PARAMETERIZED (APPROXIMATE) DEFECTIVE COLORING*

RÉMY BELMONTE†, MICHAEL LAMPIS‡, AND VALIA MITSOU§

Abstract.
In DEFECTIVE COLORING we are given a graph $G = (V, E)$ and two integers $\chi_{\mathrm{d}}, \Delta^*$ and are asked if we can partition V into χ_{d} color classes, so that each class induces a graph of maximum degree Δ^*. We investigate the complexity of this generalization of COLORING with respect to several well-studied graph parameters, and show that the problem is W-hard parameterized by treewidth, pathwidth, tree-depth, or feedback vertex set, if $\chi_{\mathrm{d}} = 2$. As expected, this hardness can be extended to larger values of χ_{d} for most of these parameters, with one surprising exception: we show that the problem is FPT parameterized by feedback vertex set for any $\chi_{\mathrm{d}} \neq 2$, and hence 2-coloring is the only hard case for this parameter. In addition to the above, we give an ETH-based lower bound for treewidth and pathwidth, showing that no algorithm can solve the problem in $n^{o(\mathrm{pw})}$, essentially matching the complexity of an algorithm obtained with standard techniques.

We complement these results by considering the problem's approximability and show that, with respect to Δ^*, the problem admits an algorithm which for any $\epsilon > 0$ runs in time $(\mathrm{tw}/\epsilon)^{O(\mathrm{tw})}$ and returns a solution with exactly the desired number of colors that approximates the optimal Δ^* within $(1 + \epsilon)$. We also give a $(\mathrm{tw})^{O(\mathrm{tw})}$ algorithm which achieves the desired Δ^* exactly while 2-approximating the minimum value of χ_{d}. We show that this is close to optimal, by establishing that no FPT algorithm can (under standard assumptions) achieve a better than 3/2-approximation to χ_{d}, even when an extra constant additive error is also allowed.

Key words. Defective Coloring, Improper Coloring, Parameterized Complexity, Treewidth.

AMS subject classifications. 68Q17, 68R10, 68W25

1. Introduction.

DEFECTIVE COLORING is the following problem: we are given a graph $G = (V, E)$, and two integer parameters $\chi_{\mathrm{d}}, \Delta^*$, and are asked whether there exists a partition of V into at most χ_{d} sets (color classes), such that each set induces a graph with maximum degree at most Δ^*. DEFECTIVE COLORING, which is also sometimes referred to in the literature as IMPROPER COLORING, is a natural generalization of the classical COLORING problem, which corresponds to the case $\Delta^* = 0$. The problem was introduced more than thirty years ago [2, 17], and since then has attracted a great deal of attention [1, 4, 6, 13, 14, 16, 24, 26, 29, 33, 36, 37].

From the point of view of applications, DEFECTIVE COLORING is particularly interesting in the context of wireless communication networks, where the assignment of colors to vertices often represents the assignment of frequencies to communication nodes. In many practical settings, the requirement of traditional coloring that all neighboring nodes receive distinct colors is too rigid, as a small amount of interference is often tolerable, and may lead to solutions that need drastically fewer frequencies. DEFECTIVE COLORING allows one to model this tolerance through the parameter Δ^*. As a result the problem's complexity has been well-investigated in graph topologies motivated by such applications, such as unit-disk graphs and various classes of grids

*Submitted to the editors 29/10/2018. A conference version of this paper appeared in STACS 2018.

Funding: All authors were supported by the GRAPA – Graph Algorithms for Parameterized Approximation – 38593YJ PHC Sakura Project and the PARAGA – Parameterized Approximation Graph Algorithms CNRS-JSPS project. The third author was also supported by ANR project HOSIGRA ANR-17-CE40-0022.

†University of Electro-Communications, Chofu, Tokyo, 182-8585, Japan, (remy.belmonte@uec.ac.jp).

‡Université Paris-Dauphine, PSL University, CNRS, LAMSADE, Paris, France, michail.lampis@dauphine.fr.

§Université de Paris, IRIF, CNRS, Paris, France, vmitsou@irif.fr.

[5, 7, 8, 10, 27, 28]. For more background we refer to [23, 32].

In this paper we study Defective Coloring from the point of view of parameterized complexity [18, 19, 22, 43]. The problem is of course NP-hard, even for small values of $\chi_{\mathrm{d}}, \Delta^*$, as it generalizes Coloring. For the same reason, it is also NP-hard to even approximate either χ_{d} or Δ^* (see Lemma 2.1). We are therefore strongly motivated to bring to bear the powerful toolbox of structural graph parameters, such as treewidth, which have proved extremely successful in tackling other intractable hard problems. Indeed, Coloring is one of the success stories of this domain, since the complexity of this flagship problem with respect to treewidth (and related parameters pathwidth, feedback vertex set, vertex cover) is by now extremely well-understood [31, 40, 41]. We pose the natural question of whether similar success can be achieved for Defective Coloring, or whether the addition of Δ^* significantly alters the complexity behavior of the problem. Such results are not yet known for Defective Coloring, except for the fact that it was observed in [9] that the problem admits (by standard techniques) a roughly $(\chi_{\mathrm{d}}\Delta^*)^{\mathrm{tw}}$-time algorithm, where tw is the graph's treewidth. In parameterized complexity terms, this shows that the problem is FPT parameterized by $\mathrm{tw} + \Delta^*$. One of our main motivating questions is whether this running time can be improved qualitatively (is the problem FPT parameterized only by tw?) or quantitatively.

Our first result is to establish that the problem is W-hard not just for treewidth, but also for several much more restricted structural graph parameters, such as pathwidth, tree-depth, and feedback vertex set. We recall that for Coloring, the standard ${\chi_{\mathrm{d}}}^{\mathrm{tw}}$ algorithm is FPT by tw, as graphs of bounded treewidth also have bounded chromatic number (Lemma 2.2). Our result shows that the complexity of the problem changes drastically with the addition of the new parameter Δ^*, and it appears likely that tw must appear in the exponent of Δ^* in the running time, even when Δ^* is large. More strongly, we establish this hardness even for the case $\chi_{\mathrm{d}} = 2$, which corresponds to the problem of partitioning a graph into two parts so as to minimize their maximum degree. This identifies Defective Coloring as another member of a family of generalizations of Coloring (such as Equitable Coloring or List Coloring) which are hard for treewidth [21].

As one might expect, the W-hardness results on Defective Coloring parameterized by treewidth (or pathwidth, or tree-depth) easily carry over for values of χ_{d} larger than 2. Surprisingly, we show that this is *not* the case for the parameter feedback vertex set, for which the only W-hard case is 2-coloring: we establish with a simple win/win argument that the problem is FPT for any other value of χ_{d}. We also show that if one considers sufficiently restricted parameters, such as vertex cover, the problem does eventually become FPT.

Our second step is to enhance the W-hardness result mentioned above with the aim of determining as precisely as possible the complexity of Defective Coloring parameterized by treewidth. Our reduction for tree-depth and feedback vertex set is quadratic in the parameter, and hence implies that no algorithm can solve the problem in time $n^{o(\sqrt{\mathrm{tw}})}$ under the Exponential Time Hypothesis (ETH) [30]. We therefore present a second reduction, which applies only to pathwidth and treewidth, but manages to show that no algorithm can solve the problem in time $n^{o(\mathrm{pw})}$ or $n^{o(\mathrm{tw})}$ under the ETH. This lower bound is tight, as it matches asymptotically the exponent given in the algorithm of [9].

To complement the above results, we also consider the problem from the point of view of (parameterized) approximation. Here things become significantly better: we give an algorithm using a technique of [39] which for any χ_{d} and error $\epsilon > 0$ runs in

Parameter	Result (Exact solution)	Ref.	Result (Approximation)	Ref.
Feedback Vertex Set	W[1]-hard for $\chi_{\mathrm{d}} = 2$ FPT for $\chi_{\mathrm{d}} \neq 2$	Thm 3.1 Thm 5.2	+1-approximation of χ_{d} in time $\mathrm{fvs}^{O(\mathrm{fvs})}$	Cor 6.7
Tree-depth	W[1]-hard for any $\chi_{\mathrm{d}} \geq 2$	Thm 3.1	W[1]-hard to color with $(3/2 - \epsilon)\chi_{\mathrm{d}} + O(1)$ colors	Thm 6.5
Treewidth, Pathwidth	No $n^{o(\mathrm{pw})}$ or $n^{o(\mathrm{tw})}$ algorithm under ETH	Thm 4.1	$(1+\epsilon)$-approximation for Δ^* in $(\mathrm{tw}/\epsilon)^{O(\mathrm{tw})}$ 2-approximation for χ_{d} in $\mathrm{tw}^{O(\mathrm{tw})}$	Thm 6.2 Thm 6.4
Vertex Cover	$\mathrm{vc}^{O(\mathrm{vc})}$ algorithm	Thm 5.3		

TABLE 1

Summary of results. Hardness results for tree-depth imply the same bounds for treewidth and pathwidth. Conversely, algorithms which apply to treewidth apply also to all other parameters.

time $(tw/\epsilon)^{O(\mathrm{tw})}n^{O(1)}$ and approximates the optimal value of Δ^* within a factor of $(1+\epsilon)$. Hence, despite the problem's W-hardness, we produce a solution arbitrarily close to optimal in FPT time.

Motivated by this algorithm we also consider the complementary approximation problem: given Δ^* find a solution that comes as close to the minimum number of colors needed as possible. By building on the approximation algorithm for Δ^*, we are able to present a $(\mathrm{tw})^{O(\mathrm{tw})}n^{O(1)}$ algorithm that achieves a 2-approximation for this problem. One can observe that this is not far from optimal, since an FPT algorithm with approximation ratio better than 3/2 would contradict the problem's W-hardness for $\chi_{\mathrm{d}} = 2$. However, this simple argument is unsatisfying, because it does not rule out algorithms with a ratio significantly better than 3/2, if one also allows a small additive error; indeed, we observe that when parameterized by feedback vertex set the problem admits an FPT algorithm that approximates the optimal χ_{d} within an additive error of just 1. To resolve this problem we present a gap-introducing version of our reduction which, for any i produces an instance for which the optimal value of χ_{d} is either $2i$, or at least $3i$. In this way we show that, when parameterized by tree-depth, pathwidth, or treewidth, approximating the optimal value of χ_{d} better than 3/2 is "truly" hard, and this is not an artifact of the problem's hardness for 2-coloring.

2. Definitions and Preliminaries.

For a graph $G = (V, E)$ and two integers $\chi_{\mathrm{d}} \geq 1$, $\Delta^* \geq 0$, we say that G admits a $(\chi_{\mathrm{d}}, \Delta^*)$-coloring if one can partition V into χ_{d} sets such that the graph induced by each set has maximum degree at most Δ^*. DEFECTIVE COLORING is the problem of deciding, given $G, \chi_{\mathrm{d}}, \Delta^*$, whether G admits a $(\chi_{\mathrm{d}}, \Delta^*)$-coloring. For $\Delta^* = 0$ this corresponds to COLORING.

We note that since DEFECTIVE COLORING generalizes COLORING, the problem is NP-hard even to approximate, with respect to both χ_{d} and Δ^*.

LEMMA 2.1. *For any constants $\rho > 1, \Delta^* \geq 0$, the following problem is NP-hard: given a graph $G = (V, E)$, and an integer χ_{d}, distinguish whether G admits a $(\chi_{\mathrm{d}}, 0)$-coloring, or whether it does not even admit a $(\rho\chi_{\mathrm{d}}, \Delta^*)$ coloring.*

Proof. We recall that Coloring is NP-hard to approximate within any constant (indeed, within any non-trivial polynomial factor [20]). For any $\rho > 1$ we can therefore produce in polynomial time a graph $G = (V, E)$ and an integer χ_{d} such that it is NP-hard to distinguish whether G can be properly colored with χ_{d} colors, or whether it needs strictly more than $\rho\chi_{\mathrm{d}}$ colors.

We construct a graph G' by replacing each vertex of G with an independent set of $\rho\chi_{\mathrm{d}}\Delta^* + 1$ vertices and each $(u, v) \in E$ by a complete bipartite graph joining the independent sets that replaced u, v. If G is χ_{d}-colorable, then G' is as well, so it admits a $(\chi_{\mathrm{d}}, 0)$-coloring. If G' admits a $(\rho\chi_{\mathrm{d}}, \Delta^*)$-coloring, we construct a coloring of G using $\rho\chi_{\mathrm{d}}$ colors by giving each $u \in V$ the color that appears most often in the independnt set that replaced u in G'. This is a valid coloring of G because if two neighbors received the same color, this color appears at least $\lceil(\rho\chi_{\mathrm{d}}\Delta^* + 1)/\rho\chi_{\mathrm{d}}\rceil \geq \Delta^* + 1$ times on two neighboring independent sets of G'. ◻

We assume the reader is familiar with basic notions in parameterized complexity, such as the classes FPT and W[1]. For the relevant definitions we refer to the standard textbooks [18, 19, 22, 43]. We rely on a number of well-known graph measures: treewidth [12], pathwidth, tree-depth [42], feedback vertex set, and vertex cover, denoted respectively as $\mathrm{tw}(G), \mathrm{pw}(G), \mathrm{td}(G), \mathrm{fvs}(G), \mathrm{vc}(G)$, where we drop G if it is clear from the context.

We recall here some standard definitions for the reader's convenience. A tree decomposition of a graph $G = (V, E)$ is a (rooted) tree $T = (X, I)$ such that each node of T is a subset of V. We call the elements of X bags. T must obey the following constraints: $\forall v \in V\ \exists B \in X$ such that $v \in B$; $\forall (u, v) \in E\ \exists B \in X$ such that $u, v \in B$; $\forall v \in V$ the bags of X that contain v induce a connected sub-tree. The width of a tree decomposition is $\max_{B \in X} |B| - 1$, and $\mathrm{tw}(G)$ is the minimum width of a tree decomposition of G. Pathwidth is defined similarly, except the decomposition is required to be a path instead of a tree.

