Generalized H-coloring and H-covering of Trees

Jiří Fiala[1,⋆], Pinar Heggernes[2], Petter Kristiansen[2], and Jan Arne Telle[2]

[1] DIMATIA and ITI[⋆⋆], Charles University, 118 00 Praha 1, Czech Republic
fiala@kam.mff.cuni.cz
[2] Department of Informatics, University of Bergen, N-5020 Bergen, Norway
{pinar, petterk, telle}@ii.uib.no

Abstract. We study $H(p, q)$-colorings of graphs, for H a fixed simple graph and p, q natural numbers, a generalization of various other vertex partitioning concepts such as H-covering. An H-cover of a graph G is a local isomorphism between G and H, and the complexity of deciding if an input graph G has an H-cover is still open for many graphs H. In this paper we show that the complexity of $H(2p, q)$-COLORING is directly related to these open graph covering problems, and answer some of them by resolving the complexity of $H(p, q)$-COLORING for all acyclic graphs H and all values of p and q.

1 Introduction

Colorings of graphs is a well-studied subject. In some cases the 'colors' to be assigned are the vertices of a fixed graph H, and model a situation where certain pairs of colors are treated specially. For example, in the well-known H-COLORING problem we ask for an assignment of 'colors' to the vertices of an input graph G such that adjacent vertices of G obtain adjacent 'colors', defining a homomorphism between G and H. H-COLORING is known to be solvable in polynomial time for bipartite H, and $\mathcal{NP}$-complete otherwise [9]. In the H-COVER problem a vertex $v \in V(G)$ is assigned a 'color' $u \in V(H)$ of the same degree, in such a way that the set of 'colors' assigned to the neighbors of v is exactly the set of 'colors' adjacent to u, defining a local isomorphism between G and H. Graph coverings come from algebraic graph theory [3], and form a special case of covering spaces from algebraic topology [17], with applications in topological graph theory [8]. The first applications in computer science were to graph recognition by parallel networks of processors [2,6]. The question of the computational complexity of H-COVER was first posed in 1989 [4], a variety of results have been shown, see e.g. [12,13,14,7], but it is still unclear what characterizes the class of simple graphs H that lead to polynomial time H-COVER problems.

⋆ Research supported in part by NATO science fellowship, administered through Norwegian Research Council project no. 143192/410 during his postdoctoral fellowship at the University of Bergen, and by Czech research grant GAČR 201/99/0242.
⋆⋆ Supported by the Ministry of Education of the Czech Republic as project LN00A056.

A wide generalization of both H-coloring and H-covering is given by the so-called $H(\sigma, \rho)$-colorings for subsets of natural numbers σ and ρ [15], and in this paper we focus on the case of $|\sigma| = |\rho| = 1$. For a fixed graph H and natural numbers p and q, the $H(p,q)$-COLORING problem studied in this paper asks if an input graph G has a mapping $f : V(G) \to V(H)$ where the neighbors of any $v \in V(G)$ are mapped to the closed neighborhood of $f(v)$, with exactly p neighbors mapped to $f(v)$, and exactly q neighbors mapped to each neighbor of $f(v)$. From the definition it is clear that $H(0,1)$-coloring is equivalent to H-covering, but it is maybe more surprising that for any simple graph H and any $p \geq 0, q \geq 1$ there exists a multigraph M such that $H(2p,q)$-coloring is equivalent to M-covering. In fact, the first graph covering problem shown to be $\mathcal{NP}$-complete in [1] was equivalent to $P_2(2,1)$-COLORING, P_2 an edge. Moreover, it is known that resolving the complexity of M-COVER for all multigraphs M (where the covering definition is somewhat more complicated) is necessary and sufficient for resolving the complexity of H-COVER for all simple graphs H [12]. Via this link, the results in this paper, on the complexity of $H(p,q)$-COLORING, contribute directly towards the partial solution of the open problem mentioned above.See Fig. 1 for an example of a tree T and values of p and q (for which $T(p,q)$-COLORING is $\mathcal{NP}$-complete), the corresponding simple graph H (for which H-COVER then is $\mathcal{NP}$-complete), and the linking multigraph M.

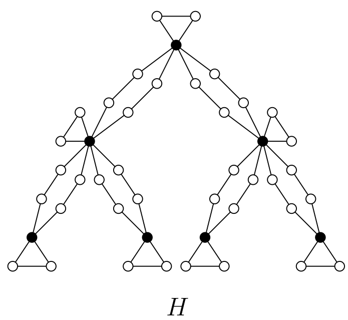

Fig. 1. A tree T (for which $T(2,2)$-COLORING is $\mathcal{NP}$-complete), the corresponding simple graph H (for which H-COVER then is $\mathcal{NP}$-complete), and the linking multigraph M

The degree refinement matrix $\mathbf{M}_G$ of a graph G gives crucial information for these problems, and it is defined as follows: The *degree refinement* of a multigraph G is the partition of its vertices into the minimum number of blocks $\mathcal{B}_G = \{B_1(G), \dots, B_t(G)\}$ for which there are constants m_{ij}, such that for all $1 \leq i,j \leq t$ each vertex in B_i has exactly m_{ij} neighbors in B_j. For a given, canonical, ordering of degree refinement blocks, the $t \times t$ matrix $\mathbf{M}_G$, $\mathbf{M}_G[i,j] = m_{ij}$, is called the *degree refinement matrix*.

We can now state our main theorem.

Theorem 1. *$T(p, q)$-COLORING, for a tree T and natural numbers p and q, is solvable in polynomial time if either:*

- *(trivial cases) all blocks [in the degree refinement of T] have size 1, or $q = 0$, or $p = 0, q = 1$,*
- *$p = 0$, $q = 2$ and either all blocks containing non-leaves have size 1, or they all have size 2,*
- *$p = 0$, $q \geq 3$ and all blocks have size 2,*
- *$p = 1$, $q = 1$ and all entries in [the degree refinement matrix] $\mathbf{M}_T$ are at most 2, or*
- *$p \geq 2$, $q = 1$, all entries in $\mathbf{M}_T$ are at most 2, and no block contains an induced edge,*

and is $\mathcal{NP}$-complete in all other cases.

The next section contains all formal definitions, and several important observations. The remainder of the paper is then devoted to the proof of Theorem 1, which has three different ingredients, split into three sections: Sect. 3 contains polynomial-time algorithms, based on finding factors in regular graphs and on reductions to 2SAT, Sect. 4 $\mathcal{NP}$-completeness reductions, and Sect. 5 various characterizations of trees, to show that all cases have been accounted for.

