Theoretical Computer Science 412 (2011) 2513–2523

Contents lists available at ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

Parameterized complexity of coloring problems: Treewidth versus vertex cover[☆]

Jiří Fiala[a,*], Petr A. Golovach[b], Jan Kratochvíl[a]

[a] *Department of Applied Mathematics and Institute for Theoretical Computer Science (ITI),[1] Charles University, Prague, Czech Republic*
[b] *School of Engineering and Computing Sciences, Durham University, South Road, Durham DH1 3LE, UK*

ARTICLE INFO

Keywords:
Graph colorings
Graph labelings
Parameterized complexity
Vertex cover

ABSTRACT

We compare the fixed parameter complexity of various variants of coloring problems (including List Coloring, Precoloring Extension, Equitable Coloring, $L(p,1)$-Labeling and Channel Assignment) when parameterized by treewidth and by vertex cover number. In most (but not all) cases we conclude that parametrization by the vertex cover number provides a significant drop in the complexity of the problems.
© 2010 Elsevier B.V. All rights reserved.

1. Introduction

An important aspect in parameterized complexity theory is the choice of the parameter for a problem. In particular, structural parameterizations measure the structural properties of the input. One of the best investigated structural parameters for graph problems is the treewidth of the input graph (see e.g. surveys [4,6]). While many problems are $\mathcal{FPT}$ when parameterized by the treewidth, there are problems that are $\mathcal{NP}$-hard even for graphs of small fixed treewidth (or even for trees). Also, there are problems that can be solved in polynomial time for graphs of bounded treewidth, but the exponent of the polynomial depends on the width (i.e., they are in $\mathcal{XP}$ if parameterized by the treewidth). Some of these problems are known to be W-hard, when parameterized by treewidth, which contributes to the fine structure of the $\mathcal{FPT}$ hierarchy. New results in this direction on $\mathcal{FPT}$-complexity of variants of graph coloring and domination problems were recently obtained in [16,10].

For problems that are difficult for graphs of bounded treewidth, it is interesting to consider different structural parameterizations that impose stronger restrictions. Fellows et al. proposed to study parametrization by the vertex cover number and they applied it to graph layout problems [15]. The goal of this paper is to pursue the road opened in [15] and apply this point of view to several variants of graph coloring problems, including problems stemming from the area of Frequency Assignment.

The *vertex cover number* of a graph G is the minimum size of a set W of vertices of G such that $I = V(G) \setminus W$ is an independent set. It is easy to see that the treewidth of a graph never exceeds its vertex cover number (an argument will be given in the next section), and thus parametrization by the vertex cover number has a chance to make problems easier. And indeed, for most (but not all) of the coloring and labeling problems considered in this paper, we conclude that parametrization by vertex cover number does make the problems more tractable.

☆ The extended abstract of this paper was presented at the 6th Annual Conference on Theory and Applications of Models of Computation TAMC'09 (Fiala et al., 2009) [19].

* Corresponding author.
E-mail addresses: fiala@kam.mff.cuni.cz (J. Fiala), petr.golovach@durham.ac.uk (P.A. Golovach), honza@kam.mff.cuni.cz (J. Kratochvíl).

1 ITI is supported by the Ministry of Education of the Czech Republic as project 1M0545.

0304-3975/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2010.10.043

Recall that a *(proper) vertex coloring* of a graph is an assignment c of colors to the vertices of the graph such that for any two adjacent vertices u and v, $c(u) \neq c(v)$. The set V_i of all vertices colored by a color i is called the i-th *color class*. Many variants of graph coloring have been intensively studied. We will consider the following decision problems:

List Coloring
Input: A graph G and for each vertex $v \in V(G)$, a list $L(v)$ of admissible colors.
Question: Is there a vertex coloring c with $c(v) \in L(v)$ for each v?

Precoloring Extension
Input: A graph G, a subset $U \subseteq V(G)$ of *precolored* vertices, a *precoloring* c_U of vertices of U, and a positive integer r.
Question: Is there a vertex coloring c of G which extends c_U (i.e. $c(v) = c_U(v)$ for each $v \in U$), using at most r colors?

(This is a special version of List Coloring when each list has either one element or contains all colors.)

Equitable Coloring
Input: A graph G and a positive integer r.
Question: Is there a vertex coloring c of G using at most r colors such that the sizes of any two color classes differ by at most one?

All these variants of graph coloring are NP-hard in the general case [27,3] and it is natural to explore their $\mathcal{FPT}$-complexity under different parameterizations as well [5].

Distance constrained labeling of graphs is a concept generalizing graph coloring that stems from the Frequency Assignment Problem. Here the colors (we prefer to call them *labels*) are nonnegative integers and requirements are posed on the difference of labels assigned to vertices that are close to each other [28,8]. In particular, given numbers p and q, an $L(p, q)$*-labeling* of span k of a graph is a labeling of its vertices by integers from $\{0, 1, \ldots, k\}$ such that the labels of adjacent vertices differ by at least p, and the labels of vertices at distance two in the graph differ by at least q. Thus we obtain the following associated problem (and we will also consider its List- and Prelabeling Extension variants defined in an obvious way):

$L(p, q)$-Labeling
Input: A graph G and a positive integer λ.
Question: Is there an $L(p, q)$-labeling l of G of the span λ?

This concept was intensively studied both for its practical motivation and for interesting theoretical properties. E.g., $L(2, 1)$-Labeling is polynomial time solvable for trees [9,23] but $\mathcal{NP}$-complete for graphs of treewidth two [17]. Moreover, for every $p > q > 1$, p and q relatively prime, $L(p, q)$-labeling becomes $\mathcal{NP}$-complete already for trees [18].

In this paper, we concentrate on the case $q = 1$; in this case it is simply required that labels assigned to vertices at distance two are distinct. Note also that in the case $p = q = 1$, $L(1, 1)$-labeling coincides with coloring the second distance power of the input graph (just beware of the offset 1 between the span of a labeling and the number of colors in a coloring), also previously intensively studied [2,13,14]. Also because of this meaningful correlation, we treat the case $p = q = 1$ in more detail and consider List $L(1, 1)$-Labeling and $L(1, 1)$-Prelabeling Extension problems as well.

Finally, we consider the Channel Assignment problem whose input is a graph equipped with integer weights on its edges, and the task is to assign nonnegative integers to its vertices so that the difference of the labels assigned to a pair of adjacent vertices is greater than or equal to the weight of the corresponding edge, while minimizing the span of the assignment (i.e., the largest label used).

Channel Assignment
Input: A graph G, a function $w : E(G) \to \mathbb{N}$, and a positive integer λ.
Question: Is there a so called channel assignment of G, i.e. a mapping $f : V(G) \to \{0, \ldots, \lambda\}$ satisfying $|f(u) - f(v)| \geq w(uv)$ for every edge $uv \in E(G)$?

This formulation also stems from the Frequency Assignment Problem, and e.g., the $L(p, q)$-Labeling problem with input graph G coincides with Channel Assignment for the second distance power of G and weights having only two values — p and q. However, note that the transition from G to its second power does not preserve bounded treewidth, so the $\mathcal{FPT}$-complexity results do not follow from one another. Yet Channel Assignment is known $\mathcal{NP}$-hard for graphs of treewidth at most three [26] (note only that in this case the size of the input is measured as $n + \log w$ where n is the number of vertices and w the maximum weight of an edge).

A comparison of known and new results on the fixed parameter complexity of the above mentioned problems for parametrization by treewidth versus parametrization by vertex cover number is summarized in Table 1.

It is readily seen that in most cases, parametrization by vertex cover number makes a problem easier with respect to parametrization by treewidth, typically making the complexity drop from W[1]-hard to $\mathcal{FPT}$ (Precoloring Extension, Equitable Coloring, $L(p, 1)$-Labeling for $p = 0$ or 1), but sometimes even from $\mathcal{NP}$-complete to $\mathcal{FPT}$ ($L(2, 1)$-Labeling and $L(1, 1)$-Prelabeling Extension). The complexity of the Channel Assignment problem drops from $\mathcal{NP}$-complete to $\mathcal{XP}$. For the List Coloring problem, we achieve matching lower bounds on the complexity, but for both parameterizations $\mathcal{W}[1]$ membership is open. The hardest of the considered problems proves to be List $L(1, 1)$-Labeling that remains $\mathcal{NP}$-complete even when parameterized by the vertex cover number.