For a rooted tree T we define its height as the number of vertices in the longest path from the root to a leaf, and its completion as the graph obtained by connecting each node to all of its ancestors. For a graph G we define $\mathrm{td}(G)$ as the minimum height of any tree whose completion contains G as a subgraph. An equivalent recursive definition is the following: $\mathrm{td}(K_1) = 1$; if G is disconnected then $\mathrm{td}(G)$ is equal to the maximum tree-depth of G's connected components; otherwise $\mathrm{td}(G) = 1 + \min_{v \in V} \mathrm{td}(G[V \setminus v])$.

A graph's feedback vertex set (respectively vertex cover) is the smallest set of vertices whose removal leaves the graph acyclic (respectively edge-less).

Lemma 2.2. *For any graph G we have $\mathrm{tw}(G) - 1 \leq \mathrm{fvs}(G) \leq \mathrm{vc}(G)$ and $\mathrm{tw}(G) \leq \mathrm{pw}(G) \leq \mathrm{td}(G) - 1 \leq \mathrm{vc}(G)$. Furthermore, any graph G admits a $(\mathrm{tw}(G) + 1, 0)$-coloring, a $(\mathrm{pw}(G) + 1, 0)$-coloring, a $(\mathrm{td}(G), 0)$-coloring, and a $(\mathrm{fvs}(G) + 2, 0)$-coloring.*

Proof. All stated relations are standard but we recall here the proofs for the sake of completeness. To obtain $\mathrm{tw}(G) - 1 \leq \mathrm{fvs}(G)$, if $S \subseteq V$ is a feedback vertex set, we can construct a tree decomposition of G by including all vertices of S in a tree decomposition (of width 1) of $G[V \setminus S]$. $\mathrm{fvs}(G) \leq vc(G)$ follows because every vertex cover is also a feedback vertex set. $\mathrm{tw}(G) \leq \mathrm{pw}(G)$ because all path decompositions are also valid tree decompositions. $\mathrm{pw}(G) \leq \mathrm{td}(G) - 1$ can be seen by recalling that, if G is connected $\exists v \in V$ such that $\mathrm{td}(G) = 1 + \mathrm{td}(G[V \setminus v])$. We can now take a path decomposition of $G[V \setminus v]$ and add v to every bag. To see that $\mathrm{td}(G) \leq vc(G) + 1$ we observe that for a vertex v that belongs in a minimum vertex cover of G we have $\mathrm{td}(G) \leq \mathrm{td}(G - v) + 1$ and $\mathrm{vc}(G) = \mathrm{vc}(G - v) + 1$, which allows us to obtain the inequality by induction.

For the coloring statements, we recall that a graph with treewidth tw is (tw + 1)-degenerate, that is, there exists an ordering of its vertices such that each vertex has at most tw + 1 neighbors among the vertices that precede it [12]. To see that $\mathrm{td}(G)$ colors suffice to color G if it is connected, we recall that $\exists v \in V$ such that $\mathrm{td}(G) = 1 + \mathrm{td}(G[V \setminus v])$, use a unique color for v and $\mathrm{td}(G) - 1$ for the rest of the graph. $\mathrm{fvs}(G) + 2$ colors are always sufficient to properly color a graph because we can use distinct colors for the feedback vertex set, and two-color the remaining forest. ◻

The Exponential Time Hypothesis (ETH) states that there exists a constant $c_3 > 1$ such that 3-SAT on instances with n variables and m clauses cannot be solved in time c_3^{n+m} [30]. For our purposes it will be sufficient to rely on a weaker form of the ETH which states that 3-SAT cannot be solved in $2^{o(n+m)}$ time. We define the k-Multi-Colored Clique problem as follows: we are given a graph $G = (V, E)$, a partition of V into k independent sets $V_1, \ldots, V_k$, such that for all $i \in \{1, \ldots, k\}$ we have $|V_i| = n$, and we are asked if G contains a k-clique. It is well-known that this problem is W[1]-hard parameterized by k, and that it does not admit any $n^{o(k)}$ algorithm, unless the ETH is false [18].

3. W-hardness for Feedback Vertex Set and Tree-depth. The main result of this section states that deciding if a graph admits a $(2, \Delta^*)$-coloring, where Δ^* is part of the input, is W[1]-hard parameterized by either fvs or td. Because of standard relations between graph parameters (Lemma 2.2), this implies also the same problem's W-hardness for parameters pw and tw. As might be expected, it is not hard to extend our proof to give hardness for deciding if a $(\chi_\mathrm{d}, \Delta^*)$-coloring exists, for any constant χ_d, parameterized by tree-depth (and hence, also treewidth and pathwidth). What is perhaps more surprising is that this cannot be done in the case of feedback vertex set. Superficially, the reason we cannot extend the reduction in this case is that one of the gadgets we use in many copies in our construction has large fvs if $\chi_\mathrm{d} > 2$. However, we give a much more convincing reason in Theorem 5.2 of Section 5 where we show that Defective Coloring is FPT parameterized by fvs for $\chi_\mathrm{d} \geq 3$, and therefore, if we could extend our reduction in this case it would prove that FPT=W[1].

The main theorem of this section is stated below. We then present the reduction in Sections 3.1, 3.2, and give the Lemmata that imply Theorem 3.1 in Section 3.3.

Theorem 3.1. *Deciding if a graph G admits a $(2, \Delta^*)$-coloring, where Δ^* is part of the input, is W[1]-hard parameterized by $\mathrm{fvs}(G)$. Deciding if a graph G admits a $(\chi_\mathrm{d}, \Delta^*)$-coloring, where $\chi_\mathrm{d} \geq 2$ is any fixed constant and Δ^* is part of the input is W[1]-hard parameterized by $\mathrm{td}(G)$.*

3.1. Basic Gadgets. Before we proceed, we present some basic gadgets that will be useful in all the reductions of this paper (Theorems 3.1, 4.1, 6.5). We first define a building block $\mathcal{T}(i, j)$ which is a graph that can be properly colored with i colors, but admits no $(i - 1, j)$-coloring (similar constructions appear in [29]). We then use this graph to build two gadgets: the Equality Gadget and the Palette Gadget (Definitions 3.4 and 3.7). Informally, for given $\chi_\mathrm{d}, \Delta^*$, the equality gadget allows us to express the constraint that two vertices v_1, v_2 of a graph must receive the same color in any valid $(\chi_\mathrm{d}, \Delta^*)$-coloring. The palette gadget will be used to express the constraint that, among three vertices v_1, v_2, v_3, there must exist two with the same color. For both gadgets we first prove formally that they express these constraints (Lemmata 3.5 and 3.8). We then show that, under certain conditions, these gadgets can be added to any graph without significantly increasing its tree-depth or feedback vertex set (Lemmata 3.6 and 3.9), that is, that we may use these gadget while maintaining a valid FPT reduction.

Below, we use K_1 to denote the graph that consists of a single isolated vertex.

DEFINITION 3.2. *Given two integers $i > 0, j \geq 0$, we define the graph $\mathcal{T}(i,j)$ recursively as follows: $T(1,j) = K_1$ for all j; for $i > 1$, $T(i,j)$ is the graph obtained by taking $(j+1)$ disjoint copies of $T(i-1,j)$ and adding to the graph a new universal vertex, that is, a vertex connected to all other vertices.*

LEMMA 3.3. *For all $i > 0, j \geq 0$ we have: $\mathcal{T}(i,j)$ admits an $(i,0)$-coloring; $\mathcal{T}(i,j)$ does not admit an $(i-1,j)$-coloring; $\mathrm{td}(\mathcal{T}(i,j)) = \mathrm{pw}(\mathcal{T}(i,j))+1 = \mathrm{tw}(\mathcal{T}(i,j))+1 = i$.*

Proof. We begin with the last statement: clearly $\mathrm{td}(\mathcal{T}(1,j)) = \mathrm{pw}(\mathcal{T}(1,j)) + 1 = \mathrm{tw}(\mathcal{T}(1,j)) + 1 = 1$, while it can be seen that $\mathrm{tw}(\mathcal{T}(i,j)) + 1 \leq \mathrm{pw}(\mathcal{T}(i,j)) + 1 \leq \mathrm{td}(\mathcal{T}(i,j)) \leq 1 + \mathrm{td}(\mathcal{T}(i-1,j))$ by removing the universal vertex. We also observe that $\mathrm{td}(\mathcal{T}(i,j)) \geq \mathrm{pw}(\mathcal{T}(i,j)) + 1 \geq \mathrm{tw}(\mathcal{T}(i,j)) + 1 \geq i$ because $\mathcal{T}(i,j)$ contains a clique of size i. The fact that $\mathcal{T}(i,j)$ admits an $(i,0)$-coloring now follows by Lemma 2.2. Finally, to see that $\mathcal{T}(i,j)$ does not admit an $(i-1,j)$-coloring, we do induction on i. Clearly, $\mathcal{T}(1,j)$ requires at least one color. Suppose now that $\mathcal{T}(i,j)$ does not admit an $(i-1,j)$-coloring but, for the sake of contradiction, $\mathcal{T}(i+1,j)$ admits an (i,j)-coloring. By assumption, each of the $j+1$ copies of $\mathcal{T}(i,j)$ contained in $\mathcal{T}(i+1,j)$ must be using all i available colors. Hence, each color appears at least $j+1$ times, which implies that there is no available color for the universal vertex. ◻

DEFINITION 3.4. *(Equality Gadget) For $i \geq 2, j \geq 0$, let $Q(u_1,u_2,i,j)$ be a graph defined as follows: Q contains $ij+1$ disjoint copies of $\mathcal{T}(i-1,j)$ as well as two vertices u_1, u_2 which are connected to all vertices except each other.*

LEMMA 3.5. *Let $G = (V,E)$ be a graph with $v_1, v_2 \in V$ and let G' be the graph obtained from G by adding to it a copy of $Q(u_1,u_2,\chi_{\mathrm{d}},\Delta^*)$ and identifying u_1 with v_1 and u_2 with v_2. Then, any $(\chi_{\mathrm{d}},\Delta^*)$-coloring of G' must give the same color to v_1, v_2. Furthermore, if there exists a $(\chi_{\mathrm{d}},\Delta^*)$-coloring of G that gives the same color to v_1, v_2, this coloring can be extended to a $(\chi_{\mathrm{d}},\Delta^*)$-coloring of G'.*

Proof. For the first statement, consider a $(\chi_{\mathrm{d}},\Delta^*)$-coloring of G' and examine the copies of $\mathcal{T}(\chi_{\mathrm{d}}-1,\Delta^*)$ contained in the equality gadget added to G. For a set $C \subseteq \{1,\ldots,\chi_{\mathrm{d}}\}$ with size $|C| = \chi_{\mathrm{d}}-1$ we say that C is contained in a copy of $\mathcal{T}(\chi_{\mathrm{d}}-1,\Delta^*)$ if all the colors of C appear in this copy in the coloring of G'. There are $\binom{\chi_{\mathrm{d}}}{\chi_{\mathrm{d}}-1} = \chi_{\mathrm{d}}$ such sets of colors C, and every copy of $\mathcal{T}(\chi_{\mathrm{d}}-1,\Delta^*)$ contains at least one by Lemma 3.3. Hence, the set of colors C that is contained in the largest number of copies is contained in at least $\lceil \frac{\chi_{\mathrm{d}}\Delta^*+1}{\chi_{\mathrm{d}}} \rceil = \Delta^*+1$ copies, therefore all its colors appear at least Δ^*+1 times. This means that v_1, v_2 cannot take any of the colors in C, and therefore must use the same color.

For the second statement, we want to extend a coloring of G to a coloring of G'. Recall that by Lemma 3.3, $\mathcal{T}(\chi_{\mathrm{d}}-1,\Delta^*)$ can be properly colored with $\chi_{\mathrm{d}}-1$ colors, and $\chi_{\mathrm{d}}-1$ colors are available if v_1, v_2 use the same colors. ◻

LEMMA 3.6. *Let $G = (V,E)$ be a graph, $S \subseteq V$, and G' be a graph obtained from G by repeated applications of the following operation: we select two vertices $v_1, v_2 \in V$ such that $v_1 \in S$, add a new copy of $Q(u_1,u_2,\chi_{\mathrm{d}},\Delta^*)$ and identify u_i with v_i, for $i \in \{1,2\}$. Then $\mathrm{td}(G') \leq \mathrm{td}(G \setminus S) + |S| + \chi_{\mathrm{d}} - 1$. Furthermore, if $\chi_{\mathrm{d}} = 2$ we have $\mathrm{fvs}(G') \leq \mathrm{fvs}(G \setminus S) + |S|$.*

Proof. For the inequality for td, we begin by observing that $\mathrm{td}(G') \leq \mathrm{td}(G' \setminus S) + |S|$, so it suffices to show that $\mathrm{td}(G' \setminus S) \leq \mathrm{td}(G \setminus S) + \chi_{\mathrm{d}} - 1$. Observe now that in $G' \setminus S$, in every copy of Q one of the vertices u_1, u_2 has been removed.

By definition, there must exist a rooted tree T_1 with $\mathrm{td}(G \setminus S)$ levels such that if we complete the tree (that is, connect each node of T_1 to all its descendants), $G \setminus S$ is a subgraph of the resulting graph. Similarly, there exists a rooted tree T_2 with $\chi_{\mathrm{d}} - 1$ levels such that $\mathcal{T}(\chi_{\mathrm{d}} - 1, \Delta^*)$ is a subgraph of its completion. We now observe that if we take T_1 and attach to each of its nodes a copy of T_2 we have a tree with $\mathrm{td}(G \setminus S) + \chi_{\mathrm{d}} - 1$ levels whose completion contains $G' \setminus S$ as a subgraph.

