Defective Coloring on Classes of Perfect Graphs

Rémy Belmonte[1], Michael Lampis[2], and Valia Mitsou[3,⋆]

[1] University of Electro-Communications, Chofu, Tokyo, 182-8585, Japan,
`remy.belmonte@uec.ac.jp`
[2] Université Paris-Dauphine, PSL Research University, CNRS, UMR 7243
LAMSADE, 75016, Paris, France, `michail.lampis@dauphine.fr`
[3] Université de Lyon, LIRIS, CNRS, UMR 5205,
Université Lyon 1, 69622, Villeurbanne, Lyon, France, `vmitsou@liris.cnrs.fr`

Abstract. In DEFECTIVE COLORING we are given a graph G and two integers χ_d, Δ^* and are asked if we can χ_d-color G so that the maximum degree induced by any color class is at most Δ^*. We show that this natural generalization of COLORING is much harder on several basic graph classes. In particular, we show that it is NP-hard on split graphs, even when one of the two parameters χ_d, Δ^* is set to the smallest possible fixed value that does not trivialize the problem ($\chi_d = 2$ or $\Delta^* = 1$). Together with a simple treewidth-based DP algorithm this completely determines the complexity of the problem also on chordal graphs.

We then consider the case of cographs and show that, somewhat surprisingly, DEFECTIVE COLORING turns out to be one of the few natural problems which are NP-hard on this class. We complement this negative result by showing that DEFECTIVE COLORING is in P for cographs if either χ_d or Δ^* is fixed; that it is in P for trivially perfect graphs; and that it admits a sub-exponential time algorithm for cographs when both χ_d and Δ^* are unbounded.

1 Introduction

In this paper we study the computational complexity of DEFECTIVE COLORING, which is also known in the literature as IMPROPER COLORING: given a graph and two parameters χ_d, Δ^* we want to color the graph with χ_d colors so that every color class induces a graph with maximum degree at most Δ^*. DEFECTIVE COLORING is a very natural generalization of GRAPH COLORING, which corresponds to the case $\Delta^* = 0$. As a result, since the introduction of this problem more than thirty years ago [13, 2] a great deal of research effort has been devoted to its study. Among the topics that have been investigated are its extremal properties [18, 30, 31, 10, 1, 20], especially on planar graphs and graphs on surfaces [14, 4, 12, 25], as well as its asymptotic behavior on random graphs [28, 29]. Lately, the problem has attracted renewed interest due to its applicability to communication networks, with the coloring of the graph modeling the assignment of frequencies to nodes and Δ^* representing some amount

of tolerable interference. This has led to the study of the problem on Unit Disk Graphs [24] as well as various classes of grids [3, 7, 5]. Weighted generalizations have also been considered [6, 23]. More background can be found in the survey by Frick [17] or Kang's PhD thesis [27].

Our main interest in this paper is to study the computational complexity of DEFECTIVE COLORING through the lens of structural graph theory, that is, to investigate the classes of graphs for which the problem becomes tractable. Since DEFECTIVE COLORING generalizes GRAPH COLORING we immediately know that it is NP-hard already in a number of restricted graph classes and for small values of χ_d, Δ^*. Nevertheless, the fundamental question we would like to pose is what is the *additional* complexity brought to this problem by the freedom to produce a slightly improper coloring. In other words, we ask what are the graph classes where even though GRAPH COLORING is easy, DEFECTIVE COLORING is still hard (and conversely, what are the classes where both are tractable). Though some results of this type are already known (for example Cowen et al. [14] prove that the problem is hard even on planar graphs for $\chi_d = 2$), this question is not well-understood. Our focus on this paper is to study DEFECTIVE COLORING on subclasses of perfect graphs, which are perhaps the most widely studied class of graphs where GRAPH COLORING is in P. The status of the problem appears to be unknown here, and in fact its complexity on interval and even proper interval graphs is explicitly posed as an open question in [24].

Chordal graphs	Cographs
NP-hard on Split if $\chi_d \geq 2$ Theorem 10	NP-hard Theorem 2
NP-hard on Split if $\Delta^* \geq 1$ Theorem 9	In P if χ_d or Δ^* is fixed Theorems 5,6
In P if χ_d, Δ^* fixed Theorem 13	Solvable in $n^{O\left(n^{4/5}\right)}$ Theorem 7
In P on Trivially perfect for any χ_d, Δ^* Theorem 4	

Table 1. Summary of results
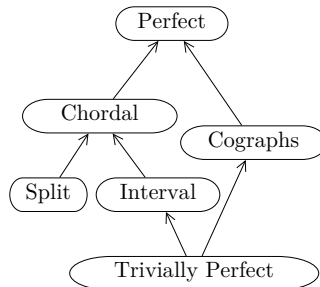
Our results revolve around two widely studied classes of perfect graphs: split graphs and cographs. For split graphs we show not only that DEFECTIVE COLORING is NP-hard, but that it remains NP-hard even if either χ_d or Δ^* is a constant with the smallest possible non-trivial value ($\chi_d \geq 2$ or $\Delta^* \geq 1$). To complement these negative results we provide a treewidth-based DP algorithm which runs in polynomial time if both χ_d and Δ^* are constant, not only for split graphs, but also for chordal graphs. This generalizes a previous algorithm of Havet et al. [24] on interval graphs.

We then go on to show that DEFECTIVE COLORING is also NP-hard when restricted to cographs. We note that this result is somewhat surprising since rel-

atively few natural problems are known to be hard for cographs. We complement this negative result in several ways. First, we show that DEFECTIVE COLORING becomes polynomially solvable on trivially perfect graphs, which form a large natural subclass of cographs. Second, we show that, unlike the case of split graphs, DEFECTIVE COLORING is in P on cographs if either χ_{d} or Δ^* is fixed. Both of these results are based on dynamic programming algorithms. Finally, by combining the previous two algorithms with known facts about the relation between χ_{d} and Δ^* we obtain a sub-exponential time algorithm for DEFECTIVE COLORING on cographs. We note that the existence of such an algorithm for split graphs is ruled out by our reductions, under the Exponential Time Hypothesis. Table 1 summarizes our results. For the reader's convenience, it also depicts an inclusion diagram for the graph classes that we mention.

2 Preliminaries and Definitions

We use standard graph theory terminology, see e.g. [16]. In particular, for a graph $G = (V, E)$ and $u \in V$ we use $N(u)$ to denote the set of neighbors of u, $N[u]$ denotes $N(u) \cup \{u\}$, and for $S \subseteq V$ we use $G[S]$ to denote the subgraph induced by the set S. A proper coloring of G with χ colors is a function $c : V \to \{1, \ldots, \chi\}$ such that for all $i \in \{1, \ldots, \chi\}$ the graph $G[c^{-1}(i)]$ is an independent set. We will focus on the following generalization of coloring:

Definition 1. *If $\chi_{\mathrm{d}}, \Delta^*$ are positive integers then a $(\chi_{\mathrm{d}}, \Delta^*)$-coloring of a graph $G = (V, E)$ is a function $c : V \to \{1, \ldots, \chi_{\mathrm{d}}\}$ such that for all $i \in \{1, \ldots, \chi_{\mathrm{d}}\}$ the maximum degree of $G[c^{-1}(i)]$ is at most Δ^*.*

We call the problem of deciding if a graph admits a $(\chi_{\mathrm{d}}, \Delta^*)$-coloring, for given parameters $\chi_{\mathrm{d}}, \Delta^*$, DEFECTIVE COLORING. For a graph G and a coloring function $c : V \to \mathbb{N}$ we say that the *deficiency* of a vertex u is $|N(u) \cap c^{-1}(c(u))|$, that is, the number of its neighbors with the same color. The deficiency of a color class i is defined as the maximum deficiency of any vertex colored with i.

We recall the following basic facts about DEFECTIVE COLORING:

Lemma 1. *([27]) For any $\chi_{\mathrm{d}}, \Delta^*$ and any graph $G = (V, E)$ with $\chi_{\mathrm{d}} \cdot \Delta^* \geq |V|$ we have that G admits a $(\chi_{\mathrm{d}}, \Delta^*)$-coloring.*

Lemma 2. *([27]) If G admits a $(\chi_{\mathrm{d}}, \Delta^*)$-coloring then $\omega(G) \leq \chi_{\mathrm{d}} \cdot (\Delta^* + 1)$.*

Let us now also give some quick reminders regarding the definitions of the graph classes we consider in this paper.

A graph $G = (V, E)$ is a *split* graph if $V = K \cup S$ where K induces a clique and S induces an independent set. A graph G is *chordal* if it does not contain any induced cycles of length four or more. It is well known that all split graphs are chordal; furthermore it is known that the class of chordal graphs contains the class of interval graphs, and that chordal graphs are perfect. For more information on these standard containments see [11].

A graph is a *cograph* if it is either a single vertex, or the disjoint union of two cographs, or the complete join of two cographs [33]. A graph is *trivially perfect* if in all induced subgraphs the maximum independent set is equal to the number of maximum cliques [21]. Trivially perfect graphs are exactly the cographs which are chordal [34], and hence are a subclass of cographs, which are a subclass of perfect graphs. We recall that GRAPH COLORING is polynomial-time solvable in all the mentioned graph classes, since it is polynomial-time solvable on perfect graphs [22], though of course for all these classes simpler and more efficient combinatorial algorithms are known.

