l) = \hat{C}_l$, for $1 \leq l \leq k$ and $f(\hat{C}) = C$, for $C \in \mathcal{D}$ is easily seen to be a bijection with the properties demanded by the definition of the partition normal form. We now arrive at our concluding claim:

Claim. ϕ is satisfiable if and only if ψ is satisfiable.

Proof. ($\Rightarrow$) Let χ be a satisfying assignment for ϕ. Extend ϕ to the new variables $\{x, y\}$ as follows: $\chi(x) = 1$ and $\chi(y) = 0$. It is easy to see that χ is satisfying for ψ.

($\Leftarrow$) This direction is immediate, as $\mathcal{C}(\phi) \subseteq \mathcal{C}(\psi)$. □

The proof that Partition Satisfiability is NP-hard follows when we put the two claims together: by appending the construction of ψ from ϕ to the formula ϕ obtained from the Disjoint Factors instance, we obtain an equivalent instance of Partition Satisfiability. This concludes the proof. We note that membership in NP is trivial — an assignment to the variables is clearly a certificate that can be verified in linear time. The lemma follows. □

Recall the Split Graph Domination problem that we introduced in the beginning of this section:

Split Graph Domination

Input:	Split graph G with split partition (K, I) and $\|K\| = \omega$.
Question:	Does there exist a dominating set D of G such that $D \subseteq K$ and every vertex v in I is nonadjacent to at least one vertex in $K \setminus D$?

We now turn to a proof that Split Graph Domination is NP-complete.

Theorem 4. Split Graph Domination *is* NP-complete.

Proof. It is straightforward to see that Split Graph Domination is in NP. We now prove that it is NP-hard by a reduction from Partition Satisfiability.

Given an instance ϕ (over the variables V) of Partition Satisfiability, we construct a split graph G with split partition (K, I) as follows. Introduce, for every variable in V, a vertex in K and for every all-positive clause of ϕ, a vertex in I: $K = \{v[x] \, : \, x \in V\}, I = \{u[C] \, : \, C \in C_P(\phi)\}$.

A pair of vertices $v[x]$ and $u[C]$ are adjacent if the variable x belongs to the clause C, that is, $x \in \nu(C)$. We also make all vertices in K pairwise adjacent and all vertices in I pairwise independent. This completes the construction.

Suppose ϕ admits a satisfying truth assignment χ. Let $D = \{v[x] \in K \, : \, \chi(x) = 1\}$. We now prove that this choice of D is a split dominating set. Consider $u[C] \in I$. There exists at least one $x \in V$ such that $x \in \nu(C)$ and $\chi(x) = 1$. Thus the corresponding vertex $v[x] \in D$, and $u[C]$ is dominated. Further, consider the all-negative clause $\hat{C}$ corresponding to C, that contains every variable in V that is not in $\nu(C)$. Since χ is a satisfying assignment, there is at least one $y \in V \setminus \nu(C)$ such that $\chi(y) = 0$. Clearly, $v[y] \notin D$, and $v[y]$ is not adjacent to $u[C]$.

Conversely, suppose there exists a dominating set $D \subseteq K$ such that each $u[C]$ in I is nonadjacent to at least one vertex in $K \setminus D$. Consider the following truth assignment χ for ϕ: $\chi(x) = 1$ if, and only if, $v[x] \in K \cap D$. We now prove that χ is a satisfying assignment. Consider any all-positive clause C. Since $u[C]$ was dominated by D, there exists a variable $x \in \nu(C)$ such that $v[x] \in D$, and thus $\chi(x) = 1$. Consider the corresponding all-negative clause $\hat{C}$. Since $K \setminus D$ contains at least one non-neighbor of $v[x]$, there exists a $y \notin \nu(C)$ such that $\chi(y) = 0$. Note that $y \notin \nu(C)$ implies that $y \in \nu(\hat{C})$. Recall that the assignment $\chi(y) = 0$ is then satisfying for $\hat{C}$, since $\hat{C}$ is an all-negative clause. □

From Theorem 3 and Theorem 4 we get

Theorem 5. *DC when restricted to split graphs is* NP-complete.