For the inequality for fvs, if $\chi_{\mathrm{d}} = 2$ the equality gadgets we have added to G contain copies of $\mathcal{T}(1, \Delta) = K_1$. If we remove S from G', and therefore remove one endpoint of each equality gadget, all these copies of K_1 become leaves, and hence do not affect the size of the graph's minimum feedback vertex set. Deleting them gives us the graph $G \setminus S$, so we conclude that $\mathrm{fvs}(G' \setminus S) = \mathrm{fvs}(G \setminus S)$ which, together with the fact that $\mathrm{fvs}(G') \leq \mathrm{fvs}(G' \setminus S) + |S|$ completes the proof. ∎

DEFINITION 3.7. *(Palette Gadget) For $i \geq 3, j \geq 0$ we define $P(u_1, u_2, u_3, i, j)$ to be the following graph: P contains $\binom{i}{2}j + 1$ copies of $\mathcal{T}(i - 2, j)$, as well as three vertices u_1, u_2, u_3 which are connected to every vertex of P except each other.*

LEMMA 3.8. *Let $G = (V, E)$ be a graph with $v_1, v_2, v_3 \in V$ and let G' be the graph obtained from G by adding to it a copy of $P(u_1, u_2, u_3, \chi_{\mathrm{d}}, \Delta^*)$ and identifying u_i with v_i for $i \in \{1, 2, 3\}$. Then, in any $(\chi_{\mathrm{d}}, \Delta^*)$-coloring of G' at least two of the vertices of $\{v_1, v_2, v_3\}$ must share a color. Furthermore, if there exists a $(\chi_{\mathrm{d}}, \Delta^*)$-coloring of G that gives the same color to two of the vertices of $\{v_1, v_2, v_3\}$, this coloring can be extended to a $(\chi_{\mathrm{d}}, \Delta^*)$-coloring of G'.*

Proof. For the first statement, consider a $(\chi_{\mathrm{d}}, \Delta^*)$-coloring of G' and examine the copies of $\mathcal{T}(\chi_{\mathrm{d}} - 2, \Delta^*)$ contained in the palette gadget added to G. For a set $C \subseteq \{1, \ldots, \chi_{\mathrm{d}}\}$ with size $|C| = \chi_{\mathrm{d}} - 2$ we say that C is contained in a copy of $\mathcal{T}(\chi_{\mathrm{d}} - 2, \Delta^*)$ if all the colors of C appear in this copy in the coloring of G'. There are $\binom{\chi_{\mathrm{d}}}{\chi_{\mathrm{d}}-2} = \binom{\chi_{\mathrm{d}}}{2}$ such sets of colors C, and every copy of $\mathcal{T}(\chi_{\mathrm{d}} - 2, \Delta^*)$ contains at least one by Lemma 3.3. Hence, the set of colors C that is contained in the largest number of copies, is contained in at least $\lceil \frac{\binom{\chi_{\mathrm{d}}}{2}\Delta^*+1}{\binom{\chi_{\mathrm{d}}}{2}} \rceil = \Delta^* + 1$ copies, therefore all its colors appear at least $\Delta^* + 1$ times. This means that v_1, v_2, v_3 cannot take any of the colors in C, and therefore have only two colors available for them. By pigeonhole principle, two of them must share a color.

For the second statement, recall that by Lemma 3.3, $\mathcal{T}(\chi_{\mathrm{d}} - 2, \Delta^*)$ can be properly colored with $\chi_{\mathrm{d}} - 2$ colors, and $\chi_{\mathrm{d}} - 2$ colors are available if v_1, v_2, v_3 use at most two colors. ∎

LEMMA 3.9. *Let $G = (V, E)$ be a graph, $S \subseteq V$, and G' be a graph obtained from G by repeated applications of the following operation: we select three vertices $v_1, v_2, v_3 \in V$ such that $v_1, v_2 \in S$, add a new copy of $P(u_1, u_2, u_3, \chi_{\mathrm{d}}, \Delta^*)$ and identify u_i with v_i, for $i \in \{1, 2, 3\}$. Then $\mathrm{td}(G') \leq \mathrm{td}(G \setminus S) + |S| + \chi_{\mathrm{d}} - 2$.*

Proof. The proof follows along the same lines as the proof of Lemma 3.6. First, we observe that $\mathrm{td}(G') \leq \mathrm{td}(G' \setminus S) + |S|$ and then show that $\mathrm{td}(G' \setminus S) \leq \mathrm{td}(G \setminus S) + \chi_{\mathrm{d}} - 2$ by taking a tree T_1 with $\mathrm{td}(G \setminus S)$ levels whose completion contains $G \setminus S$ and attaching to each node a tree T_2 with $\chi_{\mathrm{d}} - 2$ levels whose completion contains $\mathcal{T}(\chi_{\mathrm{d}} - 2, \Delta^*)$. ∎

3.2. Construction. We are now ready to present a reduction from k-MULTI-COLORED CLIQUE. In this section we describe a construction which, given an instance of this problem (G, k) as well as an integer $\chi_{\mathrm{d}} \geq 2$ produces an instance of DEFECTIVE COLORING. Recall that we assume that in the initial instance $G = (V, E)$ is given to us partitioned into k independent sets $V_1, \ldots, V_k$, all of which have size n. We will produce a graph $H(G,k,\chi_{\mathrm{d}})$ and an integer Δ^* with the property that H admits a $(\chi_{\mathrm{d}},\Delta^*)$-coloring if and only if G has a k-clique. In the next section we prove the correctness of the construction and give bounds on the values of $\mathrm{td}(H)$ and $\mathrm{fvs}(H)$ to establish Theorem 3.1.

In our new instance we set $\Delta^* = |E| - \binom{k}{2}$. Let us now describe the graph H. Since we will repeatedly use the gadgets from Definitions 3.4 and 3.7, we will use the following convention: whenever v_1, v_2 are two vertices we have already introduced to H, when we say that we add an equality gadget $Q(v_1,v_2)$, this means that we add to H a copy of $Q(u_1,u_2,\chi_{\mathrm{d}},\Delta^*)$ and then identify u_1,u_2 with v_1,v_2 respectively (similarly for palette gadgets). To ease presentation we will gradually build the graph by describing its different conceptual parts.

Palette Part: Informally, the goal of this part is to obtain two vertices (p_A,p_B) which are guaranteed to have different colors. This part contains the following:

1. Two vertices called p_A,p_B which we will call the main palette vertices.
2. Δ^* vertices called $p_A^1,p_A^2,\ldots,p_A^{\Delta^*}$ and Δ^* vertices called $p_B^1,p_B^2,\ldots,p_B^{\Delta^*}$
3. Δ^* equality gadgets $Q(p_A,p_A^1),Q(p_A,p_A^2),\ldots,Q(p_A,p_A^{\Delta^*})$, and Δ^* equality gadgets $Q(p_B,p_B^1),Q(p_B,p_B^2),\ldots,Q(p_B,p_B^{\Delta^*})$.
4. An edge between p_A,p_B.
5. The Δ^* edges $(p_A,p_A^1),(p_A,p_A^2),\ldots,(p_A,p_A^{\Delta^*})$ as well as the Δ^* edges (p_B,p_B^1), $(p_B,p_B^2),\ldots,(p_B,p_B^{\Delta^*})$.

Choice Part: Informally, the goal of this part is to encode a choice of a vertex in each V_i. To this end we make $2n$ choice vertices for each color class of the original instance. The selection will be encoded by counting how many of the first n of these vertices have the same color as p_A. Formally, this part contains the following:

6. For all $i\in\{1,\ldots,k\}$, $j\in\{1,\ldots,2n\}$ the vertex c_j^i. We call these the choice vertices.
7. For all $i\in\{1,\ldots,k\}$, the vertices g_A^i and g_B^i. We call these the guard vertices.
8. For all $i\in\{1,\ldots,k\}$, $j\in\{1,\ldots,2n\}$ edges between c_j^i and the vertices g_A^i and g_B^i.
9. For all $i\in\{1,\ldots,k\}$, we add equality gadgets $Q(p_A,g_A^i)$ and $Q(p_B,g_B^i)$.
10. If $\chi_{\mathrm{d}}\geq 3$, for all $i\in\{1,\ldots,k\}$, $j\in\{1,\ldots,2n\}$ we add a palette gadget $P(p_A,p_B,c_j^i)$.

Transfer Part: Informally, the goal of this part is to transfer the choices of the previous part to the rest of the graph. For each color class of the original instance we make $(k-1)$ "low" transfer vertices, whose deficiency will equal the choice made in the previous part, and $(k-1)$ "high" transfer vertices, whose deficiency will equal the complement of the same value. Formally, this part of H contains the following:

11. For $i_1,i_2\in\{1,\ldots,k\}$, $i_1\neq i_2$ the vertex h_{i_1,i_2} and the vertex l_{i_1,i_2}. We call these the high and low transfer vertices.
12. For $i_1,i_2\in\{1,\ldots,k\}$, $i_1\neq i_2$ and for all $j\in\{1,\ldots,n\}$ an edge from l_{i_1,i_2} to $c_j^{i_1}$.
13. For $i_1,i_2\in\{1,\ldots,k\}$, $i_1\neq i_2$ and for all $j\in\{n+1,\ldots,2n\}$ an edge from h_{i_1,i_2} to $c_j^{i_1}$.
14. For all $i_1,i_2\in\{1,\ldots,k\}$, $i_1\neq i_2$ we add an equality gadget $Q(p_A,l_{i_1,i_2})$ and an equality gadget $Q(p_A,h_{i_1,i_2})$.

Edge representation: Informally, this part contains a gadget representing each edge of G. Each gadget will contain a special vertex which will be able to receive the color of p_B if and only if the corresponding edge, that is, the edge represented by this gadget, is part of the clique. Formally, we assume that all the vertices of each V_i are numbered

$\{1, \ldots, n\}$. For each edge e of G, if e connects the vertex with index j_1 from V_{i_1} with the vertex with index j_2 from V_{i_2} (assuming without loss of generality $i_1 < i_2$) we add the following vertices and edges to H:

15. Four independent sets $L_e^1, H_e^1, L_e^2, H_e^2$ with respective sizes $n - j_1$, j_1, $n - j_2$, j_2.
16. Edges connecting the vertex l_{i_1,i_2} (respectively, $h_{i_1,i_2}, l_{i_2,i_1}, h_{i_2,i_1}$) with all vertices of the set L_e^1 (respectively the sets H_e^1, L_e^2, H_e^2).
17. A vertex c_e, connected to all vertices in $L_e^1 \cup H_e^1 \cup L_e^2 \cup H_e^2$.
18. If $\chi_{\mathrm{d}} \geq 3$, for each $v \in L_e^1 \cup H_e^1 \cup L_e^2 \cup H_e^2 \cup \{c_e\}$ we add a palette gadget $P(p_A, p_B, v)$.

Finally, once we have added a gadget (as described above) for each $e \in E$, we add the following structure to H in order to ensure that we have a sufficient number of edges included in our clique:

19. A vertex c_U (universal checker) connected to all c_e for $e \in E$.
20. An equality gadget $Q(p_A, c_U)$.

Budget-Setting: Our construction is now almost done, except for the fact that some crucial vertices have degree significantly lower than Δ^* (and hence are always trivially colorable). To fix this, we will effectively lower their deficiency budget by giving them some extra neighbors. Formally, we add the following:

21. For each guard vertex g_A^i (respectively g_B^i), we construct an independent set G_A^i (respectively G_B^i) of size $\Delta^* - n$ and connect it to g_A^i (respectively g_B^i). For each $v \in G_A^i$ (respectively G_B^i) we add an equality gadget $Q(p_A, v)$ (respectively $Q(p_B, v)$).
22. For each transfer vertex l_{i_1,i_2} (respectively h_{i_1,i_2}), we construct an independent set of size $\Delta^* - n$ and connect all its vertices to l_{i_1,i_2} (or respectively to h_{i_1,i_2}). For each vertex v of this independent set we add an equality gadget $Q(p_A, v)$.
23. For each vertex c_e we add an independent set of size Δ^* and connect all its vertices to c_e. For each vertex v of this independent set we add an equality gadget $Q(p_B, v)$.

This completes the construction of the graph H.

3.3. Correctness.

To establish Theorem 3.1 we need to establish three properties of the graph $H(G, k, \chi_{\mathrm{d}})$ described in the preceding section: that the existence of a k-clique in G implies that H admits a $(\chi_{\mathrm{d}}, \Delta^*)$-coloring; that a $(\chi_{\mathrm{d}}, \Delta^*)$-coloring of H implies the existence of a k-clique in G; and that the tree-depth and feedback vertex set of G are bounded by some function of k. These are established in the Lemmata below.

LEMMA 3.10. *For any $\chi_{\mathrm{d}} \geq 2$, if G contains a k-clique, then the graph $H(G, k, \chi_{\mathrm{d}})$ described in the previous section admits a $(\chi_{\mathrm{d}}, \Delta^*)$-coloring.*

Proof. Consider a clique of size k in G that includes exactly one vertex from each V_i. We will denote this clique by a function $f : \{1, \ldots, k\} \to \{1, \ldots, n\}$, that is, we assume that the clique contains the vertex with index $f(i)$ from V_i. We produce a $(\chi_{\mathrm{d}}, \Delta^*)$-coloring of H as follows: vertex p_A receives color 1, while vertex p_B receives color 2. All vertices for which we have added an equality gadget with one endpoint identified with p_A (respectively p_B) take color 1 (respectively 2). We use Lemma 3.5 to properly color the internal vertices of the equality gadgets.

We have still left uncolored the choice vertices c_j^i as well as the internal vertices $L_e^1, H_e^1, L_e^2, H_e^2, c_e$ of the edge gadgets. We proceed as follows: for all $i \in \{1, \ldots, k\}$ we use color 1 on the vertices c_l^i such that $l \in \{1, \ldots, f(i)\} \cup \{n+1, \ldots, 2n - f(i)\}$; we use color 2 on all remaining choice vertices. For every $e \in E$ that is contained in the clique we color all vertices of the sets $L_e^1, H_e^1, L_e^2, H_e^2$ with color 1, and c_e with color 2. For all other edges we use the opposite coloring: we color all vertices of the sets $L_e^1, H_e^1, L_e^2, H_e^2$ with color 2, and c_e with color 1. We use Lemma 3.8 to properly color the internal vertices of palette gadgets, since all palette gadgets that we add use either color 1 or color 2 twice in their endpoints. This completes the coloring.

