Finding b-Colorings Using Feedback Edges

Jakub Balabán
Faculty of Informatics, Masaryk University, Brno, Czechia

Abstract

A b-coloring of a graph is a proper vertex coloring such that each color class contains a vertex that sees all other colors in its neighborhood. The b-COLORING problem, in which the task is to decide whether a graph admits a b-coloring with k colors, is NP-complete in general but polytime solvable on trees. Moreover, it is known that b-COLORING is in XP but W[t]-hard for all $t \in \mathbb{N}$ when parameterized by tree-width. In fact, only very few parameters, such as the vertex cover number, were known to admit an FPT algorithm for b-COLORING.

In this paper, we consider a more restrictive parameter measuring similarity to trees than tree-width, namely the feedback edge number, and show that b-COLORING is fixed-parameter tractable under this parameterization. Our algorithm combines standard techniques used in parameterized algorithmics with the problem-specific ideas used in the polytime algorithm for trees. In addition, we present an FPT algorithm for b-COLORING parameterized by distance to co-cluster, which is a parameter measuring similarity to complete multipartite graphs. Finally, we make several observations based on known results, including that b-COLORING is W[1]-hard when parameterized by tree-depth.

2012 ACM Subject Classification Theory of computation → Parameterized complexity and exact algorithms

Keywords and phrases b-coloring, fixed-parameter algorithms, feedback edge number, distance to co-cluster

Digital Object Identifier 10.4230/LIPIcs.MFCS.2026.36

Related Version *Full Version*: `https://arxiv.org/abs/2512.14390` [2]

Funding Supported by the project 26-21334S of the Czech Science Foundation.
Jakub Balabán: Brno Ph.D. Talent Scholarship Holder – Funded by the Brno City Municipality

1 Introduction

The CHROMATIC NUMBER problem, in which the task is to decide whether a graph admits a proper coloring with k colors, is one of the famous 21 NP-complete problems listed by Karp [28]. One simple heuristic for this problem works as follows. We start with an arbitrary proper coloring and, in each iteration, we try to decrease the number of colors by one. We search for a color c such that for each vertex u colored with c, there is a color $c_u \neq c$ that is not present in the neighborhood of u. If we can find such a color c, then we assign the color c_u to each vertex u currently colored with c, and the obtained coloring remains proper. Otherwise, the heuristic terminates.

Irving and Manlove [22] initiated the theoretical study of this heuristic in 1999 and defined a proper coloring to be a *b-coloring* if each color class contains a vertex, a so-called *b-vertex*, that sees all other colors in its neighborhood. It is easy to see that b-colorings are exactly those colorings that cannot be improved by the heuristic. In addition, they defined the *b-chromatic number* of a graph G, denoted $\chi_b(G)$, as the largest integer k such that G admits a b-coloring with k colors; observe that the b-chromatic number describes the worst-case behavior of the heuristic. These concepts received a lot of attention since their inception [31, 13, 30, 25, 21, 9, 32, 23], see the following survey [26].

51st International Symposium on Mathematical Foundations of Computer Science (MFCS 2026).
Editors: Michal Koucký and Daniela Petrişan; Article No. 36; pp. 36:1–36:18
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

There are two natural computational problems related to b-colorings: the b-COLORING problem, which asks for a b-coloring with exactly k colors, and the b-CHROMATIC NUMBER problem, which asks whether $\chi_b(G) \geq k$. Notice that a polytime algorithm for b-COLORING can be used to solve b-CHROMATIC NUMBER in polynomial time. However, the converse does not hold in general: if $\chi(G) < k < \chi_b(G)$, where $\chi(G)$ is the chromatic number of G, then a b-coloring with k colors does not have to exist[1]. Hence, b-CHROMATIC NUMBER is, in some sense, an "easier" problem than b-COLORING.

In their seminal paper, Irving and Manlove [22] proved that b-CHROMATIC NUMBER is NP-complete on general graphs and solvable in polynomial time on trees. The NP-completeness was subsequently extended to bipartite graphs [31], chordal graphs [21], co-bipartite graphs [7], and line graphs [9]. On the positive side, polytime algorithms for b-CHROMATIC NUMBER have been designed for several classes with few P_4s [6, 38, 12] and for various tree-like classes [35], such as tree-cographs [7], cacti [11], and graphs of large girth [10]. However, almost all of these algorithms were unified by an XP algorithm for b-COLORING parameterized by clique-width, which was recently developed by Jaffke, Lima, and Lokshtanov [23]. Indeed, XP means that the algorithm runs in polynomial time on any class of bounded clique-width, and all of the above-mentioned classes (except for the graphs of large girth) have bounded clique-width [23].

Once an XP algorithm is known, the natural question is whether an FPT algorithm is possible [14]. Unfortunately, Jaffke, Lima, Sharma [24] proved that b-COLORING is XNLP-complete when parameterized by path-width, which implies that it is W[t]-hard for each $t \in \mathbb{N}$ under this parameterization. Hence, it is very unlikely that an FPT algorithm parameterized by path-width (or tree-width) exists. We emphasize that this hardness result relies on the fact that the number of colors is *not* part of the parameterization, i.e., it may be as large as the number of vertices. In fact, if we parameterize by tree-width plus the number of colors, then an FPT algorithm is possible [23, 1], see Proposition 3.

A trivial upper-bound on the b-chromatic number, called the *m-degree* [22], is defined as the largest integer k such that there are at least k vertices of degree at least $k-1$. Since the vertex cover number upper bounds both the m-degree and tree-width, there is an FPT algorithm for b-COLORING parameterized by the vertex cover number [23]. The only other structural parameters that were known to allow for an FPT algorithm for b-COLORING are the twin cover number and neighborhood diversity [24], see Figure 1.

Our contribution. The main contribution of this paper is an FPT algorithm for b-COLORING parameterized by the feedback edge number (Section 4), which is the smallest number of edges that need to be deleted to obtain an acyclic graph. This parameter has been used in several FPT algorithms [3, 37, 4, 19]. Admittedly, our algorithm is not useful for computing proper colorings with few colors, as there is a simple FPT algorithm for CHROMATIC NUMBER parameterized by the feedback edge number (thus, the aforementioned heuristic is unnecessary). However, we believe that our new algorithm is interesting for the study of b-colorings in their own right, as it is the first FPT algorithm for b-COLORING that builds on the polytime algorithm for b-CHROMATIC NUMBER on trees [22]. In particular, we generalize the notion of a *pivot*, which is, roughly speaking, a vertex that cannot be colored because all high-degree vertices (potential b-vertices) are close to it. We remark that the concept of a pivot was used also by the polytime algorithms for b-CHROMATIC NUMBER on various tree-like classes [7, 11, 10]. Hence, our algorithm can be viewed as a bridge between these polytime algorithms and the more recent FPT algorithms based on structural parameters [23, 24].

[1] For example, the bipartite graph with vertex set $\{u_i, v_i \mid i \in [4]\}$ and edge set $\{u_i v_j \mid i \neq j\}$ admits a b-coloring with two or four colors but not with three colors [22].

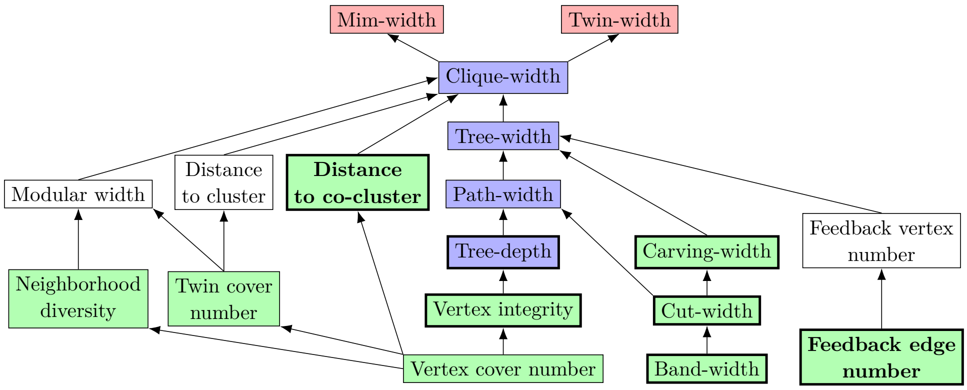

Figure 1 Parameterized complexity of b-COLORING under various structural parameters. A directed path from a parameter α to a parameter β indicates that $\beta \leq f(\alpha)$ for some computable function f. Green stands for FPT, blue is W[1]-hard and XP, red is paraNP-complete, and white means that it is unknown whether the problem is FPT or W[1]-hard under given parameterization. Our new results are marked with thick boundary, and the two parameters for which our FPT algorithms are non-trivial are written in bold text.

As a second contribution, we present an FPT algorithm for b-COLORING parameterized by distance to co-cluster (Section 3), which is the smallest number of vertices that need to be deleted to obtain a complete multipartite graph. This parameter has also been used in several FPT algorithms [29, 33, 18]. Finally, in Section 2, we observe that there are well-studied parameters that functionally upper bound tree-width and the m-degree, which yields new FPT algorithms for b-COLORING. These parameters are vertex integrity [20], band-width [15], cut-width [15], and carving-width [34], see Figure 1. We complement these algorithms with a hardness result implied by known results [24, 27]: b-COLORING is W[1]-hard when parameterized by tree-depth.