2 Definitions and Basic Observations

Let H be a fixed simple graph with k vertices $V(H) = \{u_1, u_2, \ldots, u_k\}$, and let p and q be natural numbers. An *$H(p, q)$-coloring* of a simple graph G is a partition $V_1, V_2, \ldots, V_k$ of $V(G)$, such that for all $1 \leq i, j \leq k$

$$\forall v \in V_i : |N_G(v) \cap V_j| = \begin{cases} p & \text{if } i = j \\ q & \text{if } u_i u_j \in E(H) \\ 0 & \text{otherwise} \end{cases} ,$$

where $N_G(v)$ denotes the open neighborhood of v in G. We will also view the partition as a vertex mapping $f : V(G) \to V(H)$, with $f(v) = v_i$ for $v \in V_i$. Note that this is equivalent to the definition given in Sect. 1.

In this paper we investigate the complexity of the following problem.

***H*(*p*, *q*)-COLORING**
INSTANCE: Simple graph G.
QUESTION: Does G have an $H(p, q)$-coloring?

For a simple graph H, $H(0, 1)$-COLORING is precisely the same as the H-COVER problem. Moreover, any $H(2p, q)$-coloring will correspond to an M-covering for some multigraph M, as we now show. The usual definition of topological covering spaces requires that to cover a multigraph M by a graph

G we must specify, in addition to a vertex mapping $f : V(G) \to V(M)$, also an edge mapping $g : E(G) \to E(M)$. The edge map must respect the vertex map, and for every vertex $v \in V(G)$, and every edge e incident to $f(v)$, there must be exactly one edge incident to v that is mapped to e. Moreover, for every self-loop s on $f(v)$ there must be exactly two edges (or one self-loop) incident to v mapped to s.

For a simple graph H and natural numbers p and q, let H_q^p be the multigraph obtained from H by adding p self-loops to each vertex of H, and replacing each edge of H by q multiple edges.

Observation 1 *A simple graph G has an $H(2p, q)$-coloring if and only if it has an H_q^p-cover.*

Proof. The forward direction of the proof follows from [12] which shows that even if multigraph-covering requires an edge map, this edge map always exists if the vertex map f obeys the cardinality constraint that for every vertex $v \in V(G)$ the number of neighbors of v mapping to a vertex $u \in V(H)$ is the same as the number of multiple edges between u and $f(v)$. Likewise, the number of neighbors of v mapping to $f(v)$ should be twice the number of self-loops on $f(v)$. Clearly, an $H(2p, q)$-coloring of G does satisfy these cardinality constraints as imposed by H_q^p. The other direction of the proof is trivial. □

The degree refinement and degree refinement matrix of a graph are computed in polynomial time by stepwise refinement. Start with the vertices partitioned by their degrees and refine the partition as long as two vertices in the same block do not have the same number of neighbors in some other block. Note that the center of a tree, consisting of vertices whose greatest distance from any other vertex is as small as possible, contains either one or two vertices. The following fact is folklore.

Fact 1 *If G has an H-cover f, then $\mathbf{M}_G = \mathbf{M}_H$, and f is a one-to-one mapping of the degree refinement blocks.*

Theorem 2. *If G has an $H(p, q)$-coloring, and H is a simple, connected graph, then*

$$\mathbf{M}_G = q\mathbf{M}_H + p\mathbf{I} \ .$$

Proof. If $p = 2r$ we know from Observation 1 and Fact 1 that $\mathbf{M}_G = \mathbf{M}_{H_q^r}$, and first show that $\mathbf{M}_{H_q^r} = q\mathbf{M}_H + 2r\mathbf{I}$. When computing the degree refinement of H_q^r, self-loops are inconsequential, since all vertices $u \in V(H_q^r)$ have exactly r such loops. Likewise, the edge multiplicity is also inconsequential, since for all distinct, adjacent vertices $u, u' \in V(H_q^r)$, there are exactly q edges between u and u'. This implies that the degree refinements of H and H_q^r have the same block structure. Moreover, since every vertex in $V(H)$ gets r self-loops, and every edge in $E(H)$ is replaced by q multiple edges, we have $\mathbf{M}_{H_q^r} = q\mathbf{M}_H + 2r\mathbf{I}$. The argument for odd p differs only by some technical details. □

The following fact is now evident.

Fact 2 *If G has an $H(p,q)$-coloring f, and H is connected, then $|f^{-1}(u)| = |f^{-1}(u')|$ for all $u, u' \in V(H)$.*

The *simple (p,q)-cover of G* is the graph $(G \bowtie K_q) \,\Box\, K_{p+1}$, where the graph products $\bowtie$ and $\Box$ are defined as follows (in so called Nešetřil convention):

- $V(G \bowtie G') = V(G \,\Box\, G') = V(G) \times V(G')$,
- $(v,v')(w,w') \in E(G \bowtie G')$ if and only if $vw \in E(G)$, and ($v'w' \in E(G')$ or $v' = w'$),
- $(v,v')(w,w') \in E(G \,\Box\, G')$ if and only if ($vw \in E(G)$ and $v' = w'$), or ($v'w' \in E(G')$ and $v = w$).

For an example see the simple $(2,2)$-cover of P_3 depicted in Fig. 2, P_3 the path on three vertices.

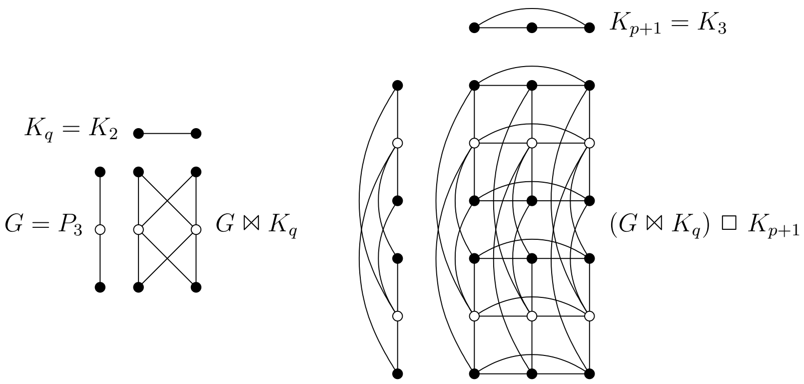

Fig. 2. Simple $(2,2)$-cover of P_3

Observe, that the operation $G \bowtie K_q$ replaces an edge of $vw \in E(G)$ with a complete bipartite graph $K_{q,q}$ on q copies of the vertices v and w. Similarly the operation $G \,\Box\, K_{p+1}$ forms a clique K_{p+1} on $p+1$ copies of every vertex $v \in V(G)$, while the edges inside the $p+1$ copies of G are maintained. Blocks in the degree refinement of the simple (p,q)-cover correspond to the blocks in the degree refinement of the original graph, as indicated in Fig. 2 by the black and white vertex colors.