We will also use the notion of treewidth for the definition of which we refer the reader to [9, 15].

3 NP-hardness on Cographs

In this section we establish that DEFECTIVE COLORING is already NP-hard on the very restricted class of cographs. To this end, we show a reduction from 4-PARTITION.

Definition 2. *In* 4-PARTITION *we are given a set A of $4n$ elements, a size function $s : A \to \mathbb{N}^+$ which assigns a value to each element, and a target integer B. We are asked if there exists a partition of A into n sets of four elements (quadruples), such that for each set the sum of its elements is exactly B.*

4-PARTITION has long been known to be strongly NP-hard, that is, NP-hard even if all values are polynomially bounded in n. In fact, the reduction given in [19] establishes the following, slightly stronger statement.

Theorem 1. 4-PARTITION *is strongly NP-complete if A is given to us partitioned into four sets of equal size A_1, A_2, A_3, A_4 and any valid solution is required to place exactly one element from each $A_i, i \in \{1, \ldots, 4\}$ in each quadruple.*

Theorem 2. DEFECTIVE COLORING *is NP-complete even when restricted to complete k-partite graphs. Therefore,* DEFECTIVE COLORING *is NP-complete on cographs.*

Proof. We start our reduction from an instance of 4-PARTITION where the set of elements A is partitioned into four equal-size sets as in Theorem 1. We first transform the instance by altering the sizes of all elements as follows: for each element $x \in A_i$ we set $s'(x) := s(x) + 5^i B + 5^5 n^2 B$ and we also set $B' := B + B \cdot \sum_{i=1}^4 5^i + 4 \cdot 5^5 n^2 B$. After this transformation our instance is "ordered", in the sense that all elements of A_{i+1} have strictly larger size than all elements of A_i. Furthermore, it is not hard to see that the answer to the problem did not change, as any quadruple that used one element from each A_i and summed up to B now sums up to B'. In addition, we observe that in the new instance the condition that exactly one element must be used from each set is imposed by the element sizes themselves: a quadruple that contains two or more elements

of A_4 will have sum strictly more than B', while one containing no elements of A_4 will have sum strictly less than B'. Similar reasoning can then be applied to A_3, A_2. We note that the element sizes now have the extra property that $s'(x) \in (B'/4 - 5B'/n^2, B'/4 + 5B'/n^2)$.

We now construct an instance of DEFECTIVE COLORING as follows. We set $\Delta^* = B'$ and $\chi_d = n$. To construct the graph G, for each element $x \in A_2 \cup A_3 \cup A_4$ we create an independent set of $s'(x)$ new vertices which we will call V_x. For each element $x \in A_1$ we construct two independent sets of $s'(x)$ new vertices each, which we will call V_x^1 and V_x^2. Finally, we turn the graph into a complete $5n$-partite graph, that is, we add all possible edges while retaining the property that all sets V_x and V_x^i remain independent.

Let us now argue for the correctness of the reduction. First, suppose that there exists a solution to our (modified) 4-PARTITION instance where each quadruple sums to B'. Number the quadruples arbitrarily from 1 to n and consider the i-th quadruple $(x_i^1, x_i^2, x_i^3, x_i^4)$ where we assume that for each $j \in \{1, \ldots, 4\}$ we have $x_i^j \in A_j$. Hence, $s'(x_i^1)$ is minimum among the sizes of the elements of the quadruple. We now use color i for all the vertices of the sets $V_{x_i^j}$ for $j \in \{2, 3, 4\}$ as well as the sets $V_{x_i^1}^1, V_{x_i^1}^2$. We continue in this way using a different color for each quadruple and thus color the whole graph (since the quadruples use all the elements of A). We observe that for any color i the vertices with maximum deficiency are those from $V_{x_i^1}^1$ and $V_{x_i^1}^2$, and all these vertices have deficiency exactly $x_i^1 + x_i^2 + x_i^3 + x_i^4 = B'$. Hence, this is a valid solution.

For the converse direction of the reduction, suppose we are given a (χ_d, Δ^*)-coloring of the graph we constructed. We first need to argue that such a coloring must have a very special structure. In particular, we will claim that in such a coloring each independent set V_x or V_x^i must be monochromatic. Towards this end we formulate a number of claims.[4]

Claim. Every color i is used on at most $5B'/4 + 25B'/n^2$ vertices.

Because of the previous claim, which states that no color appears too many times, we can also conclude that no color appears too few times.

Claim. Every color i is used on at least $5B'/4 - 50B'/n$ vertices.

Given the above bounds on the size of each color class we can now conclude that each color appears in exactly five independent sets V_x.

Claim. For each color i the graph induced by $c^{-1}(i)$ is complete 5-partite.

Claim. In any valid solution every maximal independent set of G is monochromatic.

We are now ready to complete the converse direction of the reduction. Consider the vertices of $c^{-1}(i)$, for some color i. By the previous sequence of claims

---

[4] The proofs of all claims are given in the appendix.

we know that they appear in and fully cover 5 independent sets V_x or V_x^i. We claim that for each $j \in \{2,3,4\}$ any color i is used in exactly one V_x with $x \in A_j$. This can be seen by considering the deficiency of the vertices of the smallest independent set where i appears. The deficiency of these vertices is equal to $x_i^1 + x_i^2 + x_i^3 + x_i^4$, which are the sizes of the four larger independent sets. By the construction of the modified 4-PARTITION instance, any quadruple that contains two elements of A_4 will have sum strictly greater than B'. Hence, these elements must be evenly partitioned among the color classes, and with similar reasoning the same follows for the elements of A_3, A_2.

We thus arrive at a situation where each color i appears in the independent sets $V_{x_i^4}, V_{x_i^3}, V_{x_i^2}$ as well as two of the "small" independent sets. Recall that all "small" independent sets were constructed in two copies of the same size V_x^1, V_x^2. We would now like to ensure that all color classes contain one small independent set of the form $V_{x_i^1}^1$. If we achieve this the argument will be complete: we construct the quadruple $(x_i^4, x_i^3, x_i^2, x_i^1)$ from the color class i, and the deficiency of the vertices of the remaining small independent set ensures that the sum of the elements of the quadruple is at most B'. By constructing n such quadruples we conclude that they all have sum exactly B', since the sum of all elements of the 4-PARTITION instance is (without loss of generality) exactly nB'.

To ensure that each color class contains an independent set V_x^1 we first observe that we can always exchange the colors of independent sets V_x^1 and V_x^2, since they are both of equal size (and monochromatic). Construct now an auxiliary graph with χ_{d} vertices, one for each color class and a directed edge for each $x \in A_1$. Specifically, if for $x \in A_1$ the independent set V_x^1 is colored i and the set V_x^2 is colored j we place a directed edge from i to j (note that this does not rule out the possibility of self-loops). In the auxiliary graph, each vertex that does not have a self-loop is incident on two directed edges. We would like all such vertices to end up having out-degree 1, because then each color class would contain an independent set of the form V_x^1. The main observation now is that in each weakly connected component that contains a vertex u with out-degree 0 there must also exist a vertex v of out-degree 2. Exchanging the colors of V_x^1 and V_x^2 corresponds to flipping the direction of an edge in the auxiliary graph. Hence, we can take a maximal directed path starting at v and flip all its edges, while maintaining a valid coloring of the original graph. This decreases the number of vertices with out-degree 0 and therefore repeating this process completes the proof. □

4 Polynomial Time Algorithm on Trivially Perfect Graphs

In this section, we complement the NP-completeness proof from Section 3 by giving a polynomial time algorithm for DEFECTIVE COLORING on the class of trivially perfect graphs, which form a large subclass of cographs. We will heavily rely on the following equivalent characterization of trivially perfect graphs given by Golumbic [21]:

Theorem 3. *A graph is trivially perfect if and only if it is the comparability graph of a rooted tree.*

In other words, for every trivially perfect graph G, there exists a rooted tree T such that making every vertex in the tree adjacent to all of its descendants yields a graph isomorphic to G. We refer to T as the *underlying rooted tree* of G. We recall that it is known how to obtain T from G in polynomial (in fact linear) time [34].

We are now ready to describe our algorithm. The following observation is one of its basic building blocks.

Lemma 3. *Let $G = (V, E)$ be a trivially perfect graph, T its underlying rooted tree, and $u \in V$ be an ancestor of $v \in V$ in T. Then $N[v] \subseteq N[u]$.*

Theorem 4. DEFECTIVE COLORING *can be solved in polynomial time on trivially perfect graphs.*

Proof. Given a trivially perfect graph $G = (V, E)$ with underlying rooted tree $T = (V, E')$ and two non-negative integers χ_d and Δ^*, we compute a coloring of G using at most χ_d colors and with deficiency at most Δ^* as follows. First, we partition the vertices of T (and therefore of G) into sets $V_1, \ldots, V_\ell$, where ℓ denotes the height of T, such that V_1 contains the leaves of T and, for every $2 \leq i \leq \ell$, V_i contains the leaves of $T \setminus (\bigcup_{j=1}^{i-1} V_j)$. Observe that each set V_i is an independent set in G. We now start our coloring by giving all vertices of V_1 color 1. Then, for every $2 \leq i \leq \ell$, we color the vertices of V_i by giving each of them the lowest color available, i.e., we color each vertex u with the lowest j such that u has at most Δ^* descendants colored j. If for some vertex no color is available, that is, its subtree contains at least $\Delta^* + 1$ vertices colored with each of the colors $\{1, \ldots, \chi_d\}$, then we return that G does not admit a (χ_d, Δ^*)-coloring.