4 Parameterized Algorithms

In this section we investigate the fixed-parameter tractability of the DC problem in certain graph classes. Recall that it is NP-complete to decide if a graph admits a dominator coloring with at most 4 colors [20]. It follows that in *general* graphs, the DC problem cannot be solved even in time $n^{g(k)}$ for any function $g(k)$ — that is, DC does not belong to the complexity class XP — unless P=NP. Hence DC is not FPT in general graphs unless P=NP. As we show below, however, the problem is FPT in two important classes of graphs, namely apex-minor-free graphs (which include planar graphs as a special case) and chordal graphs. Recall that it is NP-complete to decide if a planar graph admits a proper 3-coloring [18]. As a consequence, the Graph Coloring problem parameterized by the number of colors is not even in XP in planar graphs. Our result for planar graphs thus brings out a marked difference in the parameterized complexity of these two problems when restricted to planar graphs.

Apex Minor Free Graphs. We now show that the Dominator Coloring problem is FPT on apex minor free graphs. This implies, as a special case, that the problem is FPT on planar graphs. We first show that if the treewidth of the input apex minor free graph is large, then the graph has no dominator coloring with a small number of colors.

Theorem 6. [⋆] *For any apex graph H, there is a constant d_H such that any connected graph G which excludes H as a minor and has treewidth at least $d_H\sqrt{k}$ has no dominator coloring with at most k colors.*

Let (G, k) be an instance of the Dominator Coloring problem, where G excludes the apex graph H as a minor. Let $t = w_H d_H \sqrt{k}$ where d_H, w_H are the constants of Theorem 6 and Proposition 2, respectively. To solve the problem on this instance, we invoke the approximation subroutine implied by Proposition 2 on the graph G. If this subroutine returns a tree decomposition with treewidth more than t, then we return NO as the answer. Otherwise we solve the problem using the algorithm of Theorem 2, and so we have:

Theorem 7. [⋆] *The* Dominator Coloring *problem is fixed parameter tractable on apex minor free graphs.*

Chordal Graphs and Split Graphs. We now show that the Dominator Coloring problem is FPT on chordal graphs. For a special class of chordal graphs, namely split graphs, we give an FPT algorithm which runs in time single-exponential in the parameter.

Theorem 8. *The* Dominator Coloring *problem is fixed parameter tractable on chordal graphs.*

Proof. Let (G, k) be an instance of the Dominator Coloring problem, where G is chordal. The algorithm first finds a largest clique in G. If the number of vertices in this clique is more than k, then it returns NO as the answer. Otherwise it invokes the algorithm of Theorem 2 as a subroutine to solve the problem.

To see that this algorithm is correct, observe that if G contains a clique C with more than k vertices, then $\chi(G) > k$ since it requires more than k colors to properly color the subgraph C itself. It follows from Theorem 1 that $\chi_d(G) > k$, and so it is correct to return NO. A largest clique in a chordal graph can be found in linear time [19]. If the largest clique in G has size no larger than k, then — as is well known — the treewidth of G is at most $k-1$, and so the subroutine from Theorem 2 runs in at most $f((k-1), k)|V(G)| = g(k)|V(G)|$ time. Thus the algorithm solves the problem in FPT time. □

The Dominator Coloring problem can be solved in "fast" FPT time on split graphs:

Theorem 9. [⋆] *The* Dominator Coloring *problem can be solved in* $O(2^k \cdot n^2)$ *time on a split graph on* n *vertices.*

5 Conclusion and Scope

We derived several algorithmic results about the Dominator Coloring (DC) problem. We showed that the DC problem remains hard on several graph classes, including bipartite graphs, planar graphs, and split graphs. In the process we also answered, in the negative, an open problem by Chellali and Maffra [6] about the polynomial time solvability of DC on chordal graphs. Finally, we showed that though the problem cannot be solved in polynomial time on the aforementioned graph classes, it is FPT on apex minor free graphs and on chordal graphs. From Theorem 1 and from the fact that finding a constant additive approximation for the Dominating Set problem is W[2]-hard [11], it follows that the DC problem is W[2]-hard on bipartite graphs, and so also on the larger class of perfect graphs. An interesting problem which remains open is whether the DC problem is solvable in polynomial time on interval graphs.