Lemma 1. *For every graph H and every $p \geq 0$, $q \geq 1$, the simple (p,q)-cover of H has an $H(p,q)$-coloring.*

Proof. The mapping $f : ((u,a),b) \to u$, where $u \in V(H)$, $a \in V(K_q)$, $b \in V(K_{p+1})$ is an $H(p,q)$-coloring. Every vertex has p neighbors mapped to the same target, those that differ only in the b-coordinate; and for every neighbor u' of u, $((u,a),b)$ is adjacent to $((u',a'),b)$, for every $1 \leq a' \leq q$. $\Box$

3 Polynomial Cases

Theorem 3. *If $\mathbf{M}_H = \mathbf{A}_H$, the adjacency matrix of H, i.e. if all blocks in the degree refinement of H are of size 1, then $H(p,q)$-COLORING is solvable in polynomial time.*

Proof. From Theorem 2 we know what the degree refinement matrix must be. Theorem 2 describes the necessary condition, and when $\mathbf{M}_H = \mathbf{A}_H$ this is also the sufficient condition. □

As an aside we mention that for a random graph H of the model $G(n,p)$, it is almost always the case that $\mathbf{M}_H = \mathbf{A}_H$ when $0 < p = p(n) \leq \frac{1}{2}$ is such that $p^5 n/(\log n)^5 \to \infty$ [5, ch. 3].

Lemma 2. *$T(0,q)$-COLORING, $q \geq 2$, is solvable in polynomial time for every tree T whose degree refinement blocks are all of size 2.*

Proof. We reduce to 2SAT. Let G be an instance of $T(0,q)$-COLORING, $q \geq 2$, where T is a tree whose degree refinement blocks are all of size 2. We construct a formula ϕ, such that ϕ has a satisfying truth assignment if and only if G has a $T(0,q)$-coloring. It will be obvious how to transform ϕ into a set V of variables and a collection C of two-literal clauses, to form a 2SAT instance (V,C).

First, compute the degree refinements of G and T. If they do not obey the constraints of Theorem 2, reject the input G. Otherwise, let $\mathcal{B}_G = \{B_1(G), \ldots, B_t(G)\}$ and $\mathcal{B}_T = \{B_1(T), \ldots, B_t(T)\}$ be the degree refinements of G and T, respectively. Let $B_i(T) = \{\text{left}_i, \text{right}_i\}$, and note that for each vertex $v \in B_i(G)$ we must decide whether it should map to left_i (variable v FALSE) or right_i (variable v TRUE). When all blocks in the degree refinement of T are of size 2, the center of T is an edge uu', and both u and u' belong to the same degree refinement block $B_1(T)$. $B_1(T)$ is the only block containing adjacent vertices. For every pair of adjacent vertices $v, w \in B_1(G)$ we insert the subformula $(v \not\Leftrightarrow w)$ into ϕ, and for every vertex $x \in B_i(G)$, with a neighbour $y \in B_j(G)$, $j \neq i$, we insert the the subformula $(x \Leftrightarrow y)$ into ϕ.

Let $f : V(G) \to V(T)$ be a $T(0,q)$-coloring of G, and label the left and right vertex of each block $B_i(T)$ with 0 and 1, respectively. Viewing the label of $f(v)$ as a truth assignment to variable v, we get a satisfying truth assignment τ for ϕ. In the other direction of the proof we first use the degree refinement of G to determine which block $B_i(T)$ in the degree refinement of T a vertex $v \in V(G)$ must map to. We then use a truth assignment τ for ϕ to determine if v should be mapped to the left or right vertex of $B_i(T)$. □

Let S_k denote the graph $K_{1,k}$, the star on $k+1$ vertices.

Lemma 3. *$S_k(0,2)$-COLORING is solvable in polynomial time for every k.*

Proof. Let G be an instance of $S_k(0,2)$-COLORING. By Theorem 2, the vertices of G must either be of degree 2, or $2k$. The vertices of degree $2k$ must map to the central vertex of S_k. The remaining vertices must map to the k leaves, in such way that the neighborhoods of the vertices mapping to the center can be split into k disjoint pairs, where both vertices of a pair map to a unique leaf.

Contracting the vertices of degree 2 in a homeomorphic manner results in a $2k$-regular graph G'. By Petersen's theorem, G' can be split into k disjoint 2-factors. This split can be done in polynomial time, and we get an $S_k(0,2)$-coloring of G by mapping the vertices of degree 2 to the same leaf if and only if the corresponding edges belong to the same 2-factor. $\square$

Note that in the case $k = 2$ the lemma also provides a polynomial time algorithm for $P_3(0,2)$-COLORING.

Lemma 4. *$T(0,2)$-COLORING is solvable in polynomial time for every tree T if either all blocks containing non-leaves have size 1, or they all have size 2.*

Proof. Note that the condition on T implies that we have either: (1) all degree refinement blocks of size 2 or more contain only leaves, or (2) all degree refinement blocks of size not equal to 2 contain only leaves.

Case 1: Let G be an instance of $T(0,2)$-COLORING, where T is a tree such that all blocks in the degree refinement of T of size 2 or more contain only leaves.

First, compute the degree refinements of G and T. If they do not obey the constraints of Theorem 2, reject the input G. Otherwise, let $\mathcal{B}_G = \{B_1(G), \ldots, B_t(G)\}$ and $\mathcal{B}_T = \{B_1(T), \ldots, B_t(T)\}$ be the degree refinements of G and T, respectively. For each block $B_i(G)$ corresponding to a block $B_i(T)$ of size 1, all vertices of $B_i(G)$ must map to the single vertex of $B_i(T)$. For each block $B_j(G)$ corresponding to a block $B_j(T)$ of size 2 or more, the vertices of $B_j(G)$ have neighbors in exactly one other block in the degree refinement of G; this block must correspond to a block of size 1 in the degree refinement of T. Thus, for each such $B_j(G)$, the problem can be solved independently, in polynomial time by Lemma 3, and the solutions combined into an overall $T(0,2)$-coloring of G.

Case 2: Let G be an instance of $T(0,2)$-COLORING, where T is a tree such that all blocks in the degree refinement of T of size not equal to 2 contain only leaves.

First, compute the degree refinements of G and T. If they do not obey the constraints of Theorem 2, reject the input G. Otherwise, let $\mathcal{B}_G = \{B_1(G), \ldots, B_s(G), B_{s+1}(G), \ldots, B_t(G)\}$ and $\mathcal{B}_T = \{B_1(T), \ldots, B_s(T), B_{s+1}(T), \ldots, B_t(T)\}$ be the degree refinements of G and T, respectively, with $B_1(T), \ldots, B_s(T)$ as the blocks of size 2, and $B_{s+1}(T), \ldots B_t(T)$ as the remaining blocks. For the portion of G induced by the blocks $B_1(G), \ldots, B_s(G)$, the vertices can be mapped to the appropriate vertices of the portion of T induced by $B_1(T), \ldots, B_s(T)$ in polynomial time by Lemma 2. The blocks $B_{s+1}(G), \ldots, B_t(G)$ can be handled independently, in polynomial time by Lemma 3, and the solutions combined into an overall $T(0,2)$-coloring of G. $\square$

Lemma 5. *$T(p,q)$-COLORING is solvable in polynomial time for every tree T if either:*

(1) *$p = 1$, $q = 1$ and all entries in $\mathbf{M}_T$ are at most 2, or*
(2) *$p \geq 2$, $q = 1$, all entries in $\mathbf{M}_T$ are at most 2, and no block in the degree refinement of T contains an induced edge.*

Proof. Both problems can be reduced to 2SAT. We only provide proof of case 1, as the proof of case 2 is similar.