To see that the coloring we described is a $(\chi_\mathrm{d}, \Delta^*)$-coloring, first we note that by Lemmata 3.5,3.8 internal vertices of equality and palette gadgets are properly colored. Vertices p_A, p_B have exactly Δ^* neighbors with the same color; guard vertices g_A^i, g_B^i have exactly n neighbors with the same color among the choice vertices, hence exactly Δ^* neighbors with the same color overall; choice vertices have at most k neighbors of the same color, and we can assume that $k < |E| - \binom{k}{2}$; the vertex c_U has exactly $\Delta^* = |E| - \binom{k}{2}$ neighbors with color 1, since the clique contains exactly $\binom{k}{2}$ edges; all internal vertices of edge gadgets have at most one neighbor of the same color. Finally, for the transfer vertices l_{i_1,i_2} and h_{i_1,i_2}, we note that l_{i_1,i_2} (respectively h_{i_1,i_2}) has exactly $f(i_1)$ (respectively $n - f(i_1)$) neighbors with color 1 among the choice vertices. Furthermore, when $i_1 < i_2$, l_{i_1,i_2} (respectively h_{i_1,i_2}) has $|L_e^1|$ (respectively $|H_e^1|$) neighbors with color 1 in the edge gadgets, those corresponding to the edge e that belongs in the clique between V_{i_1} and V_{i_2}. But by construction $|L_e^1| = n - f(i_1)$ and $|H_e^1| = f(i_1)$, and with similar observations for the case $i_2 < i_1$ we conclude that all vertices have deficiency at most Δ^*. ∎

LEMMA 3.11. *For any $\chi_\mathrm{d} \geq 2$, if the graph $H(G, k, \chi_\mathrm{d})$ described in the previous section admits a $(\chi_\mathrm{d}, \Delta^*)$-coloring, then G contains a k-clique.*

Proof. Suppose that we are given a $(\chi_\mathrm{d}, \Delta^*)$-coloring $c : V(H) \to \{1, \ldots, \chi_\mathrm{d}\}$ of H. We first establish that $c(p_A) \neq c(p_B)$. Indeed, because of the equality gadgets added in Step 3 we have $c(p_A^1) = c(p_A^2) = \ldots = c(p_A^{\Delta^*}) = c(p_A)$ and $c(p_B^1) = c(p_B^2) = \ldots = c(p_B^{\Delta^*}) = c(p_B)$. Because of the edges added in Step 5 we then know that p_A, p_B each has at least Δ^* neighbors with the same color. Therefore, because of the edge connecting them, we conclude that $c(p_A) \neq c(p_B)$. Without loss of generality we will assume below that $c(p_A) = 1$ and $c(p_B) = 2$.

Because of the equality gadget of Step 20 we have $c(c_U) = 1$. Because c_U has degree $|E|$, we conclude that it has at least $\binom{k}{2}$ neighbors with color 2. These correspond to a set $E' \subseteq E$ of edges of the original graph with $|E'| \geq \binom{k}{2}$. We will prove that, in fact, E' induces a k-clique in G.

Let $e \in E'$ be an edge such that $c(c_e) = 2$. This implies that all the vertices of $L_e^1 \cup H_e^1 \cup L_e^2 \cup H_e^2$ must take color 1, because by Step 23 c_e already has Δ^* neighbors with color 2. In case $\chi_\mathrm{d} \geq 3$ we have also used here the fact that, by Step 18, every internal vertex of the gadget representing e must take color 1 or 2.

Suppose that $e \in E'$ connects the vertex with index j_1 in V_{i_1} to the vertex with index j_2 in V_{i_2}, $i_1 < i_2$. We first show that, for an $e' \in E$ also connecting V_{i_1} to V_{i_2} it must be that $e' \notin E'$. Suppose for contradiction that $e' \in E'$, and let j_1', j_2' be the indices of the endpoints of e'. We observe that l_{i_1,i_2} has at least $|L_e^1| + |L_{e'}^1| = 2n - j_1 - j_1'$ neighbors with color 1 in the edge gadgets, while h_{i_1,i_2} has at least $|H_e^1| + |H_{e'}^1| = j_1 + j_1'$ such neighbors. Both l_{i_1,i_2} and h_{i_1,i_2} had $\Delta^* - n$ neighbors of color 1 added in Step 22. Finally, among the $2n$ choice vertices $c_j^{i_1}$ which are neighbors of either l_{i_1,i_2} or h_{i_1,i_2} there are at least n which received color 1, because all the choice vertices have colors 1 or 2 (Step 10) and $g_B^{i_1}$, which has color 2 (Step 9), is connected to all of them and also has $\Delta^* - n$ other neighbors of color 2 (Step 21). Hence, the total number of vertices in $N(l_{i_1,i_2}) \cup N(h_{i_1,i_2})$ with color 1 is at least

$2n + 2(\Delta^* - n) + n > 2\Delta^*$, hence one of these two vertices has deficiency higher than Δ^*, contradiction. We conclude that $e' \notin E'$.

To complete the proof, let us show that the $\binom{k}{2}$ edges of E', each of which connects a different pair of parts of V, are incident on the same endpoints. Take $e \in E'$ as in the previous paragraph, and $e' \in E'$ connecting vertices with indices j_1', j_3' from the parts V_{i_1}, V_{i_3}, for $i_3 \neq i_2$. It suffices to show that $i_1 = i_1'$. Suppose for contradiction $i_1 \neq i_1'$. Consider now the vertices $l_{i_1,i_2}, h_{i_1,i_2}, l_{i_1,i_3}, h_{i_1,i_3}$, which, by similar reasoning as before, have $n - j_1$, j_1, $n - j_1'$, j_1' color-1 neighbors in the edge gadgets respectively. If there are strictly more than j_1 vertices with color 1 among the choice vertices $c_j^{i_1}$, $j \in \{1, \ldots, n\}$, then l_{i_1,i_2} would have deficiency more than Δ^*. If there are strictly more than $n - j_1$ vertices with color 1 among the choice vertices $c_j^{i_1}$, $j \in \{n+1, \ldots, 2n\}$, then h_{i_1,i_2} would have deficiency more than Δ^*. Since, by the same reasoning as previously, there are at least n vertices with color 1 among the choice vertices $c_j^{i_1}$, we conclude that there are exactly j_1 vertices with color 1 among the $c_j^{1_1}$ for $j \in \{1, \ldots, n\}$, and exactly $n - j_1$ such vertices in the rest. We can now conclude that the only way not to violate the deficiency of l_{i_1,i_3} or h_{i_1,i_3} is for $i_1 = i_1'$. ◻

LEMMA 3.12. *For any $\chi_\mathrm{d} \geq 2$, the graph $H(G, k, \chi_\mathrm{d})$ described in the previous section has $\mathrm{td}(H) = O(k^2 + \chi_\mathrm{d})$. Furthermore, if $\chi_\mathrm{d} = 2$, then $\mathrm{fvs}(H) = O(k^2)$.*

Proof. We first observe that all equality and palette gadgets added to the graph (Steps 3, 9, 10, 14, 18, 20-23) have at most one endpoint outside $\{p_A, p_B\}$. Hence, by Lemmata 3.6, 3.9, we can conclude that $\mathrm{td}(H) = td(H' \setminus \{p_A, p_B\}) + \chi_\mathrm{d} + 1$ and, for $\chi_\mathrm{d} = 2$ we have $\mathrm{fvs}(H) \leq \mathrm{fvs}(H' \setminus \{p_A, p_B\}) + 2$, where H' is the graph we obtain from H if we remove all the equality and palette gadgets. It therefore suffices to show that $\mathrm{td}(H') = O(k^2)$ and, if $\chi_\mathrm{d} = 2$, $\mathrm{fvs}(H') = O(k^2)$.

For both parameters we start by removing from the graph all the guard and transfer vertices, which are $2k + 2k(k-1) = 2k^2$ in total. We now have that all vertices $p_A^1, p_A^2, \ldots, p_A^{\Delta^*}$, vertices $p_B^1, p_B^2, \ldots, p_B^{\Delta^*}$, as well as all choice vertices are isolated. Furthermore, all vertices added to represent edges, as well as the budget-setting vertices, form a tree with root at c_U and 3 levels. We conclude that H' has $\mathrm{td}(H') \leq 2k^2 + 4$ and $\mathrm{fvs}(H') \leq 2k^2$. ◻

Theorem 3.1 now follows directly from the reduction we have described and Lemmata 3.10,3.11,3.12.

4. ETH-based Lower Bounds for Treewidth and Pathwidth. In this section we present a reduction which strengthens the results of Section 3 for the parameters treewidth and pathwidth. In particular, the reduction we present here establishes that, under the ETH, the known algorithm for DEFECTIVE COLORING for these parameters is essentially best possible.

We use a similar presentation order as in the previous section, first giving the construction and then the Lemmata that imply the result. Where possible, we re-use the gadgets we have already presented. The main theorem of this section states the following:

THEOREM 4.1. *For any fixed $\chi_\mathrm{d} \geq 2$, if there exists an algorithm which, given a graph $G = (V, E)$ and parameters $\chi_\mathrm{d}, \Delta^*$ decides if G admits a $(\chi_\mathrm{d}, \Delta^*)$-coloring in time $n^{o(\mathrm{pw})}$, then the ETH is false.*

4.1. Basic Gadgets. We use again the equality and palette gadgets of Section 3 (Definitions 3.4,3.7). Before proceeding, let us show that adding these gadgets to the graph does not increase the pathwidth too much. For the two types of gadget Q, P, we will call the vertices $u_1, u_2(, u_3)$ the endpoints of the gadget.

LEMMA 4.2. *Let $G = (V, E)$ be a graph and let G' be the graph obtained from G by repeating the following operation: find a copy of $Q(u_1, u_2, \chi_{\rm d}, \Delta^*)$, or a copy of $P(u_1, u_2, u_3, \chi_{\rm d}, \Delta^*)$; remove all its internal vertices from the graph; and add all edges between its endpoints which are not already connected. Then $\mathrm{tw}(G) \le \max\{\mathrm{tw}(G'), \chi_{\rm d}\}$ and $\mathrm{pw}(G) \le \mathrm{pw}(G') + \chi_{\rm d}$.*

Proof. First, we observe that there is a path decomposition of $Q(u_1, u_2, \chi_{\rm d}, \Delta^*)$ with width $\chi_{\rm d}$, as by Lemma 3.3 there is a path decomposition of $\mathcal{T}(\chi_{\rm d} - 1, \Delta^*)$ of width $\chi_{\rm d} - 2$, and we can add to all its bags the vertices u_1, u_2. Call this path decomposition T_Q. In the same way, there is a path decomposition of width $\chi_{\rm d}$ for $P(u_1, u_2, u_3, \chi_{\rm d}, \Delta^*)$, call it T_P.

We now take an optimal tree or path decomposition of G', call it T', and construct from it a decomposition of G. Consider a gadget $H \in \{Q, P\}$ that appears in G with endpoints $u_1, u_2(, u_3)$. Since in G' these endpoints form a clique, there is a bag in T' that contains all of them. Let B be the smallest such bag, that is, the bag that contains the smallest number of vertices. Now, if T' is a tree decomposition, we take T_H and attach it to B. If T' is a path decomposition, we insert in the decomposition immediately after B the decomposition T_H where we have added all vertices of B in all bags of T_H. It is not hard to see that in both cases the decompositions remain valid, and we can repeat this process for every H until we have a decomposition of G. ◻

4.2. Construction. We now describe a construction which, given an instance $G = (V, E),\ k$, of k-MULTI-COLORED CLIQUE and a constant $\chi_{\rm d}$ returns a graph $H(G, k, \chi_{\rm d})$ and an integer Δ^* such that H admits a $(\chi_{\rm d}, \Delta^*)$-coloring if and only if G has a k-clique, and the pathwidth of H is $O(k + \chi_{\rm d})$. We use m to denote $|E|$, and we set $\Delta^* = m - \binom{k}{2}$. As in Section 3 we present the construction in steps to ease presentation, and we use the same conventions regarding adding Q and P gadgets to the graph.

Palette Part: This part repeats steps 1-5 of the construction of Section 3. We recall that this creates two main palette vertices p_A, p_B (which are eventually guaranteed to have different colors).

Choice Part: In this part we construct a sequence of independent sets, arranged in what can be thought of as a $k \times 2m$ grid. The idea is that the choice we make in coloring the first independent set of every row will be propagated throughout the row. We can therefore encode k choices of a number between 1 and n, which will encode the clique.

6. For each $i \in \{1, \dots, k\}$, for each $j \in \{1, \dots, 2m\}$ we construct an independent set $C_{i,j}$ of size n.
7. (Backbone vertices) For each $i \in \{1, \dots, k\}$, for each $j \in \{1, \dots, 2m - 1\}$, for each $l \in \{A.B\}$ we construct a vertex $b^l_{i,j}$. We connect $b^l_{i,j}$ to all vertices of $C_{i,j}$ and all vertices of $C_{i,j+1}$.
8. For each backbone vertex $b^l_{i,j}$ added in the previous step, for $l \in \{A, B\}$, we add an equality gadget $Q(p_l, b^l_{i,j})$.

Edge Representation: In the $k \times 2m$ grid of independent sets we have constructed we devote two columns to represent each edge of G. In the remainder we assume some numbering of the edges of E with the numbers $\{1, \dots, m\}$, as well as a numbering of each V_i with the numbers $\{1, \dots, n\}$. Suppose that the j-th edge of E, where $j \in \{1, \dots, m\}$ connects the j_1-th vertex of V_{i_1} to the j_2-th vertex of V_{i_2}, where $j_1, j_2 \in \{1, \dots, n\}$ and $i_1, i_2 \in \{1, \dots, k\}$. We perform the following steps for each such edge.