Table 1
Complexity of coloring and labeling problems parameterized by treewidth and vertex cover, the results of this paper being denoted by [∗]. In the last four rows, k is the parameter (treewidth or vertex cover number).

	Treewidth	Vertex cover
List Coloring	$\mathcal{W}[1]$-hard [16]	$\mathcal{W}[1]$-hard [15, ∗]
Precoloring Extension	$\mathcal{W}[1]$-hard [16]	$\mathcal{FPT}$ [∗]
Equitable Coloring	$\mathcal{W}[1]$-hard [16]	$\mathcal{FPT}$ [∗]
$L(0, 1)$-Labeling	$\mathcal{W}[1]$-hard [∗]	$\mathcal{FPT}$ [∗]
$L(1, 1)$-Labeling	$\mathcal{W}[1]$-hard [∗]	$\mathcal{FPT}$ [∗]
$L(2, 1)$-Labeling	$\mathcal{NP}$-c for $k \geq 2$ [17]	$\mathcal{FPT}$ [∗]
List $L(1, 1)$-Labeling	$\mathcal{NP}$-c for $k \geq 2$ [∗]	$\mathcal{NP}$-c for $k \geq 4$ [∗]
$L(1, 1)$-Prelabeling Extension	$\mathcal{NP}$-c for $k \geq 2$ [∗]	$\mathcal{FPT}$ [∗]
Channel Assignment	$\mathcal{NP}$-c for $k \geq 3$ [26]	in $\mathcal{XP}$ [∗]

2. Notation and basic definitions

A parameterized problem with the input size n and a parameter k is called *fixed parameter tractable* ($\mathcal{FPT}$) if it can be solved in time $f(k) \cdot n^c$, where f is a function only depending on k and c is a constant. The basic complexity class for fixed parameter intractability is $\mathcal{W}[1]$. To show that a problem is $\mathcal{W}[1]$-hard, it is necessary to construct a parameterized reduction from a known $\mathcal{W}[1]$-hard problem. One of the basic conjectures of the parameterized complexity theory is that $\mathcal{FPT} \neq \mathcal{W}[1]$, and if this conjecture holds, then $\mathcal{W}[1]$-hard problems cannot be solved by $\mathcal{FPT}$-algorithms. We refer to the book of Downey and Fellows [11] for an excellent exposition of this concept.

We consider finite undirected graphs without loops or multiple edges. The vertex set of a graph G is denoted by $V(G)$ and its edge set by $E(G)$. A set $S \subseteq V(G)$ of pairwise adjacent vertices is called a *clique* and a set of pairwise nonadjacent vertices is called an *independent set*. For $v \in V(G)$, by $N_G(v) = \{u \in V(G) : uv \in E(G)\}$ we denote the *open neighborhood* of v, and $N_G(S) = \bigcup_{v \in S} N(v) \setminus S$ denotes the open neighborhood of a set $S \subseteq V(G)$. The *closed neighborhood* of a vertex v is $N_G[v] = N_G(v) \cup \{v\}$. The index G is often omitted if it is clear from the context.

A *tree decomposition* of a graph G is a pair (X, T) where T is a tree whose vertices we will call *nodes* and $X = (\{X_i \mid i \in V(T)\})$ is a collection of subsets (usually referred to as *bags*) of $V(G)$ such that $\bigcup_{i \in V(T)} X_i = V(G)$, both endpoints of each edge of G belong to some bag, and for each vertex of G, the bags containing it induce a connected subtree in T. The *width* of the decomposition equals $\max_{i \in V(T)}\{|X_i| - 1\}$ and the *treewidth* of G is the minimum width over all of its tree decompositions. We use notation $\mathbf{tw}(G)$ to denote the treewidth of a graph G.

A subset W of the vertex set of a graph G is a *vertex cover* of G if every edge of G has at least one end-vertex in W. The minimum size of a vertex cover is called the *vertex cover number* and is denoted by $\mathbf{vc}(G)$. The vertex cover of the minimum cardinality can be constructed by an $\mathcal{FPT}$-algorithm [7]. Given a vertex cover W, a suitable tree decomposition is a star with the central bag being W and the leaf-nodes carrying bags $W \cup \{x\}$ for $x \in V(G) - W$. Therefore, $\mathbf{tw}(G) \leq \mathbf{vc}(G)$.

3. Complexity of coloring problems

3.1. *Complexity of the* List Coloring *and* Precoloring Extension *problems*

It has been mentioned without proof in the conclusion of [15] that List Coloring remains $\mathcal{W}[1]$-hard when parameterized by the vertex cover number. We note that the problem remains hard even for a special class of *split graphs*. (A graph G is a split graph if its vertex set can be partitioned into a clique and an independent set.)

As for any chordal graph, the treewidth of a split graph is less by one than the size of its maximum clique, and so the size of the clique part of G is an upper bound on both $\mathbf{tw}(G)$ and $(\mathbf{vc}(G))$.

Theorem 1. *The* List Coloring *problem is $\mathcal{W}[1]$-hard for split graphs with the size of the maximum clique being the parameter.*

Proof. We reduce the Weighted Antimonotone 2-CNF Satisfiability problem, which is known to be $\mathcal{W}[1]$-complete [12,11]:

Instance: A Boolean formula ϕ in conjunctive normal form, where each clause consists of two negated variables.
Parameter: k.
Question: Does ϕ have a satisfying assignment of weight k (i.e., exactly k variables have value true)?

Let $x_1, \ldots, x_n$ be the variables of ϕ, and suppose that $C_1, \ldots, C_j$ are its clauses. We start our construction with a clique of k vertices $u_1, \ldots, u_k$ with identical lists of colors $\{1, \ldots, n\}$. For each clause $C_j = \overline{x}_p \vee \overline{x}_q$, a vertex v_j with the list of possible colors $\{p, q\}$ is added and joined by edges to all vertices $u_1, \ldots, u_k$. Denote the obtained graph by G. Clearly, G is a split graph and $\{u_1, \ldots, u_k\}$ is its clique of size k. We claim that ϕ has a satisfying assignment of weight k if and only if G allows a list coloring.

Suppose that the formula ϕ has a satisfying assignment such that variables $x_{i_1}, \dots, x_{i_k}$ have value true. We color the vertices $u_1, \dots, u_k$ by colors $i_1, \dots, i_k$. Each clause $C_j = \bar{x}_p \vee \bar{x}_q$ contains a literal, say $\bar{x}_p$, with value true. Then we color vertex v_j by the color p. Since $x_p =$ false, this color was not used for coloring of $u_1, \dots, u_k$, and the list coloring is correct.

Assume now that G has a list coloring, and the vertices $u_1, \dots, u_k$ are colored by k different colors $i_1, \dots, i_k$. The choice of $x_i =$ true only for $i \in \{i_1, \dots, i_k\}$ provides a satisfying assignment of ϕ of weight k. □

Since the size of the maximum clique of a split graph differs from its vertex cover number by no more than one, it also follows (perhaps somewhat surprisingly), that List Coloring of split graphs is $\mathcal{W}[1]$-hard when parameterized by the vertex cover number. In contrast with the closely related Precoloring Extension:

Theorem 2. *The* Precoloring Extension *problem is* $\mathcal{FPT}$ *when parameterized by the vertex cover number.*

Proof. Suppose that W is a vertex cover of G, and let $|W| = k$. Let $I = V(G) \setminus W$ and let X be the set of non-precolored vertices of I. Finally let $c_U : V(G) \setminus X \longrightarrow \{1, \dots, r\}$ be the given precoloring.