Future research directions. Even though the feedback edge number is a restrictive parameter, generalizing the algorithm for b-CHROMATIC NUMBER on trees turned out surprisingly non-trivial. The natural open question is whether our algorithm can be further generalized into an FPT algorithm parameterized by the feedback *vertex* number (which allows deleting vertices instead of edges).

Another natural question arising from our results is whether b-COLORING is FPT when parameterized by the distance to *cluster* (i.e., vertex-deletion distance to a disjoint union of cliques). We remark that a closely related and more restrictive parameter, namely the twin-cover number, was shown to be sufficient for an FPT algorithm [24]. This parameter provides an additional condition, which guarantees that the vertices contained in the same clique have the same neighbors among the vertices that need to be deleted. Is this additional condition necessary or is b-COLORING parameterized by the distance to cluster in FPT?

2 Preliminaries

For integers i and j, we let $[i, j] := \{a \in \mathbb{Z} \mid i \leq a \leq j\}$ and $[i] := [1, i]$. Let $f\colon A \to B$ be a function. The *range* of f, denoted range(f), is the set $\{b \in B \mid \exists a \in A\colon f(a) = b\}$. For $A' \subseteq A$, we denote the function $\{(a, b) \in f \mid a \in A'\}$ by $f \restriction A'$. For $a \in A$ and $b \in B$, we define $f[a \mapsto b]$ to be the function $f \setminus \{(a, f(a))\} \cup \{(a, b)\}$. Instead of $f[a_1 \mapsto b_1][a_2 \mapsto b_2]$, we write $f[a_1 \mapsto b_1, a_2 \mapsto b_2]$.

Let G be a simple undirected graph. We assume familiarity with basic concepts in graph theory [16] and parameterized algorithmics [14]. Given a set of vertices $U \subseteq V(G)$, we will use $\overline{U}$ to denote the set $V(G) \setminus U$, $G[U]$ to denote the graph induced on U, and $G - U$ to denote the graph $G[\overline{U}]$. Similarly, for an edge set $F \subseteq E(G)$, $G - F$ denotes the graph obtained from G by removing the edges in F. The *length* of a path or a cycle is the number of edges it contains. The *distance* between two vertices u and v, denoted $\mathrm{dist}_G(u, v)$, is the length of the shortest path between them (or ∞ if no such path exists). If $v \in V(G)$ and $U \subseteq V(G)$ is non-empty, then $\mathrm{dist}_G(v, U) = \min\{\mathrm{dist}_G(u, v) \mid u \in U\}$. For $U \subseteq V(G)$ and $v \in V(G)$, a v-U path is a path P in G with endpoints v and u such that $V(P) \cap U = \{u\}$.

For $u \in V(G)$, we define $N_G(u) := \{v \in V(G) \mid uv \in E(G)\}$ and $N_G[u] := N_G(u) \cup \{u\}$. Similarly, for $S \subseteq V(G)$, we define $N_G(S) = \bigcup_{u\in S} N_G(u)$ and $N_G[S] = N_G(S) \cup S$. Moreover, we define $N^S_G(u) = N_G(u) \cap S$. Vertices u and v are *twins* if $N(u) \setminus \{v\} = N(v) \setminus \{u\}$. When G is clear from the context, we will omit the subscript and write, e.g., $\mathrm{dist}(u, v)$ or $N(u)$.

If $\sim$ is an equivalence relation on $V(G)$, then $G/{\sim}$ is the graph such that $V(G/{\sim})$ are the equivalence classes of $\sim$, and $CD \in E(G/{\sim})$ if and only if there are vertices $u, v \in V(G)$ such that $u \in C$, $v \in D$, and $uv \in E(G)$.

Structural parameters. Given a graph class $\mathcal{G}$ and a graph G, we say that $S \subseteq V(G)$ is a *$\mathcal{G}$-modulator of G* if $G - S \in \mathcal{G}$. An integer p is the *distance to $\mathcal{G}$ of G* if p is the size of a minimum $\mathcal{G}$-modulator of G. A *cluster graph* is a disjoint union of cliques and a *co-cluster graph* is a complement of a cluster graph, i.e., a complete multipartite graph. We remark that a minimum cluster-modulator of graph can be computed in FPT time.

▶ **Theorem 1** ([8]). *Given an n-vertex graph G with distance to cluster p, a minimum cluster-modulator $S \subseteq V(G)$ can be computed in time $\mathcal{O}(1.92^p \cdot n^2)$.*

An edge set $F \subseteq E(G)$ is called a *feedback edge set* if $G - F$ is acyclic, and the *feedback edge number* of G is the size of a minimum feedback edge set in G. A *dangling path* in G is a path of vertices which all have degree 2 in G, and a *dangling tree* in G is an induced subtree in G which becomes a connected component after removing at most one edge from G. The following lemma states that a graph of bounded feedback edge number can be decomposed into dangling trees, dangling paths, and a bounded-size subgraph. This lemma was implicitly used in [3].

▶ **Lemma 2.** *If G is a connected graph with feedback edge number p that contains no dangling trees, then there is a set $S \subseteq V(G)$ such that $|S| \leq 4p$, and if $\mathcal{P}$ is the set of connected components of $G - S$, then $|\mathcal{P}| \leq 4p$ and each $P \in \mathcal{P}$ is a dangling path in G. Moreover, S can be found in polynomial time.*

Proof sketch. Fix a minimum feedback edge set F. Define S to be the endpoints of F plus vertices of degree at least 3 in $G - F$. ◀

Colorings. A (partial) *k-coloring of G* is a (partial) function $\chi \colon V(G) \to [k]$. We say that χ is *proper* if $\chi(u) \neq \chi(v)$ for each $uv \in E(G)$. We say that $u \in V(G)$ is a *b-vertex in χ* if $\chi(N[u]) = [k]$, and that χ is a *k-b-coloring* if it is proper and for each color $c \in [k]$, there is a b-vertex u in χ such that $\chi(u) = c$. A coloring is a *b-coloring* if it is a k-b-coloring for some integer k. The decision problem b-COLORING receives as input a graph G and an integer k and asks whether there is a k-b-coloring of G.

It can be easily shown that b-COLORING is FPT when parameterized by tree-width and the number of colors.

▶ **Proposition 3** ([23, 1]). *The b-COLORING problem can be solved in time $2^{\mathcal{O}(w \cdot k)} \cdot |V(G)|$, where G is the input graph, w is its tree-width, and k is the number of colors.*

Proposition 3 was used in [23] to show that b-COLORING is FPT when parameterized by the vertex cover number. Now we use this proposition to obtain two other FPT algorithms. The *vertex integrity* of a graph G is the smallest integer p such that there is a set $S \subsetneq V(G)$ such that for each connected component H of $G - S$, we have $|S \cup V(H)| \leq p$; the relationship of vertex integrity to other parameters is depicted in Figure 1.

▶ **Proposition 4.** *The b-COLORING problem can be solved in time $2^{\mathcal{O}(p^2)} \cdot n$, where n is the number of vertices in the input graph and p is its vertex integrity.*

Proof. Let (G, k) be the input instance, where G is a graph with n vertices and vertex integrity p. First, we use the algorithm from [17] to compute a set $S \subsetneq V(G)$ certifying the vertex integrity of G in time $\mathcal{O}(p^{p+1} \cdot n)$. Using S, it is easy to compute a path decomposition of width at most p. If $k \leq p$, then we may use Proposition 3 to solve the problem. Otherwise, we answer that G admits no k-b-coloring. Indeed, in such a coloring, there would have to be a b-vertex $u \in V(G) \setminus S$ of degree at least $k - 1 \geq p$, which is impossible. ◀

The second parameter we consider is *carving-width* [34]. This width parameter is certified by a so-called *carving*: a subcubic tree T and a bijection between the leaves of T and the vertices of G (we will not need the full definition). Note that the following proposition implies that b-COLORING is FPT when parameterized by one of two other well-known parameters [15], namely *cut-width* and *band-width*; see Figure 1.

▶ **Proposition 5.** *The b-COLORING problem can be solved in time $f(w) \cdot n$, where n is the number of vertices in the input graph, w is its carving-width, and f is a computable function.*

Proof. Let (G, k) be the input instance, where G is a graph with n vertices and carving-width w. First, we use the algorithm from [36] to compute a carving C certifying the carving-width of G in time $f_0(w) \cdot n$ for some computable function f_0. Using C, we can construct a tree decomposition of width at most $3w$ [36]. If $k \leq w + 1$, then we may use Proposition 3 to solve the problem. Otherwise, we answer that G admits no k-b-coloring. Indeed, in such a coloring, there would have to be a b-vertex of degree at least $w + 1$, and there are no such vertices in graphs of carving-width w [5]. ◀

In [24], it was proven that b-COLORING is XNLP-complete when parameterized by path-width. They obtained this result by reducing from CIRCULATING ORIENTATION parameterized by path-width. Since this problem is W[1]-hard when parameterized by tree-depth [27, Theorem 6.1.1.] and it can be easily observed that the reduction to b-COLORING by [24] preserves not only bounded path-width but also bounded tree-depth, we obtain the following result.

