

Parameterized Complexity of Equitable Coloring*

Guilherme de C. M. Gomes Carlos V. G. C. Lima Vinícius F. dos Santos[†]

Departamento de Ciência da Computação, Universidade Federal de Minas Gerais, Belo Horizonte, Brazil

received 1st Nov. 2018, revised 15th Apr. 2019, accepted 4th May 2019.

A graph on n vertices is equitably k -colorable if it is k -colorable and every color is used either $\lfloor n/k \rfloor$ or $\lceil n/k \rceil$ times. Such a problem appears to be considerably harder than vertex coloring, being NP-complete even for cographs and interval graphs. In this work, we prove that it is W[1]-hard for block graphs and for disjoint union of split graphs when parameterized by the number of colors; and W[1]-hard for $K_{1,4}$ -free interval graphs when parameterized by treewidth, number of colors and maximum degree, generalizing a result by Fellows et al. (2014) through a much simpler reduction. Using a previous result due to Dominique de Werra (1985), we establish a dichotomy for the complexity of equitable coloring of chordal graphs based on the size of the largest induced star. Finally, we show that EQUITABLE COLORING is FPT when parameterized by the treewidth of the complement graph.

Keywords: Equitable Coloring, Parameterized Complexity, Treewidth, Chordal Graphs

1 Introduction

EQUITABLE COLORING is a variant of the classical VERTEX COLORING problem, where we not only want to partition an n vertex graph into k independent sets, but also that each of these sets has either $\lfloor n/k \rfloor$ or $\lceil n/k \rceil$ vertices. The smallest integer k for which G admits an equitable k -coloring is called the *equitable chromatic number* of G .

An extensive survey was conducted by Lih (2013), where many of the results on EQUITABLE COLORING of the last 50 years were assembled. Most of them, however, are upper bounds on the equitable chromatic number. Such bounds are known for: bipartite graphs, trees, split graphs, planar graphs, outerplanar graphs, low degeneracy graphs, Kneser graphs, interval graphs, random graphs and some forms of graph products.

Almost all complexity results for EQUITABLE COLORING arise from a related problem, known as BOUNDED COLORING, an observation given by Bodlaender and Fomin (2005). On BOUNDED COLORING, we ask that the size of the independent sets be bounded by an integer ℓ , which is not necessarily a

*Work supported by Brazilian projects CNPq 311013/2015-5, CNPq Universal 421660/2016-3, FAPEMIG, and Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

[†]Email addresses: gcm.gomes@dcc.ufmg.br (Gomes), carloslima@dcc.ufmg.br (Lima), viniussantos@dcc.ufmg.br (Santos).

function on k or n . Among the known results for BOUNDED COLORING, we have that the problem is solvable in polynomial time for: split graphs (Chen et al., 1996), complements of interval graphs (Bodlaender and Jansen, 1995), complements of bipartite graphs (Bodlaender and Jansen, 1995), forests (Baker and Coffman, 1996), and trees (Baker and Coffman, 1996; Jarvis and Zhou, 2001). While the algorithm for BOUNDED COLORING on trees was first presented by Baker and Coffman (1996), Jarvis and Zhou (2001) compute the minimum number of colors required to color a tree using at most ℓ colors using a novel characterization. For cographs, there is a polynomial-time algorithm when the number of colors k is fixed, otherwise the problem is NP-complete (Bodlaender and Jansen, 1995); the same is also valid for bipartite graphs and interval graphs (Bodlaender and Jansen, 1995). A consequence of the difficulty of BOUNDED COLORING for cographs is the difficulty of the problem for graphs of bounded cliquewidth. In complements of comparability graphs, even if we fix $\ell \geq 3$, BOUNDED COLORING remains NP-hard (Lonc, 1992), implying that the problem is paraNP-hard for the parameter ℓ . Fellows et al. (2011) show that EQUITABLE COLORING parameterized by treewidth and number of colors is W[1]-hard and, in (Bodlaender and Fomin, 2005), an XP algorithm parameterized by treewidth is given for both EQUITABLE COLORING and BOUNDED COLORING.

In this work, we perform a series of reductions proving that EQUITABLE COLORING is W[1]-hard for different subclasses of chordal graphs. In particular, we show that the problem parameterized by the number of colors is W[1]-hard for block graphs and for the disjoint union of split graphs. Moreover, the problem remains W[1]-hard for $K_{1,4}$ -free interval graphs even if we parameterize it by treewidth, number of colors and maximum degree. These results generalize the proof given by Fellows et al. (2011) that EQUITABLE COLORING is W[1]-hard when parameterized by treewidth and number of colors. A result given by de Werra (1985) guarantees that every $K_{1,3}$ -free graph can be equitable k -colored if k is at least the chromatic number. Since VERTEX COLORING can be solved in polynomial time on chordal graphs, we trivially have a polynomial-time algorithm for EQUITABLE COLORING of $K_{1,3}$ -free chordal graphs. This allows us to establish a dichotomy for the computational complexity of EQUITABLE COLORING of chordal graphs based on the size of the largest induced star.