We reduce the problem to a list coloring problem for the subgraph H of G induced by $W \cup X$. For each vertex $v \in X$, we set $L(v) := \{1, \dots, r\}$. If $w \in W$ is precolored, then we set $L(w) := \{c_U(w)\}$, otherwise $L(w) := \{1, \dots, r\} \setminus c_U(N(w))$, i.e., we exclude the colors of precolored neighbors of w. Clearly, G allows a precoloring extension with at most r colors if and only if H has a feasible list coloring. We distinguish two cases.

If $r > k$ then vertices of X are irrelevant since they can be always list-colored by a greedy algorithm. Similarly for vertices $w \in W$ such that $|L(w)| \geq k$. Hence H is L-colorable if and only if the subgraph F of H induced by the vertices of W with $|L(w)| < k$ is L-colorable. But this can be checked in time $O(k^k)$ since F has at most k vertices and each of them can be colored by at most k colors.

If $r \leq k$ then $|L(w)| \leq k$ for any vertex $w \in W$. We consider all colorings of W, and their extensions to H by the greedy algorithm. Since W has at most k^k colorings, the running time is $O(k^{k+1}n)$.

Since H can be constructed in time $O(r(n+m))$ where $n = |V(G)|$ and $m = |E(G)|$, the total running time of the algorithm is $O(k^{k+1}n + r(n + m))$. □

3.2. *Complexity of the* Equitable Coloring *problem*

The third variant of graph coloring we want to explore is Equitable Coloring. When showing that this problem becomes easy when parameterized by the vertex cover number, we utilize the approach used in [15]. The main idea is to reduce our problem to the integer linear programming problem that is $\mathcal{FPT}$ when parameterized by the number of variables. Formally, we use the following problem:

p-Variable Integer Linear Programming Feasibility
Input: A $q \times p$ matrix A with integer elements, an integer vector $b \in \mathbb{Z}^q$.
Parameter: p.
Question: Is there a vector $x \in \mathbb{Z}^p$ such that $A \cdot x \leq b$?

It was proved by Lenstra [25] that this problem is $\mathcal{FPT}$, and this algorithmic result was improved afterwards by different authors (see, e. g., a survey [1]). We are going to use it in the following form:

Theorem 3 ([25,24,20]). *The* p-Variable Integer Linear Programming Feasibility *problem can be solved using* $O(p^{2.5p+o(p)} \cdot L)$ *arithmetic operations and space polynomial in* L*, where* L *is the number of bits of the input.*

We now focus on the Equitable Coloring problem.

Theorem 4. *The* Equitable Coloring *problem is* $\mathcal{FPT}$ *when parameterized by the vertex cover number.*

Proof. Let W be a vertex cover of size k of a graph G on n vertices, and let $\{I_1, \dots, I_s\}$ be the partition of $I = V(G) \setminus W$ according to their neighborhoods, i.e., such that any two vertices $u, v \in I$ belong to the same I_i if and only if $N(u) = N(v)$. Note that $s \leq 2^k - 1$ (assuming that G is connected).

Let r be the required number of colors. Set $t = \lfloor \frac{n}{r} \rfloor$. Any equitable coloring of G contains $a = n - rt$ color classes of cardinality $t + 1$ and $b = r - a$ color classes of cardinality t. We distinguish two cases:

If $r \leq k$, then for each proper coloring $V_1, \dots, V_r$ of W we construct a system of linear integer inequalities with sr variables $x_{i,j}$, $i \in \{1, \dots, s\}$ and $j \in \{1, \dots, r\}$, where $x_{i,j}$ will express the number of vertices of color j in the set I_i:

$$\begin{cases} x_{i,j} \geq 0, & \\ x_{i,j} = 0, & \text{if color } j \text{ is used in } N(I_i), \\ x_{1,j} + \cdots + x_{s,j} = t + 1 - |W \cap V_j|, & \text{if } j \in \{1, \dots, a\}, \\ x_{1,j} + \cdots + x_{s,j} = t - |W \cap V_j|, & \text{if } j \in \{a+1, \dots, r\}, \\ x_{i,1} + \cdots + x_{i,r} = |I_i| & \text{for every } i \in \{1, \dots, s\}. \end{cases}$$

It can be easily seen that this problem has an integer solution if and only if there is an equitable coloring of G which extends the starting coloring of W. Since W has at most k^k colorings and the number of variables is at most $k(2^k - 1)$, the Equitable Coloring problem can be solved in $\mathcal{FPT}$-time.

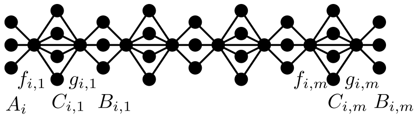

Fig. 1. Construction of a chain of graphs F.

If $r > k$ then we impose the following assumptions on the desired equitable coloring: Vertices of W are colored by colors $\{1, \dots, k\}$ and there exists an integer l between $\max\{0, k-b\}$ and $\min\{k, a\}$ such that the color classes $V_1, \dots, V_l$ contain $t+1$ vertices and the color classes $V_{l+1}, \dots, V_k$ only t vertices.

By permuting the names of colors, every equitable coloring of G gives rise to a coloring satisfying the above conditions. On the other hand, if a partial coloring of G with k colors exists, such that all vertices of W are colored and the conditions are satisfied, then it can be extended to an equitable coloring of G: Any color from the set $\{k+1, \dots, r\}$ can be used for coloring of an arbitrary vertex of I.

Hence we consider each possible coloring of W by colors $1, \dots, k$, and for every l such that $\max\{0, k-b\} \le l \le \min\{k, a\}$, we construct a system of linear integer inequalities with sk variables $x_{i,j}$, $i \in \{1, \dots, s\}$ and $j \in \{1, \dots, k\}$, where $x_{i,j}$ will express the number of vertices of color j in the set I_i:

$$\begin{cases} x_{i,j} \ge 0, & \\ x_{i,j} = 0, & \text{if color } j \text{ used in } N(I_i), \\ x_{1,j} + \cdots + x_{s,j} = t + 1 - |W \cap V_j|, & \text{if } j \in \{1, \dots, l\}, \\ x_{1,j} + \cdots + x_{s,j} = t - |W \cap V_j|, & \text{if } j \in \{l+1, \dots, k\}, \\ x_{i,1} + \cdots + x_{i,k} \le |I_i| & \text{for every } i \in \{1, \dots, s\}. \end{cases}$$

Since there are at most k^k colorings of W, and the number of variables is bounded by $k(2^k - 1)$, we again conclude that the problem is solvable in $\mathcal{FPT}$ time. □

4. Complexity of the $L(p, 1)$-Labeling problems

4.1. Parametrization by treewidth

In this section we consider the $L(p, 1)$-Labeling problems for $p = 0, 1$. It was proved in [17] that the $L(2, 1)$-Labeling problem is $\mathcal{NP}$-complete even for graphs of treewidth two (and this result can be extended for any fixed $p \ge 2$). On the other hand, it was shown in [29] that the $L(1, 1)$-Labeling problem can be solved by a dynamic programming algorithm in time $O(\Delta^{2^{8(t+1)+1}} \cdot n) + O(n^3)$, for n-vertex graphs of treewidth at most t with maximum degree Δ, and the same holds for $L(0, 1)$-Labeling. We show here that it is impossible to solve these problems in $\mathcal{FPT}$-time unless $\mathcal{FPT} = \mathcal{W}[1]$.

Theorem 5. *The $L(0, 1)$-Labeling and $L(1, 1)$-Labeling problems are $\mathcal{W}[1]$-hard when parameterized by the treewidth.*

Proof. As a sample of a hardness proof, we show the result for $L(1, 1)$-Labeling, the proof for $L(0, 1)$-Labeling is analogous.

For positive integers l and λ such that $l \le \lambda - 1$, we first construct an auxiliary graph $F = F(l, \lambda)$ on the vertex set $V(F) = \{a_1, \dots, a_l, b_1, \dots, b_l, c_1, \dots, c_{\lambda-l-1}, f, g\}$. Vertices f and g are adjacent and together dominate all the other vertices: f is adjacent to all a_i's and all c_j's, and g is adjacent to all b_i's and all c_j's. No other edges are present.

