Graphs and Combinatorics (2018) 34:457–475
https://doi.org/10.1007/s00373-018-1886-8

ORIGINAL PAPER

Dynamic F-free Coloring of Graphs

Piotr Borowiecki[1] · **Elżbieta Sidorowicz**[2]

Received: 26 May 2017 / Revised: 27 November 2017 / Published online: 15 March 2018

Abstract A problem of graph F-free coloring consists in partitioning the vertex set of a graph such that none of the resulting sets induces a graph containing a fixed graph F as an induced subgraph. In this paper we consider dynamic F-free coloring in which, similarly as in online coloring, the graph to be colored is not known in advance; it is gradually revealed to the coloring algorithm that has to color each vertex upon request as well as handle any vertex recoloring requests. Our main concern is the greedy approach and characterization of graph classes for which it is possible to decide in polynomial time whether for the fixed forbidden graph F and positive integer k the greedy algorithm ever uses more than k colors in dynamic F-free coloring. For various classes of graphs we give such characterizations in terms of a finite number of minimal forbidden graphs thus solving the above-mentioned problem for the so-called F-trees when F is 2-connected, and for classical trees, when F is a path of order 3 (the latter variant is also known as subcoloring or 1-improper coloring).

Keywords Graph coloring · Subcoloring · Improper coloring · Greedy algorithm · Grundy number

✉ Piotr Borowiecki
pborowie@eti.pg.gda.pl

Elżbieta Sidorowicz
e.sidorowicz@wmie.uz.zgora.pl

1 Department of Algorithms and System Modeling, Faculty of Electronics, Telecommunications and Informatics, Gdańsk University of Technology, Ul. Narutowicza 11/12, 80-233 Gdańsk, Poland

2 Department of Discrete Mathematics and Theoretical Computer Science, Faculty of Mathematics, Computer Science and Econometrics, University of Zielona Góra, Ul. prof. Z. Szafrana 4a, 65-516 Zielona Gora, Poland

Mathematics Subject Classification 68R10 · 05C85 · 05C15 · 05C75

1 Introduction

Graphs have proven to be a powerful model of complex networks in various domains, ranging from telecommunication to chemistry and social networks. In their earlier studies researches usually focused on static networks rather than their dynamic behavior. However, many real world networks are not static but inherently evolve over time (see, e.g., Newman [27]); each and every minute, new web pages are added, people become members of new social networks thereby creating new nodes and forming new groups (clusters) having certain properties. In this paper we join the concept of evolving graph structure with the dynamic maintenance of temporal partitions of its vertex set into the subsets that induce graphs with specified structural properties. In addition to practical applications, significant motivation for our research on the dynamic variant of graph partitioning comes from the area of designing of graph classes along with polynomial-time approximation algorithms (see, e.g., Borowiecki [6]). In the above context, as a model for dynamic partitioning, we use the dynamic F-free coloring of graphs.

1.1 Dynamic F-free Coloring

We consider simple, finite, undirected graphs $G = (V, E)$ with the vertex set V, edge set E and *order* $n = |V(G)|$. For graphs G and F we say that G *contains* F if there exists an induced subgraph of G isomorphic to F. Otherwise, we say that G is F-free. It is known that every class of F-free graphs is *hereditary*, i.e., every induced subgraph of an F-free graph is F-free. If from the fact that connected components of a graph G are F-free it follows that G is F-free, then the class of F-free graphs is *additive*. We will also use the fact that if a class of F-free graphs is additive, then F is connected. All classes of graphs that we consider are additive and hereditary.

An *F-free coloring* of a graph is a partition of its vertex set into subsets called *color classes* such that each color class induces an F-free graph. In the classical coloring, called *proper coloring*, adjacent vertices cannot be colored with the same color, which is equivalent to K_2-free coloring (K_p denotes the complete graph of order p). Indeed, in proper coloring each color class is an independent set, or equivalently, induces a K_2-free graph.

In this paper we consider the *dynamic F-free coloring*. The dynamic model of F-free coloring can be interpreted as a game of *Presenter* and *Painter*; the two uncooperative players. According to [7] the game can be described as follows. In a sequence of moves, Presenter gradually reveals the structure of a graph, i.e., in each move Presenter presents some number of new vertices along with the edges between them as well as the edges between new vertices and previously known ones. In the same move Presenter can also discolor arbitrary vertices, and then request Painter to immediately color the vertices of Presenter's choice. More formally, let $r > 0$ and let $\sigma_1, \ldots, \sigma_r$ denote subsequent moves of Presenter. Let G_1 be an empty graph. For $i > 1$, by G_i and G_{i+1} we denote the graphs known to Painter before and after σ_i, respectively. The move σ_i is defined by (D_i, U_i, E_i, C_i), where D_i is the set of vertices that