2 Preliminaries

We used standard graph theory notation. Define $[k] = \{1, \dots, k\}$ and 2^S the powerset of S . A k -coloring φ of a graph G is a function $\varphi : V(G) \mapsto [k]$. Alternatively, a k -coloring is a k -partition $V(G) \sim \{\varphi_1, \dots, \varphi_k\}$ such that $\varphi_i = \{u \in V(G) \mid \varphi(u) = i\}$. A set $X \subseteq V(G)$ is *monochromatic* if $|\bigcup_{u \in X} \{\varphi(u)\}| = 1$. A k -coloring is said to be *equitable* if, for every $i \in [k]$, $\lfloor n/k \rfloor \leq |\varphi_i| \leq \lceil n/k \rceil$. A k -coloring of G is *proper* if no edge of G is monochromatic, that is, if φ_i is an independent set for every $i \in [k]$. Unless stated, all colorings are proper.

The *disjoint union*, or simply *union*, of two graphs $G \cup H$ is a graph such that $V(G \cup H) = V(G) \cup V(H)$ and $E(G \cup H) = E(G) \cup E(H)$. The *join* of two graphs $G \oplus H$ is the graph given by $V(G \oplus H) = V(G) \cup V(H)$ and $E(G \oplus H) = E(G) \cup E(H) \cup \{uv \mid u \in V(G), v \in V(H)\}$. A k -*connected component* is a connected component such that k vertices must be removed for it to become disconnected. A component is *biconnected* if $k = 2$. A graph is a *block graph* if and only if every biconnected component is a clique; it is a *split graph* if and only if $V(G)$ can be partitioned in a clique and an independent set. The *length* of a path P_n on n vertices is the number of edges it contains, that is, $n - 1$. The *diameter* of a graph is the length of the longest minimum path between any two vertices of the graph.

A *tree decomposition* of a graph G is defined as $\mathbb{T} = (T, \mathcal{B} = \{B_j \mid j \in V(T)\})$, where T is a tree and

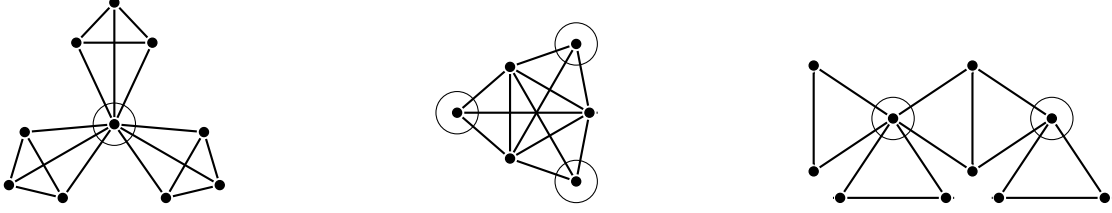


Fig. 1: A (2, 4)-flower, a (2, 4)-antiflower, and a (2, 2)-trem.

$\mathcal{B} \subseteq 2^{V(G)}$ is a family where: $\bigcup_{B_j \in \mathcal{B}} B_j = V(G)$; for every edge $uv \in E(G)$ there is some B_j such that $\{u, v\} \subseteq B_j$; for every $i, j, q \in V(T)$, if q is in the path between i and j in T , then $B_i \cap B_j \subseteq B_q$. Each $B_j \in \mathcal{B}$ is called a *bag* of the tree decomposition. The *width* of a tree decomposition is defined as the size of a largest bag minus one. The *treewidth* $\text{tw}(G)$ of a graph G is the smallest width among all valid tree decompositions of G (Downey and Fellows, 2013). If \mathbb{T} is a rooted tree, by G_x we will denote the subgraph of G induced by the vertices contained in any bag that belongs to the subtree of \mathbb{T} rooted at bag x . An algorithmically useful property of tree decompositions is the existence of a so called *nice tree decompositions* of width $\text{tw}(G)$.

Definition 1. *Nice tree decomposition* A tree decomposition \mathbb{T} of G is said to be nice if it is a tree rooted at, say, the empty bag $r(T)$ and each of its bags is from one of the following four types:

1. Leaf node: a leaf x of \mathbb{T} with $B_x = \emptyset$.
2. Introduce node: an inner bag x of \mathbb{T} with one child y such that $B_x \setminus B_y = \{u\}$.
3. Forget node: an inner bag x of \mathbb{T} with one child y such that $B_y \setminus B_x = \{u\}$.
4. Join node: an inner bag x of \mathbb{T} with two children y, z such that $B_x = B_y = B_z$.