This procedure can clearly be performed in polynomial time and, if it returns a solution, it uses at most χ_d colors. Furthermore, whenever the procedure uses color i on a vertex u it is guaranteed that u has deficiency at most Δ^* among currently colored vertices. Since any neighbor of u that is currently colored with i must be a descendant of u, by Lemma 3 this guarantees that the deficiency of all vertices remains at most Δ^* at all times.

It now only remains to prove that the algorithm concludes that G cannot be colored with χ_d colors and deficiency Δ^* only when no such coloring exists. For this we will rely on the following claim which states that any valid coloring can be "sorted".

Claim. If G admits a (χ_d, Δ^*)-coloring, then there exists a (χ_d, Δ^*)-coloring of G c such that, for every two vertices $u, v \in V(G)$, if v is a descendant of u, then $c(v) \leq c(u)$.

It follows from the previous claim that if a (χ_d, Δ^*)-coloring exists, then a sorted (χ_d, Δ^*)-coloring exists where ancestors always have colors at least as high as their descendants. We can now argue that our algorithm also produces a sorted coloring, with the extra property that whenever it sets $c(u) = i$ we

7

know that *any* sorted $(\chi_{\mathrm{d}}, \Delta^*)$-coloring of G must give color at least i to u. This can be shown by induction on i: it is clear for the vertices of V_1 to which the algorithm gives color 1; and if the algorithm assigns color i to u, then u has $\Delta^* + 1$ descendants which (by inductive hypothesis) must have color at least $i - 1$ in any valid sorted coloring of G. □

5 Algorithms on Cographs

In this section we present algorithms that can solve DEFECTIVE COLORING on cographs in polynomial time if either Δ^* or χ_{d} is bounded; both algorithms rely on dynamic programming. After presenting them we show how their combination can be used to obtain a sub-exponential time algorithm for DEFECTIVE COLORING on cographs.

5.1 Algorithm for Small Deficiency

We now present an algorithm that solves DEFECTIVE COLORING in polynomial time on cographs if Δ^* is bounded. Before we proceed, let us sketch the main ideas behind the algorithm. Given a $(\chi_{\mathrm{d}}, \Delta^*)$-coloring c of a graph G, we define the *type* of a color class i, as the pair of two integers (s_i, d_i) where $s_i := \min\{|c^{-1}(i)|, \Delta^* + 1\}$ and d_i is the maximum degree of $G[c^{-1}(i)]$. In other words, the type of a color class is characterized by its size (up to value $\Delta^* + 1$) and the maximum deficiency of any of its vertices. We observe that there are at most $(\Delta^* + 1)^2$ possible types in a valid $(\chi_{\mathrm{d}}, \Delta^*)$-coloring, because s_i only takes values in $\{1, \ldots, \Delta^* + 1\}$ and d_i in $\{0, \ldots, \Delta^*\}$.

We can now define the *signature* of a coloring c as a tuple which contains one element for every possible color type (s, d). This element is the number of color classes in c that have type (s, d), and hence is a number in $\{0, \ldots, \chi_{\mathrm{d}}\}$. We can conclude that there are at most $(\chi_{\mathrm{d}} + 1)^{(\Delta^* + 1)^2}$ possible signatures that a valid $(\chi_{\mathrm{d}}, \Delta^*)$-coloring can have. Our algorithm will maintain a binary table which states for each possible signature if the current graph admits a $(\chi_{\mathrm{d}}, \Delta^*)$-coloring with this signature. The obstacle now is to describe a procedure which, given two such tables for graphs G_1, G_2 is able to generate the table of admissible signatures for their union and their join.

Theorem 5. *There is an algorithm which decides if a cograph admits a $(\chi_{\mathrm{d}}, \Delta^*)$-coloring in time $O^* \left(\chi_{\mathrm{d}}^{O((\Delta^*)^4)} \right)$.*

5.2 Algorithm for Few Colors

In this section we provide an algorithm that solves DEFECTIVE COLORING in polynomial time on cographs if χ_{d} is bounded. The type of a color class i is defined in a similar manner as in the first paragraph of Section 5.1, with the only difference that the first coordinate of the output pair takes values in $\{0, \ldots, \Delta^* + 1\}$. The signature S of a coloring c is now a function $S :$

8

$\{1, \ldots, \chi_{\mathrm{d}}\} \to \{0, \ldots, \Delta^* + 1\} \times \{0, \ldots, \Delta^*\}$, which takes as input a color class and returns its type. Once again, we should maintain a table T of size less than $(\Delta^* + 2)^{2\chi_{\mathrm{d}}}$ for which $T(S) = 1$ if and only if there is a $(\chi_{\mathrm{d}}, \Delta^*)$-coloring of signature S for the current graph G. As in the previous section, we shall describe how to compute table T of a graph G which is the union or the join of two graphs G_1 and G_2 whose tables T_1 and T_2 are known.

Theorem 6. *There is an algorithm which decides if a cograph admits a $(\chi_{\mathrm{d}}, \Delta^*)$-coloring in time $O^*\big((\Delta^*)^{O(\chi_{\mathrm{d}})}\big)$.*

5.3 Sub-Exponential Time Algorithm

We now combine the algorithms of Sections 5.1 and 5.2 in order to obtain a sub-exponential time algorithm for cographs.

Theorem 7. DEFECTIVE COLORING *can be solved in time* $n^{O\left(n^{4/5}\right)}$ *on cographs.*

6 Split and Chordal Graphs

In this section we present the following results: first, we show that DEFECTIVE COLORING is hard on split graphs even when Δ^* is a fixed constant, as long as $\Delta^* \geq 1$; the problem is of course in P if $\Delta^* = 0$. Then, we show that DEFECTIVE COLORING is hard on split graphs even when χ_{d} is a fixed constant, as long as $\chi_{\mathrm{d}} \geq 2$; the problem is of course trivial if $\chi_{\mathrm{d}} = 1$. These results completely describe the complexity of the problem when one of the two relevant parameters is fixed. We then give a treewidth-based procedure through which we obtain a polynomial-time algorithm even on chordal graphs when $\chi_{\mathrm{d}}, \Delta^*$ are bounded (in fact, the algorithm is FPT parameterized by $\chi_{\mathrm{d}} + \Delta^*$). Hence these results give a complete picture of the complexity of the problem on chordal graphs: the problem is still hard when one of $\chi_{\mathrm{d}}, \Delta^*$ is bounded, but becomes easy if both are bounded.

Let us also remark that both of the reductions we present are linear. Hence, under the Exponential Time Hypothesis [26], they establish not only NP-hardness, but also unsolvability in time $2^{o(n)}$ for DEFECTIVE COLORING on split graphs, for constant values of χ_{d} or Δ^*. This is in contrast with the results of Section 5.3 on cographs.

6.1 Hardness for Bounded Deficiency

In this section we show that DEFECTIVE COLORING is NP-hard for any fixed value $\Delta^* \geq 1$. We first show hardness for $\Delta^* = 1$, then we tweak our reduction in order to make it work for larger Δ^*.

We will reduce from 3CNFSAT. Suppose we are given a CNF formula f where $X = \{x_1, \ldots, x_n\}$ are the variables and $C = \{c_1, \ldots, c_m\}$ are the clauses and each clause contains exactly 3 literals. We construct a split graph $G = (V, E)$,

where $\{U, Z\}$ is a partition of V with U inducing a clique of $4n$ vertices and Z inducing an independent set of $m + 4n$ vertices, such that having a satisfying assignment $s : X \to \{T, F\}$ for f implies a $(2n, 1)$-coloring $c : V \to \{1, \ldots, 2n\}$ for G and vice versa.

The construction is as follows. For every variable $x_i, i \in \{1, \ldots, n\}$ we construct a set of four vertices $U_i = \{u_i^A, u_i^B, u_i^C, u_i^D\}$ which are part of the clique vertices U (that is, for all $i \in \{1, \ldots, n\}$ and $k \in \{A, B, C, D\}$, vertices u_i^k are fully connected). We also construct four vertices $Z_i = \{z_i^A, z_i^B, z_i^C, z_i^D\}$ in the independent set Z. Furthermore, for each clause $c_j, j \in \{1, \ldots, m\}$ we construct a vertex v_j in the independent set. Last we add every edge between U and Z save for the following non-edges: for every $i \in \{1, \ldots, n\}, k \in \{A, B, C, D\}$, z_i^k does not connect to vertices $u_i^{k'}, k' \neq k$ and for every $i \in \{1, \ldots, n\}, j \in \{1, \ldots, m\}$, if clause c_j contains variable x_i then: if x_i appears positive then v_j does not connect to u_i^A, u_i^B, whereas if it appears negative then v_j does not connect to u_i^A, u_i^C. This completes the construction.