As the vertex f is of degree λ, all labels $\{0, \dots, \lambda\}$ need to be used in any $L(1, 1)$-labeling of f together with its neighbors. The same argument holds also for each g.

We form a *chain* of graphs $F(l, \lambda)$ by merging every b_i with the corresponding a_i of the consequent copy (see Fig. 1). The following fact follows easily by the induction on the length of a chain:

Observation 6. *For any $L(1, 1)$-labeling of a chain of graphs $F(l, \lambda)$ of the span λ, the labels used on $a_1, \dots, a_l$ are distinct. These labels are identical with the set of labels used on $b_1, \dots, b_l$ of the last graph $F(l, \lambda)$ in the chain. Also any labeling of $a_1, \dots, a_l$ by different labels from the set $\{0, \dots, \lambda\}$ can be extended to an $L(1, 1)$-labeling of the whole chain of span λ.*

We reduce from the Equitable Coloring problem. It was proved in [16] that it is $\mathcal{W}[1]$-hard when parameterized both by the treewidth and r. Let G be a graph on n vertices $u_1, \dots, u_n$ with m edges, for which an equitable coloring by r colors is questioned. Assume without loss of generality that r divides n, and let $l = \frac{n}{r}$. Define $\lambda = n + m + 1$.

Let $(\{X_i \mid i \in V(T)\}, T)$ be a tree decomposition of G. We assume that T has maximum degree three, node 1 is a leaf, $|X_1| = 1$, and that for any two adjacent nodes i and j of T, $|X_i \setminus X_j| + |X_j \setminus X_i| \le 1$ (this may only increase the size of T linearly).

Choose a walk $P = 1 \dots s$ in T that visits every node of T at least once and at most three times. Let $e_1, \dots, e_m$ be the edges of G reordered in the order as they first occur in the bags $X_1, \dots, X_s$.

Now we take r disjoint chains of graphs $F(l, \lambda)$ of length m. We denote the set of vertices $\{a_1, \dots, a_l\}$ of the first copy of F in the i-th chain by A_i. The set $\{b_1, \dots, b_l\}$ of the j-th copy of F in the i-th chain will be denoted by $B_{i,j}$ and in an analogous manner we use symbols $C_{i,j}$, $f_{i,j}$, and $g_{i,j}$.

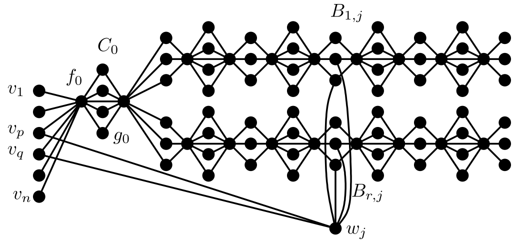

Fig. 2. Construction of H.

We proceed by adding a copy of $F(n, \lambda)$. We distinguish vertices $a_1, \ldots, a_n$ of this copy by renaming them to $v_1, \ldots, v_n$. Vertices $b_1, \ldots, b_n$ are merged in a one-to-one manner to the vertices of the union $A_1 \cup \ldots \cup A_r$. In an analogous way we use notation f_0, g_0 and C_0 for the remaining vertices in $F(n, \lambda)$.

For each edge $e_j = u_p u_q$ in the graph G, we introduce a new vertex w_j. We make w_j adjacent to vertices v_p, v_q, and to $l-1$ vertices of each set $B_{i,j}$ with $i \in \{1, \ldots, r\}$ (hence each w_j is of degree $2 + (l-1)r$).

Denote the obtained graph by H (see Fig. 2). We argue for the correctness of the reduction in the following lemma.

Lemma 7. *The graph G has an equitable coloring by r colors if and only if H has an $L(1, 1)$-labeling of span λ.*

Proof. Suppose that G has an equitable coloring using r colors. Each color i used on G will correspond to labels $\alpha_{i,1}, \ldots, \alpha_{i,l}$ in H. We further use $m+2$ additional labels $\beta_1, \ldots, \beta_m$ and γ_1, γ_2. We obtain the desired $L(1, 1)$-labeling of H of span $n + m + 1 = \lambda$ as follows:

- The vertices of $\{v_p : u_p$ is colored by color $i\}$ are labeled by different labels $\alpha_{i,1}, \ldots, \alpha_{i,l}$.
- For $i \in \{1, \ldots, r\}$, vertices of A_i are labeled by $\alpha_{i,1}, \ldots, \alpha_{i,l}$.
- Each vertex w_j is labeled by β_j.
- If w_j is adjacent to v_p where v_p has been labeled by some $\alpha_{i,y}$, we use the same label $\alpha_{i,y}$ on the only non-neighbor of w_j in $B_{i,j}$.
- The labeling of all other vertices are deduced from Observation 6, in particular:
  - labels of the sets $B_{i,j}$ are completed arbitrarily to $\alpha_{i,1}, \ldots, \alpha_{i,l}$,
  - the vertices of $C_{i,j}$ are labeled by $\alpha_{k,y}$ with $k \neq i$, and by β_x, $x = 1, 2, \ldots, m$, and
  - all vertices $f_{i,j}$ together with f_0 are labeled by γ_1 and the vertex g_0 together with all $g_{i,j}$ are labeled by γ_2.

If a vertex w_j is adjacent to vertices v_p and v_q, it means that the corresponding vertices u_p and u_q are adjacent in G (they form the edge e_j), the labels $\alpha_{i,j}$ of v_p and v_q differ in the subscript i. Hence it suffices that w_j has one non-neighbor in each set $B_{i,j}$ for the labeling to be well defined. It is a routine check that we have constructed a correct $L(1, 1)$-labeling of H.

Assume now that H has an $L(1, 1)$-labeling of the span λ. The vertices of the sets A_i must be labeled by different labels, and the same labels are used for labeling the vertices $v_1, \ldots, v_n$ by Observation 6. As above we denote the labels used for vertices of each set A_i by $\alpha_{i,1}, \ldots, \alpha_{i,l}$.

Now, we color a vertex u_p of G by color i if the corresponding v_p is labeled in H by some $\alpha_{i,y}$. Observe that each color is used exactly on l vertices, and thus forms an equitable coloring. Assume for a contradiction that for some edge $e_j = u_p u_q \in E(G)$, the vertices u_p and u_q are colored by the same color i. It means that the vertices v_p and v_q are labeled by some labels $\alpha_{i,x}$ and $\alpha_{i,y}$. By Observation 6 the vertices of $B_{i,j}$ are labeled by $\alpha_{i,1}, \ldots, \alpha_{i,l}$. By the construction of H, the vertex w_j has $l-1$ neighbors in $B_{i,j}$. Hence at least one of them has the same label as v_p or as v_q, a contradiction. □

To conclude the proof of the theorem, it is necessary to prove that H has a bounded treewidth.

Lemma 8. *The treewidth of the graph H is at most $(2r+2)\mathbf{tw}(G) + 3r + 2$.*

Proof. We construct a tree decomposition of H from the tree decomposition $(\{X_i \mid i \in V(T)\}, T)$ of G of width t. For the same tree T, we introduce sets Y_i by first putting v_j to Y_i if and only if $u_j \in X_i$ for $j = 1, \ldots, n$, and then adding f_0, g_0 to each Y_i. Then we alter the tree by adding some leaves and obtain the decomposition of H by adding further vertices to the bags.

The changes are performed inductively by following the walk $P = 1 \ldots s$. Since $|X_1| = 1$, this bag contains no edge of G. Let $Y_1 := Y_1 \cup \{f_{1,1}, \ldots, f_{r,1}\}$. For each vertex $z \in C_0 \cup \bigcup_{i=1}^{r} A_i$, a new leaf node d_z adjacent to 1 is added in T. We define the corresponding bag as $Y_{d_z} := N_H[z]$ (it contains 3 vertices).

For the induction we use an auxiliary variable h, initialized by $h := 1$.