3 Subclasses of Chordal Graphs

All of our reductions involve the BIN-PACKING problem, which is NP-hard in the strong sense (Garey and Johnson, 1979) and $W[1]$ -hard when parameterized by the number of bins (Jansen et al., 2013). In the general case, the problem is defined as: given a set of positive integers $A = \{a_1, \dots, a_n\}$, called *items*, and two integers k and B , can we partition A into k bins such that the sum of the elements of each bin is at most B ? We shall use a version of BIN-PACKING where each bin sums *exactly* to B . This second version is equivalent to the first, even from the parameterized point of view; it suffices to add $kB - \sum_{j \in [n]} a_j$ unitary items to A . For simplicity, by BIN-PACKING we shall refer to the second version, which we formalize as follows.

BIN-PACKING

Instance: A set of n items A and a bin capacity B .

Parameter: The number of bins k .

Question: Is there a k -partition φ of A such that, $\forall i \in [k], \sum_{a_j \in \varphi_i} a_j = B$?

The idea for the following reductions is to build one gadget for each item a_j of the given BIN-PACKING instance, perform their disjoint union, and equitably k -color the resulting graph. The color given to the

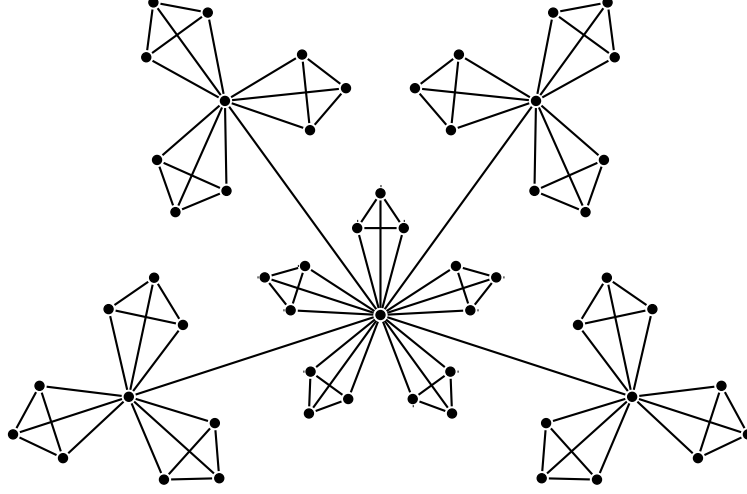


Fig. 2: EQUITABLE COLORING instance built on Theorem 2 corresponding to the BIN-PACKING instance $A = \{2, 2, 2, 2\}$, $k = 3$ and $B = 4$.

circled vertices in Figure 1 control the bin to which the corresponding item belongs to. Each reduction uses only one of the three gadget types. Since every gadget is a chordal graph, their treewidth is precisely the size of the largest clique minus one, that is, k , which is also the number of desired colors for the built instance of EQUITABLE COLORING.

3.1 Disjoint union of Split Graphs

Definition 2. An (a, k) -antiflower is the graph $F_-(a, k) = K_{k-1} \oplus \left(\bigcup_{i \in [a+1]} K_1 \right)$, that is, it is the graph obtained after performing the disjoint union of $a + 1$ K_1 's followed by the join with K_{k-1} .

Theorem 1. EQUITABLE COLORING of the disjoint union of split graphs parameterized by the number of colors is $W[1]$ -hard.

Proof: Let $\langle A, k, B \rangle$ be an instance of BIN-PACKING and G a graph such that $G = \bigcup_{j \in [n]} F_-(a_j, k)$. Note that $|V(G)| = \sum_{j \in [n]} |F_-(a_j, k)| = \sum_{j \in [n]} k + a_j = nk + kB$. Therefore, in any equitable k -coloring of G , each color class has $n + B$ vertices. Define $F_j = F_-(a_j, k)$ and let C_j be the corresponding K_{k-1} . We show that there is an equitable k -coloring ψ of G if and only if $\varphi = \langle A, k, B \rangle$ is a YES instance of BIN-PACKING.

Let φ be a solution to BIN-PACKING. For each $a_j \in A$, we do $\psi(C_j) = [k] \setminus \{i\}$ if $a_j \in \varphi_i$. We color each vertex of the independent set of F_j with i and note that all remaining possible proper colorings of the gadget use each color the same number of times. Thus, $|\psi_i| = \sum_{j|a_j \in \varphi_i} (a_j + 1) + \sum_{j|a_j \notin \varphi_i} 1 = \sum_{j|a_j \in \varphi_i} (a_j + 1) + \sum_{j \in [n]} 1 - \sum_{j|a_j \in \varphi_i} 1 = n + B$.