Let G be an instance of $T(1,1)$-COLORING, where T is a tree such that all entries in $\mathbf{M}_T$ are at most 2. We construct a set V of variables and a formula ϕ, such that ϕ has a satisfying truth assignment if and only if G has a $T(1,1)$-coloring. It will be obvious how to transform ϕ into a collection C of two-literal clauses, to form a 2SAT instance (V, C).

First, compute the degree refinements of G and T. If they do not obey the constraints of Theorem 2, reject the input G. Otherwise, let $\mathcal{B}_G = \{B_1(G), \ldots, B_t(G)\}$ and $\mathcal{B}_T = \{B_1(T), \ldots, B_t(T)\}$ be the degree refinements of G and T, respectively. Let $B_1(T)$ be the block containing the center of T, and level the blocks of $\mathcal{B}_T$ according to their distance from $B_1(T)$, with $B_1(T)$ as level 1. $\mathcal{B}_G$ is given the same leveling. For each vertex $v \in B_i(G)$ we must decide which vertex of $B_i(T)$ it should map to. When all entries in the degree refinement matrix M_T are at most 2, a vertex $u \in B_i(T)$ can have at most 2 neighbours in $B_j(T)$, and $B_1(T)$ is the only block that can contain adjacent vertices. A block $B_i(T)$ on level l, $l > 1$, can therefore contain at most 2^l vertices. Every vertex v in a block on level l is represented by l variables $v_1, \ldots, v_l$. For every vertex $v \in B_1(G)$ we insert the clause (v_1) into ϕ; for every vertex $w \in B_i(G)$ on level l, with two children $x, x' \in B_j(G)$ on level $l+1$, we insert clauses $(x_1 \Leftrightarrow w_1), \ldots, (x_l \Leftrightarrow w_l), (x'_1 \Leftrightarrow w_1), \ldots, (x'_l \Leftrightarrow w_l), (x_{l+1} \not\Leftrightarrow x'_{l+1})$ into ϕ; and for every vertex $y \in B_i(G)$ on level l, with only one child $z \in B_j(G)$ on level $l+1$, we insert clauses $(z_1 \Leftrightarrow y_1), \ldots, (z_l \Leftrightarrow y_l), (z_{l+1})$ into ϕ.

Let $f : V(G) \to V(T)$ be a $T(1,1)$-coloring of G, and label the vertices of T as follows: label vertices in the center of T with 1, for all other vertices concatenate the label of its parent with 0 if the vertex is the left child of its parent, in its block, and concatenate the label of its parent with 1 if it is the right, or only, child. For a vertex $v \in B_i(G)$ on level l, viewing the label of $f(v)$ as a truth assignment to the variables $v_1, \ldots, v_l$, digit by digit, we get a satisfying truth assignment τ for ϕ.

In the other direction of the proof we first use the degree refinement of G to determine which block $B_i(T)$ in the degree refinement of T a vertex $v \in V(G)$ must map to. If the center of T is an edge, T consists of two subtrees T_{left} and T_{right}, and we must decide whether a vertex v should map to the left or right subtree. In this case $B_1(G)$ induces a 2-regular graph, which is split by following the cycles, mapping two adjacent vertices to the left vertex of the center, two to the right, and so on. This split of $B_1(G)$ propagates down through the rest of G, and we map one part to T_{left}, the other to T_{right}. Finally, we use a truth assignment τ for ϕ and its restriction to variables $v_1, \ldots v_l$, to determine which vertex $u \in B_i(T)$ a vertex $v \in B_i(G)$ on level l must map to, in the same manner as described above. $\square$

4 $\mathcal{NP}$-complete Cases

In this section all remaining $T(p,q)$-COLORING problems are shown to be $\mathcal{NP}$-complete. Due to lack of space we give the full proof only for Lemma 6, the remaining proofs can be found in Appendix A. To resolve the complexity of the open H-COVER problems mentioned in the introduction, it will probably be necessary to generalize Lemma 6 to all graphs, hence its proof is of special interest.

Recall that the simple (p,q)-cover of H has $|V(H)| \cdot q(p+1)$ vertices. For simplicity we write $u_{(i-1)q+j}$ for the vertex $((u,a),b)$, when a is the i-th vertex of K_q, and b is the j-th vertex of K_{p+1}.

Lemma 6. *If T' is a tree isomorphic to a connected component of the subtree of T induced by some subset of degree refinement blocks $\mathcal{B}'_T \subseteq \mathcal{B}_T$, and the degree refinement of T' is identical to $\mathcal{B}'_T$ restricted to T', then $T'(p,q)$-COLORING reduces to $T(p,q)$-COLORING, for every p and q.*

Proof. We can assume $q \neq 0$, otherwise both $T'(p,q)$-COLORING and $T(p,q)$-COLORING are solvable in polynomial time, and the reduction follows trivially.

For every $u \in T'$ let T_u denote the component of $(V(T), E(T) \setminus E(T'))$ containing u, i.e., the part of T which hangs from u, but which does not belong to T'. Clearly trees T_u and $T_{u'}$ are isomorphic if u and u' belong to the same degree refinement block.

Let G' be an instance of $T'(p,q)$-COLORING. We may assume, that the blocks in the degree refinement of G', $\mathcal{B}_{G'} = \{B_1(G'), \ldots, B_t(G')\}$, correspond to the blocks $\mathcal{B}'_T$, otherwise G' has no $T'(p,q)$-coloring. We construct a graph G which will have a $T(p,q)$-coloring if and only if G' has a $T'(p,q)$-coloring.

For every vertex $v \in V(G')$ there is a gadget F_v. For $v \in B_i(G')$ F_v is constructed by taking the simple (p,q)-cover of T_u, for an arbitrary vertex $u \in B_i(T')$, and removing the edges connecting the $q(p+1)$ copies $u_1, \ldots, u_{q(p+1)}$ of vertex u. Note that gadgets F_v and $F_{v'}$ are isomorphic for vertices v and v' from the same degree refinement block. G is constructed by making $q(p+1)$ disjoint copies of G, where the a-th copy of a vertex $v \in V(G')$ is labeled v_a. For every vertex $v \in V(G)$ we insert the gadget F_v, and identify vertex v_a from the a-th copy of G and vertex u_a from F_v.