Suppose that we have already made modifications of the tree decomposition for a subwalk $1 \ldots i_{j-1}$ of P. Recall that the edges of G are listed the order in which they occur in the bags $X_1, \ldots, X_s$. Let $E_j = \{e_x, \ldots, e_y\}$ be the set of edges that first occur in X_{i_j}.

If $E_j = \emptyset$ then we set $Y_{i_j} := Y_{i_j} \cup \{f_{h,1}, \ldots, f_{h,r}\}$ (if $h > m$ this set is assumed to be empty), and consider the next node of the walk.

Suppose that $E_j \neq \emptyset$. Then we set

$$Y_{i_j} := Y_{i_j} \cup \{w_x, \dots, w_y\} \cup \bigcup_{p=x}^{y} (\{f_{p,1}, \dots, f_{p,r}\} \cup \{g_{p,1}, \dots, g_{p,r}\}) \cup \{f_{y+1,1}, \dots, f_{y+1,r}\}$$

(if $y = m$ then the last set is empty). Since $|X_{i_j} \setminus X_{i_{j-1}}| \leq 1$, E_j contains at most t edges, and we added at most $t(2r+1)+r$ vertices. For each vertex $z \in \bigcup_{q=1}^{r} \bigcup_{p=x}^{y} (B_{q,p} \cup C_{q,p})$, a leaf node d_z is added to the tree and joined to i_j, and gets associated with the bag $Y_{d_z} := N_H[z]$ (note that $|Y_{d_z}| \leq 4$). Finally we set $h := y + 1$.

It is easy to check that we constructed a valid tree decomposition of H. The walk P visits any vertex at most three times. When it visits a node i the first time, we add at most $t(2r+1)+r$ vertices to the initial set Y_i, and when the walk goes to i the second or the third time then at most r vertices are added. Therefore, each bag Y_i contains at most $t+1+2+t(2r+1)+r+r+r$ vertices, and $\mathbf{tw}(H) \leq (2r+2)t + 3r + 2$. □

Recall that Equitable Coloring is $\mathcal{W}[1]$-hard when parameterized *both* by the treewidth and r. So Lemma 8 completes the proof of the theorem for the case of $L(1,1)$-labeling.

The argument for $L(0,1)$-labeling is almost the same. The only difference is that in the definition of $F(l, \lambda)$, the edge fg is missing while both f and g are adjacent to one additional common neighbor $c_{\lambda+1-l}$. □

4.2. Parametrization by vertex cover

In the case of $L(p,1)$-Labeling problems, the decrease in complexity is most visible (note that for $p > 1$ even from NP-hardness to FPT):

Theorem 9. *For every p, the $L(p,1)$-Labeling problem is $\mathcal{FPT}$ when parameterized by the vertex cover number.*

Proof. As in the case of the Equitable Coloring problem we use reductions to systems of integer linear inequalities. The cases of $p = 0, 1$ are considerably simpler than the case $p > 1$, since we have to be careful about the linear ordering of the label space in the latter case.

We first consider the case of $p = 1$. Assume that W is a vertex cover of size k and $I = \bigcup_{i=1}^{s} I_i$ is the partition of $V(G) \setminus W$ according to neighborhoods as in the proof of Theorem 4. Note that in any valid $L(1,1)$-labeling, the vertices of I_i have to be labeled by different labels.

If $\lambda < k$ then an $L(1,1)$-labeling of G exists only if $|I_i| \leq k$. In this case, $n \leq k2^k$ and the $L(1,1)$-Labeling problem can be decided in time $O(k^{k2^k})$.

Hence, it remains to consider the case of $\lambda \geq k$. Let $\{X_0, X_1, \dots, X_t\}$ be the collection of all possible set systems on $\{I_1, \dots, I_s\}$ such that wheneverI_j and $I_{j'}$ are distinct elements of a system X_i then $N(I_j) \cap N(I_{j'}) = \emptyset$. In particular we assume that $X_0 = \emptyset$.

Any $L(1,1)$-labeling of G of span λ determines a partition of the set $\{0, \dots, \lambda\}$ into sets $L_0, \dots, L_t$ as follows: For each $i = 1, \dots, t$ the labels in L_i are exactly those labels that are used once on a single vertex of every $I_j \in X_i$ and on no other vertices of I. Consequently, L_0 is the set of labels that are not used on I. Note that some sets L_i may be empty.

We try all partial $L(1,1)$ labelings of $G[W]$ of span at most $k-1$ (by permuting the labels if necessary we may assume that only labels $0, 1, \dots, k-1$ are used on W). There are at most k^k such labelings. For each of them we further try the at most t^k placements of the labels $0, \dots, k-1$ into the sets $L_1, \dots, L_t$. (If a label r is placed in some L_i then r should not be used as a label for a neighbor of elements of X_i nor for another vertex adjacent to such a neighbor.) The total number of such nonisomorphic partial labelings is bounded by $(tk)^k$.

For a fixed partial labeling let a_i be the number of labels placed in L_i. We decide whether this partial labeling can be extended to the entire graph G by deciding the feasibility of the following system of linear integer inequalities with $t+1$ variables $x_0, \dots, x_t$:

$$\begin{cases} x_0 + \cdots + x_t \leq \lambda + 1, & \\ \quad x_i \geq a_i, & \text{for } i \in \{0, \dots, t\}, \\ \displaystyle\sum_{i: I_j \in X_i} x_i = |I_j|, & \text{for } j \in \{1, \dots, s\}. \end{cases}$$

Similarly to the proof of Theorem 4, each variable x_i denotes the number of labels in L_i in the desired labeling. The correspondence between a solution of the system and a valid $L(1,1)$-labeling is straightforward: A partial labeling of W (given together with placements of the labels used into the sets L_i) can easily extended to a valid $L(1,1)$-labeling by adding new extra $x_i - a_i$ labels and using them (arbitrarily) on the so far unlabeled vertices of X_i, one vertex in each $I_j \in X_i$.

By the above discussion and according to Theorem 3, the total running time of this process is

$$O(kn + (tk)^k((s+t+2)^{2.5(s+t+2)+o(s+t+2)}(s+t+2)(t+1+\log n))) = O(kn + 2^{2^{2^{k+1}}} \log n).$$

The first summand stands for the partition of I into sets I_j. The next term stands for at most $(tk)^k$ partial labelings. The last term corresponds to the procedure of deciding feasibility of a system of $s+t+2$ inequalities with $t+1$ variables — the matrix is 0/1 valued and the entries on the right hand side are bounded by n, thus each requiring $O(\log n)$ bits. This completes the proof for the case $p = 1$.

For the case of $p = 0$ we need only minor modifications. We try all at most k^k feasible labelings of W, i.e. labelings where vertices at distance two are labeled by different labels. For each such partial labeling, we consider all at most $(t+1)^k$ placements of $0, \ldots, k-1$ in sets $L_0, \ldots, L_t$ satisfying the condition that if a label r is placed in L_i then r is not used as a label of a vertex adjacent to a neighbor of an element in X_i.

For $p > 1$, we use the same approach but must be a little more careful, since it is necessary to take into account the ordering of the labels. We use the same notation $W, k, I, \{I_1, \ldots, I_s\}, \{X_0, \ldots, X_t\}$ as in the previous cases.

If $\lambda < 2pk$ then the existence of an $L(p, 1)$-labeling of G of span λ can be decided by brute-force enumeration in constant time: In any valid $L(p, 1)$-labeling, the labels used on the vertices of each I_i are all different, and thus G may have at most $k + 2psk$ vertices.

Assume that $\lambda \geq 2pk$. We claim that any $L(p, 1)$-labeling of G can be transformed into a labeling of the same span λ with the additional property that the labels of W are in the set $\{0, \ldots, p(2k-1)\} \cup \{\lambda - p(2k-1), \ldots, \lambda\}$.