Now, let ψ be an equitable k -coloring of G . Note that $|\psi_i| = n + B$ and that the independent set of an antiflower is monochromatic. For each $j \in [n]$, $a_j \in \varphi_i$ if $i \notin \psi(C_j)$. That is, $n + B = |\psi_i| = \sum_{j|i \notin C_j} (a_j + 1) + \sum_{j|i \in C_j} 1 = \sum_{j|i \notin C_j} (a_j + 1) + \sum_{j \in [n]} 1 - \sum_{j|i \notin C_j} 1 = \sum_{j|i \notin C_j} a_j + n$, from which we conclude that $\sum_{j|i \notin C_j} a_j = B$. \square

3.2 Block Graphs

We now proceed to the parameterized complexity of block graphs. Conceptually, the proof follows a similar argumentation as the one developed in Theorem 1; in fact, we are able to show that even restricting the problem to graphs of diameter at least four is not enough to develop an FPT algorithm, unless $\text{FPT} = \text{W}[1]$.

Definition 3. An (a, k) -flower is the graph $F(a, k) = K_1 \oplus \left(\bigcup_{i \in [a+1]} K_{k-1} \right)$, that is, it is obtained from the union of $a + 1$ cliques of size $k - 1$ followed by a join with K_1 .

Theorem 2. EQUITABLE COLORING of block graphs of diameter at least four parameterized by the number of colors and treewidth is $\text{W}[1]$ -hard.

Proof: Let $\langle A, k, B \rangle$ be an instance of BIN-PACKING, $\forall k \in [n]$, $F_j = F(a_j, k + 1)$, $F_0 = F(B, k + 1)$ and, for $j \in \{0\} \cup [n]$, let y_j be the universal vertex of F_j . Define a graph G such that $V(G) = V \left(\bigcup_{j \in \{0\} \cup [n]} V(F_j) \right)$ and $E(G) = \{y_0 y_j \mid j \in [n]\} \cup E \left(\bigcup_{j \in \{0\} \cup [n]} E(F_j) \right)$. Looking at Figure 2, it is easy to see that any minimum path between a non-universal vertex of F_a and a non-universal vertex of F_b , $a \neq b \neq 0$ has length four. We show that $\langle A, k, B \rangle$ is an YES instance if and only if G is equitably $(k + 1)$ -colorable.

$$\begin{aligned} |V(G)| &= |V(F_0)| + \sum_{j \in [n]} |V(F_j)| &&= k(B + 1) + 1 + \sum_{j \in [n]} (1 + k(a_j + 1)) \\ &= kB + k + n + k^2 B + kn + 1 &&= (k + 1)(kB + n + 1) \end{aligned}$$

Given a k -partition φ of A that solves our instance of BIN-PACKING, we construct a coloring ψ of G such that $\psi(y_j) = i$ if $a_j \in \varphi_i$ and $\psi(y_0) = k + 1$. Using a similar argument to the previous theorem, after coloring each y_j , the remaining vertices of G are automatically colored. For ψ_{k+1} , note that $|\psi_{k+1}| = 1 + \sum_{j \in [n]} (a_j + 1) = kB + n + 1 = \frac{|V(G)|}{k+1}$. It remains to prove that every other color class ψ_i also has $\frac{|V(G)|}{k+1}$ vertices.

$$\begin{aligned} |\psi_i| &= B + 1 + \sum_{j \mid y_j \notin \psi_i} (a_j + 1) + \sum_{j \mid y_j \in \psi_i} 1 &&= B + 1 + \sum_{j \in [n]} (a_j + 1) - \sum_{j \mid y_j \in \psi_i} a_j \\ &= B + 1 + kB + n - B &&= kB + n + 1 \end{aligned}$$

For the converse we take an equitable $(k + 1)$ -coloring of G and suppose, without loss of generality, that $\psi(y_0) = k + 1$ and, consequently, for every other y_i , $\psi(y_i) \neq k + 1$. To build our k -partition φ of A , we say that $a_j \in \varphi_i$ if $\psi(y_j) = i$. The following equalities show that $\sum_{a_j \in \varphi_i} a_j = B$ for every i , completing the proof.

$$\begin{aligned}
|\psi_i| &= B + 1 + \sum_{j|y_j \in \psi_i} 1 + \sum_{j|y_j \notin \psi_i} (a_j + 1) &= B + 1 + \sum_{j \in [n]} (a_j + 1) - \sum_{j|y_j \in \psi_i} a_j \\
kB + n + 1 &= B + 1 + kB + n - \sum_{j|y_j \in \psi_i} a_j &\Rightarrow B = \sum_{j|y_j \in \psi_i} a_j
\end{aligned}$$