For an example of the construction of G see Fig. 3. The example shows a reduction from $P_2(2,1)$-COLORING, where P_2 appears as a block in T indicated by the white vertices. For a white vertex $u \in V(T)$, the tree T_u is depicted in the upper right corner together with the gadget F_v, the dotted edges connecting copies of u are removed. An instance G' and the constructed graph G are depicted in the lower part. The dashed edges are those that belong to the copies of G', while the solid edges belong to the gadgets F_v.

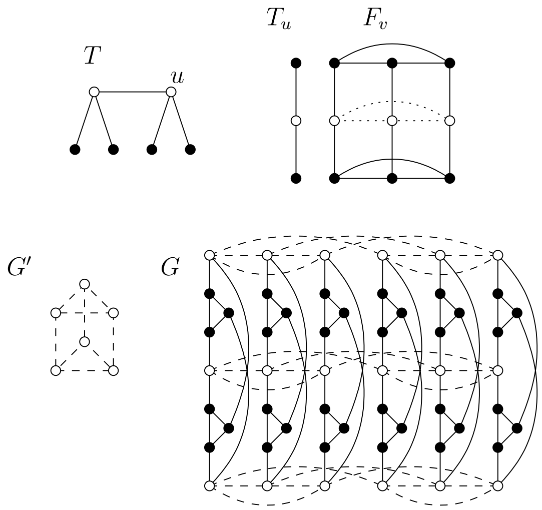

Fig. 3. Constructed graph G has a $T(2,1)$-coloring if and only if G' has $T'(2,1)$-coloring, for $T' = P_2$

Let $\mathcal{B}_G = \{B_1(G), \dots, B_t(G)\}$ be the degree refinement of G. We claim, that each $B_i(G)$ corresponds to the block $B_i(T) \in \mathcal{B}_T$. Every vertex $v \in B_i(G')$ is connected to exactly p neighbors inside $B_i(G')$, in G; every copy of v_a is therefore connected to the same p neighbors inside the a-th copy of G'. Similarly, v_a has the correct number of neighbors in every other block $B_j(G)$. Take any $u \in B_i(T)$, for $B_j(G') \in \mathcal{B}_{G'}$, $|N(v_a) \cap B_j(G)| = |N(v) \cap B_j(G')| = q \cdot |N(u) \cap B_j(T)|$ holds inside each copy of G', and for $B_j(G) \notin \mathcal{B}_{G'}$ we get the same equality, $|N(v_a) \cap B_j(G)| = q \cdot |N(u) \cap B_j(T)|$, due to the construction of the simple (p,q)-cover F_v of T_u. The properties of F_v assure the same for its vertices.

If G has a $T(p,q)$-coloring f, its restriction to a single copy of G' is a $T'(p,q)$-coloring. Only vertices of T' (or its isomorphic copy in T) may appear as colors of G', because the blocks in the degree refinement of T and G are in one-to-one correspondence if any such $T(p,q)$-coloring exists. Conversely, any $T'(p,q)$-coloring f' of G' can be extended to a $T(p,q)$-coloring f of G; we use the same mapping on each copy of G', i.e. $f(v_a) = f'(v)$ for all $1 \leq a \leq q(p+1)$, and extend it to each F_v as described in Lemma 1. $\square$

Lemma 7. *If no block in the degree refinement of H contains an edge, then $H(p,q)$-COLORING reduces to $H(p+1,q)$-COLORING, for every p and q.*

Lemma 8. *$P_2(p,q)$-COLORING is $\mathcal{NP}$-complete if $p \geq 1$, $q \geq 1$, except for the case $p = q = 1$.*

Lemma 9. *$P_3(p,q)$-COLORING is $\mathcal{NP}$-complete if either:*

(1) $p = 0$, $q \geq 3$, *or*
(2) $p = 1$, $q = 2$.

Lemma 10. *$S_k(0,q)$-COLORING is $\mathcal{NP}$-complete for $k \geq 2$ if $q \geq 3$.*

Lemma 11. *$S_k(p,q)$-COLORING is $\mathcal{NP}$-complete for $k \geq 3$ if either:*

(1) $p = 1$, $q = 1$, *or*
(2) $p = 1$, $q = 2$.

Lemma 12. *$T(0,2)$-COLORING is $\mathcal{NP}$-complete for every tree T whose degree refinement consists of three blocks, one of size 1 and the others of size at least 2.*

Lemma 13. *If T is a tree with one degree refinement block consisting of exactly one vertex u, and $2T$ is the tree made from two disjoint copies of T, T_1 and T_2, joined by the edge u_1u_2, then $T(p,q)$-COLORING reduces to $2T(p,q)$-COLORING, for every p and q.*

5 Completing the Proof

In the following we write $v \approx_G v'$ if vertices $v, v' \in V(G)$ belong to the same block in the degree refinement of G.

Lemma 14. *For any tree T with degree refinement $\mathcal{B}_T$, one of the following cases applies:*

(1) *All blocks of $\mathcal{B}_T$ contain only one vertex,*
(2) *there exists a block $B_i(T) \in \mathcal{B}$ of size 2, whose vertices induce an edge,*
(3) *there exists two adjacent blocks $B_i(T), B_j(T) \in \mathcal{B}$ whose vertices induce a disjoint union of stars $S_k, k \geq 3$, or*
(4) *for all pairs of adjacent blocks $B_i(T), B_j(T) \in \mathcal{B}_T$, their vertices either induce a perfect matching or a disjoint union of paths $P_3 = S_2$, and at least one pair inducing a P_3 exists in T.*

Proof. Assume T has at least two distinct vertices $u \approx_T u'$, otherwise the first case applies. Since T is connected, u and u' are connected by a path $P = v_1, \ldots, v_k$ with $u = v_1$ and $u' = v_k$ as its terminals. The fact $u \approx_T u'$ implies $v_i \approx_T v_{k+1-i}$. Therefore, if P is of odd length, its center $v_{\lfloor k/2 \rfloor}, v_{\lceil k/2 \rceil}$ induces an edge and the second case applies.

Otherwise vertices of two adjacent blocks always induce a disjoint union of stars $S_k, k \geq 1$. The last two cases distinguish whether an induced star S_k, $k \geq 3$ exists, or not. $\square$

Proof of Theorem 1. Throughout this case study we assume that the degree refinement of the tree T contains at least one block of size 2 or more, otherwise a polynomial-time algorithm follows from Theorem 3. There are nine cases depending on the values of p and q.