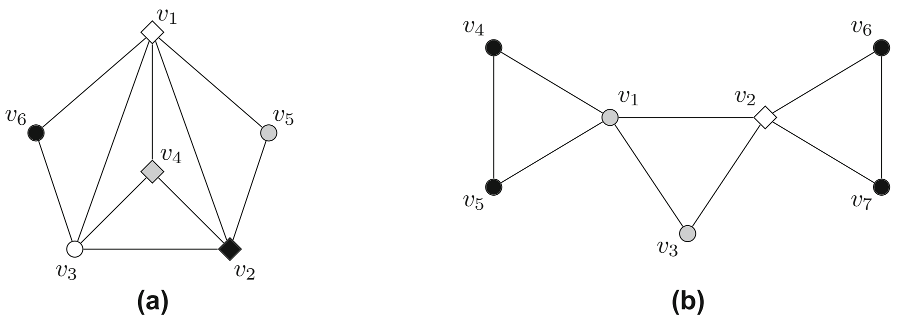

Fig. 1 Examples of greedy dynamic F-free colorings: **a** $F = K_2$ (proper coloring), and **b** $F = K_3$ (triangle-free coloring)

lose their colors, U_i is the set of new vertices, E_i is the set of edges uv such that $u \in U_i, v \in V(G_i) \cup U_i$, and C_i is the set of vertices that have to be immediately colored by Painter in response to σ_i. Naturally, $G_{i+1} = (V(G_i) \cup U_i, E(G_i) \cup E_i)$. Note that there is no loss of generality in assuming that $C_i \neq \emptyset$. We also allow U_i to be empty, i.e., at some stage of the game Presenter and Painter may continue discoloring and coloring the same graph. Similarly, Presenter does not have to discolor vertices, i.e., $D_i = \emptyset$ is allowed. In our setting Painter does not know the graph in advance, nor the order in which vertices will be colored or discolored. The goal of Presenter is to find a sequence of moves that force Painter to use as many colors as possible, while Painter aims at minimizing the number of colors.

Example 1.1 Consider dynamic F-free coloring of graphs in Fig. 1 under assumption that Painter acts greedily, i.e., always uses the smallest possible color. Note that in proper coloring of a graph presented in Fig. 1a Painter is forced to color each vertex with a different color. To achieve this goal Presenter reveals vertices $v_1, \ldots, v_4$ and requests Painter to color each vertex as soon as it is revealed (in other words $U_i = C_i = \{v_i\}$ for $i \in \{1, \ldots, 4\}$). In his next move, Presenter discolors v_2 and asks for coloring of just revealed vertex v_5. Now, if Presenter requests coloring of v_2 (for the second time), then since v_5 is already colored 2, the vertex v_2 will be colored 5. Next, Presenter discolors v_1 and asks for coloring of the new vertex v_6. Finally, the request for coloring of v_1 forces Painter to introduce color 6. The example in Fig. 1b presents K_3-free coloring with 3 colors. Consider the following sequence of coloring and discoloring requests (we assume that each vertex is revealed just before it is colored for the first time). Namely, requests for coloring of v_1, v_2 and v_3 (in order) result in the assignment of color 1 to v_1 and v_2, and forces Painter to use color 2 for v_3 (for otherwise we would have a monochromatic K_3 induced by $\{v_1, v_2, v_3\}$). Next, after discoloring of v_1 and v_2, Painter is asked to color $v_4, \ldots, v_7$ (naturally, the first color will be used for all of them). Consequently, v_1 will be colored 2 and because of the two triangles induced by $\{v_1, v_2, v_3\}$ and $\{v_2, v_6, v_7\}$ the third color will be forced in v_2. □

Dynamic F-free coloring is a natural generalization of both offline and online F-free coloring models. Namely, the problem of offline coloring of a graph $G = (V, E)$ can be seen as a one-move game with $\sigma_1 = (\emptyset, V, E, V)$, while online coloring is a game such that for every move: $D_i = \emptyset$, the set U_i is a singleton, and $C_i = U_i$. In the above-mentioned setting, the dynamic K_2-free coloring has been considered in [7].