□

3.3 Interval Graphs without some induced stars

Before proceeding to our last reduction, we present a polynomial time algorithm to equitably k -color a claw-free chordal graph G . To do this, given a partial k -coloring φ of G , denote by $G[\varphi]$ the subgraph of G induced by the vertices colored with φ , define φ_- as the set of colors used $\lfloor |V(G[\varphi])|/k \rfloor$ times in φ and φ_+ the remaining colors. If k divides $|V(G[\varphi])|$, we say that $\varphi_+ = \emptyset$. Our goal is to color G one maximal clique (say Q) at a time and keep the invariant that, the new vertices introduced by Q can be colored a subset of the elements of L_- . To do so, we rely on the fact that, for claw-free graphs, the maximal connected components of the subgraph induced by any two colors form either cycles, which cannot happen since G is chordal, or paths. By carefully choosing which colors to look at, we find odd length paths that can be greedily recolored to restore our invariant.

Lemma 1. *There is an $\mathcal{O}(n^2)$ -time algorithm to equitably k -color a claw-free chordal graph or determine that no such coloring exists.*

Proof: We proceed by induction on the number n of vertices of G , and show that G is equitably k -colorable if and only if its maximum clique has size at most k . The case $n = 1$ is trivial. For general n , take one of the leaves of the clique tree of G , say Q , a simplicial vertex $v \in Q$ and define $G' = G \setminus \{v\}$. By the inductive hypothesis, there is an equitable k -coloring of G' if and only if $k \geq \omega(G')$. If $k < \omega(G')$ or $k < |Q|$, G can't be properly colored.

Now, since $k \geq \omega(G) \geq |Q|$, take an equitable k -coloring φ' of G' and define $Q' = Q \setminus \{v\}$. If $|\varphi'_- \setminus \varphi'(Q')| \geq 1$, we can extend φ' to φ using one of the colors of $\varphi'_- \setminus \varphi'(Q')$ to greedily color v . Otherwise, note that $\varphi'_+ \setminus \varphi'(Q') \neq \emptyset$ because $k \geq \omega(G')$. Now, take some color $c \in \varphi'_- \cap \varphi'(Q')$, $d \in \varphi'_+ \setminus \varphi'(Q')$; by our previous observation, we know that $G'[\varphi_c \cup \varphi_d]$ has $C = \{C_1, \dots, C_l\}$ connected components, which in turn are paths. Now, take $C_i \in C$ such that C_i has odd length and both endvertices are colored with d ; said component must exist since $d \in \varphi'_+$ and $c \in \varphi'_-$. Moreover, $C_i \cap Q' = \emptyset$, we can swap the colors of each vertex of C_i and then color v with d ; neither operation makes an edge monochromatic.

As to the complexity of the algorithm, at each step we may need to select c and d – which takes $\mathcal{O}(k)$ time – construct C , find C_i and perform its color swap, all of which take $\mathcal{O}(n)$ time. Since we need to color n vertices and $k \leq n$, our total complexity is $\mathcal{O}(n^2)$. □

The above algorithm was not the first to solve EQUITABLE COLORING for claw-free graphs; this was accomplished by [de Werra \(1985\)](#) which implies that, for any claw-free graph G , $\chi_=(G) = \chi_*(G) = \chi(G)$.

Theorem 3 ([de Werra \(1985\)](#)). *If G is claw-free and k -colorable, then G is equitably k -colorable.*

However, (de Werra, 1985) is not easily accessible, as it is not in any online repository. Moreover, the given algorithm has no clear time complexity and, as far as we were able to understand the proof, its running time would be $\mathcal{O}(k^2n)$, which, for $k = f(n)$, is worse than the algorithm we present in Lemma 1.

Definition 4. Let $\mathcal{Q} = \{Q_1, Q'_1, \dots, Q_a, Q'_a\}$ be a family of cliques such that $Q_i \simeq Q'_i \simeq K_{k-1}$ and $Y = \{y_1, \dots, y_a\}$ be a set of vertices. An (a, k) -trem is the graph $H(a, k)$ where $V(H(a, k)) = \mathcal{Q} \cup Y$ and $E(H(a, k)) = E\left(\bigcup_{i \in [a]} (Q_i \cup Q'_i) \oplus y_i\right) \cup E\left(\bigcup_{i \in [a-1]} y_i \oplus Q_{i+1}\right)$.

Theorem 4. Let G be a $K_{1,r}$ -free interval graph. If $r \geq 4$, EQUITABLE COLORING of G parameterized by treewidth, number of colors and maximum degree is $W[1]$ -hard. Otherwise, the problem is solvable in polynomial time.