A. $p = 0$.
  - A1. $q = 1$. This case is equivalent to the tree-isomorphism problem, which is solvable in polynomial time, even if T is not fixed.
  - A2. $q = 2$. Assume T does not satisfy the conditions of Lemma 4. Since the center of any tree is always of size at most 2, the degree refinement of T contains two adjacent blocks $B_i(T)$ and $B_j(T)$, with $|B_i(T)| < |B_j(T)|$, and neither $B_i(T)$ nor $B_j(T)$ contain leaves. Hence, $B_j(T)$ is adjacent to another block $B_k(T)$, $|B_j(T)| \leq |B_k(T)|$. A connected component T' of the subtree of T induced by $B_i(T) \cup B_j(T) \cup B_k(T)$ satisfies the properties of Lemma 12, possibly with the application of Lemma 13. We apply Lemma 6 to T' to show $\mathcal{NP}$-completeness for T.
  - A3. $q \geq 3$. Assume T does not satisfy the conditions of Lemma 2. The degree refinement of T must then contain two adjacent blocks $B_i(T)$ and $B_j(T)$, with $|B_i(T)| \neq |B_j(T)|$. Any connected component T' of the subtree induced by $B_i \cup B_j$ is either isomorphic to S_k, $k \geq 2$, or to two such stars linked as described in Lemma 13. $\mathcal{NP}$-completeness for T follows from Lemmata 10 and 6.

B. $p = 1$.
  - B1. $q = 1$. Assume T does not satisfy the conditions of Lemma 5. Since $m_{ij} \geq 3$, any connected component T' of T, restricted to $B_i(T) \cup B_j(T)$, is either isomorphic to a star S_k, $k \geq 3$, or two such stars linked as described in Lemma 13. $\mathcal{NP}$-completeness for T follows from Lemmata 11 and 6.
  - B2. $q = 2$. By Lemma 14, T either contains (in the sense of Lemma 6) a block-induced subtree isomorphic to
    - P_2, in which case $\mathcal{NP}$-completeness for T follows from Lemma 8,
    - S_k, $k \geq 3$, in which case $\mathcal{NP}$-completeness for T follows from Lemma 11, or
    - $P_3 = S_2$, in which case $\mathcal{NP}$-completeness for T follows from Lemma 9.
  - B3. $q \geq 3$. As for B2 above. For a P_2 contained in T $\mathcal{NP}$-completeness follows from Lemma 8, for P_3 from Lemmata 9 and 7, and for S_k, $k \geq 3$ from Lemmata 10 and 7.

C. $p \geq 2$.
  - C1. $q = 1$. Assume T does not satisfy the conditions of Lemma 5. The degree refinement of T must then either contain a block $B_i(T)$ inducing a P_2, or a pair of blocks $B_i(T)$ and $B_j(T)$ inducing an S_k, $k \geq 3$. In the former case $\mathcal{NP}$-completeness for T follows from Lemma 8, in the latter from Lemmata 10 and 7.
  - C2. $q = 2$. As for B2 above. For a P_2 contained in T $\mathcal{NP}$-completeness follows from Lemma 8, for P_3 from Lemmata 9 and Lemma 7, and for S_k, $k \geq 3$ from Lemmata 11 and 7.
  - C3. $q \geq 3$. As for C2 above, with Lemma 10 instead of Lemma 11. □

References

1. James Abello, Michael R. Fellows, and John C. Stillwell. On the complexity and combinatorics of covering finite complexes. *Australasian Journal of Combinatorics*, 4:103–112, 1991.
2. Dana Angluin. Local and global properties in networks of processors (extended abstract). In *Conference Proceedings of the Twelfth Annual ACM Symposium on Theory of Computing*, pages 82–93, Los Angeles, California, April 28–30 1980.
3. Norman Biggs. *Algebraic Graph Theory*. Cambridge University Press, 1974.
4. Hans L. Bodlaender. The classification of coverings of processor networks. *Journal of Parallel and Distributed Computing*, 6:166–182, 1989.
5. Béla Bollobás. *Random Graphs*. Academic Press, 1985.
6. Bruno Courcelle and Yves Métivier. Coverings and minors: Applications to local computations in graphs. *European Journal of Combinatorics*, 15:127–138, 1994.
7. Jiří Fiala and Jan Kratochvíl. Partial covers of graphs. To appear in *Discussiones Mathematicae Graph Theory*.
8. Jonathan L. Gross and Thomas W. Tucker. *Topological Graph Theory*. J. Wiley and Sons, 1987.
9. Pavol Hell and Jaroslav Nešetřil. On the complexity of H-colouring. *Journal of Combinatorial Theory, Series B*, 48(1):92–110, 1990.
10. Ian Holyer. The NP-completeness of edge coloring. *SIAM Journal on Computing*, 10(4):718–720, 1981.
11. Jan Kratochvíl. k-in-m-satisfiability with q occurrences. Unpublished manuscript, 1997.
12. Jan Kratochvíl, Andrzej Proskurowski, and Jan Arne Telle. Complexity of colored graph covers I. Colored directed multigraphs. In *Graph-Theoretic Concepts in Computer Science, 23rd International Workshop, WG'97, Berlin, Germany, June 18–20 1997, Proceedings*, volume 1335 of *Lecture Notes in Computer Science*, pages 242–257. Springer-Verlag, 1997.
13. Jan Kratochvíl, Andrzej Proskurowski, and Jan Arne Telle. Covering regular graphs. *Journal of Combinatorial Theory, Series B*, 71(1):1–16, 1997.
14. Jan Kratochvíl, Andrzej Proskurowski, and Jan Arne Telle. Complexity of graph covering problems. *Nordic Journal of Computing*, 5(3):173–195, 1998.
15. Petter Kristiansen and Jan Arne Telle. Generalized H-coloring of graphs. In *Algorithms and Computation, 11th International Conference, ISAAC 2000, Taipei, Taiwan, December 18–20 2000, Proceedings*, volume 1969 of *Lecture Notes in Computer Science*, pages 456–466. Springer-Verlag, 2000.
16. D. Leven and Z. Galil. NP-completeness of finding the chromatic index of regular graphs. *Journal of Algorithms*, 4(1):35–44, 1983.
17. William S. Massey. *Algebraic Topology: An Introduction*. Harcourt, Brace and World, 1967. Reprinted 1977, Springer-Verlag.

Appendix A $\mathcal{NP}$-completeness Reductions

Lemma 7. *If no block in the degree refinement of H contains an edge, then $H(p,q)$-COLORING reduces to $H(p+1,q)$-COLORING, for every p and q.*

Proof. The problem is obviously in $\mathcal{NP}$. We reduce from $H(p,q)$-COLORING. Let G be an instance of $H(p,q)$-COLORING. We create an instance G' of $H(p+1,q)$-COLORING, such that G' has an $H(p+1,q)$-coloring if and only if G has an $H(p,q)$-coloring. G' is made up of two copies of G, G_1 and G_2, with every vertex $v_1 \in V(G_1)$ connected to the corresponding vertex $v_2 \in V(G_2)$ by an edge.