▶ **Proposition 6.** *The b-COLORING problem is W[1]-hard when parameterized by the tree-depth of the input graph.*

3 Distance to Co-Cluster

In this section, we prove the following theorem.

▶ **Theorem 7.** *The b-COLORING problem can be solved in time $2^{2^{\mathcal{O}(p)}} \cdot n^{\mathcal{O}(1)}$, where p is the distance to co-cluster of the input n-vertex graph.*

Let us fix an instance of the b-COLORING problem (G, k). Let $S \subseteq V(G)$ be a minimum co-cluster-modulator of G, let $p = \min(|S|, k)$, and let $\mathcal{U}$ be the set containing all maximal independent sets of $G - S$. The main idea of our algorithm is to partition $\mathcal{U}$ into a bounded number of types based on the connection to S, and then to carefully argue that two sets of the same type are "interchangeable" w.r.t. any b-coloring (see Lemma 13).

First, we define the types. If many vertices have the same neighborhood in S, then we do not need to record the precise number of such vertices, which makes the number of types bounded by a function of p.

▶ **Definition 8.** *A set $U \in \mathcal{U}$ has* type $t\colon 2^S \to [0, p+1]$ *if for each $A \subseteq S$, we have $t(A) = \min(p+1, |\{u \in U \mid N^S(u) = A\}|)$. Let T be the set of all types.*

Now we define *signatures*, which will be used to describe each proper k-coloring ψ of G in terms of the colors used on S. The first component of a signature, χ, will be the restriction of ψ to S. By permuting the colors, we may assume that $\mathrm{range}(\chi) \subseteq [p]$. The other parts of a signature describe how colors in $[p]$ behave in $G - S$ according to ψ. Let $U_1, \ldots, U_q \in \mathcal{U}$ be the sets such that some color in $[p]$ is used on some set U_i; the function τ describes the types of these sets. Since ψ is proper, each color in $[p]$ is used on at most one such set: the function λ describes which set it is (if any). Finally, the function ξ describes, for each color $c \in [p]$, the connections to S of vertices in $G - S$ colored with c.

▶ **Definition 9.** *A* signature *is a tuple $\sigma = (\chi, q, \tau, \lambda, \xi)$, where:*
- *$\chi\colon S \to [p]$ is a proper coloring of $G[S]$;*
- *$q \in [0, p]$ is an integer;*
- *$\tau\colon [q] \to T$, $\lambda\colon [p] \to [0, q]$, and $\xi\colon [p] \to 2^{2^S}$ are functions.*

A signature is required to have the following properties.
1. *It holds that $[q] \subseteq \mathrm{range}(\lambda)$.*
2. *For each $t \in T$, $|\tau^{-1}(t)|$ is at most the number of sets of type t in $\mathcal{U}$.*
3. *For each $i \in [q]$ and $A \subseteq S$, we have $|\{c \in [p] : \lambda(c) = i \wedge A \in \xi(c)\}| \leq \tau(i)(A)$.*

We remark that the last two properties required by Definition 9 ensure that the signature is "realizable" by some coloring: Property 2 states that there are enough sets of each type, and Property 3 states that there are enough vertices for each neighborhood A in S. Now we formally define the correspondence between signatures and colorings of G.

▶ **Definition 10.** *Let $\psi\colon V(G) \to [k]$ be a partial coloring and $\sigma = (\chi, q, \tau, \lambda, \xi)$ be a signature. We say that ψ* has signature σ *(or that it is a σ*-coloring*) if $\chi = \psi \upharpoonright S$ and:*
1. *there is a set $\mathcal{U}_\psi = \{U_1, \ldots, U_q\} \subseteq \mathcal{U}$ such that $\psi^{-1}([p]) \subseteq S \cup \bigcup \mathcal{U}_\psi$;*
2. *for each $i \in [q]$, $\tau(i)$ is the set-type of U_i;*
3. *for each $c \in [p]$, if $\lambda(c) = 0$, then $\psi^{-1}(c) \subseteq S$, and if $\lambda(c) = i \neq 0$, then $c \in \psi(U_i)$;*
4. *for each $c \in [p]$, if $\lambda(c) = i \neq 0$, then $\xi(c) = \{A \subseteq S \mid \exists u \in U_i : \psi(u) = c \wedge N^S(u) = A\}$;*

We say that ψ is a *minimal σ-coloring* if it is a σ-coloring, $\psi(V(G)) \subseteq [p]$, and for each $c \in [p]$ and $A \in \xi(c)$, there is a *unique* vertex $u \in U_{\lambda(c)}$ such that $\psi(u) = c$ and $N^S(u) = A$. It can be easily observed that minimal σ-colorings are exactly inclusion-wise minimal colorings that have signature σ. Next, we prove that a minimal σ-coloring can be efficiently computed.

▶ **Lemma 11.** *For a signature $\sigma = (\chi, q, \tau, \lambda, \xi)$, a minimal σ-coloring ψ exists and can be computed in polynomial time. Moreover, the number of vertices colored by ψ is $2^{\mathcal{O}(p)}$.*

Proof sketch. We define $\psi \upharpoonright S = \chi$. For each $i \in [q]$, we find a set $U_i \in \mathcal{U}$ of type $\tau(i)$ (different set for each i). For each $c \in [p]$ such that $\lambda(c) = i \neq 0$ and each $A \in \xi(c)$, we assign c to a single vertex $u \in U_i$ such that $N^S(u) = A$. ◀

The following lemma states the "converse" of Lemma 11: each coloring has a signature.

▶ **Lemma 12.** *If ψ is a k-b-coloring of G, then there is a signature $\sigma = (\chi, q, \tau, \lambda, \xi)$ such that ψ is a σ-coloring (up to permutation of colors in ψ).*

Proof sketch. All components of σ can be easily defined based on ψ. ◀

The following lemma is the technical core of this section. It says that any minimal σ-coloring can be extended to a b-coloring (if at least one b-coloring with signature σ exists).

▶ **Lemma 13.** *If $\sigma = (\chi_S, q, \tau, \lambda, \xi)$ is a signature, $\psi\colon V(G) \to [k]$ is a b-coloring of G with signature σ, and $\chi\colon V(G) \to [p]$ is a minimal σ-coloring, then there is a b-coloring $\chi'\colon V(G) \to [k]$ of G such that $\chi \subseteq \chi'$.*

Proof sketch. Let $\mathcal{U}_\chi = \{U_1, \dots, U_q\}$ and $\mathcal{U}_\psi = \{V_1, \dots, V_q\}$ be the sets as per Definition 10. For $u, v \in V(G)$, we say that $u \sim_0 v$ if $u, v \in W$ for some $W \in \mathcal{U}_\chi \cup \mathcal{U}_\psi$, u and v are twins, and $\psi(u) = \psi(v)$. Let $\sim$ be the equivalence relation on $V(G)$ generated by $\sim_0$, let $H = G/{\sim}$, and let $\eta\colon G \to H$ be the natural homomorphism, i.e., $\eta(u) = [u]_\sim$. It can be seen that there is a k-b-coloring ψ_H of H such that $\psi = \psi_H \circ \eta$. Now we carefully define *another* homomorphism $h\colon G \to H$ and then we set $\chi' = \psi_H \circ h$, see Figure 2. ◀

$$\begin{tikzcd}
& K_k & K_p \arrow[l, hook'] & \\
G \arrow[ur, "\psi"] \arrow[r, "\eta"] & H \arrow[u, "\psi_H"'] & G \arrow[l, "h"] \arrow[ul, "\chi'"'] & G_\chi \arrow[l, hook'] \arrow[ul, "\chi"'] \\
& & G[S] \arrow[ull, hook'] \arrow[ur, hook] &
\end{tikzcd}$$

Figure 2 A commutative diagram depicting the proof of Lemma 13. The arrows are graph homomorphisms: colorings are viewed as homomorphisms into cliques. G_χ is the subgraph of G induced by the vertices on which χ is defined. The unnamed arrows are inclusion maps.

We will need one more definition.

▶ **Definition 14.** *Let $\sigma = (\chi_S, q, \tau, \lambda, \xi)$ be a signature and $\chi\colon V(G) \to [p]$ be a minimal σ-coloring. We say that $B \subseteq V(G)$ is a χ-*candidate subset* if $|B| = p$, $\chi(B) = [p]$, and for each $u \in B$, it holds that $[p] \subseteq \chi(N[u])$. Let us fix a χ-candidate subset B. We say that $U \in \mathcal{U}$ is a (χ, B)-*candidate* if each $u \in B$ has a neighbor $v \in U$ that is uncolored by χ, and there is a vertex $u \in U$ that is uncolored by χ such that $[p] \subseteq \chi(N(u))$. We say that U is χ-*flexible* if each $u \in U$ uncolored by χ has a twin $v \in U$ that is colored by χ.*

We have defined two properties of sets in $\mathcal{U}$: (χ, B)-*candidates* are those that may contain a b-vertex for some color in $[p+1, k]$, and χ-*flexible* are those that *do not have to* contain such a b-vertex. If some set will have neither of these properties, then χ cannot be extended into a b-coloring of G such that B is a set containing a b-vertex for each color in $[p]$. Another issue occurs when there are too many or too few (χ, B)-candidates that are not χ-flexible: the reason is that we need to assign each color in $[p+1, k]$ to exactly one (χ, B)-candidate. The following lemma precisely states the sufficient and necessary conditions for a b-coloring extending χ to exist.