Lemma 4. *Given a satisfying assignment $s : X \to \{T, F\}$ for f we can always construct a $(2n, 1)$-coloring $c : V \to \{1, \ldots, 2n\}$.*

Proof. Let us first assign colors to the clique vertices. We are going to use two distinct colors for every quadruple U_i. The way we choose to color vertices in U_i should depend on the assignment $s(x_i)$: if $s(x_i) = T$ then $c(u_i^A) = c(u_i^B) = 2i - 1$ and $c(u_i^C) = c(u_i^D) = 2i$; if $s(x_i) = F$ then $c(u_i^A) = c(u_i^C) = 2i - i$ and $c(u_i^B) = c(u_i^D) = 2i$. Observe that we have consumed the entire supply of the $2n$ available colors on coloring U and for every color $l \in \{1, \ldots, 2n\}$ we have that $|c^{-1}(l) \cap U| = 2$.

In order to finish coloring the independent set Z, we can only reuse colors that have already appeared in U. If for some $z \in Z$ there exists a color l such that $c^{-1}(l) \cap N(z) = \emptyset$, that is if both vertices of color l in U are non-neighbors of z, then we can assign $c(z) = l$. Remember that a vertex in Z_i is a non-neighbor of exactly three vertices in U_i, thus two of them should be using the same color. Additionally, if s is a satisfying assignment for f, then for every c_j there is at least one satisfied literal, say $(\neg)x_i$ and by the construction of G and the assignment of colors on U we should be able once again to find two vertices in U_i having the same color that v_j does not connect to, these should be u_i^A and, depending on s, either u_i^B or u_i^C. $\qquad\square$

Lemma 5. *Given a $(2n, 1)$-coloring $c : V \to \{1, \ldots, 2n\}$ of G, we can produce a satisfying assignment $s : X \to \{T, F\}$ for f.*

Proof. First, observe that, since $\Delta^* = 1$, for any color $l \in \{1, \ldots, 2n\}$ we have that $|c^{-1}(l) \cap U| \leq 2$. Since there are at most $2n$ colors in use and $|U| = 4n$, that means that the color classes of c should induce a matching of size $2n$ in the clique. The above imply that for any $z \in Z$ with $c(z) = l$ there exist $u, u' \in U$ with $c(u) = c(u') = l$ which are non-neighbors of z.

We can now make the following claim:

Claim. For any $u, u' \in U$, if $c(u) = c(u')$ then there exists i such that $u, u' \in U_i$.

Proof. This is a consequence of vertices in Z_i having exactly three non-neighbors in U all of them belonging to U_i. More precisely, for any $k \in \{A, B, C, D\}$, $c(z_i^k) = l$ for some color l implies that $\exists k_1, k_2 \neq k$ such that $c(u_i^{k_1}) = c(u_i^{k_2})(= l)$. Similarly, the fact that $c(z_i^{k_1}) = l'$ for some color l' together with the fact that $|c^{-1}(l) \cap U| = 2$ gives us that $c(u_i^k) = c(u_i^{k'})(= l')$, where of course $u_i^k, u_i^{k'}$ are the only vertices of U colored l'. $\qquad\square$

The above claim directly provides the assignment: if $c(u_i^A) = c(u_i^B)$ then set $s(x_i) = T$, else $s(x_i) = F$.

Claim. The assignment s as described above satisfies f.

Proof. By construction, for all $j \in \{1, \ldots, m\}$, vertex v_j should be a non-neighbor to six vertices of U. At least two of them, call them u, u' should have the same color as v_j. From the previous claim, u, u' should belong to the same group U_i. By construction $u = u_i^A$ and $u' \in \{u_i^B, u_i^C\}$. Consider that $u' = u_i^B$ (similar arguments hold when $u' = u_i^C$). Since $c(u_i^A) = c(u_i^B)$, the assignment should set $s(x_i) = T$. Observe now that c_j, which by construction contains literal x_i, should be satisfied. $\qquad\square$

This concludes the proof. $\qquad\square$

Lemmata 4, 5 prove the following Theorem:

Theorem 8. DEFECTIVE COLORING *is NP-hard on split graphs for $\Delta^* = 1$.*

To show hardness for $\Delta^* \geq 2$, all we need to do is slightly change the above construction so that we are now forced to create bigger color classes. Namely, we add $2(\Delta^* - 1)n$ more vertices to U which we divide into $2n$ sets U_i^D and U_i^A and which we fully connect to each other and to previous vertices of U. We also remove vertices z_i^B, z_i^C from Z_i. Last, for $k \in \{A, D\}$, we connect vertices of U_i^k to all vertices in Z save for the following: U_i^D does not connect to z_i^A and U_i^A does not connect to z_i^D and to v_j if variable x_i appears in clause c_j.

Lemma 6. *Given a satisfying assignment s for f we can always construct a $(2n, \Delta^*)$-coloring c.*

Lemma 7. *Given a $(2n, \Delta^*)$-coloring c, we can produce a satisfying assignment s for f.*

The main theorem of this section follows from Lemmata 6,7 and Theorem 8.

Theorem 9. DEFECTIVE COLORING *is NP-hard on split graphs for any fixed $\Delta^* \geq 1$.*

6.2 Hardness for Bounded Number of Colors

Theorem 10. DEFECTIVE COLORING *is NP-complete on split graphs for every fixed value of $\chi_d \geq 2$.*

6.3 A Dynamic Programming Algorithm

In this section we present an algorithm which solves the problem efficiently on chordal graphs when χ_d and Δ^* are small. Our main tool is a treewidth-based procedure, as well as known connections between the maximum clique size and treewidth of chordal graphs.

Theorem 11. DEFECTIVE COLORING *can be solved in time* $(\chi_d \Delta^*)^{O(tw)} n^{O(1)}$ *on any graph G with n vertices if a tree decomposition of width tw of G is supplied with the input.*

We now recall the following theorem connecting $\omega(G)$ and $\mathrm{tw}(G)$ for chordal graphs.

Theorem 12. *([32, 8]) In chordal graphs $\omega(G) = tw(G) + 1$. Furthermore, an optimal tree decomposition of a chordal graph can be computed in polynomial time.*

Together with Lemma 2 this gives the following algorithm for chordal graphs.

Theorem 13. DEFECTIVE COLORING *can be solved in time* $(\chi_d \Delta^*)^{O(\chi_d \Delta^*)} n^{O(1)}$ *in chordal graphs.*

Proof. We use Theorem 12 to compute an optimal tree decomposition of the input graph and its maximum clique size. If $\omega(G) > \chi_d(\Delta^* + 1)$ then we can immediately reject by Lemma 2. Otherwise, we know that $\mathrm{tw}(G) \leq \chi_d(\Delta^* + 1)$ from Theorem 12, so we apply the algorithm of Theorem 11. □

7 Conclusions

Our results indicate that DEFECTIVE COLORING is significantly harder than GRAPH COLORING, even on classes where the latter is easily in P. Though we have completely characterized the complexity of the problem on split and chordal graphs, its tractability on interval and proper interval graphs remains an interesting open problem as already posed in [24].

Beyond this, the results of this paper point to several potential future directions. First, the algorithms we have given for cographs are both XP parameterized by χ_d or Δ^*. Is it possible to obtain FPT algorithms? On a related question, is it possible to obtain a faster sub-exponential time algorithm for DEFECTIVE COLORING on cographs? Second, is it possible to find other natural classes of graphs, beyond trivially perfect graphs, which are structured enough to make DEFECTIVE COLORING tractable? Finally, in this paper we have not considered the question of approximation algorithms. Though in general DEFECTIVE COLORING is likely to be quite hard to approximate (as a consequence of the hardness of GRAPH COLORING), it seems promising to also investigate this question in classes where GRAPH COLORING is in P.

References

1. Nirmala Achuthan, N. R. Achuthan, and M. Simanihuruk. On minimal triangle-free graphs with prescribed k-defective chromatic number. *Discrete Mathematics*, 311(13):1119–1127, 2011.

2. James A Andrews and Michael S Jacobson. On a generalization of chromatic number. *Congressus Numerantium*, 47:33–48, 1985.

3. Júlio Araújo, Jean-Claude Bermond, Frédéric Giroire, Frédéric Havet, Dorian Mazauric, and Remigiusz Modrzejewski. Weighted improper colouring. *J. Discrete Algorithms*, 16:53–66, 2012.

4. Dan Archdeacon. A note on defective colorings of graphs in surfaces. *Journal of Graph Theory*, 11(4):517–519, 1987.

5. Claudia Archetti, Nicola Bianchessi, Alain Hertz, Adrien Colombet, and François Gagnon. Directed weighted improper coloring for cellular channel allocation. *Discrete Applied Mathematics*, 182:46–60, 2015.

6. Jørgen Bang-Jensen and Magnús M. Halldórsson. Vertex coloring edge-weighted digraphs. *Inf. Process. Lett.*, 115(10):791–796, 2015.

7. Jean-Claude Bermond, Frédéric Havet, Florian Huc, and Cláudia Linhares Sales. Improper coloring of weighted grid and hexagonal graphs. *Discrete Math., Alg. and Appl.*, 2(3):395–412, 2010.

8. Hans L. Bodlaender. A partial k-arboretum of graphs with bounded treewidth. *Theor. Comput. Sci.*, 209(1-2):1–45, 1998.

9. Hans L. Bodlaender and Arie M. C. A. Koster. Combinatorial optimization on graphs of bounded treewidth. *Comput. J.*, 51(3):255–269, 2008.