Proof: Once again, let $\langle A, k, B \rangle$ be an instance of BIN-PACKING, define $\forall j \in [n]$, $H_j = H(a_j, k)$ and let Y_j be the set of cut-vertices of H_j . The graph G is defined as $G = \bigcup_{j \in [n]} V(H_j)$. By the definition of an (a, k) -trem, we note that the vertices with largest degree are the ones contained in $Y_j \setminus \{y_a\}$, which have degree equal to $3(k-1)$. We show that $\langle A, k, B \rangle$ is a YES instance if and only if G is equitably k -colorable, but first note that $|V(G)| = \sum_{j \in [n]} |V(H_j)| = \sum_{j \in [n]} a_j + 2a_j(k-1) = kB + 2(k-1)kB = k(2kB - B)$.

Given a k -partition φ of A that solves our instance of BIN-PACKING, we construct a coloring ψ of G such that, for each $y \in Y_j$, $\psi(y) = i$ if and only if $a_j \in \varphi_i$. Using a similar argument to the other theorems, after coloring each Y_j , the remaining vertices of G are automatically colored, and we have $|\psi_i| = \sum_{j \in \varphi_i} a_j + \sum_{j \notin \varphi_i} 2a_j = B + 2(k-1)B = 2kB - B$.

For the converse we take an equitable k -coloring of G and observe that, for every $j \in [n]$, $|\psi(Y_j)| = 1$. As such, to build our k -partition φ of A , we say that $a_j \in \varphi_i$ if and only if $\psi(Y_j) = \{i\}$. Thus, since $|\psi| = 2kB - B$, we have that $2kB - B = \sum_{j \in \varphi_i} a_j + \sum_{j \notin \varphi_i} 2a_j = \sum_{j \in [n]} a_j - \sum_{j \in \varphi_i} a_j = 2kB - \sum_{j \in \varphi_i} a_j$, from which we conclude that $B = \sum_{j \in \varphi_i} a_j$. \square

4 Clique Partitioning

Since EQUITABLE COLORING is $W[1]$ -hard when simultaneously parameterized by many parameters, we are led to investigate a related problem. Much like EQUITABLE COLORING is the problem of partitioning G in k' independent sets of size $\lceil n/k' \rceil$ and $k - k'$ independent sets of size $\lfloor n/k' \rfloor$, one can also attempt to partition \overline{G} in cliques of size $\lceil n/k' \rceil$ or $\lfloor n/k' \rfloor$. A more general version of this problem is formalized as follows:

CLIQUE PARTITIONING

Instance: A graph G and two positive integers k and r .

Question: Can G be partitioned in k cliques of size r and $\frac{n-rk}{r-1}$ cliques of size $r-1$?

We note that both MAXIMUM MATCHING (when $k \geq n/2$) and TRIANGLE PACKING (when $k < n/2$) are particular instances of CLIQUE PARTITIONING, the latter being FPT when parameterized by k (Fellows et al., 2005). As such, we will only be concerned when $r \geq 3$. To the best of our efforts, we were unable to provide an FPT algorithm for CLIQUE PARTITIONING when parameterized by k and r , even if we fix $r = 3$. However, the situation is different when parameterized by the treewidth of G , and we obtain an algorithm running in $2^{\text{tw}(G)} n^{\mathcal{O}(1)}$ time for the corresponding counting problem, #CLIQUE PARTITIONING.

The key ideas for our bottom-up dynamic programming algorithm are quite straightforward. First, cliques are formed only when building the tables for forget nodes. Second, for join nodes, we can safely consider only the combination of two partial solutions that have empty intersection on the covered vertices (i.e. that have already been assigned to some clique). Finally, both join and forget nodes can be computed using fast subset convolution (Björklund et al., 2007). For each node $x \in \mathbb{T}$, our algorithm builds the table $f_x(S, k')$, where each entry is indexed by a subset $S \subseteq B_x$ that indicates which vertices of B_x have already been covered, an integer k' recording how many cliques of size r have been used, and stores how many partitions exist in G_x such that only $B_x \setminus S$ is yet uncovered. If an entry is inconsistent (e.g. $S \not\subseteq B_x$), we say that $f(S, K') = 0$.

Theorem 5. *There is an algorithm that, given a nice tree decomposition of an n -vertex graph G of width tw , computes the number of partitions of G in k cliques of size r and $\frac{n-rk}{r-1}$ cliques of size $r-1$ in time $\mathcal{O}^*(2^{\text{tw}})$ time.*

Proof: *Leaf node:* Take a leaf node $x \in \mathbb{T}$ with $B_x = \emptyset$. Since the only one way of covering an empty graph is with zero cliques, we compute f_x with:

$$f_x(S, k') = \begin{cases} 1, & \text{if } k' = 0 \text{ and } S = \emptyset; \\ 0, & \text{otherwise.} \end{cases}$$

Introduce node: Let x be an introduce node, y its child and $v \in B_x \setminus B_y$. Due to our strategy, introduce nodes are trivial to solve; it suffices to define $f_x(S, k') = f_y(S, k')$. If $v \in S$, we simply define $f_x(S, k') = 0$.