If G has an $H(p,q)$-coloring $f : V(G) \to V(H)$, we get an $H(p+1,q)$-coloring $f' : V(G') \to V(H)$ of G' by taking $f'(v_1) = f'(v_2) = f(v)$. Under any $H(p+1,q)$-coloring $f' : V(G') \to V(H)$ of G', both v_1 and v_2 must map to the same vertex, we get a $H(p,q)$-coloring f of G by taking $f(v) = f'(v_1)$. □

Lemma 8. *$P_2(p,q)$-COLORING is $\mathcal{NP}$-complete if $p \geq 1$, $q \geq 1$, except for the case $p = q = 1$.*

Proof. $P_2(p,q)$-COLORING is equivalent to BLACK/WHITE(p,q)-COLORING which asks if the vertices of a $(p+q)$-regular graph can be colored black and white such that every vertex has exactly p neighbors of the same color and q of the other color. It has been shown to be $\mathcal{NP}$-complete when p and q are positive and not both 1, see [7]. □

In the reduction below we use the following problem:

[(l,m,n)-SAT] ***l*-in-*m*-SATISFIABILITY WITH *n* OCCURRENCES**
INSTANCE: Set V of variables, collection C of clauses over V such that:
(1) every clause contains exactly m distinct, positive, variables, and
(2) every variable occurs in exactly n clauses.
QUESTION: Is C l-in-m satisfiable, that is, is there a truth assignment for C such that every clause has exactly l variables that evaluate to TRUE?

(l,m,n)-SAT was shown to be $\mathcal{NP}$-complete for every fixed l, m, n such that $0 < l < m$ and $n \geq 3$ by Kratochvíl in [11].

Lemma 9. *$P_3(p,q)$-COLORING is $\mathcal{NP}$-complete if either:*

(1) $p = 0$, $q \geq 3$, *or*
(2) $p = 1$, $q = 2$.

Proof. The problem is obviously in $\mathcal{NP}$. We reduce from (l,m,n)-SAT in both cases.

Case 1: We reduce from $(q,2q,q)$-SAT. Let (V,C) be an instance of $(q,2q,q)$-SAT, $q \geq 3$. We create an instance G of $P_3(0,q)$-COLORING, such that G has a $P_3(0,q)$-coloring if and only if a satisfying truth assignment exists for C. G is constructed by replacing every variable $v \in V$ with a variable vertex v_v, and every clause $c \in C$ with a clause vertex v_c. Clause vertices are connected to the variable vertices corresponding to the variables occurring in the clause.

Let τ be a satisfying truth assignment for the $(q, 2q, q)$-SAT instance, we get a $P_3(0, q)$-coloring of the corresponding graph by mapping the clause vertices v_c to the center vertex of the P_3, and the variable vertices v_v to one end-vertex if $\tau(v) = \text{TRUE}$, and to the other end-vertex if $\tau(v) = \text{FALSE}$. A reversal works for the other direction of the proof.

Case 2: We reduce from $(2, 4, 4)$-SAT. Let (V, C) be an instance of $(2, 4, 4)$-SAT with an even number of clauses. We create an instance G of $P_3(1, 2)$-COLORING, such that G has a $P_3(1, 2)$-coloring if and only if a satisfying truth assignment exists for C. G is constructed by replacing every variable $v \in V$ with a variable gadget G_v. G_v consists of two vertices v_v and v'_v connected by an edge. For each clause $c \in C$ there is a clause vertex v_c connected to the variable gadgets corresponding to the variables occurring in the clause. The variable gadget is connected in such a way that every vertex of the gadget is connected to exactly two clause vertices, the connecting vertex can be chosen arbitrarily. Each clause vertex is in addition connected to exactly one other [arbitrarily chosen] clause vertex.

Let τ be a satisfying truth assignment for the $(2, 4, 4)$-SAT instance, we get a $P_3(1, 2)$-coloring of the corresponding graph by mapping the clause vertices v_c to the center vertex of the P_3, and the variable gadgets G_v to one end-vertex if $\tau(v) = \text{TRUE}$, and to the other end-vertex if $\tau(v) = \text{FALSE}$. A reversal works for the other direction of the proof. $\square$

Lemma 10. *$S_k(0, q)$-COLORING is $\mathcal{NP}$-complete for $k \geq 2$ if $q \geq 3$.*

Proof. The problem is obviously in $\mathcal{NP}$. The star S_2 is equal to the path P_3 so we know that $S_2(0, q)$-COLORING is $\mathcal{NP}$-complete if $q \geq 3$. We reduce from $S_k(0, q)$-COLORING to $S_{k+1}(0, q)$-COLORING.

Let G be an instance of $S_k(0, q)$-COLORING. We create an instance G' of $S_{k+1}(0, q)$-COLORING, such that G' has an $S_{k+1}(0, q)$-coloring if and only if G has an $S_k(0, q)$-coloring. G' is constructed by adding several copies of the complete bipartite graphs $K_{q,(k+1)q-1}$ to G. These new vertices of low degree q will have no further neighbors and must therefore map to the leaves in any $S_{k+1}(0, q)$-coloring. The new vertices of high degree must map to the central vertex of the star and will each have one more new vertex as a neighbor, these neighbors will act as connectors to G. Note that the connectors of a single copy of $K_{q,(k+1)q-1}$ must all map to the same leaf of S_{k+1}. By arranging the new copies of $K_{q,(k+1)q-1}$ in a cycle and for each adjacent pair of copies having a pair of vertices sharing the same connector, we guarantee that all connectors must map to the same leaf of the star (which will be the $k + 1$st leaf). In this way, by taking c copies of $K_{q,(k+1)q-1}$ we get c shared connectors and $c(q - 2)$ non-shared connectors, lacking respectively $q-2$ and $q-1$ neighbors that will be assigned from G. Assuming G has n vertices of high degree kq, that each needs q new neighbors, we would like to choose c so that $c(q-2)+c(q-2)(q-1) = nq$, in other words, so that the number of edges needed for connectors, is the same as the number of new edges needed for G. Resolving, we get $c(q-2) = n$, so we can do this if the number of central vertices of G divides $q-2$. If it does not, we first increase G as follows. Since for any $x \geq q$ we have a bipartite q-regular graph with x vertices in each partition class, we also have a connected, simple graph F_x with x 'central' vertices of degree kq having an $S_k(0,q)$-coloring. Moreover, a graph G has an $S_k(0,q)$-coloring if and only if $G \cup F_x$ does. Thus, if $n \equiv z \mod (q-2)$, then $n+x \equiv 0 \mod (q-2)$ for $x = q(q-2)+(q-2-z) \geq q$ and we consider instead as input to the $S_k(0,q)$-coloring problem the graph $G \cup F_x$, and in this graph the number of new edges to the $k+1$st leaf matches the number of connector edges needed. As we have argued alongside the construction, if the constructed graph has an $S_{k+1}(0,q)$-coloring, then the connectors must all map to the same leaf, so that the coloring induced on the copy of G is an $S_k(0,q)$-coloring. Conversely, if G has an $S_k(0,q)$-coloring it is easy to see that the constructed graph G' has an $S_{k+1}(0,q)$-coloring.