10. Oleg V. Borodin, Alexandr V. Kostochka, and Matthew Yancey. On 1-improper 2-coloring of sparse graphs. *Discrete Mathematics*, 313(22):2638–2649, 2013.

11. Andreas Brandstädt, Van Bang Le, and Jeremy P Spinrad. *Graph classes: a survey*. SIAM, 1999.

12. I. Choi and L. Esperet. Improper coloring of graphs on surfaces. *ArXiv e-prints*, March 2016.

13. L. J. Cowen, R. H. Cowen, and D. R. Woodall. Defective colorings of graphs in surfaces: Partitions into subgraphs of bounded valency. *Journal of Graph Theory*, 10(2):187–195, 1986.

14. Lenore Cowen, Wayne Goddard, and C. Esther Jesurum. Defective coloring revisited. *Journal of Graph Theory*, 24(3):205–219, 1997.

15. Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015.

16. Reinhard Diestel. *Graph Theory, 4th Edition*, volume 173 of *Graduate texts in mathematics*. Springer, 2012.

17. Marietjie Frick. A survey of (m, k)-colorings. *Annals of Discrete Mathematics*, 55:45–57, 1993.

18. Marietjie Frick and Michael A. Henning. Extremal results on defective colorings of graphs. *Discrete Mathematics*, 126(1-3):151–158, 1994.

19. Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1979.

20. Wayne Goddard and Honghai Xu. Fractional, circular, and defective coloring of series-parallel graphs. *Journal of Graph Theory*, 81(2):146–153, 2016.

21. Martin Charles Golumbic. Trivially perfect graphs. *Discrete Mathematics*, 24(1):105–107, 1978.

22. Martin Grötschel, László Lovász, and Alexander Schrijver. *Geometric algorithms and combinatorial optimization.* Springer4060 XII, 362 S (Berlin [ua]), 1988.

23. Bjarki Agust Gudmundsson, Tómas Ken Magnússon, and Björn Orri Sæmundsson. Bounds and fixed-parameter algorithms for weighted improper coloring. *Electr. Notes Theor. Comput. Sci.*, 322:181–195, 2016.

24. Frédéric Havet, Ross J. Kang, and Jean-Sébastien Sereni. Improper coloring of unit disk graphs. *Networks*, 54(3):150–164, 2009.

25. Frédéric Havet and Jean-Sébastien Sereni. Improper choosability of graphs and maximum average degree. *Journal of Graph Theory*, 52(3):181–199, 2006.

26. Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity? *J. Comput. Syst. Sci.*, 63(4):512–530, 2001.

27. Ross J. Kang. *Improper colourings of graphs.* PhD thesis, University of Oxford, UK, 2008.

28. Ross J. Kang and Colin McDiarmid. The t-improper chromatic number of random graphs. *Combinatorics, Probability & Computing*, 19(1):87–98, 2010.

29. Ross J. Kang, Tobias Müller, and Jean-Sébastien Sereni. Improper colouring of (random) unit disk graphs. *Discrete Mathematics*, 308(8):1438–1454, 2008.

30. Jaehoon Kim, Alexandr V. Kostochka, and Xuding Zhu. Improper coloring of sparse graphs with a given girth, I: (0, 1)-colorings of triangle-free graphs. *Eur. J. Comb.*, 42:26–48, 2014.

31. Jaehoon Kim, Alexandr V. Kostochka, and Xuding Zhu. Improper coloring of sparse graphs with a given girth, II: constructions. *Journal of Graph Theory*, 81(4):403–413, 2016.

32. Neil Robertson and Paul D. Seymour. Graph minors. II. algorithmic aspects of tree-width. *J. Algorithms*, 7(3):309–322, 1986.

33. Dieter Seinsche. On a property of the class of n-colorable graphs. *Journal of Combinatorial Theory, Series B*, 16(2):191–193, 1974.

34. Jing-Ho Yan, Jer-Jeong Chen, and Gerard J. Chang. Quasi-threshold graphs. *Discrete Applied Mathematics*, 69(3):247–255, 1996.

A   Omitted Proofs

A.1   Proofs of Lemmata 1 and 2

Proof (Lemma 1). Partition V arbitrarily into χ_d sets of size at most $\lceil |V|/\chi_d \rceil$ and color each set with a different color. The maximum deficiency of any vertex is at most $\left\lceil \frac{|V|}{\chi_d} \right\rceil - 1 \le \frac{|V|}{\chi_d} \le \Delta^*$. $\qquad\square$

Proof (Lemma 2). For the sake of contradiction, assume that G has a clique of size $\chi_d \cdot (\Delta^* + 1) + 1$, then any coloring of G with χ_d colors must give the same color to strictly more than $\Delta^* + 1$ vertices of the clique, which implies that these vertices have deficiency at least $\Delta^* + 1$. $\qquad\square$

A.2   Full Proof of Theorem 1

Proof (Theorem 1). We start our reduction from an instance of 4-PARTITION where the set of elements A is partitioned into four equal-size sets as in Theorem 1. We first transform the instance by altering the sizes of all elements as follows: for each element $x \in A_i$ we set $s'(x) := s(x) + 5^i B + 5^5 n^2 B$ and we also set $B' := B + B \cdot \sum_{i=1}^{4} 5^i + 4 \cdot 5^5 n^2 B$. After this transformation our instance is "ordered", in the sense that all elements of A_{i+1} have strictly larger size than all elements of A_i. Furthermore, it is not hard to see that the answer to the problem did not change, as any quadruple that used one element from each A_i and summed up to B now sums up to B'. In addition, we observe that in the new instance the condition that exactly one element must be used from each set is imposed by the element sizes themselves: a quadruple that contains two or more elements of A_4 will have sum strictly more than B', while one containing no elements of A_4 will have sum strictly less than B'. Similar reasoning can then be applied to A_3, A_2. We note that the element sizes now have the extra property that $s'(x) \in (B'/4 - 5B'/n^2, B'/4 + 5B'/n^2)$.

We now construct an instance of DEFECTIVE COLORING as follows. We set $\Delta^* = B'$ and $\chi_d = n$. To construct the graph G, for each element $x \in A_2 \cup A_3 \cup A_4$ we create an independent set of $s'(x)$ new vertices which we will call V_x. For each element $x \in A_1$ we construct two independent sets of $s'(x)$ new vertices each, which we will call V_x^1 and V_x^2. Finally, we turn the graph into a complete $5n$-partite graph, that is, we add all possible edges while retaining the property that all sets V_x and V_x^i remain independent.

Let us now argue for the correctness of the reduction. First, suppose that there exists a solution to our (modified) 4-PARTITION instance where each quadruple sums to B'. Number the quadruples arbitrarily from 1 to n and consider the i-th quadruple $(x_i^1, x_i^2, x_i^3, x_i^4)$ where we assume that for each $j \in \{1, \ldots, 4\}$ we have $x_i^j \in A_j$. Hence, $s'(x_i^1)$ is minimum among the sizes of the elements of the quadruple. We now use color i for all the vertices of the sets $V_{x_i^j}$ for $j \in \{2, 3, 4\}$ as well as the sets $V_{x_i^1}^1, V_{x_i^1}^2$. We continue in this way using a different color for each quadruple and thus color the whole graph (since the quadruples use all the

elements of A). We observe that for any color i the vertices with maximum deficiency are those from $V^1_{x^1_i}$ and $V^2_{x^1_i}$, and all these vertices have deficiency exactly $x^1_i + x^2_i + x^3_i + x^4_i = B'$. Hence, this is a valid solution.

For the converse direction of the reduction, suppose we are given a $(\chi_\mathrm{d}, \Delta^*)$-coloring of the graph we constructed. We first need to argue that such a coloring must have a very special structure. In particular, we will claim that in such a coloring each independent set V_x or V^i_x must be monochromatic. Towards this end we formulate a number of claims.

Claim. Every color i is used on at most $5B'/4 + 25B'/n^2$ vertices.

Proof. We will assume that i is used at least $5B'/4 + 25B'/n^2 + 1$ times and obtain a contradiction. Since the size of the largest independent set V_x is at most $B'/4 + 5B'/n^2$ we know that color i must appear in at least six different independent sets. Among the independent sets in which i appears let V_x be the one in which it appears the minimum number of times. The deficiency of a vertex colored with i in this set is at least $\frac{5}{6}|c^{-1}(i)| \geq \frac{25B'}{24} > B' = \Delta^*$. $\square$

Because of the previous claim, which states that no color appears too many times, we can also conclude that no color appears too few times.

Claim. Every color i is used on at least $5B'/4 - 50B'/n$ vertices.

Proof. First, note that $|V| \geq 5nB'/4 - 25B'/n$ because we have created $5n$ independent sets each of which has size more than $B'/4 - 5B'/n^2$. By the previous claim any color $j \neq i$ has $|c^{-1}(j)| \leq 5B'/4 + 25B'/n^2$. Therefore $\sum_{j \neq i}|c^{-1}(j)| \leq (n-1)(5B'/4 + 25B'/n^2)$. We have $|c^{-1}(i)| = |V| - \sum_{j \neq i}|c^{-1}(j)| \geq \frac{5nB'}{4} - \frac{25B'}{n} - (n-1)\frac{5B'}{4} - (n-1)\frac{25B'}{n^2} = \frac{5B'}{4} - \frac{50B'}{n} + \frac{25B'}{n^2} > \frac{5B'}{4} - \frac{50B'}{n}$. $\square$

Given the above bounds on the size of each color class we can now conclude that each color appears in exactly five independent sets V_x.