9. We construct four independent sets $H_j^1, L_j^1, H_j^2, L_j^2$ with respective sizes $n - j_1, j_1, n - j_2, j_2$.
10. We construct four vertices $h_j^1, l_j^1, h_j^2, l_j^2$. We connect h_j^1 (respectively l_j^1, h_j^2, l_j^2) with all vertices of H_j^1 (respectively L_j^1, H_j^2, L_j^2).
11. We connect h_j^1 to all vertices of $C_{i_1,2j-1}$, l_j^1 to all vertices of $C_{i_1,2j}$, h_j^2 to all vertices of $C_{i_2,2j-1}$, l_j^2 to all vertices of $C_{i_2,2j}$.
12. We add equality gadgets $Q(p_A, h_j^1), Q(p_A, l_j^1), Q(p_A, h_j^2), Q(p_A, l_j^2)$.
13. We add a checker vertex c_j and connect it to all vertices of $H_j^1 \cup L_j^1 \cup H_j^2 \cup L_j^2$.

Validation and Budget-Setting: Finally, we add a vertex that counts how many edges we have included in our clique, as well as appropriate vertices to diminish the deficiency budget of various parts of our construction.

14. We add a universal checker vertex c_U and connect it to all vertices c_j added in step 13. We add an equality gadget $Q(p_A, c_U)$.
15. For every vertex c_j added in step 13 we construct an independent set of size Δ^* and connect all its vertices to c_j. For each vertex v in this set we add an equality gadget $Q(p_B, v)$.
16. For each vertex constructed in step 10 ($h_j^1, l_j^1, h_j^2, l_j^2$), we construct an independent set of size $\Delta^* - n$ and connect it to the vertex. For each vertex v of this independent set we add an equality gadget $Q(p_A, v)$.
17. For each backbone vertex $b_{i,j}^l$, with $l \in \{A, B\}$, we construct an independent set of size $\Delta^* - n$ and connect it to $b_{i,j}^l$. For each vertex v of this independent set we add an equality gadget $Q(p_l, v)$.
18. If $\chi_\mathrm{d} \geq 3$, for each vertex v added in steps 6-17 we add a palette gadget $P(p_A, p_B, v)$.

4.3. Correctness.

LEMMA 4.3. *For any $\chi_\mathrm{d} \geq 2$, if G contains a k-clique then the graph $H(G, k, \chi_\mathrm{d})$ described in the previous section admits a $(\chi_\mathrm{d}, \Delta^*)$-coloring.*

Proof. Suppose that G has a k-clique, given by a function $f : \{1, \ldots, k\} \to \{1, \ldots, n\}$, meaning that the clique contains vertex $f(i)$ from the set V_i. We color H as follows: p_A receives color 1, p_B receives color 2, and all vertices on which we have attached equality gadgets receive the appropriate color, according to Lemma 3.5. By Lemmata 3.5,3.8 we can extend this coloring to the internal vertices of equality and palette gadgets. For every independent set $C_{i,j}$, we color $f(i)$ of its vertices with 1 if j is odd, otherwise we color $n - f(i)$ of its vertices with 1; we color the remaining vertices of independent sets $C_{i,j}$ with 2. For the j-th edge of E, if it is contained in the clique then we color c_j with 2 and $H_j^1, L_j^1, H_j^2, L_j^2$ with 1, otherwise we color c_j with 1 and $H_j^1, L_j^1, H_j^2, L_j^2$ with 2. This completes the coloring.

To see that this coloring is valid, observe that the vertices in the palette part have each at most Δ^* neighbors of the same color; the backbone vertices $b_{i,j}^l$ have exactly Δ^* neighbors of the same color ($f(i)$ in one grid independent set and $n - f(i)$ in the other, plus $\Delta^* - n$ from step 17); the vertices $l_j^1, h_j^1, l_j^2, h_j^2$ if the j-th edge belongs to the clique have exactly Δ^* neighbors with the same color; the same vertices for an edge that does not belong to the clique have strictly fewer than Δ^* neighbors of the same color; all vertices c_j have at most Δ^* neighbors with the same color; and vertex c_U has $m - \binom{k}{2} = \Delta^*$ neighbors with the same color. ∎

LEMMA 4.4. *For any $\chi_\mathrm{d} \geq 2$, if the graph $H(G, k, \chi_\mathrm{d})$ described in the previous* *section admits a* $(\chi_{\mathrm{d}}, \Delta^*)$*-coloring, then* G *contains a* k*-clique.*

Proof. Suppose that we have a valid $(\chi_{\mathrm{d}}, \Delta^*)$-coloring of H. As in Lemma 3.11 we can assume that p_A, p_B receive distinct colors, without loss of generality, colors 1 and 2 respectively. Because of step 18 we can assume that all the main vertices of the graph also receive colors 1 or 2. Because of the equality gadget added in 14 we know that vertex c_U received color 1. Since it has m neighbors, there must exist at least $m - \Delta^* = \binom{k}{2}$ vertices c_j which received color 2. We call the corresponding edges of G the selected edges and we will eventually prove that they induce a clique.

We define a set of k vertices of G, one from each V_i, as follows: in V_i we select the vertex $f(i)$ if there are $f(i)$ vertices with color 1 in $C_{i,1}$. We call these k vertices the selected vertices of G.

We now observe that if there are $f(i)$ vertices with color 1 in $C_{i,j}$, then there are $n - f(i)$ vertices with color 1 in $C_{i,j+1}$. To see this observe that if there were more than $n - f(i)$ vertices with color 1 in $C_{i,j+1}$ this would violate vertex $b^A_{i,j}$, which also has color 1 and is connected to $C_{i,j} \cup C_{i,j+1}$. If there were fewer, this would violate the vertex $b^B_{i,j}$, which has color 2. Hence, for any $j \in \{1, \ldots, m\}$ we have that $C_{i,2j-1}$ contains $f(i)$ vertices with color 1, while $C_{i,2j}$ contains $n - f(i)$ vertices with color 1.

We now want to show that every selected edge is incident on two selected vertices to complete the proof. Consider a c_j that corresponds to a selected edge. Since c_j received color 2, because of step 15 all vertices of $H^1_j, L^1_j, H^2_j, L^2_j$ must have color 1. Consider now the vertices h^1_j, l^1_j, which also have color 1 because of step 12. If h^1_j is connected to $C_{i_1,2j-1}$ and l^1_j is connected to $C_{i_1,2j}$, then h^1_j has $(\Delta^* - n) + |H^1_j| + f(i_1)$ neighbors with color 1, while l^1_j has $(\Delta^* - n) + |L^1_j| + n - f(i_1)$ such neighbors. But $|L^1_j| = n - |H^1_j|$. We therefore have $f(i_1) \leq n - |H^1_j|$ as well as $f(i_1) \geq |L^1_j| = n - |H^1_j|$. Therefore, $f(i_1) = |L^1_j|$ and this implies by construction that edge j is incident on vertex $f(i_1)$ of V_{i_1}. ∎

LEMMA 4.5. *For the graph* $H(G, k, \chi_{\mathrm{d}})$ *described in the previous section* $\mathrm{pw}(H) = O(k + \chi_{\mathrm{d}})$.

Proof. We first invoke Lemma 4.2 to replace all palette and equality gadgets with edges. It suffices to show that the pathwidth of the resulting graph is $O(k)$. We continue by removing from the graph the vertices p_A, p_B, c_U. This does not decrease the pathwidth by more than 3, since these vertices can be added to all bags. In the remaining graph we remove all leaves and isolated vertices. It is not hard to see that this does not decrease pathwidth by more than 1, since if we find a path decomposition of the remaining graph, we can reinsert the leaves as follows: for each leaf v we find the smallest bag in the decomposition that contains its neighbor and insert after it a copy of the same bag with v added. We note that removing all leaves deletes from the graph all vertices added for budget-setting, as well as the remaining vertices of the palette part.

What remains then is to bound the pathwidth of the graph induced by the backbone vertices $b^l_{i,j}$, the choice vertices in sets $C_{i,j}$, and the edge representation vertices. We construct a backbone of a path decomposition as follows: for each $j \in \{1, \ldots, m\}$ we construct a bag that contains all $b^l_{i,2j-1}, b^l_{i,2j}$, and $b^l_{i,2j+1}$ (if they exist), as well as $h^1_j, l^1_j, h^2_j, l^2_j, c_j$. We connect these bags in a path in increasing order of j. All these bags have with at most $O(k)$.

We now observe that for every remaining vertex of the graph, there is a bag in the path decomposition that we have constructed that contains all its neighbors. We therefore do the following: for every remaining vertex v, we find the smallest bag of the path decomposition that contains its neighborhood, and insert after it a copy of this bag with v added. This process results in a valid path decomposition, and it does not increase the size of the largest bag by more than 1. ∎

The proof of Theorem 4.1 now follows directly from Lemmata 4.3,4.4,4.5.

5. Exact Algorithms for Treewidth and Other Parameters. In this section we present several exact algorithms for Defective Coloring. Theorem 5.1 gives a treewidth-based algorithm which can be obtained using standard techniques. We assume that the reader is familiar with dynamic programming on tree decompositions, as described in standard textbooks [18]. Essentially the same algorithm was already sketched in [9], but we give another version here for the sake of completeness and because it is a building block for the approximation algorithm of Theorem 6.2. Theorem 5.2 uses a win/win argument to show that the problem is FPT parameterized by fvs when $\chi_{\mathrm{d}} \neq 2$ and therefore explains why the reduction presented in Section 3 only works for 2 colors. Theorem 5.3 uses a similar argument to show that the problem is FPT parameterized by vc (for any χ_{d}).

Theorem 5.1. *There is an algorithm which, given a graph $G = (V, E)$, parameters $\chi_{\mathrm{d}}, \Delta^*$, and a tree decomposition of G of width* tw*, decides if G admits a $(\chi_{\mathrm{d}}, \Delta^*)$-coloring in time $(\chi_{\mathrm{d}}\Delta^*)^{O(\mathrm{tw})} n^{O(1)}$.*

Proof. The algorithm uses standard dynamic programming techniques, so we sketch some of the details. We assume we are given a nice tree decomposition, as defined in [12]. For each bag B_t of the decomposition we denote by $B_t^{\downarrow}$ the set of vertices included in bags in the sub-tree of the decomposition rooted at B_t. We will maintain in each bag B_t a dynamic programming table $D_t \subseteq (\{1, \dots, \chi_{\mathrm{d}}\} \times \{0, \dots, \Delta^*\})^{|B_t|}$. Informally, each element $s \in (\{1, \dots, \chi_{\mathrm{d}}\} \times \{0, \dots, \Delta^*\})^{|B_t|}$ is the signature of a partial solution: we interpret s as a function which, for each vertex in B_t tells us its color, as well as the number of neighbors this vertex has in $B_t^{\downarrow} \setminus B_t$ that share the same color. The invariant we want to maintain is that $s \in D_t$ if and only if there exists a coloring of $B_t^{\downarrow}$ with signature s. We can now build the DP table inductively:

- For a Leaf node $B_t = \{u\}$, D_t contains all signatures $s = (c_u, 0)$, for any $c_u \in \{1, \dots, \chi_{\mathrm{d}}\}$.
- For an Introduce node B_t with child $B_{t'}$ such that $B_t = B_{t'} \cup \{u\}$, for any $s' \in D_{t'}$, and for any $c_u \in \{1, \dots, \chi_{\mathrm{d}}\}$, we add to D_t a signature s which agrees with s' on $B_{t'}$ and contains the pair $(c_u, 0)$ for vertex u.
- For a Forget node B_t with child $B_{t'}$ such that $B_t = B_{t'} \setminus \{u\}$ for every signature $s' \in D_{t'}$ we do the following: let (c_u, d_u) be the pair contained in s' corresponding to vertex u. Let $S_u \subseteq B_{t'}$ be the set of vertices of $B_{t'}$ which are given color c_u according to s' and which are neighbors of u. We check two conditions: first that $d_u + |S_u| \leq \Delta^*$; second, that for all $v \in S_u$ such that s' contains the pair (c_u, d_v) we have $d_v \leq \Delta^* - 1$. If both conditions hold, we add to D_t a signature s that agrees with s' on $B_t \setminus S_u$, and that for each $v \in S_u$ such that s' returns (c_u, d_v), returns the pair $(c_u, d_v + 1)$.
- For a Join node B_t with children B_{t_1}, B_{t_2}, (such that $B_t = B_{t_1} = B_{t_2}$) we do the following: for each $s_1 \in D_{t_1}$ and each $s_2 \in D_{t_2}$ we check the following two conditions for all $u \in B_t$: if s_1 returns (c_{u_1}, d_{u_1}) for u and s_2 returns (c_{u_2}, d_{u_2}) we check if $c_{u_1} = c_{u_2}$; and we check if $d_{u_1} + d_{u_2} \leq \Delta^*$. If both conditions hold for all $u \in B_t$ we say that s_1, s_2 are compatible, and we add to D_t a signature s which for $u \in B_t$ contains the pair $(c_{u_1}, d_{u_1} + d_{u_2})$.

It is not hard to see that the above operations can be performed in time polynomial in the size of the table, which is upper-bounded by $(\chi_{\mathrm{d}}(\Delta^*+1))^{\mathrm{tw}}$. We can then prove by induction that a signature appears in a table D_t if and only if a coloring with this signature exists for $B_t^{\downarrow}$. If we assume, without loss of generality, that the root bag contains a single vertex, we can check if the graph admits a $(\chi_{\mathrm{d}}, \Delta^*)$-coloring by checking if the table of the root bag is non-empty. ∎

THEOREM 5.2. DEFECTIVE COLORING *is FPT parameterized by* fvs *for* $\chi_{\mathrm{d}} \neq 2$. *More precisely, there exists an algorithm which given a graph* $G = (V, E)$*, parameters* $\chi_{\mathrm{d}}, \Delta^*$*, with* $\chi_{\mathrm{d}} \neq 2$*, and a feedback vertex set of* G *of size* fvs*, decides if* G *admits a* $(\chi_{\mathrm{d}}, \Delta^*)$*-coloring in time* $\mathrm{fvs}^{O(\mathrm{fvs})} n^{O(1)}$.