▶ **Lemma 15.** *Let $\sigma = (\chi_S, q, \tau, \lambda, \xi)$ be a signature and $\chi\colon V(G) \to [p]$ be a minimal σ-coloring. There is a b-coloring $\psi\colon V(G) \to [k]$ of G with signature σ such that $\chi \subseteq \psi$ if and only if χ is proper and there is a χ-candidate subset B and a set $\mathcal{C} \subseteq \mathcal{U}$ such that $|\mathcal{C}| = k - p$, all sets in $\mathcal{C}$ are (χ, B)-candidates, and all sets in $\mathcal{U} \setminus \mathcal{C}$ are χ-flexible. Moreover, given B and $\mathcal{C}$ satisfying these properties, the b-coloring ψ can be computed in polynomial time.*

Proof sketch. First, assume that ψ exists. We define B to be a set containing a b-vertex u for each color $c \in [p]$ in ψ (we must ensure that $\chi(u)$ is defined). Consequently, we define $\mathcal{C}$ to be the minimal subset of $\mathcal{U}$ such that $\psi^{-1}([p+1, k]) \subseteq \bigcup \mathcal{C}$.

Conversely, given B and $\mathcal{C}$, we define ψ so that for each $U \in \mathcal{C}$, all vertices from U are colored with some color $c \in [p+1, k]$ (different color for each U). Since U is a (χ, B)-candidate, it contains a b-vertex for c in ψ, and each vertex of B gains a neighbor colored with c. Moreover, sets in $\mathcal{U} \setminus \mathcal{C}$ can be colored with colors from $[p]$ since they are χ-flexible. ◀

Using Lemmas 13 and 15, we are able to finish the proof of Theorem 7.

Proof sketch of Theorem 7. First, we compute a minimum co-cluster-modulator S of the input graph G using Theorem 1. Second, we try all signatures σ, and we find a minimal σ-coloring χ of G using Lemma 11. If χ is not proper, we discard it and continue with the next signature. Otherwise, we try all χ-candidate subsets B, and we compute for each set $U \in \mathcal{U}$ whether it is χ-flexible or a (χ, B)-candidate. If there is a set $U \in \mathcal{U}$ that is neither χ-flexible nor a (χ, B)-candidate, we reject B. Otherwise, we compute the sets $\mathcal{C}_0 = \{U \in \mathcal{U} \mid U \text{ is a } (\chi, B)\text{-candidate but not } \chi\text{-flexible}\}$ and $\mathcal{C}_1 = \{U \in \mathcal{U} \mid U \text{ is a } \chi\text{-flexible } (\chi, B)\text{-candidate}\}$. If $|\mathcal{C}_0| > k - p$ or $|\mathcal{C}_0| + |\mathcal{C}_1| < k - p$, we reject B. Otherwise, we can find a set $\mathcal{C}$ such that $|\mathcal{C}| = k - p$ and $\mathcal{C}_0 \subseteq \mathcal{C} \subseteq \mathcal{C}_0 \cup \mathcal{C}_1$, which allows us to compute a b-coloring of G by Lemma 15. If we do not find a b-coloring for any σ and B, we reject (G, k).

The correctness of the algorithm follows from the lemmas proven in this section. Regarding the running time, observe that there are $2^{2^{\mathcal{O}(p)}}$ signatures and χ-candidate subsets B. For the latter, this holds because the number of vertices colored by a minimal σ-coloring is $2^{\mathcal{O}(p)}$ by Lemma 11 (and each vertex of B is colored by χ). ◀

4 Feedback Edge Number

In this section, we prove the following theorem.

▶ **Theorem 16.** *The b-Coloring problem can be solved in time $2^{\mathcal{O}(p^2)} \cdot n^{\mathcal{O}(1)}$, where p is the feedback edge number of the input n-vertex graph.*

A key notion used in the proof of Theorem 16 is a *pivot*, which originates in the polytime algorithm for b-Chromatic Number on trees [22]. In order to smoothly introduce pivots to the reader, we now briefly describe the original **algorithm for trees**.

Let T be a tree and let k be the m-degree of T; recall that k is an upper bound on the b-chromatic number of T. Informally, this upper bound can be achieved "almost always": the only exception are so-called *pivoted* trees, which can be recognized in linear time [22]. Let us call vertices of degree at least $k-1$ *candidates*; notice that candidates are those vertices that may serve as b-vertices in a k-coloring of T. We say that T is *pivoted* if there are exactly k candidates and there is a vertex u, a so-called *pivot*, such that:

1. u is not a candidate;
2. each candidate is adjacent to u or to another candidate that is adjacent to u;
3. each candidate that is adjacent to u and to another candidate has degree $k-1$.

It is not hard to see that a pivoted tree does not admit a k-b-coloring: each candidate would have to be a b-vertex and a vertex of degree $k-1$ can be a b-vertex only if it sees no color twice, which means that no color may be assigned to the pivot u. It was shown that $\chi_b(T) = k-1$ if T is pivoted, and $\chi_b(T) = k$ otherwise [22].

4.1 Overview of the proof of Theorem 16

Let G be a graph with feedback edge number p. Our goal is to decide whether G admits a b-coloring with k colors. If χ is a partial coloring of G, then a χ-*candidate* is a vertex that may become a b-vertex in a k-coloring extending χ (for example, if χ colors no vertices, then χ-candidates are vertices of degree at least $k-1$). The first step is to compute a set $S \subseteq V(G)$ of size $\mathcal{O}(p)$ such that all cycles in G have an edge in $G[S]$ and each vertex not in S is close to at most one vertex of S. For simplicity, assume that $p = |S|$. The second step is to guess a coloring χ of S and a set $B \subseteq S$; now it suffices to decide whether there is a b-coloring ψ of G such that ψ extends χ, and each vertex in B is a b-vertex in ψ. We may assume that $\chi(S) = [p]$ and $\chi(B) = [b]$ for some integer $b \leq p$.

The next step is to choose b-vertices for colors in $[b+1, k]$. What needs to be carefully handled are the colors in $[b+1, p]$ because we need to ensure that ψ will be proper and that vertices in B can become b-vertices. For example, if a vertex $u \in B$ has degree $k-1$ and it has a neighbor of color p in χ, then the b-vertex for p cannot be adjacent to u. We begin by guessing a so-called *color plan*, which is a function determining the location of b-vertices for the colors in $[b+1, p]$. Crucially, the number of color plans will be small. The next step is to compute a so-called *color realization* ρ, which is an injective assignment of colors in $[b+1, k]$ to χ-candidates outside of S. We show that using a "valid" color plan, we can compute a color realization such that each chosen candidate can indeed become a b-vertex. More precisely, each vertex in $B \cup \text{range}(\rho)$ will be χ_ρ-candidate, where $\chi_\rho = \chi \cup \rho^{-1}$ is the coloring obtained by "extending" χ according to ρ.

Now it may occur that the coloring χ_ρ we have constructed has a "pivot", i.e., a vertex that cannot be assigned any color, see the above described algorithm for b-Chromatic Number on trees for an idea what a pivot is. Sometimes, the realization ρ can be slightly modified to get rid of the pivot. However, this is not straightforward because we must ensure that each vertex in $B \cup \text{range}(\rho')$ is a $\chi_{\rho'}$-candidate, where ρ' is the new realization. On the other hand, sometimes the pivot is "unavoidable", and we must try a different pair (χ, B). In fact, there are more properties we need ρ to have but achieving "pivot-freeness" will be the most challenging.

Once we have a "feasible" color realization ρ, we are ready to extend χ_ρ into a b-coloring of G. First, we color the neighborhoods of all chosen candidates. Roughly speaking, we proceed in a BFS-order from S, but there may be an "exceptional" candidate that needs to be handled first. Finally, we color the remainder of the graph. This last step is also non-trivial because there may be a vertex which already sees all colors. However, one property we require from ρ will ensure that this cannot happen.

For a structured overview of the proof, see Algorithm 2.

4.2 Technical definitions and lemmas

Let us fix an instance (G, k) of the b-Coloring problem, and let p_G be the feedback edge number of G. We will assume that $k \geq 96p_G + 18$ since otherwise we could simply use Proposition 3.

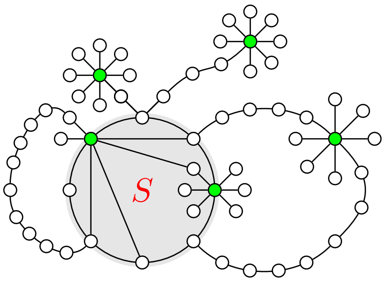

Figure 3 A depiction of the fen-core S, see Definition 17. If $k \leq 9$, then the green vertices may be used as b-vertices. Note that in contrast to this figure, $G[S]$ may be disconnected.

First, we define a small vertex set S, which will play a central role in the algorithm, see Figure 3. To handle the connected components of G which are trees, we also define a superset S^+ of S.