Claim. For each color i the graph induced by $c^{-1}(i)$ is complete 5-partite.

Proof. First, observe that by the previous claim, there must exist at least 5 sets V_x or V^i_x that intersect $c^{-1}(i)$, because $|c^{-1}(i)| \geq 5B'/4 - O(B'/n)$ while the size of each such set is at most $B'/4 + O(B'/n^2)$; therefore, the size of any four sets is strictly smaller than $|c^{-1}(i)|$ (assuming of course that n is sufficiently large). Suppose now that $c^{-1}(i)$ intersects 6 different sets, and consider the independent set V_x on which color i appears at least once but a minimum number of times. A vertex colored i in this set will have deficiency at least $\frac{5}{6}(\frac{5B'}{4} - \frac{50B'}{n}) = \frac{25B'}{24} - O(\frac{B'}{n})$, which is strictly greater than B' for sufficiently large n. Hence, color i appears in exactly 5 independent sets. $\square$

Claim. In any valid solution every maximal independent set of G is monochromatic.

16

Proof. Consider color i, which by the previous claim appears in exactly 5 independent sets. Suppose that one of these is not monochromatic, say colors i, j appear in V_x. If i appears in at most $|V_x|/2$ vertices of V_x then we obtain a contradiction as follows: the total number of times i is used in the graph is at most $|c^{-1}(i)| \leq 4(\frac{B'}{4} + \frac{5B'}{n^2}) + \frac{1}{2}(\frac{B'}{4} + \frac{5B'}{n^2})$, where the first term uses the general upper bound on the size of all other independent sets where i appears, and the second term uses the same upper bound on $|V_x|$. Thus, $|c^{-1}(i)| \leq \frac{9B'}{8} + O(\frac{B'}{n^2})$ which is strictly smaller than $\frac{5B'}{4} - \frac{50B'}{n}$, the minimum number of times that i must be used (for sufficiently large n). We thus conclude that i must use strictly more than half of the vertices of V_x. But then we can repeat the same argument for color j, which is now the minority color in V_x. Hence we conclude that the independent sets where i is used are monochromatic. □

We are now ready to complete the converse direction of the reduction. Consider the vertices of $c^{-1}(i)$, for some color i. By the previous sequence of claims we know that they appear in and fully cover 5 independent sets V_x or V_x^i. We claim that for each $j \in \{2, 3, 4\}$ any color i is used in exactly one V_x with $x \in A_j$. This can be seen by considering the deficiency of the vertices of the smallest independent set where i appears. The deficiency of these vertices is equal to $x_i^1 + x_i^2 + x_i^3 + x_i^4$, which are the sizes of the four larger independent sets. By the construction of the modified 4-PARTITION instance, any quadruple that contains two elements of A_4 will have sum strictly greater than B'. Hence, these elements must be evenly partitioned among the color classes, and with similar reasoning the same follows for the elements of A_3, A_2.

We thus arrive at a situation where each color i appears in the independent sets $V_{x_i^4}, V_{x_i^3}, V_{x_i^2}$ as well as two of the "small" independent sets. Recall that all "small" independent sets were constructed in two copies of the same size V_x^1, V_x^2. We would now like to ensure that all color classes contain one small independent set of the form $V_{x_i^1}^1$. If we achieve this the argument will be complete: we construct the quadruple $(x_i^4, x_i^3, x_i^2, x_i^1)$ from the color class i, and the deficiency of the vertices of the remaining small independent set ensures that the sum of the elements of the quadruple is at most B'. By constructing n such quadruples we conclude that they all have sum exactly B', since the sum of all elements of the 4-PARTITION instance is (without loss of generality) exactly nB'.

To ensure that each color class contains an independent set V_x^1 we first observe that we can always exchange the colors of independent sets V_x^1 and V_x^2, since they are both of equal size (and monochromatic). Construct now an auxiliary graph with χ_d vertices, one for each color class and a directed edge for each $x \in A_1$. Specifically, if for $x \in A_1$ the independent set V_x^1 is colored i and the set V_x^2 is colored j we place a directed edge from i to j (note that this does not rule out the possibility of self-loops). In the auxiliary graph, each vertex that does not have a self-loop is incident on two directed edges. We would like all such vertices to end up having out-degree 1, because then each color class would contain an independent set of the form V_x^1. The main observation now is that in each weakly connected component that contains a vertex u with out-degree 0 there must also exist a vertex v of out-degree 2. Exchanging the colors of V_x^1 and

V_x^2 corresponds to flipping the direction of an edge in the auxiliary graph. Hence, we can take a maximal directed path starting at v and flip all its edges, while maintaining a valid coloring of the original graph. This decreases the number of vertices with out-degree 0 and therefore repeating this process completes the proof. □

A.3 Proof of Lemma 3

Proof (Lemma 3). Any vertex $w \in N[v]$ must be either a descendant of v, in which case it is also a descendant of u and $w \in N[u]$, or another ancestor of v. However, because T is a tree, if w is an ancestor of v, then w is either an ancestor or a descendant of u. □

A.4 Full Proof of Theorem 4

Proof (Theorem 4). Given a trivially perfect graph $G = (V, E)$ with underlying rooted tree $T = (V, E')$ and two non-negative integers χ_d and Δ^*, we compute a coloring of G using at most χ_d colors and with deficiency at most Δ^* as follows. First, we partition the vertices of T (and therefore of G) into sets $V_1, \ldots, V_\ell$, where ℓ denotes the height of T, such that V_1 contains the leaves of T and, for every $2 \le i \le \ell$, V_i contains the leaves of $T \setminus (\bigcup_{j=1}^{i-1} V_j)$. Observe that each set V_i is an independent set in G. We now start our coloring by giving all vertices of V_1 color 1. Then, for every $2 \le i \le \ell$, we color the vertices of V_i by giving each of them the lowest color available, i.e., we color each vertex u with the lowest j such that u has at most Δ^* descendants colored j. If for some vertex no color is available, that is, its subtree contains at least $\Delta^* + 1$ vertices colored with each of the colors $\{1, \ldots, \chi_d\}$, then we return that G does not admit a (χ_d, Δ^*)-coloring.

This procedure can clearly be performed in polynomial time and, if it returns a solution, it uses at most χ_d colors. Furthermore, whenever the procedure uses color i on a vertex u it is guaranteed that u has deficiency at most Δ^* among currently colored vertices. Since any neighbor of u that is currently colored with i must be a descendant of u, by Lemma 3 this guarantees that the deficiency of all vertices remains at most Δ^* at all times.

It now only remains to prove that the algorithm concludes that G cannot be colored with χ_d colors and deficiency Δ^* only when no such coloring exists. For this we will rely on the following claim which states that any valid coloring can be "sorted".

Claim. If G admits a (χ_d, Δ^*)-coloring, then there exists a (χ_d, Δ^*)-coloring of G c such that, for every two vertices $u, v \in V(G)$, if v is a descendant of u, then $c(v) \le c(u)$.

Proof. Let us consider an arbitrary (χ_d, Δ^*)-coloring $c^* : V(G) \to \{1, \ldots, \chi_d\}$ of G. We describe a process which, as long as there exist $u, v \in V$ with u an ancestor of v and $c^*(u) < c^*(v)$ transforms c^* to another valid coloring which is closer to having the desired property. So, suppose that such a pair u, v exists,

and furthermore, if many such pairs exist, suppose that we select a pair where u is as close to the root of T as possible. As a result, we can assume that no ancestor u' of u has color $c^*(u)$, because otherwise we would have started with the pair u', v.

We will now consider two cases. Assume first that there exists a vertex x such that $c^*(x) = c^*(v)$ and x is an ancestor of u. We claim that swapping the colors of u and v yields a new coloring of G with deficiency at most Δ^*. The only affected vertices are those colored $c^*(u)$ or $c^*(v)$. Regarding color $c^*(u)$, because by Lemma 3 $N[v] \subseteq N[u]$ and color $c^*(u)$ was moved from u to v, the deficiency of every vertex colored $c^*(u)$ in $V \setminus \{u, v\}$ is at most as high as it was before, and the deficiency of v is at most as high as the deficiency of u in c^*. Regarding color $c^*(v)$ we observe that the deficiency of vertex x remains unchanged, since both u, v are its neighbors, and the same is true for all ancestors of x. Since the deficiency of x is at most Δ^*, by Lemma 3, the deficiency of every descendant of x colored with $c^*(v)$ is also at most Δ^*.

For the remaining case, suppose that no ancestor of u uses color $c^*(v)$. Recall that we have also assumed that no ancestor of u uses color $c^*(u)$. We therefore transform the coloring as follows: in the subtree rooted at u we exchange colors $c^*(u)$ and $c^*(v)$ (that is, we color all vertices currently colored with $c^*(u)$ with $c^*(v)$ and vice-versa). Because no ancestor of u uses either of these two colors, this exchange does not affect the deficiency of any vertex.