Forget node: For a forget node x with child y and forgotten vertex v , we formulate the computation of $f_x(S, k')$ as the subset convolution of two functions as follows:

$$f_x(S, k') = f_y(S \cup \{v\}, k') + \sum_{A \subseteq S} f_y(S \setminus A, k' - 1)g_r(A, v) + \sum_{A \subseteq S} f_y(S \setminus A, k')g_{r-1}(A, v)$$

$$g_l(A, v) = \begin{cases} 1, & \text{if } A \text{ is a clique of size } l \text{ contained in } N[v] \text{ and } v \in A; \\ 0, & \text{otherwise.} \end{cases}$$

The above computes, for every $S \subseteq B_x$ and every clique A (that contains v) of size r or $r-1$ contained in $N[v] \cap B_y \cap S$, if $S \setminus A$ and some k'' is a valid entry of f_y , or if v had been previously covered by another clique (first term of the sum). Directly computing the last two terms of the equation, for each pair (S, k') , yields a total running time of the order of $\sum_{|S|=0}^{\text{tw}} \binom{\text{tw}}{|S|} 2^{|S|} = (1+2)^{\text{tw}} = 3^{\text{tw}}$ for each forget node. However, using the fast subset convolution technique described by Björklund et al. (2007), we can compute the above equation in time $\mathcal{O}^*(2^{|B_x|}) = \mathcal{O}^*(2^{\text{tw}})$.

Correctness follows directly from the hypothesis that f_y is correctly computed and that, for every $A \subseteq B_x$, $g_r(A, v)g_{r-1}(A, v) = 0$. For the running time, we can pre-compute both g_r and g_{r-1} in $\mathcal{O}(2^{\text{tw}r^2})$, so their values can be queried in $\mathcal{O}(1)$ time. As such, each forget node takes $\mathcal{O}(2^{\text{tw}r^2}k)$ time, since we can compute the subset convolutions of $f_y * g_r$ and $f_y * g_{r-1}$ in $\mathcal{O}(2^{\text{tw}r^2})$ time each. The additional factor of k comes from the second coordinate of the table index.

Join node: Take a join node x with children y and z . Since we want to partition our vertices, the cliques we use in G_y and G_z must be completely disjoint and, consequently, the vertices of B_x covered in B_y and B_z must also be disjoint. As such, we can compute f_x through the equation:

$$f_x(S, k') = \sum_{k_y + k_z = k'} \sum_{A \subseteq S} f_y(A, k_y) f_z(S \setminus A, k_z)$$

Note that we must sum over the integer solutions of the equation $k_y + k_z = k'$ since we do not know how the cliques of size r are distributed in G_x . To do that, we compute the subset convolution $f_y(\cdot, k_y) * f_z(\cdot, k_z)$. The time complexity of $\mathcal{O}(2^{\text{tw}} \text{tw}^3 k^2)$ follows directly from the complexity of the fast subset convolution algorithm, the range of the outermost sum and the range of the second parameter of the table index.

For the root x , we have $f_x(\emptyset, k) \neq 0$ if and only if $G_x = G$ can be partitioned in k cliques of size r and the remaining vertices in cliques of size $r - 1$. Since our tree decomposition has $\mathcal{O}(n \text{tw})$ nodes, our algorithm runs in time $\mathcal{O}(2^{\text{tw}} \text{tw}^4 k^2 n)$.

To recover a solution given the tables f_x , start at the root node with $S = \emptyset$, $k' = k$ and let $\mathcal{Q} = \emptyset$ be the cliques in the solution. We shall recursively extend \mathcal{Q} in a top-down manner, keeping track of the current node x , the set of vertices S and the number k' of K_r 's used to cover G_x . Our goal is to keep the invariant that $f_x(S, k') \neq 0$.

Introduce node: Due to the hypothesis that $f_x(S, k') \neq 0$ and the way that f_x is computed, it follows that $f_y(S, k') \neq 0$.

Forget node: Since the current entry is non-zero, there must be some $A \subseteq S$ such that exactly one of the products $f_y(S \setminus A, k' - 1) g_r(A, v)$, $f_y(S \setminus A, k') g_{r-1}(A, v)$ is non-zero and, in fact, any such A suffices. To find this subset, we can iterate through 2^S in $\mathcal{O}(2^{\text{tw}})$ time and test both products to see if any of them is non-zero. Note that the chosen $A \cup \{v\}$ will be a clique of size either r or $r - 1$, and thus, we can set $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{A \cup \{v\}\}$.