Proof. We use a win/win argument. First, note that we can assume that $\chi_{\mathrm{d}} \geq 3$, since if $\chi_{\mathrm{d}} = 1$ the problem is trivial. Furthermore, if $\chi_{\mathrm{d}} \geq \mathrm{fvs}+2$ then we can produce a $(\chi_{\mathrm{d}}, \Delta^*)$-coloring by giving a distinct color to each vertex of the feedback vertex set and properly two-coloring the remaining graph. Hence, we assume in the remainder that $3 \leq \chi_{\mathrm{d}} \leq \mathrm{fvs} + 2$.

Now, if $\Delta^* \leq \mathrm{fvs}$, then we can use the algorithm of Theorem 5.1. Because of Lemma 2.2 this algorithm will run in time $\mathrm{fvs}^{O(\mathrm{fvs})} n^{O(1)}$.

Finally, suppose that $\Delta^* > \mathrm{fvs}$. In this case the answer is always Yes. To see this we can produce a coloring as follows: we use a single color for all the vertices of the feedback vertex set. Since $\chi_{\mathrm{d}} \geq 3$, there are at least two other colors available, so we use them to properly color the remaining forest. This is a valid $(\chi_{\mathrm{d}}, \Delta^*)$-coloring, since the only vertices that may have neighbors of the same color belong in the feedback vertex set, and these can have at most $\mathrm{fvs} - 1 < \Delta^*$ neighbors with the same color. ∎

THEOREM 5.3. DEFECTIVE COLORING *is FPT parameterized by* vc. *More precisely, there exists an algorithm which, given a graph* $G = (V, E)$*, parameters* $\chi_{\mathrm{d}}, \Delta^*$*, and a vertex cover of* G *of size* vc*, decides if* G *admits a* $(\chi_{\mathrm{d}}, \Delta^*)$*-coloring in time* $\mathrm{vc}^{O(\mathrm{vc})} n^{O(1)}$.

Proof. The proof is essentially identical to that of Theorem 5.2. We can assume that $\chi_{\mathrm{d}} \leq \mathrm{vc}$ (otherwise we use a distinct color for each vertex of the vertex cover, and a single color for the independent set), and that $\chi_{\mathrm{d}} \geq 2$ (otherwise the problem is trivial). If $\Delta^* \leq \mathrm{vc}$ we can use the algorithm of Theorem 5.1, otherwise we can use a single color for the vertex cover and another for the independent set. ∎

6. Approximation Algorithms and Lower Bounds.

We now present two approximation algorithms which run in FPT time parameterized by treewidth. The first algorithm (Theorem 6.2) is an *FPT approximation scheme* which, given a desired number of colors χ_{d}, is able to approximate the minimum feasible value of Δ^* for this value of χ_{d} arbitrarily well (that is, within a factor $(1+\epsilon)$). The second algorithm, which also runs in FPT time parameterized by treewidth, given a desired value for Δ^*, produces a solution that approximates the minimum number of colors χ_{d} within a factor of 2.

These results raise the question of whether it is possible to approximate χ_{d} as well as we can approximate Δ^*, that is, whether there exists an algorithm which comes within a factor $(1+\epsilon)$ (rather than 2) of the optimal number of colors. As a first response, one could observe that such an algorithm probably cannot exist, because the problem is already hard when $\chi_{\mathrm{d}} = 2$, and therefore an FPT algorithm with multiplicative error less than $3/2$ would imply that FPT=W[1]. However, this does not satisfactorily settle the problem as it does not rule out an algorithm that achieves a much better approximation ratio, if we allow it to also have a small additive error in the number of colors. Indeed, as we observe in Corollary 6.7, it is possible to obtain an algorithm which runs in FPT time parameterized by feedback vertex set and has an additive error of only 1, as a consequence of the fact that the problem is FPT for $\chi_{\mathrm{d}} \geq 3$. This poses the question of whether we can design an FPT algorithm parameterized by treewidth which, given a $(\chi_{\mathrm{d}}, \Delta^*)$-colorable graph, produces a coloring with $\rho\chi_{\mathrm{d}} + O(1)$ colors, for $\rho < 3/2$.

In the second part of this section we settle this question negatively by showing, using a recursive construction that builds on Theorem 3.1, that such an algorithm cannot exist. More precisely, we present a gap-introducing version of our reduction: the ratio between the number of colors needed to color Yes and No instances remains 3/2, even as the given χ_{d} increases. This shows that the "correct" multiplicative approximation ratio for this problem really lies somewhere between 3/2 and 2, or in other words, that there are significant barriers impeding the design of a better than 3/2 FPT approximation for χ_{d}, beyond the simple fact that 2-coloring is hard.

6.1. Approximation Algorithms. Our first approximation algorithm, which is an approximation scheme for the optimal value of Δ^*, relies on a method introduced in [39] (see also [3, 34, 35]), and a theorem of [11]. The high-level idea is the following: intuitively, the obstacle that stops us from obtaining an FPT running time with the dynamic programming algorithm of Theorem 5.1 is that the dynamic program is forced to store some potentially large values for each vertex. More specifically, to characterize a partial solution we need to remember not just the color of each vertex in a bag, but also how many neighbors with the same color this vertex has already seen (which is a value that can go up to Δ^*). The main trick now is to "round" these values in order to decrease the number of possible states a vertex can be found in. To do this, we select an appropriate value δ (polynomial in $\frac{\epsilon}{\log n}$), and try to replace every value that the dynamic program would calculate with the next higher integer power of $(1+\delta)$. This has the advantage of limiting the number of possible values from Δ^* to $\log_{(1+\delta)} \Delta^* \approx \frac{\log \Delta^*}{\delta}$, and this is sufficient to obtain the promised running time. The problem is now that the rounding we applied introduces an approximation error, which is initially a factor of at most $(1+\delta)$, but may increase each time we apply an arithmetic operation as part of the algorithm. To show that this error does not get out of control we show that in any bag of the tree all values stored are within a factor $(1+\delta)^h$ of the correct ones, where h is the height of the bag. We then use a theorem of Bodlaender and Hagerup [11] which states that any tree decomposition can be balanced in such a way that its height is at most $O(\log n)$, and as a result we obtain that all values are sufficiently close to being correct.

The second algorithm we present in this section (Theorem 6.4) uses the approximation scheme for Δ^* to obtain an FPT 2-approximation for χ_{d}. The idea here is that, given a $(\chi_{\mathrm{d}}, \Delta^*)$-colorable graph, we first produce a $(\chi_{\mathrm{d}}, (1+\epsilon)\Delta^*)$-coloring using the algorithm of Theorem 6.2, and then apply a procedure which uses 2 colors for each color class of this solution but manages to divide by two the number of neighbors with the same color of every vertex. This is achieved with a simple polynomial-time local search procedure.

THEOREM 6.1. *[11] There is a polynomial-time algorithm which, given a graph $G = (V, E)$ and a tree decomposition of G of width* tw*, produces a tree decomposition of G of width at most* $3\mathrm{tw} + 2$ *and height* $O(\log n)$.

THEOREM 6.2. *There is an algorithm which, given a graph $G = (V, E)$, parameters $\chi_{\mathrm{d}}, \Delta^*$, a tree decomposition of G of width* tw*, and an error parameter $\epsilon > 0$, either returns a $(\chi_{\mathrm{d}}, (1+\epsilon)\Delta^*)$-coloring of G, or correctly concludes that G does not admit a*

$(\chi_{\mathrm{d}}, \Delta^*)$*-coloring, in time* $(\mathrm{tw}/\epsilon)^{O(\mathrm{tw})} n^{O(1)}$.

Proof. Our first step is to invoke Theorem 6.1 to obtain a tree decomposition of width $O(\mathrm{tw})$ and height $O(\log n)$. We then define a value $\delta = \frac{\epsilon}{\log^2 n}$ and the set $\Sigma = \{0\} \cup \{(1+\delta)^i \mid i \in \mathbb{N}, (1+\delta)^i \leq (1+\epsilon)\Delta^*\}$. In other words, the set Σ contains (in addition to 0), all positive integer powers of $(1+\delta)$ with value at most $(1+\epsilon)\Delta^*$. We note that $|\Sigma| \leq 1 + \log_{(1+\delta)}((1+\epsilon)\Delta^*) = O(\log \Delta^*/\delta)$, where we have used the properties $\log_a b = \ln b / \ln a$, and $\ln(1+x) \geq x/2$ for x a sufficiently small positive constant (that is, for sufficiently large n). Taking into account the value of δ we have selected, and the fact that $\Delta^* \leq n$, we have $|\Sigma| = O(\log^3 n/\epsilon)$.

We now follow the outline of the algorithm of Theorem 5.1, with the difference that we now define a DP table for bag B_t as $D_t \subseteq (\{1, \ldots, \chi_{\mathrm{d}}\} \times \Sigma)^{|B_t|}$. Again, we interpret the elements of D_t as functions which, for each vertex in B_t return a color and an *approximate* number of neighbors that have the same color as this vertex in $B_t^{\downarrow} \setminus B_t$.

More precisely, if a bag B_t is at height h (that is, its maximum distance from a leaf bag in the sub-tree rooted at B_t is h) we will maintains the following two invariants:

1. If there exists a coloring $\mathbf{c}$ of $B_t^{\downarrow}$ such that all vertices of $B_t^{\downarrow} \setminus B_t$ have at most Δ^* neighbors of the same color, and all vertices of B_t have at most Δ^* neighbors of the same color in $B_t^{\downarrow} \setminus B_t$, then there exists $s \in D_t$ which assigns the same colors as $\mathbf{c}$ to B_t; and which, if $u \in B_t$ has d'_u neighbors with the same color in $B_t^{\downarrow} \setminus B_t$ in $\mathbf{c}$, returns value $d_u \leq (1+\delta)^h d'_u$ for vertex u, where $d_u \in \Sigma$.
2. If there exists a signature $s \in D_t$, then there exists a coloring $\mathbf{c}$ of $B_t^{\downarrow}$ such that all vertices of $B_t^{\downarrow} \setminus B_t$ have at most $(1+\epsilon)\Delta^*$ neighbors of the same color; all vertices of B_t take in $\mathbf{c}$ the colors described in s; if s dictates that a vertex $u \in B_t$ has d_u neighbors with the same color in $B_t^{\downarrow} \setminus B_t$, then u has at most d_u neighbors with the same color in $B_t^{\downarrow} \setminus B_t$ according to coloring $\mathbf{c}$.

The first of the two properties above implies that, if there exists a $(\chi_{\mathrm{d}}, \Delta^*)$-coloring of G, the algorithm will be able to find some entry in the table of the root bag that will allows us to construct a $(\chi_{\mathrm{d}}, (1+\delta)^H)$-coloring, where H is the height of the tree decomposition. We recall now that $H = O(\log n)$, therefore, $(1+\delta)^H \leq e^{\delta H} \leq e^{O(\epsilon/\log n)} \leq 1+\epsilon$. Hence, if we establish the first property, we know that if a $(\chi_{\mathrm{d}}, \Delta^*)$-coloring exists, the algorithm will be able to find a $(\chi_{\mathrm{d}}, (1+\epsilon)\Delta^*)$-coloring. Conversely, the second property assures us that, if the algorithm places a signature s in a DP table, there must exist a coloring that matches this signature.

In order to establish these invariants we must make a further modification to the algorithm of Theorem 5.1. We recall that the algorithm makes some arithmetic calculation in Forget nodes (where the value d_v of neighbors of the forgotten node with the same color is increased by 1); and in Join nodes (where values d_{u_1}, d_{u_2} corresponding to the same node are added). The problem here is that even if the values stored are integer powers of $(1+\delta)$, the results of these additions are not necessarily such integer powers. Hence, our algorithm will simply "round up" the result of these additions to the closest integer power of $(1+\delta)$. Formally, instead of the value $d_v + 1$ we use the value $(1+\delta)^{\lceil \log_{(1+\delta)}(d_v+1) \rceil}$, and instead of the value $d_{u_1} + d_{u_2}$ we use the value $(1+\delta)^{\lceil \log_{(1+\delta)}(d_{u_1}+d_{u_2}) \rceil}$.

We can now establish the two properties by induction. The two interesting cases are Forget and Join nodes. For a Join node of height h and the first property, if we have established by induction that for the two values d_{u_1}, d_{u_2} stored in the children's tables we have $d_{u_1} \leq (1+\delta)^{h-1} d'_{u_1}$, $d_{u_2} \leq (1+\delta)^{h-1} d'_{u_2}$, where d'_{u_1}, d'_{u_2} are as described in the first property, then $d_{u_1} + d_{u_2} \le (1+\delta)^{h-1}(d'_{u_1} + d'_{u_2})$. However, for the new value we calculate we have $d_u \le (1+\delta)(d_{u_1} + d_{u_2}) \le (1+\delta)^h(d'_{u_1} + d'_{u_2}) = (1+\delta)^h d'_u$. For the second property, observe that since we always round up, the value stored in the table will always be at least as high as the true number of neighbors of a vertex in the coloring $\mathbf{c}$. Calculations are similar for Forget nodes.