▶ **Definition 17.** *Let $S \subseteq V(G)$ be a set. An S-outer* path *P is a path in G such that $V(P) \cap S = \emptyset$ and $V(P) \cap N(S) = \{u, v\}$, where u and v are the endpoints of P. We say that S is a* fen-core *if each cycle in G has at least one edge in $G[S]$, there are at most two (u, S)-paths for each $u \in \overline{S}$, each S-outer path has length at least 7, and $|S| \leq 32p_G$. An* extension *of S is a minimal set $S^+ \supseteq S$ such that each connected component of G contains a vertex of S^+.*

Observe that if S is a fen-core, then no vertex in $\overline{S}$ has two neighbors in S because it would induce an S-outer path of length 0. Now we show that S can be efficiently computed.

▶ **Lemma 18.** *A fen-core exists and it can be found in polynomial time.*

Proof sketch. Remove all dangling trees from G. Apply Lemma 2 to each connected component of G, and define S to be the union of all obtained sets of vertices. Now it suffices to add all S-outer paths of length at most 6 to S. ◀

Let us fix a fen-core S and an extension S^+ of S, and let $p = |S|$. From now on, we may view p as the parameter of our algorithm because $p \in \mathcal{O}(p_G)$. Note that $k \geq 3p + 18$.

Now we present a key definition. Informally, a vertex is a χ-candidate if it may become a b-vertex in a coloring of G that extends a partial coloring χ. We also define an *S-profile*, which consists of a coloring of S and a set $B \subseteq S$ of vertices that should become b-vertices.

▶ **Definition 19.** *Let $U \subseteq V(G)$ and let χ be a proper coloring of $G[U]$. The* χ-redundancy *of a vertex $v \in V(G)$ is the integer $\mathsf{red}_\chi(v) = |N[v] \setminus U| + |\chi(N[v])| - k$. If $\mathsf{red}_\chi(u) \geq 0$, we say that u is a* χ-candidate. *If $\mathsf{red}_\chi(u) = 0$, we say that u is* χ-tight.

Let $\chi\colon S \to [p]$ be a proper coloring of $G[S]$, and let $B \subseteq S$ be a set such that each vertex in B is a χ-candidate and $\chi(B) = [b]$, where $b = |B|$. We say that $\zeta := (\chi, B)$ is an S-profile. *We call a k-b-coloring ψ of G* described by *ζ if $\chi = \psi \restriction S$, all vertices in B are b-vertices in ψ, and no color in $[b+1, p]$ has a b-vertex in S.*

Let us fix an S-profile $\zeta := (\chi, B)$. Let $K = \{u \in \overline{S} \mid \mathsf{red}_\chi(u) \geq 0\}$ and $K^+ = K \cup B$. The reader should think of K^+ as the set of our candidates for b-vertices, in a sense formalized by the following observation.

▶ **Observation 20.** *If ψ is a k-b-coloring of G described by ζ, then each color $c \in [k]$ has a b-vertex $u \in K^+$ in ψ. Moreover, if $|K^+| = k$, then all vertices in K^+ are b-vertices in ψ.*

Proof. Let $c \in [k]$ be a color and let $u \in V(G)$ be a b-vertex for c in ψ. If $u \in S \setminus B$, then by Definition 19, $c \in [b]$ and there is another b-vertex for c in ψ that is in B. Hence, we may assume $u \notin S \setminus B$. By definition of a b-vertex, we know that $|\psi(N[u])| = k$. Since $|N[u] \setminus S| + |\chi(N[u])| \geq |\psi(N[u])|$, we deduce that the χ-redundancy of u is non-negative, which implies that $u \in K^+$. The second part of the statement follows by the pigeonhole principle. ◀

Next, we define a *link*, which is a vertex that cannot become a b-vertex if the two linked vertices have the same color. We also define a *pivot*, generalizing the definition from the algorithm for trees. In particular, ψ-links in our definition correspond to vertices of degree $k-1$ in the algorithm for trees.

▶ **Definition 21.** *Let ψ be a partial coloring of G such that $\chi \subseteq \psi$. If $u, v \in V(G)$ are distinct vertices, then we say that u and v are* ψ-linked *if there is a ψ-tight vertex $w \in N(u) \cap N(v) \cap K^+$. We also say that w is a (ψ, u, v)-link.*

Let $D \subseteq V(G)$ be a set. We say that a vertex $u \in \overline{S \cup D}$ is a D-pivot *if for each $v \in D$, either $uv \in E(G)$ or there is a vertex $w \in N(u) \cap N(v) \cap D$. The vertices in $N(u) \cap N(D) \cap D$ will be called* (u, D)-links. *We say that u is a* (ψ, D)-pivot *if u is a D-pivot and each (u, D)-link is a (ψ, u, v)-link for some $v \in D$.*

▶ **Observation 22.** *If ψ is a k-b-coloring of G described by (χ, B), $u \in V(G)$, $v \in \overline{S}$, and $w \in K^+$ is the (χ, u, v)-link and also a b-vertex in ψ, then $\psi(u) \neq \psi(v)$.*

Proof. Let u, v, w, and ψ be as in the statement. Let $f\colon [k] \to \chi(N[w]) \cup (N[w] \setminus S)$ be a function such that for $c \in \chi(N[w])$, we have $f(c) = c$, and otherwise $f(c)$ is a vertex of $N[w]$ colored with c by ψ. Since w is a b-vertex in ψ, f is indeed a well-defined function. By Definition 21, w is χ-tight, which means that $|N[w] \setminus S| + |\chi(N[w])| = k$ and that f is a bijection. Suppose for contradiction that $c := \psi(u) = \psi(v)$. If $u \in S$, then $f(c) = c$, and otherwise $f(c) = u$ since f is a bijection. Both cases lead to a contradiction since $f(c) = v$ (and $u \neq v$ by Definition 21). ◀

Now we define a *color plan*, which is a function deciding, for each color $c \in [b+1, p]$, whether the b-vertex for c should be in $N(S)$, and if yes, which vertex of S it should be adjacent to (it may be adjacent to at most one vertex of S by Definition 17). The b-vertices that should not be in $N(S)$ must be in $K^* := K \setminus N(S)$, and the color plan assigns $*$ to their colors. Only some color plans may be used to produce a b-coloring: we call such plans *valid*.

▶ **Definition 23.** *6A function $\pi\colon [b+1, p] \to S \sqcup \{*\}$ is called a* color plan *of ζ. If $c \in [b+1, p]$ is a color, then π is* c-critical *if $\pi(c) = *$ and each vertex in K^* is χ-linked to a vertex in $\chi^{-1}(c)$. We say that π is* critical *if it is c-critical for some $c \in [b+1, p]$. We say that π is* valid *if:*

1. *$|\pi^{-1}(*)| \leq |K^*|$; and*
2. *for each $u \in S$, we have $|\pi^{-1}(u)| \leq |N(u) \cap K|$; and*
3. *for each $u \in S$, if $\chi(u) \in [b+1, p]$, then $\pi(\chi(u)) \neq u$; and*
4. *for each $u \in B$, we have $|\pi^{-1}(u) \cap \chi(N(u))| \leq \textsf{red}_\chi(u)$; and*
5. *if π is critical, then $|K^+| > k$ and there is a vertex $u \in S$, called a* π-anchor, *such that $|\pi^{-1}(u)| < |N(u) \cap K|$ and there is a vertex in K^* χ-linked to u.*

Let us present an intuition behind the five properties of a valid color plan. First, Properties 1 and 2 ensure that there are enough vertices for the plan to be realized. Second, a violation of Property 3 would mean that the obtained coloring cannot be proper. Third, Property 4 ensures that each vertex in B can still become a b-vertex. Finally, if π is c-critical, then the b-vertex for c must be linked to a vertex in S colored with c, which means that the link cannot be a b-vertex. Property 5 ensures that there exists such a link that does not have to be a b-vertex, which will enable us to find a b-vertex for c.

There are three possible reasons for the non-existence of a b-coloring described by ζ. The most obvious reason is that there are not enough candidates, i.e., $|K^+| < k$. The second natural reason is that there is no valid color plan of ζ. In the following definition, we describe the third possible reason: informally, it says that there is an "unavoidable" pivot.

▶ **Definition 24.** *We say that ζ is* candidate-failing *if $|K^+| < k$ and* plan-failing *if there is no valid color plan of ζ. We say that ζ is* pivot-failing *if there is a set $D \subseteq K \cup S$ and a vertex $u \in \overline{S \cup D}$ such that u is a (χ, D)-pivot, $K \subseteq D$, $[b] \subseteq \chi(D)$, and (a) $|K^+| \le k$ or (b) $\{v, w\} \cap S \neq \emptyset$ for every $v, w \in D$ such that w is the (χ, u, v)-link.*

We say that ζ is failing *if it is candidate-failing, pivot-failing or plan-failing.*

Definition 24 provides a sufficient and necessary condition for a b-coloring described by ζ to exist: ζ must not be failing. In the following lemma, we show the necessity.