Join node: The reasoning for join nodes is similar to forget nodes, however, we only need to determine which states to look at in the child nodes. That is, for each integer solution to $k_y + k_z = k'$ and for each $A \subseteq S$, we check if both $f_y(A, k_y) f_z(S \setminus A, k_z)$ is non-zero; in the affirmative, we compute the solution for both children with the respective entries. Any such triple (A, k_y, k_z) that satisfies the condition suffices.

Clearly, retrieving the solution takes $\mathcal{O}(2^{\text{tw}} k)$ time per node, yielding a running time of $\mathcal{O}^*(2^{\text{tw}})$. \square

Corollary 1. *Equitable coloring is FPT when parameterized by the treewidth of the complement graph.*

5 Conclusions

In this work, we investigated the EQUITABLE COLORING problem. We developed novel parameterized reductions from BIN-PACKING, which is $\text{W}[1]$ -hard when parameterized by number of bins. These reductions showed that EQUITABLE COLORING is $\text{W}[1]$ -hard in three more cases: (i) if we restrict the problem to block graphs and parameterize by the number of colors, treewidth and diameter; (ii) on the disjoint union of split graphs, a case where the connected case is polynomial; (iii) EQUITABLE COLORING of $K_{1,r}$ interval graphs, for any $r \geq 4$, remains hard even if we parameterize by the number of colors, treewidth and maximum degree. This, along with a previous result by [de Werra \(1985\)](#), establishes a dichotomy based on the size of the largest induced star: for $K_{1,r}$ -free graphs, the problem is solvable

in polynomial time if $r \leq 2$, otherwise it is $W[1]$ – hard. These results significantly improve the ones by Fellows et al. (2011) through much simpler proofs and in very restricted graph classes.

Since the problem remains hard even for many natural parameterizations, we resorted to a more exotic one – the treewidth of the complement graph. By applying standard dynamic programming techniques on tree decompositions and the fast subset convolution machinery of Björklund et al. (2007), we obtain an FPT algorithm when parameterized by the treewidth of the complement graph.

Natural future research directions include the identification and study of other uncommon parameters that may aid in the design of other FPT algorithms. Revisiting CLIQUE PARTITIONING when parameterized by k and r is also of interest, since its a related problem to EQUITABLE COLORING and the complexity of its natural parameterization is yet unknown.

Acknowledgements

We are grateful to the two anonymous reviewers whose comments/suggestions helped us to improve and clarify the paper.

References

- B. S. Baker and E. G. Coffman. Mutual exclusion scheduling. *Theoretical Computer Science*, 162(2): 225–243, 1996.
- A. Björklund, T. Husfeldt, P. Kaski, and M. Koivisto. Fourier meets möbius: fast subset convolution. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, pages 67–74. ACM, 2007.
- H. L. Bodlaender and F. V. Fomin. *Equitable colorings of bounded treewidth graphs*, volume 349, pages 22 – 30. 2005. doi: <https://doi.org/10.1016/j.tcs.2005.09.027>. Graph Colorings.
- H. L. Bodlaender and K. Jansen. Restrictions of graph partition problems. part I. *Theoretical Computer Science*, 148(1):93 – 109, 1995. ISSN 0304-3975.
- B.-L. Chen, M.-T. Ko, and K.-W. Lih. *Equitable and m -bounded coloring of split graphs*, pages 1–5. Springer Berlin Heidelberg, Berlin, Heidelberg, 1996.
- D. de Werra. Some uses of hypergraphs in timetabling. *Asia-Pacific Journal of Operational Research*, 2 (1):2–12, 1985.
- R. G. Downey and M. R. Fellows. *Fundamentals of parameterized complexity*, volume 4. Springer, 2013.
- M. Fellows, P. Heggernes, F. Rosamond, C. Sloper, and J. A. Telle. Finding k disjoint triangles in an arbitrary graph. In J. Hromkovič, M. Nagl, and B. Westfechtel, editors, *Graph-Theoretic Concepts in Computer Science*, pages 235–244, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.
- M. R. Fellows, F. V. Fomin, D. Lokshtanov, F. Rosamond, S. Saurabh, S. Szeider, and C. Thomassen. On the complexity of some colorful problems parameterized by treewidth. *Information and Computation*, 209(2):143 – 153, 2011.

- M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1979.
- K. Jansen, S. Kratsch, D. Marx, and I. Schlotter. Bin packing with fixed number of bins revisited. *Journal of Computer and System Sciences*, 79(1):39 – 49, 2013.
- M. Jarvis and B. Zhou. Bounded vertex coloring of trees. *Discrete Mathematics*, 232(1-3):145–151, 2001.
- K.-W. Lih. Equitable coloring of graphs. In *Handbook of combinatorial optimization*, pages 1199–1248. Springer, 2013.
- Z. Lonc. *On complexity of some chain and antichain partition problems*, pages 97–104. Springer Berlin Heidelberg, Berlin, Heidelberg, 1992.