Because of the above we have an algorithm that runs in time polynomial in $|D_t| = (\chi_{\rm d}|\Sigma|)^{O(\mathrm{tw})}$. We can assume without loss of generality that $\chi_{\rm d} \le \mathrm{tw}+1$, otherwise by Lemma 2.2 the graph can be easily properly colored. By the observations of $|\Sigma|$, specifically the fact that $|\Sigma| = O(\log \Delta^*/\delta) = O(\log^3 n/\epsilon)$, we therefore have that the running time is $(\mathrm{tw} \log n/\epsilon)^{O(\mathrm{tw})}$. A well-known win/win argument allows us to obtain the promised bound as follows: if $\mathrm{tw} \le \sqrt{\log n}$, this running time is in fact polynomial in $n, 1/\epsilon$, so we are done; if $\sqrt{logn} \le \mathrm{tw}$ then $\log n \le \mathrm{tw}^2$ and the running time is upper bounded by $(\mathrm{tw}/\epsilon)^{O(\mathrm{tw})}$. ∎

For our second approximation algorithm, we first state a helpful lemma.

LEMMA 6.3. *There exists a polynomial-time algorithm which, given a graph with maximum degree Δ, produces a two-coloring of that graph where all vertices have at most $\Delta/2$ neighbors of the same color.*

Proof. We run what is essentially a local search algorithm for MAX CUT. Initially, color all vertices with color 1. Then, as long as there exists a vertex u such that the majority of its neighbors have the same color as u, we change the color of u. We continue with this process until all vertices have a majority of their neighbors with a different color. In that case the claim follows. To see that this procedure terminates in polynomial time, observe that in each step we increase the number of edges that connect vertices of different colors. ∎

Combining Lemma 6.3 with the algorithm of Theorem 6.4 gives the following result:

THEOREM 6.4. *There is an algorithm which, given a graph $G = (V, E)$, parameters $\chi_{\rm d}, \Delta^*$, and a tree decomposition of G of width* tw, *either returns a $(2\chi_{\rm d}, \Delta^*)$-coloring of G, or correctly concludes that G does not admit a $(\chi_{\rm d}, \Delta^*)$-coloring, in time* $(\mathrm{tw})^{O(\mathrm{tw})} n^{O(1)}$.

Proof. We assume without loss of generality that Δ^* is sufficiently large (e.g. $\Delta^* \ge 20$), otherwise we can solve the problem exactly by using the fact that $\chi_{\rm d}$ is bounded by tw (by Lemma 2.2) and the algorithm of Theorem 5.1. We invoke the algorithm of Theorem 6.2, setting $\epsilon = 1/10$. The algorithm runs in the promised running time. If it reports that G does not admit a $(\chi_{\rm d}, \Delta^*)$-coloring, we output the same answer and we are done.

Suppose that the algorithm of Theorem 6.2 returned a $(\chi_{\rm d}, \frac{11}{10}\Delta^*)$-coloring of G. We transform this to a $(2\chi_{\rm d}, \Delta^*)$-coloring by using Lemma 6.3.

We consider each color class in the returned coloring of G separately. Each class induces a graph with maximum degree $\frac{11}{10}\Delta^*$. According to Lemma 6.3, we can two-color this graph so that no vertex has more than $\frac{11}{20}\Delta^* \le \Delta^*$ neighbors with the same color. We produce such a two-coloring for the graph induced by each color class using two new colors. Hence, the end result is a $(2\chi_{\rm d}, \frac{11}{20}\Delta^*)$-coloring of G, which is also a valid $(2\chi_{\rm d}, \Delta^*)$-coloring. ∎

6.2. Hardness of Approximation. The main result of this section is that $\chi_{\rm d}$ cannot be approximated with a factor better than 3/2 in FPT time (for parameters tree-depth, pathwidth, or treewidth), even if we allow the algorithm to also have a constant additive error. We remark that an FPT algorithm with additive error 1 is easy to obtain for feedback vertex set (Corollary 6.7).

THEOREM 6.5. *For any fixed* $\chi_{\mathrm{d}} > 0$*, if there exists an algorithm which, given a graph* $G = (V, E)$ *and a* $\Delta^* \geq 0$*, correctly distinguishes between the case that* G *admits a* $(2\chi_{\mathrm{d}}, \Delta^*)$*-coloring, and the case that* G *does not admit a* $(3\chi_{\mathrm{d}} - 1, \Delta^*)$*-coloring in FPT time parameterized by* $\mathrm{td}(G)$*, then FPT=W[1].*

Proof. First, observe that the theorem already follows for $\chi_{\mathrm{d}} = 1$ by Theorem 3.1, which states that it is W[1]-hard parameterized by $\mathrm{td}(G)$ to decide if a graph admits a $(2, \Delta^*)$-coloring. Let G^1 be the graph produced in the reduction of Theorem 3.1. By repeated composition we will construct, for any χ_{d}, a graph $G^{\chi_{\mathrm{d}}}$ such that either $G^{\chi_{\mathrm{d}}}$ admits a $(2\chi_{\mathrm{d}}, \Delta^*)$-coloring, or it does not admit a $(3\chi_{\mathrm{d}} - 1, \Delta^*)$-coloring, depending on whether G^1 admits a $(2, \Delta^*)$-coloring.

Suppose that we have constructed the graph $G^{\chi_{\mathrm{d}}}$, for some χ_{d}. We describe how to build the graph $G^{\chi_{\mathrm{d}}+1}$. We start with a copy of G^1, which we call the main part of our construction. We will add to this many disjoint copies of $G^{\chi_{\mathrm{d}}}$ and appropriately connect them to G^1 to obtain $G^{\chi_{\mathrm{d}}+1}$.

Recall that the graph G^1 contains two palette vertices p_A, p_B, each connected to Δ^* neighbors p_j^i, $i \in \{1, \ldots, \Delta^*\}$, $j \in \{A, B\}$ with both edges and equality gadgets. Furthermore, recall that for two colors, an equality gadget with endpoints p_j, p_j^i is an independent set on $2\Delta^* + 1$ vertices which are common neighbors of p_j and p_j^i.

For each $j \in \{A, B\}$, each $i \in \{1, \ldots, \Delta^*\}$, and each internal vertex v of the equality gadget $Q(p_j, p_j^i)$ added in step 3 we add to the main graph $\binom{3\chi_{\mathrm{d}}+2}{3\chi_{\mathrm{d}}}\Delta^* + 1$ disjoint copies of $G^{\chi_{\mathrm{d}}}$ and connect all their vertices to p_j, p_j^i, and v.

Now, for every vertex v of G^1 that is not part of the palette (that is, every vertex that was not constructed in steps 1-5), we add another $\binom{3\chi_{\mathrm{d}}+2}{3\chi_{\mathrm{d}}}\Delta^* + 1$ disjoint copies of $G^{\chi_{\mathrm{d}}}$ and connect all their vertices to p_A, p_B, and v.

This completes the construction. We now need to establish three properties: that if G^1 admits a $(2, \Delta^*)$-coloring then $G^{\chi_{\mathrm{d}}+1}$ admits a $(2\chi_{\mathrm{d}} + 2, \Delta^*)$-coloring; that if G^1 does not admit a $(2, \Delta^*)$-coloring then $G^{\chi_{\mathrm{d}}+1}$ does not admit a $(3\chi_{\mathrm{d}} + 2, \Delta^*)$-coloring; and that the tree-depth of $G^{\chi_{\mathrm{d}}+1}$ did not increase too much.

We proceed by induction and assume that all the above have been shown for $G^{\chi_{\mathrm{d}}}$. For the first property, if G^1 admits a $(2, \Delta)$-coloring and $G^{\chi_{\mathrm{d}}}$ admits a $(2\chi_{\mathrm{d}}, \Delta^*)$-coloring, then we can construct a coloring of $G^{\chi_{\mathrm{d}}+1}$ by taking the same coloring with $2\chi_{\mathrm{d}}$ colors for all the copies of $G^{\chi_{\mathrm{d}}}$, and using two new colors to color the main graph G^1.

For the second property, suppose that we know that a $(3\chi_{\mathrm{d}} - 1, \Delta^*)$-coloring of $G^{\chi_{\mathrm{d}}}$ implies the existence of a $(2, \Delta^*)$-coloring of G^1. We want to show that a $(3\chi_{\mathrm{d}} + 2, \Delta^*)$-coloring of $G^{\chi_{\mathrm{d}}+1}$ also implies a $(2, \Delta^*)$-coloring of G^1. Suppose then that we have such a $(3\chi_{\mathrm{d}} + 2, \Delta^*)$-coloring of $G^{\chi_{\mathrm{d}}+1}$. If a copy of $G^{\chi_{\mathrm{d}}}$ included in $G^{\chi_{\mathrm{d}}+1}$ uses at most $3\chi_{\mathrm{d}} - 1$ colors, we are done, since this implies the existence of a $(2, \Delta^*)$-coloring of G^1. Therefore, assume that all copies of $G^{\chi_{\mathrm{d}}+1}$ use at least $3\chi_{\mathrm{d}}$ colors.

Consider now two vertices p_j, p_j^i, for some $j \in \{A, B\}$, $i \in \{1, \ldots, \Delta^*\}$. We claim that they must receive the same color. To see this, take an internal vertex v of the equality gadget $Q(p_j, p_j^i)$ and recall that we have added $\binom{3\chi_{\mathrm{d}}+2}{3\chi_{\mathrm{d}}}\Delta^* + 1$ disjoint copies of $G^{\chi_{\mathrm{d}}}$ connected to p_j, p_j^i, v. Hence, there is some set of $3\chi_{\mathrm{d}}$ colors that appears in at least $\Delta^* + 1$ of these copies, and therefore cannot be used in p_j, p_j^i, v. Therefore, if p_j, p_j^i do not share a color, all the $2\Delta^* + 1$ internal vertices of the equality gadget share the color of one of the two, which violates the correctness of the coloring. We conclude that p_A has Δ^* neighbors with its own color, as does p_B, therefore, since they are connected, p_A, p_B use distinct colors.

Consider now any other vertex v of the main graph. Again, we have added $\binom{3\chi_d+2}{3\chi_d}\Delta^* + 1$ disjoint copies of G^{χ_d} connected to p_A, p_B, v, hence there is a set of $3\chi_d$ colors which appears in Δ^*+1 copies and is therefore not used by p_A, p_B, v. Since there are $3\chi_d + 2$ colors overall and p_A, p_B use distinct colors, we conclude that v uses either the color of p_A or that of p_B. Hence, the coloring of G^{χ_d+1} contains a 2-coloring of G^1.

For the final property, suppose that $\mathrm{td}(G^{\chi_d}) \le \chi_d \mathrm{td}(G^1) + 2\chi_d$. We want to establish that $\mathrm{td}(G^{\chi_d+1}) \le (\chi_d+1)\mathrm{td}(G^1) + 2\chi_d + 2$. To see this, we construct a tree for G^{χ_d+1} as follows, the two top vertices are p_A, p_B, and below these we place a tree whose completion contains G^1 (hence we have at most $\mathrm{td}(G^1) + 2$ levels now). For every copy of G^{χ_d} that was connected to p_A, p_B, and a vertex v, we find v and attach below it a tree whose completion contains G^{χ_d}. Similarly, for every copy of G^{χ_d} attached to p_j, p_j^i, and a vertex v, for some $j \in \{A, B\}$, $i \in \{1, \dots, \Delta^*\}$, one of the vertices v, p_j^i is a descendant of the other in the current tree (since they are connected); we attach a tree containing G^{χ_d} to this descendant. The total number of levels of the tree is therefore $\mathrm{td}(G^1) + 2 + \mathrm{td}(G^{\chi_d}) \le (\chi_d+1)\mathrm{td}(G^1) + \chi_d + 2$, as desired. ∎

COROLLARY 6.6. *For any constants $\delta_1, \delta_2 > 0$, if there exists an algorithm which, given a graph $G = (V, E)$ that admits a (χ_d, Δ^*)-coloring and parameters χ_d, Δ^*, is able to produce a $((\frac{3}{2} - \delta_1)\chi_d + \delta_2, \Delta^*)$-coloring of G in FPT time parameterized by $\mathrm{td}(G)$, then FPT=W[1].*

Proof. Fix some constants δ_1, δ_2. We invoke Theorem 6.5 with $\chi_d = \lceil \frac{\delta_2+1}{\delta_1} \rceil$. The graph produced either admits a $(2\chi_d, \Delta)$-coloring or does not admit a $(3\chi_d - 1, \Delta)$-coloring. Suppose that the algorithm described in this corollary exists. Then, in the former case it produces a coloring with at most $(\frac{3}{2} - \delta_1) \cdot 2\lceil \frac{\delta_2+1}{\delta_1} \rceil + \delta_2 = 3\lceil \frac{\delta_2+1}{\delta_1} \rceil - 2\delta_1 \lceil \frac{\delta_2+1}{\delta_1} \rceil + \delta_2 \le 3\chi_d - 2(\delta_2+1) + \delta_2 \le 3\chi_d - 1$ colors. Hence, the algorithm would be able to distinguish the two cases of a W[1]-hard problem. ∎

COROLLARY 6.7. *There is an algorithm which, given a graph $G = (V, E)$, parameters χ_d, Δ^*, and a feedback vertex set of G of size* fvs, *either returns a $(\chi_d + 1, \Delta^*)$-coloring of G, or correctly concludes that G does not admit a (χ_d, Δ^*)-coloring, in time $(\mathrm{fvs})^{O(\mathrm{fvs})} n^{O(1)}$.*

Proof. If $\chi_d \ge 3$ we simply invoke Theorem 5.2. If $\chi_d = 2$ we invoke the same algorithm with $\chi_d = 3$. If the algorithm produces a coloring, we output that as the solution, otherwise we can report that no (χ_d, Δ^*)-coloring exists. ∎

7. Conclusions.

In this paper we classified the complexity of DEFECTIVE COLORING with respect to some of the most well-studied graph parameters, given essentially tight ETH-based lower bounds for pathwidth and treewidth, and explored the parameterized approximability of the problem. Though this gives a good first overview of the problem's parameterized complexity landscape, there are several questions worth investigating next. First, is it possible to make the lower bounds of Section 4 even tighter, by precisely determining the base of the exponent in the algorithm's dependence? This would presumably rely on a stronger complexity assumption such as the SETH, as in [41]. Second, can we determine the complexity of the problem with respect to other structural parameters, such as clique-width [15], modular-width [25], or neighborhood diversity [38]? For some of these parameters the existence of FPT algorithms is already ruled out by the fact that DEFECTIVE COLORING is NP-hard on cographs [9], however the complexity of the problem is unknown if we also add χ_{d} or Δ^* as a parameter. Finally, it would be very interesting to close the gap between 2 and 3/2 on the performance of the best treewidth-parameterized FPT approximation for χ_{d}.