In the reduction below we use the following problem:

[k-EC] ***k*-EDGE-COLORING**
INSTANCE: Graph G.
QUESTION: Can $E(G)$ be partitioned into k' disjoint sets $E_1, E_2, \ldots, E_{k'}$, with $k' \leq k$, such that, for $1 \leq i \leq k'$, no two edges in E_i share a common endpoint in G?

If G is a k-regular graph, the question becomes whether each vertex is incident to k distinctly colored edges. This last problem was shown to be $\mathcal{NP}$-complete for $k = 3$ in [10] and for $k \geq 3$ in [16]. We get the following result for the complexity of $S_k(1,1)$-COLORING and $S_k(1,2)$-COLORING.

Lemma 11. *$S_k(p,q)$-COLORING is $\mathcal{NP}$-complete for $k \geq 3$ if either:*

(1) *$p = 1$, $q = 1$, or*
(2) *$p = 1$, $q = 2$.*

Proof. Both problems are obviously in $\mathcal{NP}$. We reduce from k-EC on k-regular graphs in both cases.

Case 1: Let G be an instance of k-EC, such that G is k-regular. We construct a graph G', such that G' has an $S_k(1,1)$-coloring if and only if G is k-edge-colorable. G' is made up of two copies of G, G_1 and G_2, with every vertex $v_1 \in V(G_1)$ connected to the corresponding vertex $v_2 \in V(G_2)$ by an edge. In addition, every edge $vw \in E(G_1) \cup E(G_2)$ is subdivided twice, that is, every edge vw becomes a path v, v', w', w.

Let c be the center vertex of the star S_k, $l_1, l_2, \ldots, l_k$ the leaves, and let $f : E(G) \to \{1, 2, \ldots, k\}$ be a k-edge-coloring of G. We get an $S_k(1,1)$-coloring from f by mapping all vertices $v \in V(G_1) \cup V(G_2)$ to c, and the subdivision vertices of the edge vw to l_k if and only if $f(vw) = k$. In the other direction of the proof, we can let the $S_k(1,1)$-coloring induced on G_1 define the k-edge-coloring, since for every edge vw of G_1, and hence of G, both vertices v' and w' on the subdivided path v, v', w', w must map to the same leaf, and we use this leaf as the color of the edge.

Case 2: The reduction to $S_k(1,2)$-COLORING is identical except that for each edge $vw \in E(G_1) \cup E(G_2)$ we also add edges vw' and $v'w$ to the v, v', w', w-path. $\square$

Lemma 12. *$T(0,2)$-COLORING is $\mathcal{NP}$-complete for every tree T whose degree refinement consists of three blocks, one of size 1 and the others of size at least 2.*

Proof. The problem is obviously in $\mathcal{NP}$. We reduce from (l,m,n)-SAT. Let $B_1(T)$, $B_2(T)$, and $B_3(T)$ be the blocks in question, and assume $|B_1(T)| = 1$, $|B_2(T)| = k \geq 2$, and $|B_3(T)| = kt$, $t \geq 1$. We distinguish between the following two cases: (1) $k = 2$, and (2) $k \geq 3$.

Case 1: When $k = 2$ we reduce from $(2,4,4)$-SAT. Let (V, C) be an instance of $(2,4,4)$-SAT. We create an instance G of $T(0,2)$-COLORING, such that G has a $T(0,2)$-coloring if and only if a satisfying truth assignment exists for C. G is constructed by replacing every variable $v \in V$ with a variable gadget G_v. G_v consists of a complete bipartite graph $K_{2,2t}$ where the vertices of the first partition are labeled v_v and v'_v. Vertices v_v and v'_v are designated variable vertices. For each clause $c \in C$ there is a clause vertex v_c connected to the variable gadgets corresponding to the variables occurring in the clause. The variable gadget is connected in such a way that each variable vertex is connected to exactly two clause vertices, the connecting vertex can be chosen arbitrarily.

Let $B_2(T) = \{\text{left}_2, \text{right}_2\}$, and let τ be a satisfying truth assignment for the $(2,4,4)$-SAT instance. We get a $T(0,2)$-coloring of G by mapping the clause vertices v_c to the only vertex of $B_1(T)$, the v_v- and v'_v-vertices of the variable gadgets G_v to left_2 if $\tau(v) = \text{FALSE}$, and to right_2 if $\tau(v) = \text{TRUE}$. The remaining vertices of the variable gadgets $K_{2,2t}$ are mapped in pairs to the t distinct neighbors of left_2 or right_2, respectively. A reversal works for the other direction of the proof.

Case 2: When $k \geq 3$ we start with the graph G from the transformation described for case 1, and add $k-2$ connected bipartite gadgets G_g. $G_g = (V_1, V_2, E)$ is such that V_1 consists of $|C|$ vertices of degree $2t$ and V_2 consists of $|C| \cdot t$ vertices of degree 2. The gadgets G_g are connected to G by connecting each vertex of V_1 to two distinct clause vertices, and each clause vertex to two vertices from V_1. This forms an instance G' of $T(0,2)$-COLORING.

In any $T(0,2)$-coloring of G', the vertices of each V_1 must map to the same vertex of $B_2(T)$, due to the connectedness of the gadgets G_g. Hence, two arbitrarily chosen vertices from $B_2(T)$ can be used as colors determining the truth assignment to the variables corresponding to the vertices of the original graph G. $\square$

Lemma 13. *If T is a tree with one degree refinement block consisting of exactly one vertex u, and $2T$ is the tree made from two disjoint copies of T, T_1 and T_2, joined by the edge u_1u_2, then $T(p,q)$-COLORING reduces to $2T(p,q)$-COLORING, for every p and q.*

Proof. The problem is obviously in $\mathcal{NP}$. We reduce from $T(p,q)$-COLORING. Let G be an instance of $T(p,q)$-COLORING. We create an instance G' of $2T(p,q)$-COLORING, such that G' has a $2T(p,q)$-coloring if and only if G has a $T(p,q)$-coloring. G' is constructed by making $2q$ disjoint copies of G, $G_1, \ldots G_{2q}$. For every vertex $v \in V(G)$ we form a complete bipartite graph $K_{q,q}$ on the sets $\{v_1, \ldots, v_q\}$ and $\{v_{q+1}, \ldots, v_{2q}\}$.

For any $2T(p,q)$-coloring $f' : V(G') \to V(2T)$ of G', its restriction to a single copy of G is a $T(p,q)$-coloring. In the opposite direction, any $T(p,q)$-coloring $f : V(G) \to V(T)$ of G can be extended to G' by using the mapping to T_1 on the first q copies of G, and to T_2 on the last q copies of G. $\square$