▶ **Lemma 25.** *If ζ is failing, then there is no k-b-coloring of G described by ζ.*

Proof sketch. Suppose for contradiction that ζ is failing and there is a k-b-coloring ψ of G described by ζ. By Observation 20, $|K^+| \ge k$, which means that ζ is plan-failing or pivot-failing. If ζ is plan-failing, then we obtain a contradiction by constructing a valid color plan π based on ψ. Namely, for each color $c \in [b+1, p]$, if there is a b-vertex $v \in N(u) \cap \psi^{-1}(c)$ for some $u \in S$, then we set $\pi(c) = u$, and otherwise we set $\pi(c) = *$.

Now suppose that ζ is pivot-failing. Let D and u be as in Definition 24, and let $c := \psi(u)$. If $c \in [b]$, let $v \in D$ be a vertex such that $\chi(v) = c$, and otherwise, let v be the b-vertex for c in K (v exists by Observation 20). In both cases, we have $v \in D$. Since ψ is proper, we have $uv \notin E(G)$. Hence, u and v are χ-linked via a vertex $w \in K^+$. Since $\psi(u) = \psi(v)$ and $u \notin S$, we may use Observation 22 to deduce that w is not a b-vertex in ψ. By Observation 20, we obtain $|K^+| > k$, which means that $\{v, w\} \cap S \neq \emptyset$, see Definition 24. If $w \in S$, then $w \in K^+ \cap S = B$, and w would be a b-vertex in ψ, which it is not. Hence, $w \notin S$ and $v \in S$. It can be observed that $D \cap S = \{v\}$ (otherwise there would be a short S-outer path, which would contradict Definition 17). Observe that $b \le 1$ because $[b] \subseteq \chi(D)$. If $v', w' \in D$ are such that w' is the (χ, u, v')-link, then $\{v', w'\} \cap S \neq \emptyset$ by Definition 24, which implies $v' = v$. Hence, $D \setminus N(u) = \{v\}$. Since $|K| = |K^+| - b > k - b \ge k - 1$, we have $|K| \ge k$. Since $K \subseteq D$ and $v \notin K$, we have $K \subseteq N(u)$. Hence, u has degree at least k, which implies $u \in K$; a contradiction with $u \notin D$. ◀

Now we define a *color realization* ρ, which is a function selecting concrete vertices as b-vertices for colors in $[b+1, k]$, based on some color plan. We also define *damage-freeness*, which is a property of ρ saying that the chosen vertices do not interfere with each other.

▶ **Definition 26.** *An injective function $\rho \colon [b+1, k] \to K$ is called a* color realization. *Let $B_\rho := B \cup \operatorname{range}(\rho)$, $S_\rho := S \cup B_\rho$, and $\chi_\rho := \chi \cup \rho^{-1}$; note that χ_ρ is a k-coloring of $G[S_\rho]$. We say that ρ* realizes *a color plan π (or is a color realization of π) if for each $c \in [b+1, p]$:*

(a) *$\pi(c) = *$ if and only if $\rho(c) \in K^*$;*

(b) *for each $u \in S$, it holds that $\pi(c) = u$ if and only if $\rho(c) \in N(u)$.*

A vertex u is damaged *in ρ if there is a color $c \in [b+1, p]$ and a vertex $v \in \chi^{-1}(c)$ such that $\rho(c) \in K^*$ and v and $\rho(c)$ are χ-linked via u. We say that ρ is* damage-free *if no vertex in $\operatorname{range}(\rho)$ is damaged in ρ.*

The following lemma says that the vertices chosen by a "good" color realization can become b-vertices.

▶ **Lemma 27.** *If ρ is a damage-free color realization of a valid color plan π, then:*
(a) *χ_ρ is a proper partial coloring of G; and*
(b) *each vertex $u \in B_\rho$ is a χ_ρ-candidate.*

Proof sketch. Observe that (a) holds because χ is proper, $\chi \subseteq \chi_\rho$, and because Property 3 of Definition 23 is satisfied. Hence, it suffices to focus on (b). If $u \in B$, then u is a χ_ρ-candidate by Property 4: informally, $\pi^{-1}(u) \cap \chi(N(u))$ is the set of colors "seen twice" by u in χ_ρ. On the other hand, if $u \notin B$, then u is a χ_ρ-candidate because ρ is damage-free. ◀

Now we define when a color realization ρ is *feasible*, which is a property that ensures that χ_ρ can be extended into a b-coloring. Indeed, Lemma 27 says that damage-freeness ensures that each vertex in B_ρ can be made a b-vertex individually but not necessarily that they can all become b-vertices *at the same time*. The definition of feasibility is rather technical so we first give intuition for each of its component.

Informally, a ρ-pivot u is a vertex that cannot be assigned any color: each color c is either in its neighborhood or u is linked to a vertex colored with c (this is analogous to the role of a pivot in the algorithm for trees). Similarly, a vertex u is ρ-blocked by c if c cannot be added to the neighborhood of u: each uncolored vertex either has a neighbor colored with c or it is linked to a vertex colored with c. Finally, *almost* 13*-safeness* is a property used to ensure that high-degree vertices that are not chosen to become b-vertices can be colored at the end of the algorithm.

▶ **Definition 28.** *Let ρ be a color realization. A vertex $u \in \overline{S_\rho}$ is a ρ-*pivot *if there is a set $D \subseteq S_\rho$ such that $\chi_\rho(D) = [k]$ and u is a (χ_ρ, D)-pivot. We say that D is ρ-*pivoted *by u. We say that ρ is* pivot-free *if there is no ρ-pivot.*

*A vertex $u \in B_\rho$ is ρ-*blocked *by a color $c \in [k]$ if $c \notin \chi_\rho(N[u])$ and for each $v \in N(u) \setminus S_\rho$, there is a vertex $w \in \chi_\rho^{-1}(c)$ such that $w \in N(v)$ or w is χ_ρ-linked to v via a vertex in B_ρ. We say that ρ is* block-free *if no vertex $u \in B_\rho$ is ρ-blocked by any color $c \in [k]$.*

*A vertex $u \in V(G)$ is ℓ-*safe *in ρ if $|\{c \in [p+1,k] \mid \operatorname{dist}(u, S^+) < \operatorname{dist}(\rho(c), S^+)\}| \leq \ell$. We say that ρ is ℓ-*safe *if all vertices in $K \setminus B_\rho$ are ℓ-safe in ρ. We say that ρ is* almost ℓ-safe *if all vertices in $K \setminus B_\rho$ are ℓ-safe in ρ, possibly except for one vertex v such that either $v \in N(B_\rho)$ or v has at most $p+2$ neighbors in $N(B_\rho)$.*

We say that ρ is feasible *if it is damage-free, almost 13-safe, pivot-free, and block-free, and it realizes a valid color plan.*

The following lemma is the technical core of the proof.

▶ **Lemma 29.** *If ζ is not failing, then given a valid color plan π of ζ, a feasible color realization can be computed in polynomial time.*

Proof sketch. First, we compute a damage-free and almost 0-safe color realization ρ, as described in Algorithm 1. In a nutshell, the algorithm computes an auxiliary bipartite graph H such that an edge cu means that a vertex u can be colored with color c, see line 5. In the correctness proof, we use Hall's theorem to show that a maximal matching in H covers all colors present in H, which is needed for line 11. However, not all colors are present in H; other colors are handled on lines 9, 10, and 13. Concerning lines 9 and 13, we must show that the required vertex v can always be found. We remark that almost 0-safeness is ensured by line 13 (in fact, ρ is even 0-safe unless π is critical). See [2] for more details.

Algorithm 1 Constructing a damage-free and 0-safe color realization.

Input: A graph G, an integer k, a fen-core S, an S-profile (χ, B), a valid color plan π.

if π *is c-critical for some* $c \in [b+1, p]$ **then**
  let u_c be a π-anchor and let $v_c \in K^*$ be χ-linked to u_c via w_c
  // u_c, v_c, and w_c exist by Definition 23 since π is valid
  $U := K \setminus \{w_c\}$;
else $U := K$
let H be a bipartite graph with parts $C^* := \pi^{-1}(*)$ and K^* such that $E(H) = \{cu \mid c \in C^*, u \in K^*, u \text{ is not } \chi\text{-linked to a vertex in } \chi^{-1}(c)\}$
let μ be a maximal matching in H
for c **from** $b+1$ **to** p **do**
  if $\pi(c) = u$ *for some vertex* $u \in S$ **then**
    $\rho(c) := v$, where $v \in (U \cap N(u)) \setminus \mathrm{range}(\rho)$
  if π *is c-critical* **then** $\rho(c) := v_c$
  else if $\pi(c) = *$ **then** $\rho(c) := u$, where $u \in K^*$ is the vertex matched with c in μ
let $\prec$ be an ordering of U such that if $u \prec v$, then $\mathrm{dist}(u, S^+) \leq \mathrm{dist}(v, S^+)$
for c **from** $p+1$ **to** k **do** $\rho(c) := v$, where v is the $\prec$-smallest vertex in $U \setminus \mathrm{range}(\rho)$
return ρ

If the color realization ρ found by Algorithm 1 is feasible, we are done. Otherwise, we need to slightly modify ρ to obtain a feasible realization ρ'. There are two types of modifications we need: swaps and shifts. We say that ρ' is obtained by *swapping* two colors $c, d \in [b+1, k]$ if $\rho' = \rho[c \mapsto \rho(d), d \mapsto \rho(c)]$, and that ρ' is obtained by *shifting* a color $c \in [b+1, k]$ if $\rho' = \rho[c \mapsto v]$ for some vertex $v \in K \setminus \mathrm{range}(\rho)$. We need to show "safeness" of these operations: for example, if ρ is ℓ-safe and ρ' obtained by a swap, then ρ' is $(\ell+2)$-safe. The fact that feasibility requires almost 13-safeness gives us a limitation on the number of modifications we can make. We must be very careful, as we must ensure that ρ' is damage-free, pivot-free, and block-free. We elaborate on pivot-freeness, as it is the most challenging to achieve (but it can be achieved since ζ is not pivot-failing).