REFERENCES

[1] N. Achuthan, N. R. Achuthan, and M. Simanihuruk, *On minimal triangle-free graphs with prescribed k-defective chromatic number*, Discrete Mathematics, 311 (2011), pp. 1119–1127, https://doi.org/10.1016/j.disc.2010.08.013, http://dx.doi.org/10.1016/j.disc.2010.08.013.

[2] J. A. Andrews and M. S. Jacobson, *On a generalization of chromatic number*, Congressus Numerantium, 47 (1985), pp. 33–48.

[3] E. Angel, E. Bampis, B. Escoffier, and M. Lampis, *Parameterized power vertex cover*, Discrete Mathematics & Theoretical Computer Science, 20 (2018), http://dmtcs.episciences.org/4873.

[4] P. Angelini, M. A. Bekos, F. D. Luca, W. Didimo, M. Kaufmann, S. G. Kobourov, F. Montecchiani, C. N. Raftopoulou, V. Roselli, and A. Symvonis, *Vertex-coloring with defects*, J. Graph Algorithms Appl., 21 (2017), pp. 313–340, https://doi.org/10.7155/jgaa.00418, http://dx.doi.org/10.7155/jgaa.00418.

[5] J. Araújo, J. Bermond, F. Giroire, F. Havet, D. Mazauric, and R. Modrzejewski, *Weighted improper colouring*, J. Discrete Algorithms, 16 (2012), pp. 53–66, https://doi.org/10.1016/j.jda.2012.07.001, http://dx.doi.org/10.1016/j.jda.2012.07.001.

[6] D. Archdeacon, *A note on defective colorings of graphs in surfaces*, Journal of Graph Theory, 11 (1987), pp. 517–519, https://doi.org/10.1002/jgt.3190110408, http://dx.doi.org/10.1002/jgt.3190110408.

[7] C. Archetti, N. Bianchessi, A. Hertz, A. Colombet, and F. Gagnon, *Directed weighted improper coloring for cellular channel allocation*, Discrete Applied Mathematics, 182 (2015), pp. 46–60, https://doi.org/10.1016/j.dam.2013.11.018, http://dx.doi.org/10.1016/j.dam.2013.11.018.

[8] J. Bang-Jensen and M. M. Halldórsson, *Vertex coloring edge-weighted digraphs*, Inf. Process. Lett., 115 (2015), pp. 791–796, https://doi.org/10.1016/j.ipl.2015.05.007, http://dx.doi.org/10.1016/j.ipl.2015.05.007.

[9] R. Belmonte, M. Lampis, and V. Mitsou, *Defective coloring on classes of perfect graphs*, CoRR, abs/1702.08903 (2017), http://arxiv.org/abs/1702.08903.

[10] J. Bermond, F. Havet, F. Huc, and C. L. Sales, *Improper coloring of weighted grid and hexagonal graphs*, Discrete Math., Alg. and Appl., 2 (2010), pp. 395–412, https://doi.org/10.1142/S1793830910000747, http://dx.doi.org/10.1142/S1793830910000747.

[11] H. L. Bodlaender and T. Hagerup, *Parallel algorithms with optimal speedup for bounded treewidth*, SIAM J. Comput., 27 (1998), pp. 1725–1746, https://doi.org/10.1137/S0097539795289859, http://dx.doi.org/10.1137/S0097539795289859.

[12] H. L. Bodlaender and A. M. C. A. Koster, *Combinatorial optimization on graphs of bounded treewidth*, Comput. J., 51 (2008), pp. 255–269, https://doi.org/10.1093/comjnl/bxm037, http://dx.doi.org/10.1093/comjnl/bxm037.

[13] O. V. Borodin, A. V. Kostochka, and M. Yancey, *On 1-improper 2-coloring of sparse graphs*, Discrete Mathematics, 313 (2013), pp. 2638–2649, https://doi.org/10.1016/j.disc.2013.07.014, http://dx.doi.org/10.1016/j.disc.2013.07.014.

[14] I. Choi and L. Esperet, *Improper coloring of graphs on surfaces*, ArXiv e-prints, (2016), https://arxiv.org/abs/1603.02841.

[15] B. Courcelle, J. A. Makowsky, and U. Rotics, *Linear time solvable optimization problems on graphs of bounded clique-width*, Theory Comput. Syst., 33 (2000), pp. 125–150, https://doi.org/10.1007/s002249910009, http://dx.doi.org/10.1007/s002249910009.

[16] L. Cowen, W. Goddard, and C. E. Jesurum, *Defective coloring revisited*, Journal of Graph Theory, 24 (1997), pp. 205–219, https://doi.org/10.1002/(SICI)1097-0118(199703)24:3⟨205::AID-JGT2⟩3.0.CO;2-T, http://dx.doi.org/10.1002/(SICI)1097-0118(199703)24:3⟨205::AID-JGT2⟩3.0.CO;2-T.

[17] 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 (1986), pp. 187–195, https://doi.org/10.1002/jgt.3190100207, http://dx.doi.org/10.1002/jgt.3190100207.

[18] M. Cygan, F. V. Fomin, L. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, and S. Saurabh, *Parameterized Algorithms*, Springer, 2015, https://doi.org/10.1007/978-3-319-21275-3, http://dx.doi.org/10.1007/978-3-319-21275-3.

[19] R. G. Downey and M. R. Fellows, *Fundamentals of Parameterized Complexity*, Texts in Computer Science, Springer, 2013, https://doi.org/10.1007/978-1-4471-5559-1, http://dx.doi.org/10.1007/978-1-4471-5559-1.

[20] U. Feige and J. Kilian, *Zero knowledge and the chromatic number*, J. Comput. Syst. Sci., 57 (1998), pp. 187–199, https://doi.org/10.1006/jcss.1998.1587, https://doi.org/10.1006/jcss.1998.1587.

[21] M. R. Fellows, F. V. Fomin, D. Lokshtanov, F. A. Rosamond, S. Saurabh, S. Szeider, and C. Thomassen, *On the complexity of some colorful problems parameterized by treewidth*, Inf. Comput., 209 (2011), pp. 143–153, https://doi.org/10.1016/j.ic.2010.11.026, http://dx.doi.org/10.1016/j.ic.2010.11.026.

[22] J. Flum and M. Grohe, *Parameterized Complexity Theory*, Texts in Theoretical Computer Science. An EATCS Series, Springer, 2006, https://doi.org/10.1007/3-540-29953-X, http://dx.doi.org/10.1007/3-540-29953-X.

[23] M. Frick, *A survey of (m, k)-colorings*, Annals of Discrete Mathematics, 55 (1993), pp. 45–57.

[24] M. Frick and M. A. Henning, *Extremal results on defective colorings of graphs*, Discrete Mathematics, 126 (1994), pp. 151–158, https://doi.org/10.1016/0012-365X(94)90260-7, http://dx.doi.org/10.1016/0012-365X(94)90260-7.

[25] J. Gajarský, M. Lampis, and S. Ordyniak, *Parameterized algorithms for modular-width*, in Parameterized and Exact Computation - 8th International Symposium, IPEC 2013, Sophia Antipolis, France, September 4-6, 2013, Revised Selected Papers, G. Gutin and S. Szeider, eds., vol. 8246 of Lecture Notes in Computer Science, Springer, 2013, pp. 163–176, https://doi.org/10.1007/978-3-319-03898-8_15, http://dx.doi.org/10.1007/978-3-319-03898-8_15.

[26] W. Goddard and H. Xu, *Fractional, circular, and defective coloring of series-parallel graphs*, Journal of Graph Theory, 81 (2016), pp. 146–153, https://doi.org/10.1002/jgt.21868, http://dx.doi.org/10.1002/jgt.21868.

[27] B. A. Gudmundsson, T. K. Magnússon, and B. O. Sæmundsson, *Bounds and fixed-parameter algorithms for weighted improper coloring*, Electr. Notes Theor. Comput. Sci., 322 (2016), pp. 181–195, https://doi.org/10.1016/j.entcs.2016.03.013, http://dx.doi.org/10.1016/j.entcs.2016.03.013.

[28] F. Havet, R. J. Kang, and J. Sereni, *Improper coloring of unit disk graphs*, Networks, 54 (2009), pp. 150–164, https://doi.org/10.1002/net.20318, http://dx.doi.org/10.1002/net.20318.

[29] F. Havet and J. Sereni, *Improper choosability of graphs and maximum average degree*, Journal of Graph Theory, 52 (2006), pp. 181–199, https://doi.org/10.1002/jgt.20155, http://dx.doi.org/10.1002/jgt.20155.

[30] R. Impagliazzo, R. Paturi, and F. Zane, *Which problems have strongly exponential complexity?*, J. Comput. Syst. Sci., 63 (2001), pp. 512–530, https://doi.org/10.1006/jcss.2001.1774, http://dx.doi.org/10.1006/jcss.2001.1774.

[31] L. Jaffke and B. M. P. Jansen, *Fine-grained parameterized complexity analysis of graph coloring problems*, CoRR, abs/1701.06985 (2017), http://arxiv.org/abs/1701.06985.

[32] R. J. Kang, *Improper colourings of graphs*, PhD thesis, University of Oxford, UK, 2008, http://ora.ox.ac.uk/objects/uuid:a93d8303-0eeb-4d01-9b77-364113b81a63.

[33] R. J. Kang and C. McDiarmid, *The* t*-improper chromatic number of random graphs*, Combinatorics, Probability & Computing, 19 (2010), pp. 87–98, https://doi.org/10.1017/S0963548309990216, http://dx.doi.org/10.1017/S0963548309990216.

[34] I. Katsikarelis, M. Lampis, and V. T. Paschos, *Structurally parameterized d-scattered set*, in Graph-Theoretic Concepts in Computer Science - 44th International Workshop, WG 2018, Cottbus, Germany, June 27-29, 2018, Proceedings, A. Brandstädt, E. Köhler, and K. Meer, eds., vol. 11159 of Lecture Notes in Computer Science, Springer, 2018, pp. 292–305, https://doi.org/10.1007/978-3-030-00256-5_24, https://doi.org/10.1007/978-3-030-00256-5_24.

[35] I. Katsikarelis, M. Lampis, and V. T. Paschos, *Structural parameters, tight bounds, and approximation for (k, r)-center*, Discrete Applied Mathematics, 264 (2019), pp. 90–117, https://doi.org/10.1016/j.dam.2018.11.002, https://doi.org/10.1016/j.dam.2018.11.002.

[36] J. Kim, A. V. Kostochka, and X. Zhu, *Improper coloring of sparse graphs with a given girth, I: (0, 1)-colorings of triangle-free graphs*, Eur. J. Comb., 42 (2014), pp. 26–48, https://doi.org/10.1016/j.ejc.2014.05.003, http://dx.doi.org/10.1016/j.ejc.2014.05.003.

[37] J. Kim, A. V. Kostochka, and X. Zhu, *Improper coloring of sparse graphs with a given girth, II: constructions*, Journal of Graph Theory, 81 (2016), pp. 403–413, https://doi.org/10.1002/jgt.21886, http://dx.doi.org/10.1002/jgt.21886.

[38] M. Lampis, *Algorithmic meta-theorems for restrictions of treewidth*, Algorithmica, 64 (2012), pp. 19–37, https://doi.org/10.1007/s00453-011-9554-x, http://dx.doi.org/10.1007/s00453-011-9554-x.

[39] M. Lampis, *Parameterized approximation schemes using graph widths*, in Automata, Languages,

and Programming - 41st International Colloquium, ICALP 2014, Copenhagen, Denmark, July 8-11, 2014, Proceedings, Part I, J. Esparza, P. Fraigniaud, T. Husfeldt, and E. Koutsoupias, eds., vol. 8572 of Lecture Notes in Computer Science, Springer, 2014, pp. 775–786, https://doi.org/10.1007/978-3-662-43948-7_64, http://dx.doi.org/10.1007/978-3-662-43948-7_64.

[40] M. Lampis, *Finer tight bounds for coloring on clique-width*, in 45th International Colloquium on Automata, Languages, and Programming, ICALP 2018, July 9-13, 2018, Prague, Czech Republic, I. Chatzigiannakis, C. Kaklamanis, D. Marx, and D. Sannella, eds., vol. 107 of LIPIcs, Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2018, pp. 86:1–86:14, https://doi.org/10.4230/LIPIcs.ICALP.2018.86, https://doi.org/10.4230/LIPIcs.ICALP.2018.86.

[41] D. Lokshtanov, D. Marx, and S. Saurabh, *Known algorithms on graphs on bounded treewidth are probably optimal*, in Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2011, San Francisco, California, USA, January 23-25, 2011, D. Randall, ed., SIAM, 2011, pp. 777–789, https://doi.org/10.1137/1.9781611973082.61, http://dx.doi.org/10.1137/1.9781611973082.61.

[42] J. Nesetril and P. O. de Mendez, *Tree-depth, subgraph coloring and homomorphism bounds*, Eur. J. Comb., 27 (2006), pp. 1022–1041, https://doi.org/10.1016/j.ejc.2005.01.010, http://dx.doi.org/10.1016/j.ejc.2005.01.010.

[43] R. Niedermeier, *Invitation to Fixed-Parameter Algorithms*, Oxford Lecture Series in Mathematics and Its Applications, OUP Oxford, 2006, https://books.google.fr/books?id=mAA_OkeJxhsC.