To prove this claim, consider a situation when two labels $a, b \in \{0, \ldots, \lambda\}$ are such that their difference with all labels used on W is at least p. If $a < b$ then we can permute labels of G by replacing $a \to a+1 \to \cdots \to b \to a$ to obtain a valid $L(p, 1)$-labeling of G: The $2p-1$ labels that differed by at most p from the label of a vertex of W were either all shifted or they all stayed in their places, so no labeling constraint along an edge with at least one endpoint in W got violated. And any permutation of labels keeps the labels of vertices with a common neighbor distinct. Using such relabelings (and the symmetric ones), one can "sweep" the labels of W towards the ends of the interval $\{0, \ldots, \lambda\}$ so that the difference of any two consecutive ones (in each of the two groups) is at most $2p$.

Similarly as in the proof for $p = 1$ we try all partial $L(p, 1)$-labelings of $G[W]$ using labels $\{0, \ldots, 2pk\} \cup \{\lambda - 2pk, \ldots, \lambda\}$ such that the difference between labels of W and $2pk$ or $\lambda - 2pk$ is at least $p - 1$, and all distributions of the labels $\{0, \ldots, 2pk\} \cup \{\lambda - 2pk, \ldots, \lambda\}$ into the classes L_i. The completion of each partial labeling to the entire graph G is achieved with labels from the set $\{r+1, \ldots, \lambda - r - 1\}$. The existence of such an extension is decided with the help of a system of integer inequalities analogous to that in the case $p = 1$. Note that the labels of yet unlabeled vertices are at least p apart from their neighbors in W due to the assumptions posed on the partial labelings. The overall time complexity of this method is again $O(kn + 2^{2^{2^{k+1}}} \log n)$. □

5. Labeling as coloring of the distance power

We have already mentioned that the special case of $L(p, q)$-Labeling for $p = q = 1$ coincides with the coloring of the second distance power of the input graph. As such, it has attracted the attention of many graph theorists, and we also want to reserve some extra space to refining our results from the previous section. In particular, we will pay closer attention to the List and Prelabeling variants of the problem. We prefer to stay in the *labeling* setting because the FPT result holds for general p. The hardness results are new and interesting just for $p = 1$, since it was known that the List $L(p, 1)$-Labeling and $L(p, 1)$-Prelabeling Extension problems for $p \geq 2$ are $\mathcal{NP}$-complete for graphs of treewidth two [22,17]. We thus obtain a complete characterization of the computational complexity for all values of p.

Theorem 10. *The* List $L(1, 1)$-Labeling *problem is $\mathcal{NP}$-complete for graphs of treewidth at most two, as well as for graphs of vertex cover number at most four, even if all lists have at most three elements.*

Proof. We reduce the 3-Satisfiability problem:

Instance: A Boolean formula ϕ in conjunctive normal form.
Question: Does ϕ have a satisfying truth assignment?

This problem is known to be $\mathcal{NP}$-complete even when restricted to formulas where each clause contains two or three literals and every variable occurs in exactly three clauses — once positive and twice negated [21].

Let $x_1, \ldots, x_n$ be Boolean variables, and let $C_1, \ldots, C_m$ be the clauses of ϕ. We start our construction with four vertices $a_0, \ldots, a_3$ with lists of labels $L(a_i) := \{i\}$, for $i = 0, \ldots, 3$.

For each $i \in \{1, \ldots, n\}$, we introduce vertices y_i, u_i, v_i, w_i, s_i, where y_i is adjacent to a_0 and a_1, u_i is adjacent to a_0, v_i is adjacent to a_1, w_i is adjacent to a_0 and a_2, and finally s_i is adjacent to a_1 and a_3.

We define lists of labels of these vertices as follows: $L(y_i) := \{4i, 4i+3\}$, $L(u_i) := \{4i+1, 4i+3\}$, $L(v_i) := \{4i+2, 4i+3\}$, $L(w_i) := \{4i, 4i+1\}$ and $L(s_i) := \{4i, 4i+2\}$.

For each $j \in \{1, \ldots, m\}$, the vertex c_j is added, and for every literal l in the clause C_j one integer is included to $L(c_j)$: if $l = x_i$ then $4i$ is included, if $t = \bar{x}_i$ then $4i+1$ is included if it is the first occurrence of this literal in ϕ, and $4i+2$ is included for the second occurrence.

Denote the obtained graph by G (see Fig. 3). Clearly, $\mathbf{tw}(G) = 2$ and $\mathbf{vc}(G) = 4$. We claim that ϕ can be satisfied if and only if G has a list $L(1, 1)$-labeling.

Suppose that variables $x_1, \ldots, x_n$ have values for which the formula ϕ is satisfied. If $x_i =$ `true` then the vertices y_i, u_i, v_i, w_i and s_i are labeled by $4i$, $4i+3$, $4i+3$, $4i+1$ and $4i+2$. If $x_i =$ `false` then we use for these five vertices labels $4i+3$, $4i+1$, $4i+2$, $4i$ and $4i$.

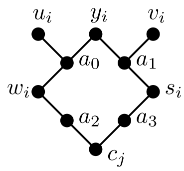

Fig. 3. Construction of G.

Every clause C_j contains positively valued literal l. If l is a variable x_i then the vertex c_j gets label $4i$. Otherwise, i.e. when $l = \bar{x}_i$, then c_j is labeled either by $4i + 1$, if it is the first occurrence of l in ϕ, or by $4i + 2$ for the second occurrence.

Vertices a_i are labeled by the unique possible labels. It is straightforward to verify that we get a valid list $L(1, 1)$-labeling of G.

Assume now that G allows a list $L(1, 1)$-labeling. If y_i is labeled by $4i$ then the variable x_i is set to the value true, and if y_i is labeled by $4i + 3$ then $x_i :=$ false.

Suppose that c_j is labeled by $4i$ for some i (i.e. the clause C_j contains literal x_i). Therefore w_i is labeled by $4i + 1$, u_i is labeled by $4i + 3$ and y_i is labeled by $4i$. It means that $x_i =$ true and the clause C_j is satisfied.

If c_j is labeled by $4i+1$ (i.e. the clause C_j contains literal $\bar{x}_i$ as its first occurrence in ϕ) then w_i is labeled by $4i$, y_i is labeled by $4i + 3$.

Finally, if c_j is labeled by $4i + 2$ (i.e. C_j contains the second occurrence of $\bar{x}_i$) then s_i is labeled by $4i$, and y_i is labeled by $4i + 3$ as well.

In the last two cases we get that $x_i =$ false, hence C_j is satisfied. □

It is easy to obtain the same result for the $L(1, 1)$-Prelabeling Extension problem for graphs of bounded treewidth:

Theorem 11. *The $L(1, 1)$-Prelabeling Extension problem is $\mathcal{NP}$-complete for graphs of treewidth at most two.*

Proof. We reduce the List $L(1, 1)$-Labeling problem for graphs of treewidth two. Let G be a graph with lists of labels $L(v)$. Let $\lambda = \max \bigcup_{v \in V(G)} L(v) + 1$. For each vertex $v \in V(G)$, we construct a star $K_{1,\lambda-|L(v)|}$. The central vertex of this star is prelabeled by λ, and the leaves are prelabeled by labels from $\{0, \ldots, \lambda - 1\} \setminus L(v)$. Then the central vertex is joined with the vertex v by an edge. The vertex v remains unlabeled. Denote the obtained graph by H. Obviously, G has a list $L(1, 1)$-labeling if and only if the prelabeling of G can be extended to an $L(1, 1)$-labeling of span λ. □

Somewhat surprisingly, the complexity of $L(1, 1)$-Prelabeling Extension differs when parameterized by treewidth or vertex cover number. While the $L(1, 1)$-Prelabeling Extension problem is difficult for graphs of bounded treewidth, it becomes tractable when parameterized by the vertex cover number, even for general p.

Theorem 12. *For every p, the $L(p, 1)$-Prelabeling Extension problem is in the class $\mathcal{FPT}$ when parameterized by the vertex cover number.*

Proof. We follow an analogous approach as in the proof of Theorem 9 and use the same notation of $W, k, I, \{I_1, \ldots, I_s\}$, $\{X_0, \ldots, X_t\}$.