Suppose that u is a ρ-pivot and recall that u is a (χ_ρ, D)-pivot for some set $D \subseteq S_\rho$ by Definition 28. However, u is not necessarily a (χ, D)-pivot. Indeed, if a (u, D)-link v has two neighbors of the same color in χ_ρ, one in S and one in $\mathrm{range}(\rho)$, then v is not χ-tight. The case when u is not (χ, D)-pivot is the easier one: a single swap suffices to obtain the desired realization ρ', see Section 4.3.3 in [2]. From now on, suppose that u is a (χ, D)-pivot.

The basic idea how to modify ρ so that u is no longer a ρ-pivot is simple: choose a color $c \in [p+1, k]$ and move it far away from u (via a swap or a shift). After such a movement, we obtain a color realization ρ' such that u is not a ρ'-pivot because there is only one vertex colored with c, namely $\rho'(c)$. The second possibility is to move c to u itself; this is, of course, possible only if $u \in K$. We formalize this idea as follows: let $Q \subseteq S \cup K$ be the maximal set such that u is a (χ, Q)-pivot. Observe that $D \subseteq Q$. If $K \nsubseteq Q$, then we can simply move c to a vertex in $K \setminus Q$. The more interesting case occurs when $K \subseteq Q$. Now we use the fact that ζ is not pivot-failing. By Definition 24, we can find two vertices $v, w \in Q \setminus S$ such that v is the (χ, u, w)-link. Using a few swaps and/or shifts, we can achieve the situation in which $\rho'(c) = w$ and $v \notin \mathrm{range}(\rho')$; recall that $c \in [p+1, k]$. Now it can again be easily observed that u is not a ρ'-pivot. In other words, u can be safely colored with c.

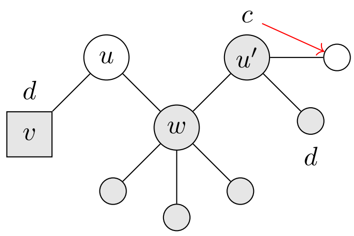

Figure 4 A case in the proof of Lemma 29 in which a shift (depicted by the red arrow) destroys a pivot u but a different pivot u' emerges. The gray vertices are in S_ρ and $v \in S$. The vertex w is χ-tight and for each color $c' \in [b+1, k] \setminus \{c, d\}$, we have $\rho(c') \in N[w]$.

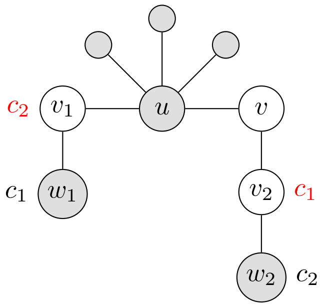

Figure 5 A case in which there is an exceptional vertex $u \in B_\rho$, which needs to be handled first in the proof of Lemma 30. The gray vertices are in B_ρ and the depicted w_1-w_2 path is a subpath of a long S-outer path such that $\mathrm{dist}(w_1, S) = \mathrm{dist}(w_2, S)$. We have $c_1, c_2 \in [k]$, $\rho(c_1) = w_1$, $\rho(c_2) = w_2$, and each color $c \in [k] \setminus \{c_1, c_2\}$ is present in $N[u]$, i.e., $c \in \chi_\rho(N[u])$. Moreover, u is χ_ρ-tight. Hence, if w_1 and w_2 are processed before u, then v_1 (resp. v_2) might be colored with c_2 (resp. c_1), as depicted in red. If this occurs, v becomes a ψ-pivot, which is a problem because it cannot be colored.

If our goal was to find a color realization ρ' such that u is not a ρ'-pivot, then we would be done. Unfortunately, there may be a ρ'-pivot u' such that $u' \neq u$. We are able to show that if ρ' is obtained by a swap, than such a vertex u' cannot exist. However, this is not true for a shift: a case in which a different pivot emerges is depicted in Figure 4. Hence, we must be very careful which color c we choose to shift. For more details, see Section 4.3 in [2]. ◀

The following lemma shows how to color $N(B_\rho)$.

▶ **Lemma 30.** *Given a feasible color realization ρ, we can compute a partial k-b-coloring $\psi' \colon S_\rho \cup N(B_\rho) \to [k]$ described by ζ such that $\psi' \supseteq \chi_\rho$ in polynomial time.*

Proof sketch. We iteratively build a coloring ψ. Initially, we set $\psi := \chi_\rho$. In each iteration, we choose a different vertex $u \in B_\rho$, and color $N(u)$ so that u becomes a b-vertex. Our invariant is that all vertices in B_ρ are ψ-candidates; this holds at the beginning by Lemma 27. We process the vertices in B_ρ in order of increasing distance from S^+ (recall that S^+ equals S plus one vertex from each component of G that is a tree). However, there may be an exceptional vertex $u \in B_\rho$, which needs to be handled first. In particular, the issue may occur when u lies on an S-outer path and the two u-S paths have very similar lengths, see Figure 5 for an illustration. Nevertheless, all vertices in B_ρ can be processed using the pivot- and block-freeness of ρ. Once $N(B_\rho)$ is colored, we set $\psi' := \psi$. For more details, see Section 4.4 in [2]. ◀

Now we show how to color the remainder of the graph to obtain a (total) k-b-coloring.

▶ **Lemma 31.** *If ρ is a feasible color realization and $\psi_0 \colon S_\rho \cup N(B_\rho) \to [k]$ is a partial b-coloring described by ζ, then there is a (total) k-b-coloring ψ described by ζ. Moreover, ψ can be computed in polynomial time given ψ_0.*

Proof sketch. We need to assign colors to vertices uncolored by ψ_0. Since ρ is feasible, it is almost 13-safe. Observe that it suffices to discuss vertices of degree at least k and that all such vertices are in $K \setminus B_\rho$. If the exceptional vertex mentioned in the definition of almost ℓ-safeness exists, then we color this vertex first (its properties specified in Definition 28 make it possible). We color the remaining vertices from $K \setminus B_\rho$ in order of increasing distance from S^+; this is possible because they are all 13-safe and $k \geq 3p + 18$. ◀

Finally, we are ready to prove Theorem 16.

Proof sketch of Theorem 16. The algorithm is summarized in Algorithm 2. Its correctness follows from the lemmas proven in this section. Let us now analyze its running time. Observe that for graphs with p_G feedback edges, we can easily compute a tree-decomposition of width $p_G + 1$. Hence, if $k < 96p_G + 18$, then the running time is $2^{\mathcal{O}(p_G^2)}$ by Proposition 3 (which will actually dominate the running time).

Assume $k \geq 96p_G + 18$ and recall that $p \in \mathcal{O}(p_G)$ by Definition 17. Observe that the number of S-profiles is at most $p^p \cdot 2^p \in p^{\mathcal{O}(p)}$ and the number of color plans is at most $(p+1)^{p-b} \in p^{\mathcal{O}(p)}$. The conditions on lines 4 and 7 can clearly be verified in polynomial time. Since Lemmas 29, 30, and 31 also guarantee polynomial time, we achieve the desired running time. ◀

Algorithm 2 Finding a b-coloring.

Input: A graph G with feedback edge number p_G, an integer k
Output: A k-b-coloring of G or NO if it does not exist.
1 **if** $k < 96p_G + 18$ **then** use the tree-width based algorithm, see Proposition 3
2 compute a fen-core S of G using Lemma 18
3 **for each** *S-profile* $\zeta := (\chi, B)$ **do**
4   **if** *ζ is candidate-failing or pivot-failing, see Definition 24* **then**
5   **continue** with next S-profile
6   **for each** *color plan π of ζ* **do**
7   **if** *π is not valid, see Definition 23* **then continue** with next color plan
8   compute a feasible color realization ρ using π and Lemma 29
9   compute a partial coloring ψ_0 of G using ρ and Lemma 30
10   extend ψ_0 to a k-b-coloring ψ of G using Lemma 31
11   **return** ψ
12 **return** NO

References

1 Davi Andrade and Ana Silva. (sub)fall coloring and b-coloring parameterized by treewidth. In *Anais do VII Encontro de Teoria da Computação*, pages 69–72, Porto Alegre, RS, Brasil, 2022. SBC. doi:10.5753/etc.2022.222982.

2 Jakub Balabán. Finding b-colorings using feedback edges. *CoRR*, abs/2512.14390, 2025. doi:10.48550/arXiv.2512.14390.

3 Jakub Balabán, Robert Ganian, and Mathis Rocton. Computing twin-width parameterized by the feedback edge number and vertex integrity. *SIAM J. Discret. Math.*, 39(4):2102–2141, 2025. doi:10.1137/24M1705329.