References

1. S. Arnborg, J. Lagergren, and D. Seese. Easy problems for tree-decomposable graphs. *Journal of Algorithms*, 12(2):308–340, 1991.
2. S. Arumugam, J. Bagga, and K. R. Chandrasekar. On dominator colorings in graphs. *Manuscript*, 2010.
3. H. L. Bodlaender. A linear time algorithm for finding tree-decompositions of small treewidth. *SIAM Journal on Computing*, 25:1305–1317, 1996.
4. H. L. Bodlaender, S. Thomassé, and A. Yeo. Kernel bounds for disjoint cycles and disjoint paths. In *ESA*, volume 5757 of *LNCS*, pages 635–646, 2009.
5. R. B. Borie, G. R. Parker, and C. A. Tovey. Automatic Generation of Linear-Time Algorithms from Predicate Calculus Descriptions of Problems on Recursively Constructed Graph Families. *Algorithmica*, 7:555–581, 1992.

6. M. Chellali and F. Maffray. Dominator colorings in some classes of graphs. *Graphs and Combinatorics*, pages 1–11, 2011.
7. B. Courcelle. The monadic second-order logic of graphs. i. recognizable sets of finite graphs. *Information and Computation*, 85(1):12–75, 1990.
8. B. Courcelle. The expression of graph properties and graph transformations in monadic second-order logic. In G. Rozenberg, editor, *Handbook of Graph Grammars and Computing by Graph Transformations, Volume 1: Foundations*, chapter 5. World Scientific, 1997.
9. B. Courcelle and M. Mosbah. Monadic second-order evaluations on tree-decomposable graphs. *Theoretical Computer Science*, 109(1–2):49–82, 1993.
10. R. G. Downey and M. R. Fellows. *Parameterized Complexity*. Springer-Verlag, New York, 1999.
11. R. G. Downey, M. R. Fellows, C. McCartin, and F. Rosamond. Parameterized approximation of dominating set problems. *Information Processing Letters*, 109(1):68 – 70, 2008.
12. U. Feige. A threshold of ln n for approximating set cover. *Journal of the ACM*, 45(4):634–652, 1998.
13. U. Feige, M. Hajiaghayi, and J. R. Lee. Improved approximation algorithms for minimum-weight vertex separators. *SIAM Journal on Computing*, 38(2):629–657, 2008.
14. J. Flum and M. Grohe. *Parameterized Complexity Theory*. Texts in Theoretical Computer Science. An EATCS Series. Springer-Verlag, Berlin, 2006.
15. F. V. Fomin, P. A. Golovach, and D. M. Thilikos. Contraction bidimensionality: The accurate picture. In *ESA*, volume 5757 of *LNCS*, pages 706–717, 2009.
16. F. V. Fomin and D. M. Thilikos. Dominating sets in planar graphs: Branch-width and exponential speed-up. *SIAM Journal on Computing*, 36(2):281–309, 2006.
17. M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP–Completeness*. Freeman, San Francisco, 1979.
18. M. R. Garey, D. S. Johnson, and L. Stockmeyer. Some simplified *NP*-complete graph problems. *Theoretical Computer Science*, 1:237–267, 1976.
19. F. Gavril. Algorithms for minimum coloring, maximum clique, minimum covering by cliques, and maximum independent set of a chordal graph. *SIAM Journal on Computing*, 1(2):180–187, 1972.
20. R. Gera. On dominator coloring in graphs. In *Graph Theory Notes of New York*, pages 25–30. LII, 2007.
21. R. Gera. On the dominator colorings in bipartite graphs. In *ITNG*, pages 1–6. IEEE, 2007.
22. R. Gera, C. Rasmussen, and S. Horton. Dominator colorings and safe clique partitions. *Congressus Numerantium*, 181(7-9):19–32, 2006.
23. M. M. Halldórsson. A still better performance guarantee for approximate graph coloring. *Information Processing Letters*, 45(1):19–23, 1993.
24. S. Hedetniemi, S. Hedetniemi, A. McRae, and J. Blair. Dominator colorings of graphs. *Preprint*, 2006.
25. D. S. Johnson. Approximation algorithms for combinatorial problems. *Journal of Computer and System Sciences*, 9(3):256–278, 1974.
26. L. Lovász. On the ratio of optimal integral and fractional covers. *Discrete Mathematics*, 13:383–390, 1975.
27. C. Lund and M. Yannakakis. On the hardness of approximating minimization problems. *Journal of the ACM*, 41(5):960–981, 1994.