As in the proof of Theorem 9, we explore all extensions of the given prelabeling to the set W and also to some vertices of I. The newly labeled vertices of I may be labeled only by labels that are used on W.

In the given partial labeling, let b_i, for $0 \le i \le t$ be the number of distinct labels that are used exclusively on elements of X_i. In particular, b_0 is the difference between $\lambda + 1$ and the number of labels of the partial labeling. Furthermore, let a_i be the number of distinct labels used on X_i (as b_i is defined) but this time restricted only to the set of labels used on W.

We decide whether the partial labeling can be extended to the entire graph G by a system of linear inequalities on variables indexed by two indices. A variable $x_{i,i'}$ is present if $i, i' \in \{0, \ldots, t\}$ and $X_{i'} \subseteq X_i$. The value of $x_{i,i'}$ stands for the number of distinct labels that are used on $X_{i'}$ and that will be used to label vertices of the system X_i (at one vertex from each set $I_j \in X_i$, of course).

$$\begin{cases} \displaystyle\sum_{i:X_{i'} \subseteq X_i} x_{i,i'} = b_{i'}, & \text{for } i' \in \{0, \ldots, t\}, \\ \displaystyle\sum_{i':X_{i'} \subseteq X_i} x_{i,i'} \ge a_i, & \text{for } i \in \{0, \ldots, t\}, \\ \displaystyle\sum_{i:I_j \in X_i} \sum_{i':X_{i'} \subseteq X_i} x_{i,i'} = |I_j|, & \text{for } j \in \{1, \ldots, s\}. \end{cases}$$

The first set of equalities expresses the fact that the numbers of labels that are used on extensions of a system $X_{i'}$ sums up to $b_{i'}$. This set of equations also captures the property that the whole span does not exceed λ.

The second and third sets of (in-)equalities correspond directly to those of Theorem 9, the only difference is that the final number of labels used for X_i is obtained as the sum of all extensions over partial labelings defined on $X_{i'} \subseteq X_i$.

The number of variables is bounded by $O(t^2)$, the number of inequalities by $2 + 2t + s$ so the overall time complexity is $O(kn + 2^{2^{2^{2k+1}}} \log n)$. □

6. Complexity of the Channel Assignment problem

Recall that for the Channel Assignment problem, the span of the assignment – as a part of the input – is measured in binary encoding. Since Channel Assignment is known to be NP-complete for graphs of treewidth 3, the following theorem proves a drop in complexity under parametrization by the vertex cover number. However, it does not settle its FPT status, and we thus leave this question as an open problem.

Theorem 13. *For every k, the* Channel Assignment *problem is solvable in polynomial time for graphs of vertex cover number bounded by k.*

Proof. We will actually show that the minimum span of a feasible labeling can be computed in polynomial time, if the input graph has bounded vertex cover number. Towards this end suppose that G comes equipped with a weight function $w : E(G) \longrightarrow Z^+$ with all weights at most $w_{\max}$. Let $V(G) = W \cup I$ be a partition into a vertex cover W of size k and an independent set I. Our approach is based on the following minimality argument:

Lemma 14. *If G, w allows a channel assignment of span λ, then in every assignment $f : V(G) \longrightarrow \{0, 1, \ldots, \lambda\}$ that minimizes the sum $\sum_{x\in V(G)} f(x)$, every vertex $u \in W$ fulfills one of the following conditions:*

1. *$f(u) = 0$, or*
2. *there is an $x \in I$ such that $f(x) = 0$, $xu \in E(G)$ and $f(u) = w(xu)$, or*
3. *there is a $v \in W$ such that $uv \in E(G)$ and $f(u) = f(v) + w(uv)$, or*
4. *there are $x \in I$, $v \in W$, such that $vx, xu \in E(G)$ and $f(x) = f(v) + w(vx)$ and $f(u) = f(x) + w(xu)$.*

Note that because of the nonnegativity of w, it is always $f(x) \le f(u)$, or $f(v) \le f(u)$, or $f(v) \le f(x) \le f(u)$ in the above listed cases.

Proof. If there exists any channel assignment of span λ, consider such a one that minimizes the sum

$$\sum_{x\in V(G)} f(x).$$

Clearly, $\min_{x\in V(G)} f(x) = 0$ and none of the labels can be decreased by 1. That means that for every $u \in V(G)$ such that $f(u) > 0$, there is a $y \in N(u)$ such that $f(y) = f(u) - w(uy)$. The claim above then follows as a case analysis whether $y \in W$ or $y \in I$ for a particular $u \in W$. □

Suppose G, w allows a channel assignment of span λ, and consider a (hypothetical) assignment $f : V(G) \longrightarrow \{0, 1, \ldots, \lambda\}$ that minimizes the sum $\sum_{x\in V(G)} f(x)$. Construct an auxiliary directed graph $\widetilde{G}$ with vertex set $W \cup X$ for some $X \subset I$ as follows: For every $u \in W$, follow one of the following rules (if more than one are applicable, choose an arbitrary one)

1. if there is an $x \in I$ such that $f(x) = 0$, $xu \in E(G)$ and $f(u) = w(xu)$, then add one such x to X and the arc ux to $E(\widetilde{G})$,
2. if there is a $v \in W$ such that $uv \in E(G)$ and $f(u) = f(v) + w(uv)$, then add one such arc uv to $E(\widetilde{G})$,
3. if there are $x \in I$, $v \in W$, such that $vx, xu \in E(G)$ and $f(x) = f(v) + w(vx)$ and $f(u) = f(x) + w(xu)$, then add one such vertex x to X, and the arcs ux, xv to $E(\widetilde{G})$.

This auxiliary graph is a directed forest, all sinks are labeled 0, and all other vertices have outdegree 1 (if some $u \in W$ had outdegree 0, then reassigning $f'(u) = f(u) - 1$ would yield a valid assignment with a smaller sum of labels). Let us call such a directed graph a *scenario*. Since each vertex of W which is not a sink can be adjacent to $n - k$ sinks from I or can be adjacent to at most $k - 1$ vertices of W or can be connected by directed paths of length two to a vertex in W in at most $(n - k)(k - 1)$ ways, the number of possible scenarios is at most $(k(n - k + 1))^k = O(k^k n^k)$.

For each scenario, we check if it extends to a valid channel assignment and what would be the minimum span of such an extension. For a particular scenario with vertex set $W \cup X$, the labels of the vertices $W \cup X$ are uniquely determined by the scenario. We first check all $O(k^2)$ edges between the vertices of $W \cup X$, and then we attend to the vertices of $I \setminus X$. Let $u_1, u_2, \ldots, u_k$ be an ordering of W determined by the scenario such that $f(u_1) \le f(u_2) \le \cdots \le f(u_k)$. For each vertex $z \in I \setminus X$, we check whether it fits in some interval $[f(u_i)..f(u_{i+1})]$. This can be done in time linear in $(\log w_{\max} + \log n)k$ by checking if $\max_{j\le i}(f(u_j) + w(u_j z)) \le \min_{j\ge i+1}(f(u_j) - w(u_j z))$. If none of these intervals is available for $f(z)$, and neither is the interval $[0..f(u_1)]$, we have to label z by $f(z) = \min_{1\le j\le k}(f(u_j) + w(u_j z))$. Finally we compute the maximum of all labels used to get the span. In this way we compute the minimum possible span of a channel assignment in time $O(k^{k+2} n^{k+1}(\log w_{\max} + \log n))$. □

7. Concluding remarks

7.1 The fixed parameter complexity of the Channel Assignment problem when parameterized by vertex cover number is not solved by the XP membership to a full satisfaction. Is the problem in FPT? Or W[1]-hard? We see this as the main open problem in the area of parametrization of coloring and labeling problems by vertex cover.