4 Michael J. Bannister, Sergio Cabello, and David Eppstein. Parameterized complexity of 1-planarity. *J. Graph Algorithms Appl.*, 22(1):23–49, 2018. doi:10.7155/JGAA.00457.

5 Rémy Belmonte, Pim van 't Hof, Marcin Kaminski, Daniël Paulusma, and Dimitrios M. Thilikos. Characterizing graphs of small carving-width. *Discret. Appl. Math.*, 161(13-14):1888–1893, 2013. doi:10.1016/J.DAM.2013.02.036.

6 Flavia Bonomo, Guillermo Durán, Frédéric Maffray, Javier Marenco, and Mario Valencia-Pabon. On the b-coloring of cographs and P_4-sparse graphs. *Graphs Comb.*, 25(2):153–167, 2009. doi:10.1007/S00373-008-0829-1.

7 Flavia Bonomo, Oliver Schaudt, Maya Stein, and Mario Valencia-Pabon. b-coloring is np-hard on co-bipartite graphs and polytime solvable on tree-cographs. *Algorithmica*, 73(2):289–305, 2015. doi:10.1007/S00453-014-9921-5.

8 Anudhyan Boral, Marek Cygan, Tomasz Kociumaka, and Marcin Pilipczuk. A fast branching algorithm for cluster vertex deletion. *Theory of Computing Systems*, 58(2):357–376, 2016. doi:10.1007/S00224-015-9631-7.

9 Victor A. Campos, Carlos Vinícius G. C. Lima, Nicolas Almeida Martins, Leonardo Sampaio Rocha, Marcio Costa Santos, and Ana Silva. The b-chromatic index of graphs. *Discret. Math.*, 338(11):2072–2079, 2015. doi:10.1016/J.DISC.2015.04.026.

10 Victor A. Campos, Carlos Vinícius G. C. Lima, and Ana Silva. Graphs of girth at least 7 have high b-chromatic number. *Eur. J. Comb.*, 48:154–164, 2015. doi:10.1016/J.EJC.2015.02.017.

11 Victor A. Campos, Cláudia Linhares Sales, Frédéric Maffray, and Ana Silva. b-chromatic number of cacti. *Electron. Notes Discret. Math.*, 35:281–286, 2009. doi:10.1016/J.ENDM.2009.11.046.

12 Victor A. Campos, Cláudia Linhares Sales, Rudini Menezes Sampaio, and Ana Karolinna Maia. Maximization coloring problems on graphs with few p_4. *Discret. Appl. Math.*, 164:539–546, 2014. doi:10.1016/J.DAM.2013.10.031.

13 Sylvie Corteel, Mario Valencia-Pabon, and Juan-Carlos Vera. On approximating the b-chromatic number. *Discrete Applied Mathematics*, 146(1):106–110, 2005. doi:10.1016/J.DAM.2004.09.006.

14 Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015. doi:10.1007/978-3-319-21275-3.

15 Josep Díaz, Jordi Petit, and Maria Serna. A survey of graph layout problems. *ACM Computing Surveys (CSUR)*, 34(3):313–356, 2002. doi:10.1145/568522.568523.

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

17 Pål Grønås Drange, Markus S. Dregi, and Pim van 't Hof. On the computational complexity of vertex integrity and component order connectivity. *Algorithmica*, 76(4):1181–1202, 2016. doi:10.1007/S00453-016-0127-X.

18 Esther Galby, Liana Khazaliya, Fionn Mc Inerney, Roohani Sharma, and Prafullkumar Tale. Metric dimension parameterized by feedback vertex set and other structural parameters. *SIAM Journal on Discrete Mathematics*, 37(4):2241–2264, 2023. doi:10.1137/22M1510911.

19 Robert Ganian and Sebastian Ordyniak. The power of cut-based parameters for computing edge-disjoint paths. *Algorithmica*, 83(2):726–752, 2021. doi:10.1007/S00453-020-00772-W.

20 Tesshu Hanaka, Michael Lampis, Manolis Vasilakis, and Kanae Yoshiwatari. Parameterized Vertex Integrity Revisited. In Rastislav Královič and Antonín Kučera, editors, *49th International Symposium on Mathematical Foundations of Computer Science (MFCS 2024)*, volume 306 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 58:1–58:14, Dagstuhl, Germany, 2024. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.MFCS.2024.58.

21 Frédéric Havet, Cláudia Linhares Sales, and Leonardo Sampaio Rocha. b-coloring of tight graphs. *Discret. Appl. Math.*, 160(18):2709–2715, 2012. doi:10.1016/J.DAM.2011.10.017.

22 Robert W. Irving and David F. Manlove. The b-chromatic number of a graph. *Discret. Appl. Math.*, 91(1-3):127–141, 1999. doi:10.1016/S0166-218X(98)00146-2.

23 Lars Jaffke, Paloma T. Lima, and Daniel Lokshtanov. b-coloring parameterized by clique-width. *Theory Comput. Syst.*, 68(4):1049–1081, 2024. doi:10.1007/S00224-023-10132-0.

24 Lars Jaffke, Paloma T. Lima, and Roohani Sharma. Structural parameterizations of b-coloring. In Satoru Iwata and Naonori Kakimura, editors, *34th International Symposium on Algorithms and Computation, ISAAC 2023, December 3-6, 2023, Kyoto, Japan*, volume 283 of *LIPIcs*, pages 40:1–40:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPIcs.ISAAC.2023.40.

25 Marko Jakovac and Sandi Klavžar. The b-chromatic number of cubic graphs. *Graphs and Combinatorics*, 26(1):107–118, 2010. doi:10.1007/S00373-010-0898-9.

26 Marko Jakovac and Iztok Peterin. The b-chromatic number and related topics - A survey. *Discret. Appl. Math.*, 235:184–201, 2018. doi:10.1016/J.DAM.2017.08.008.

27 Jack de Jong. Analysing flow-like problems parameterised by tree-depth. Master's thesis, Utrecht University, 2024.

28 Richard M. Karp. Reducibility among combinatorial problems. In Raymond E. Miller and James W. Thatcher, editors, *Proceedings of a symposium on the Complexity of Computer Computations, held March 20-22, 1972, at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York, USA*, The IBM Research Symposia Series, pages 85–103. Plenum Press, New York, 1972. doi:10.1007/978-1-4684-2001-2_9.

29 Christian Komusiewicz, Dieter Kratsch, and Van Bang Le. Matching cut: Kernelization, single-exponential time fpt, and exact exponential algorithms. *Discret. Appl. Math.*, 283:44–58, 2020. doi:10.1016/J.DAM.2019.12.010.

30 Mekkia Kouider and Manouchehr Zaker. Bounds for the b-chromatic number of some families of graphs. *Discrete Mathematics*, 306(7):617–623, 2006. doi:10.1016/J.DISC.2006.01.012.

31 Jan Kratochvíl, Zsolt Tuza, and Margit Voigt. On the b-chromatic number of graphs. In Ludek Kucera, editor, *Graph-Theoretic Concepts in Computer Science, 28th International Workshop, WG 2002, Cesky Krumlov, Czech Republic, June 13-15, 2002, Revised Papers*, volume 2573 of *Lecture Notes in Computer Science*, pages 310–320. Springer, 2002. doi:10.1007/3-540-36379-3_27.

32 Rafael A Melo, Michell F Queiroz, and Marcio C Santos. A matheuristic approach for the b-coloring problem using integer programming and a multi-start multi-greedy randomized metaheuristic. *European Journal of Operational Research*, 295(1):66–81, 2021. doi:10.1016/J.EJOR.2021.02.049.

33 I. Vinod Reddy. On structural parameterizations of load coloring. *Discrete Applied Mathematics*, 342:181–189, 2024. doi:10.1016/j.dam.2023.09.001.

34 Paul D. Seymour and Robin Thomas. Call routing and the ratcatcher. *Combinatorica*, 14(2):217–241, 1994. doi:10.1007/BF01215352.

35 Ana Shirley Ferreira Da Silva. *The b-chromatic number of some tree-like graphs. (Le nombre b-chromatique de quelques classes de graphes généralisant les arbres)*. PhD thesis, Joseph Fourier University, Grenoble, France, 2010. URL: https://tel.archives-ouvertes.fr/tel-00544757.

36 Dimitrios M. Thilikos, Maria J. Serna, and Hans L. Bodlaender. Constructive linear time algorithms for small cutwidth and carving-width. In D. T. Lee and Shang-Hua Teng, editors, *Algorithms and Computation, 11th International Conference, ISAAC 2000, Taipei, Taiwan, December 18-20, 2000, Proceedings*, volume 1969 of *Lecture Notes in Computer Science*, pages 192–203. Springer, 2000. doi:10.1007/3-540-40996-3_17.

37 Johannes Uhlmann and Mathias Weller. Two-layer planarization parameterized by feedback edge set. *Theor. Comput. Sci.*, 494:99–111, 2013. doi:10.1016/J.TCS.2013.01.029.

38 Clara Inés Betancur Velasquez, Flavia Bonomo, and Ivo Koch. On the b-coloring of P_4-tidy graphs. *Discret. Appl. Math.*, 159(1):60–68, 2011. doi:10.1016/J.DAM.2010.10.002.