We can now repeat this procedure as follows: as long as there is a conflicting pair u, v, with u an ancestor of v and $c^*(u) < c^*(v)$ we select such a pair with u as close to the root as possible and, if there are several such pairs, we select the one with maximum $c^*(v)$. We perform the transformation explained above on this pair and then repeat. It is not hard to see that every vertex will be used at most once as an ancestor in this transformation, because after the transformation it will have the highest color in its subtree. Hence we will eventually obtain the claimed property. □

It follows from the previous claim that if a (χ_d, Δ^*)-coloring exists, then a sorted (χ_d, Δ^*)-coloring exists where ancestors always have colors at least as high as their descendants. We can now argue that our algorithm also produces a sorted coloring, with the extra property that whenever it sets $c(u) = i$ we know that *any* sorted (χ_d, Δ^*)-coloring of G must give color at least i to u. This can be shown by induction on i: it is clear for the vertices of V_1 to which the algorithm gives color 1; and if the algorithm assigns color i to u, then u has $\Delta^* + 1$ descendants which (by inductive hypothesis) must have color at least $i - 1$ in any valid sorted coloring of G. □

A.5 Proof of Theorem 5

Proof (Theorem 5). We use the ideas sketched above. Specifically, we say that a coloring signature S is a function $\{1, \ldots, \Delta^* + 1\} \times \{0, \ldots, \Delta^*\} \to \{0, \ldots, \chi_d\}$ and a coloring c has signature S if for any $(s, d) \in \{1, \ldots, \Delta^* + 1\} \times \{0, \ldots, \Delta^*\}$ the number of color classes with type (s, d) in c is $S((s, d))$. Our algorithm will

maintain a binary table T with the property that, for S a possible coloring signature we have $T(S) = 1$ if and only if there exists a (χ_d, Δ^*)-coloring of G with signature S. The size of T is therefore at most $(\chi_d + 1)^{(\Delta^*+1)^2}$.

It is not hard to see how to compute T if G consists of a single vertex: the only color class then has type $(1, 0)$, so the only possible signature is the one that sets $S((1, 0)) = 1$ and $S((s, d)) = 0$ otherwise.

Now, suppose that G is either the union or the join of two graphs G_1, G_2 for which our algorithm has already calculated the corresponding tables T_1, T_2. We will use the fact that for any valid (χ_d, Δ^*)-coloring c of G with signature S, its restrictions to G_1, G_2 are also valid (χ_d, Δ^*)-colorings. If these restrictions have signatures S_1, S_2 it must then be the case that $T_1(S_1) = T_2(S_2) = 1$. It follows that in order to compute all the signatures for which we must have $T(S) = 1$ it suffices to consider all pairs of signatures S_1, S_2 such that $T_1(S_1) = T_2(S_2) = 1$ and decide if it is possible to have a coloring of G with signature S whose restrictions to G_1, G_2 have signatures S_1, S_2.

Given two signatures S_1, S_2 such that $T_1(S_1) = T_2(S_2) = 1$ we would therefore like to generate all possible signatures S for colorings c of G such that S_1, S_2 represent the restriction of c to G_1, G_2. Every color class of c will either consist of vertices of only one subgraph G_1 or G_2, or it will be the result of merging a color class of G_1 with a color class of G_2. Our algorithm will enumerate all possible merging combinations between color classes of G_1 and G_2.

Let us now explain how we enumerate all merging possibilities. Let c_1, c_2 be (χ_d, Δ^*)-colorings of G_1, G_2 with signatures S_1, S_2 respectively. If G is the join of G_1, G_2 we say that type (s_1, d_1) is mergeable with type (s_2, d_2) if $s_1 + d_2 \leq \Delta^*$ and $s_2 + d_1 \leq \Delta^*$. If G is the union of G_1, G_2 we say that any pair of types is mergeable. Furthermore, if G is the join of G_1, G_2 and $(s_1, d_1), (s_2, d_2)$ are mergeable types, we say that they merge into type $(\min\{s_1 + s_2, \Delta^* + 1\}, \max\{d_1 + s_2, d_2 + s_1\})$. If G is the union of G_1, G_2 we say that types (s_1, d_1) and (s_2, d_2) merge into type $(\min\{s_1 + s_2, \Delta^* + 1\}, \max\{d_1, d_2\})$. The intuition behind these definitions is that a color class i of c_1 is mergeable with a color class j of c_2 if we can use a single color for $c_1^{-1}(i) \cup c_2^{-1}(j)$ in G, and the type of this color class is the type into which the types of i, j merge.

We now construct a bipartite graph $G'(A_1, A_2, E')$ that will help us enumerate all merging combinations. The graph consists of $(\Delta^* + 1)^2$ vertices on each side, each corresponding to a type. We place an edge between two vertices if their corresponding types are mergeable (so if G is a union of G_1, G_2 then G' is a complete bipartite graph). We also give a weight to each vertex as follows: if $u \in A_i$ corresponds to type (s, d) we set $w(u) = S_i((s, d))$. In words, the weight of a vertex that represents a type is the number of color classes of that type in the coloring of the subgraphs.

We will now enumerate all weighted matchings of G', where a weighted matching is an assignment of weights to E' such that for all vertices $u \in A_1 \cup A_2$ we have $\sum_{v \in N(u)} w((u, v)) \leq w(u)$. It is not hard to see that the total number of valid weighted matchings is at most $(\chi_d + 1)^{(\Delta^*+1)^4}$, since every edge must

receive in weight $\{0, \ldots, \chi_d\}$ and there are at most $(\Delta^* + 1)^4$ edges. This is the step that dominates the running time of our algorithm.

For each of the enumerated matchings of G' we can now calculate a signature S of a coloring of G. For each type (s, d) let $E_{(s,d)} \subseteq E'$ be the set of edges of G' whose endpoints merge into type (s, d). Let $u_i \in A_i$ be the vertices corresponding to type (s, d). We have $S((s, d)) = \sum_{i=1,2}(w(u_i) - \sum_{v \in N(u_i)} w(u_i, v)) + \sum_{e \in E_{(s,d)}} w(e)$. We now check that the signature we computed refers to a coloring with at most χ_d colors, that is, if $\sum S((s, d)) \leq \chi_d$, where $s \in \{1, \ldots, \Delta^*+1\}$ and $d \in \{0, \ldots, \Delta^*\}$. In this case we set $T(S) = 1$. The observation that completes the proof is that for all valid colorings c of G with signature S such that the restriction of c to G_1, G_2 has signatures S_1, S_2 there must exist a weighted matching for which the above procedure finds the signature S. Hence, by examining all pairs of feasible signatures S_1, S_2 we will discover all feasible signatures of G. □

A.6 Proof of Theorem 6

Proof (Theorem 6). The base case is when we introduce a single vertex u to the graph G. In this case, any coloring of u is valid, so for all $i \in \{1, \ldots, \chi_d\}$ we define a signature S_i such that $S_i(i) = (1, 0)$ and $S_i(j) = (0, 0)$ when $i \neq j$. Last, $T(S) = 1$ if and only if $S = S_i$ for any i.

Now, suppose that G is either the union or the join of two graphs G_1, G_2 for which we have already calculated their corresponding tables. Once again we just need to consider all pairs of signatures S_1, S_2 such that $T_1(S_1) = T_2(S_2) = 1$ and decide if we can have a coloring of G with signature S whose restrictions to G_1, G_2 have signatures S_1, S_2. Let S_1, S_2 be one such pair of signatures, for which $S_j(i) = (s_j^i, d_j^i)$, $j = 1, 2$. We examine the cases of union and join separately.

Let us start with the case that G is the union of G_1, G_2. Define S such that for any i, $S(i) = (\min\{s_1^i + s_2^i, \Delta^* + 1\}, \max\{d_1^i, d_2^i\})$ and set $T(S) = 1$.

The case where G is the join of G_1, G_2 is a little more complicated since we first need to check if, given two precolored graphs the outcome of their join is valid, that for all colors i, the maximum degree of $G[i]$ remains at most Δ^*. This corresponds to checking for all colors i whether $d = \max\{s_1^i + d_2^i, s_2^i + d_1^j\} \leq \Delta^*$. Given that the above is true, we define S such that for any i, $S(i) = (\min\{s_1^i + s_2^i, \Delta^* + 1\}, d)$ and set $S(T) = 1$.

The algorithm considers all pairs of elements of T_1, T_2, so it runs in time dominated by $|T_i|^2 = O^*\left((\Delta^*)^{O(\chi_d)}\right)$. □

A.7 Proof of Theorem 7

Proof (Theorem 7). First, we remind the reader that, from Lemma 1, if $\Delta^* \cdot \chi_d \geq n$ then the answer is trivially yes. Thus the interesting case is when $\Delta^* \cdot \chi_d < n$. Note that we also trivially have that $\Delta^*, \chi_d \leq n$.

If $\Delta^* \leq \sqrt[5]{n}$, then the algorithm of Section 5.1 runs in $O^*\left(\chi_d^{O((\Delta^*)^4)}\right) = n^{O\left(n^{4/5}\right)}$ time.