7.2 Several of our positive results can be generalized to graphs that can be split to small components by deleting a bounded number of vertices. More formally, for a positive integer l let us define $\mathbf{vc}_l(G)$ as the minimum size of a set W of vertices such that every connected component of $G \setminus W$ has at most l vertices. Hence for every graph G, we have $\mathbf{vc}(G) = \mathbf{vc}_1(G) \geq \mathbf{vc}_2(G) \geq \cdots$ and bounded vertex cover number implies bounded $\mathbf{vc}_l$ for $l \geq 2$ (but not vice versa). Note also that for every graph G and every l, it holds $\mathbf{tw}(G) \leq \mathbf{vc}_l(G) + l - 1$, and hence bounded $\mathbf{vc}_l$ implies bounded treewidth (but not vice versa). Using similar arguments as in our proofs of Theorems 2 and 13, we can prove the following stronger results.

Theorem 15. *The* Precoloring Extension *problem is* $\mathcal{FPT}$ *for graphs G satisfying* $\mathbf{vc}_l(G) \leq k$ *when parameterized by k and l.*

Theorem 16. *For every k and l, the* Channel Assignment *problem is solvable in polynomial time for graphs G satisfying* $\mathbf{vc}_l(G) \leq k$.

7.3 Our FPT algorithms might not attain the best possible time bounds, hence more efficient algorithms might be an interesting subject for further research.

Acknowledgements

The authors thank Henning Fernau for useful discussions leading to considerations presented in the previous paragraph. The second author was supported by the Norwegian Research Council.

References

[1] K. Aardal, R. Weismantel, L.A. Wolsey, Non-standard approaches to integer programming, Discrete Appl. Math. 123 (2002) 5–74.
[2] G. Agnarsson, M.M. Halldórsson, Coloring powers of planar graphs, SIAM J. Discrete Math. 16 (2003) 651–662.
[3] N. Alon, Restricted colorings of graphs, in: Surveys in Combinatorics (Keele), in: London Math. Soc. Lecture Note Ser., vol. 187, Cambridge University Press, Cambridge, 1993, pp. 1–33.
[4] H.L. Bodlaender, Treewidth: Characterizations, applications, and computations, in: F.V. Fomin (Ed.), Graph-Theoretic Concepts in Computer Science, Proceedings WG 2006, in: Lecture Notes in Computer Science, vol. 4271, Springer, 2006, pp. 1–14.
[5] H.L. Bodlaender, F.V. Fomin, Equitable colorings of bounded treewidth graphs, Theoret. Comput. Sci. 349 (2005) 22–30.
[6] H.L. Bodlaender, A.M.C.A. Koster, Combinatorial optimization on graphs of bounded treewidth, Comput. J. 51 (2008) 255–269.
[7] J.F. Buss, J. Goldsmith, Nondeterminism within P, SIAM J. Comput. 22 (1993) 560–572.
[8] T. Calamoneri, The $l(h, k)$-labelling problem: a survey and annotated bibliography, Comput. J. 49 (2006) 585–608.
[9] G.J. Chang, D. Kuo, The $L(2, 1)$-labeling problem on graphs, SIAM J. Discrete Math. 9 (1996) 309–316.
[10] M. Dom, D. Lokshtanov, S. Saurabh, Y. Villanger, Capacitated domination and covering: A parameterized perspective, in: M. Grohe, R. Niedermeier (Eds.), Parameterized and Exact Computation, Proceedings of IWPEC 2008, in: Lecture Notes in Computer Science, vol. 5018, Springer, 2008, pp. 78–90.
[11] R.G. Downey, M.R. Fellows, Parameterized complexity, in: Monographs in Computer Science, Springer-Verlag, New York, 1999.
[12] R.G. Downey, M.R. Fellows, Fixed-parameter tractability and completeness. II. On completeness for $W[1]$, Theoret. Comput. Sci. 141 (1995) 109–131.
[13] Z. Dvořák, D. Král, P. Nejedlý, R. Škrekovski, Coloring squares of planar graphs with girth six, European J. Combin. 29 (2008) 838–849.
[14] Z. Dvořák, R. Škrekovski, M. Tancer, List-coloring squares of sparse subcubic graphs, SIAM J. Discrete Math. 22 (2008) 139–159.
[15] M. Fellows, D. Lokshtanov, N. Misra, F.A. Rosamond, S. Saurabh, Graph layout problems parameterized by vertex cover, in: S.-H. Hong, H. Nagamochi, T. Fukunaga (Eds.), Algorithms and Computation, Proceedings of ISAAC 2008, in: Lecture Notes in Computer Science, vol. 5369, Springer, 2008, pp. 294–305.
[16] M.R. Fellows, F.V. Fomin, D. Lokshtanov, F.A. Rosamond, S. Saurabh, S. Szeider, C. Thomassen, On the complexity of some colorful problems parameterized by treewidth, in: A.W.M. Dress, Y. Xu, B. Zhu (Eds.), Combinatorial Optimization and Applications, Proceedings of COCOA 2007, in: Lecture Notes in Computer Science, vol. 4616, Springer, 2007, pp. 366–377.
[17] J. Fiala, P.A. Golovach, J. Kratochvíl, Distance constrained labelings of graphs of bounded treewidth, in: L. Caires, G.F. Italiano, L. Monteiro, C. Palamidessi, M. Yung (Eds.), Automata, Languages and Programming, Proceedings of ICALP 2005, in: Lecture Notes in Computer Science, vol. 3580, Springer, 2005, pp. 360–372.
[18] J. Fiala, P.A. Golovach, J. Kratochvíl, Computational complexity of the distance constrained labeling problem for trees (extended abstract), in: L. Aceto, I. Damgård, L.A. Goldberg, M.M. Halldórsson, A. Ingólfsdóttir, I. Walukiewicz (Eds.), Automata, Languages and Programming, Proceedings of ICALP 2008, in: Lecture Notes in Computer Science, vol. 5125, Springer, 2008, pp. 294–305.
[19] J. Fiala, P.A. Golovach, J. Kratochvíl, Parameterized complexity of coloring problems: treewidth versus vertex cover, in: J. Chen, S.B. Cooper (Eds.), Theory and Applications of Models of Computation, Proceedings of TAMC 2009, in: Lecture Notes in Computer Science, vol. 5532, Springer, 2009, pp. 221–230.
[20] A. Frank, É Tardos, An application of simultaneous Diophantine approximation in combinatorial optimization, Combinatorica 7 (1987) 49–65.
[21] M.R. Garey, D.S. Johnson, Computers and intractability, in: A Guide to the Theory of NP-completeness, in: A Series of Books in the Mathematical Sciences, W. H. Freeman and Co, San Francisco, Calif, 1979.
[22] P.A. Golovach, Systems of pairs of q-distant representatives, and graph colorings, Zap. Nauchn. Sem. S.-Peterburg. Otdel. Mat. Inst. Steklov. (POMI) 293 (2002) 5–25. 181.
[23] T. Hasunuma, T. Ishii, H. Ono, Y. Uno, A linear time algorithm for L(2, 1)-labeling of trees, in: A. Fiat, P. Sanders (Eds.), Algorithms — ESA 2009, Proceedings of ESA 2009, in: Lecture Notes in Computer Science, vol. 5757, Springer, 2009, pp. 35–46.
[24] R. Kannan, Minkowski's convex body theorem and integer programming, Math. Oper. Res. 12 (1987) 415–440.
[25] H.W. Lenstra Jr., Integer programming with a fixed number of variables, Math. Oper. Res. 8 (1983) 538–548.
[26] C. McDiarmid, B. Reed, Channel assignment on graphs of bounded treewidth, Discrete Math. 273 (2003) 183–192.
[27] Z. Tuza, Graph colorings with local constraints—a survey, Discuss. Math. Graph Theory 17 (1997) 161–228.
[28] R.K. Yeh, A survey on labeling graphs with a condition at distance two, Discrete Math. 306 (2006) 1217–1231.
[29] X. Zhou, Y. Kanari, T. Nishizeki, Generalized vertex-coloring of partial k-trees, IEICE Trans. Fundamentals Electron. Commun. Comput. Sci. E83-A (2000) 671–678.