If $\Delta^* > \sqrt[5]{n}$, then $\chi_d < n/\Delta^* < n^{\frac{4}{5}}$. In this case, the algorithm of Section 5.2 runs in $O^*\left((\Delta^*)^{O(\chi_d)}\right) = n^{O\left(n^{4/5}\right)}$ time. $\qquad \square$

A.8 Proof of Theorem 9 for $\Delta^* \geq 2$

Proof (Lemma 6). The assignment of colors on old vertices remains the same; for $u \in U_i^D$ set $c(u) = 2i$, whereas for $u \in U_i^A$, set $c(u) = 2i - 1$. Again, having a satisfying assignment for f means that literal $(\neg)x_i$ which appears in c_j is set to true by s, so with similar arguments as above we can set $c(v_j) = 2i - 1$. $\quad \square$

Proof (Lemma 7). For the other direction, as in the case $\Delta^* = 1$, we can derive now that for any color l, $|c^{-1}(l) \cap U| = \Delta^* + 1$. Having $c(z_i^D) = l$ for some color l means that at least $\Delta^* + 1$ vertices among $U_i^A \cup \{u_i^A, u_i^B, u_i^C\}$ also have color l, where $|U_i^A \cup \{u_i^A, u_i^B, u_i^C\}| = \Delta^* + 2$. This implies the following two properties:

Property 1. $\forall u$, $c(u) = l$ implies $u \in U_i^A \cup \{u_i^A, u_i^B, u_i^C\}$.
Property 2. At least two vertices out of $\{u_i^A, u_i^B, u_i^C\}$ have color l.

Claim. $c(u_i^B) \neq c(u_i^C)$

Proof. Since $c(z_i^A) = l'$ for some color l', that means that at least $\Delta^* + 1$ vertices among $U_i^D \cup \{u_i^B, u_i^C, u_i^D\}$ are given the same color l'. Observe that $\exists u \in c^{-1}(l') \cap (U_i^D \cup \{u_i^D\})$ and from Property 1, $l' \neq l$. Assuming $c(u_i^B) = c(u_i^C)$ implies $l = l'$, a contradiction. $\qquad \square$

Thus, from Property 2 and the above claim, either $U_i^A \cup \{u_i^A, u_i^B\}$ have the same color l (and $U_i^D \cup \{u_i^C, u_i^D\}$ have color l') or $U_i^A \cup \{u_i^A, u_i^C\}$ have color l (and $U_i^D \cup \{u_i^B, u_i^D\}$ have color l'). The assignment again depends on whether $c(u_i^A) = c(u_i^B)$ and the correctness follows similar ideas to the case $\Delta^* = 1$. $\quad \square$

A.9 Proof of Theorem 10

Proof (Theorem 10). We reduce from the problem 3-SET SPLITTING, which takes as input a set of elements U, called the *universe*, and a family $\mathcal{F}$ of subsets of U of size exactly 3, and asks whether there is a partition (U_1, U_2) of U such that, for every set $S \in \mathcal{F}$, we have $S \cap U_1 \neq \emptyset$ and $S \cap U_2 \neq \emptyset$. This problem is well-known to be NP-complete [19]. Given an instance $(U, \mathcal{F})$ of 3-SET SPLITTING and a positive integer χ_d, we build a split graph $G = (V, E)$ such that $V = C_1 \cup C_2 \cup C^* \cup I \cup Z_1 \cup Z_2$, with $|C_1| = |C_2| = |\mathcal{F}|$, $|C^*| = (\chi_d - 2) \cdot (|\mathcal{F}| + 2)$, $|I| = |U|$ and $|Z_1| = |Z_2| = \chi_d \cdot (|\mathcal{F}| + 2)$. We proceed by making all the vertices of $C_1 \cup C_2$ pairwise adjacent. We then associate each set S of $\mathcal{F}$ with two vertices v_S^1 and v_S^2 of C_1 and C_2 respectively, and every element x of U with a vertex w_x of I. For every pair $x \in U, S \in \mathcal{F}$, we make w_x adjacent to v_S^1 and v_S^2 if and only if $x \in S$. We complete our construction by making all the vertices of C_i adjacent to all the vertices of Z_i for $i \in \{1, 2\}$, and all the vertices of C^* adjacent to every other vertex in V. Observe that the graph we constructed is split, since $C_1 \cup C_2 \cup C^*$ induces a clique and $I \cup Z_1 \cup Z_2$ induces an independent

set. We now claim that there exists a partition (U_1, U_2) of U as described above if and only if G can be colored with at most χ_d colors and deficiency at most $\Delta^* = |\mathcal{F}| + 1$.

For the forward direction, it suffices to observe that coloring every vertex of $C_1 \cup Z_2 \cup U_1$ with color 1, every vertex of $C_2 \cup Z_1 \cup U_2$ with color 2, and coloring the vertices of C^* equitably with the remaining $\chi_d - 2$ colors produces the desired coloring of G.

For the converse, we first prove the following:

Claim. For any coloring of G with χ_d colors and deficiency at most Δ^*, all the vertices of C_1 have the same color. Similarly, all the vertices of C_2 have the same color, and this color is distinct from that of the vertices of C_1. Additionally, the remaining $\chi_d - 2$ colors are each used exactly $\Delta^* + 1$ times in C^*.

Proof. We first consider the colors given to the vertices of Z_1 and Z_2. Observe that since both sets have size $\chi_d \cdot (|\mathcal{F}| + 2) = \chi_d \cdot (\Delta^* + 1)$, there is a color c_1 that appears at least $\Delta^* + 1$ times in Z_1 and a color c_2 that appears at least $\Delta^* + 1$ times in Z_2. Since $Z_1 \subset N(u)$ for every vertex $u \in C_1 \cup C^*$, we obtain that no vertex of $C_1 \cup C^*$ uses color c_1. Using a similar argument, we obtain that no vertex of $C_2 \cup C^*$ uses color c_2.

We will first prove that $c_1 \neq c_2$. Indeed, suppose that $c_1 = c_2$. Since this color c_1 does not appear in $C_1 \cup C_2 \cup C^*$, we are left with $\chi_d - 1$ available colors for these sets, where $|C_1 \cup C_2 \cup C^*| = \chi_d \cdot |\mathcal{F}| + 2\chi_d - 4$. To obtain a contradiction observe that at least one color class should have size at least $\frac{|C_1 \cup C_2 \cup C^*|}{\chi_d - 1} > |\mathcal{F} + 2|$ for sufficiently large $\mathcal{F}$, which is more than $\Delta^* + 1$ vertices.

The above implies that C^* must be colored using at most $\chi_d - 2$ colors. Since C^* is a clique of size exactly $(\chi_d - 2) \cdot (|\mathcal{F}| + 2) = (\chi_d - 2) \cdot (\Delta^* + 1)$, it follows that C^* is colored using $\chi_d - 2$ colors, each of which are used exactly $\Delta^* + 1$ times, as desired. Last, we conclude that vertices in C_1 should only be colored c_2 and similarly vertices of C_2 should receive color c_1. □

By the previous claim C_1 and C_2 are both monochromatic and use different colors. Without loss of generality suppose that C_1 is colored with color 1 and C_2 with color 2. Since every vertex of I is adjacent to every vertex of C^*, we immediately obtain that I must be colored using only 1 or 2. It only remains to show that for every set S of $\mathcal{F}$, there exist elements $x, y \in S$ such that vertex w_x is colored with color 1 and vertex w_y is colored with color 2. Then, the coloring of I will give us a partition of U. Assume for contradiction that there exists a set S whose elements x, y and z all have the same color, say color 1. From the above claim, we know that $v_S^1 \in C_1$ uses color 1 and is adjacent to the other $\Delta^* - 2$ vertices of C_1, all of which also use color 1. Therefore, v_S^1 is adjacent to $\Delta^* - 2 + 3$ vertices using color 1, and hence has deficiency $\Delta^* + 1$, a contradiction. This concludes the proof. □

A.10 Proof of Theorem 11

Proof (Theorem 11). We describe a dynamic programming algorithm which uses standard techniques, and hence we sketch some of the details. Suppose that we are given a rooted nice tree decomposition of G (we use here the definition of nice tree decomposition given in [9]). For every bag B of the decomposition we denote by $B^{\downarrow}$ the set of vertices of G that appear in B and bags below it in the decomposition. For a coloring $c : V \rightarrow \{1, \ldots, \chi_{\mathrm{d}}\}$ we say that the partial type of a vertex $u \in B$ is a pair consisting of $c(u)$ and $|c^{-1}(c(u)) \cap N(u) \cap B^{\downarrow}|$. In words, the type of a vertex is its color and its deficiency in the graph induced by $B^{\downarrow}$. Clearly, if c is a valid coloring, any vertex can have at most $\chi_{\mathrm{d}} \cdot (\Delta^* + 1)$ types. Hence, if we define the type of B as a tuple containing the types of its vertices, any bag can have one of at most $(\chi_{\mathrm{d}} \cdot (\Delta^* + 1))^{\mathrm{tw}}$ types.

Our dynamic programming algorithm will now construct a table which for every bag B and every possible bag type decides if there is a coloring of $B^{\downarrow}$ with the specified type for which all vertices of $B^{\downarrow} \setminus B$ have deficiency at most Δ^*. The table is easy to construct for leaf bags and forget bags. For introduce bags we consider all possible colors of the new vertex, and for each color we appropriately compute its deficiency and update the deficiency of its neighbors in the bag, rejecting solutions where a vertex reaches deficiency $\Delta^* + 1$. Finally, for join bags we consider any pair of partial solutions from the two children bags that agree on the colors of all vertices of the bag and compute the deficiency of each vertex as the sum of its deficiencies in the two solutions. □