1.2 Related Research and Our Results

The idea of F-free coloring goes back at least as far as to a paper of Chartrand et al. [11]. Many variants and generalizations of basic concepts have been introduced and intensively studied over the years. Since this subject is too wide to be surveyed in a short paper, we mention just a few examples like subcoloring known also as P_3-free coloring (where P_p denotes the chordless path on p vertices), P_4-free coloring and improper coloring, and we refer to appropriate literature on other variants, e.g., many results on subcoloring can be found in Albertson et al. [2], Broere and Mynhardt [8], Fiala et al. [16] as well as in work of Gimbel and Hartman [17]. For results on P_4-free coloring see, e.g., Gimbel and Nešetřil [18] and a paper of Hoàng and Le [23], while for improper coloring we refer the reader to papers of Bermond et al. [4], Cowen et al. [15] and Havet et al. [21]. Concerning the computational complexity of F-free k-coloring problem we mention the result of Achlioptas [1] who proved that for any fixed graph F, except K_2, the problem of deciding if a given graph admits an F-free coloring with at most k-colors is $\mathbb{NP}$-complete (for a detailed study of the computational complexity of many variants of offline generalized colorings see, e.g., Broersma et al. [9]).

As we have already mentioned, dynamic coloring can be seen as a generalization of offline and online colorings, but it is also closely related to the iterative coloring, in which the coloring algorithm is allowed to decide on improvements of subsequent solutions obtained by recoloring some vertices of a known graph (see, e.g., Caramia and Dell'Olmo [10], Molloy and Reed [26]). Since in iterative approach the graph is given in advance and the coloring algorithm can decide on which part of the solution to improve, the results on greedy dynamic F-free coloring seem to be applicable in the analysis and development of vertex recoloring rules for iterative greedy F-free coloring.

In context of online model special attention was paid to the analysis of colorings that can be obtained with the greedy algorithm. For proper coloring the largest number of colors used by the greedy algorithm is a well known graph invariant, denoted by $\Gamma(G)$ and called the *Grundy number* of a graph. In this sense, Goyal and Vishwanathan [20], and Zaker [28] determined the computational complexity of a long-standing open problem posed by Hedetniemi et al. [22] (cf. Jensen and Toft [25]). Namely, they proved that given a graph G and a positive integer k it is $\mathbb{NP}$-complete to decide if $\Gamma(G) \geq k$. Recently, their result has been extended by Borowiecki [6] who proved that for F-free coloring an analogous problem is $\mathbb{NP}$-complete for every $F = K_p$ with $p \geq 3$. Despite vast literature devoted to the Grundy number, we know only few graph classes for which the greedy algorithm always outputs optimal colorings (see, e.g., Borowiecki and Rautenbach [5], and Christen and Selkow [13] for more details). As of today, determining the Grundy number is known to be polynomial for P_4-laden graphs and trees (see Araujo et al. [3] and Hedetniemi et al. [22], respectively).

In this paper we continue the above-mentioned line of investigation by considering the classes of graphs for which a "dynamic variant" of the above-mentioned problem can be solved in polynomial time. More formally, let $\mathcal{F}$ denote the class of F-free graphs. The *dynamic $\mathcal{F}$-Grundy number* of a graph G, denoted by $\Gamma_d(G, F)$, is the largest number of colors that may be required by the greedy algorithm during some dynamic F-free coloring of G. The main problem considered in this paper can be stated as follows.

Problem 1.1 (GREEDY DYNAMIC F- FREE k- COLORING)

Input: A graph G.
Question: Does $\Gamma_d(G, F) \geq k$ hold for G?

The current knowledge on minimal forbidden subgraphs characterizing the classes of F-free k-colorable graphs is very far from being complete even if we consider offline proper coloring (in this sense, some recent results in selected classes of graphs, mainly for $k \in \{3, 4\}$, can be found in Chudnovsky et al. [14], Goedgebeur and Schaudt [19] and Hoàng et al. [24]). On the other hand, for every fixed $k \geq 2$ and connected graph F, the class of graphs F-free k-colorable with the greedy algorithm can be characterized by a finite number of minimal forbidden graphs (see, Borowiecki [6]) and hence the online variant of greedy F-free k-coloring can be solved in polynomial time. Despite the fact that the structure of minimal graphs for greedy dynamic F-free coloring seems to be more involved than in the case considered in [6], we strongly believe that Problem 1.1 can also be solved in polynomial time.

In this paper, we show that for every forbidden graph F and every integer $k \geq 1$ there exists an infinite number of F-free 2-colorable graphs G for which $\Gamma_d(G, F) > k$. Consequently, the performance ratio of the greedy algorithm cannot be bounded by any constant, even if we restrict the input to F-free 2-colorable graphs. This negative result motivates our investigations in the direction of characterizing the subclasses of F-free 2-colorable graphs for which the greedy algorithm uses no more than k colors. In Sect. 2, under the assumption that F is 2-connected, for every fixed $k \geq 1$ we characterize the class of F-trees for which the greedy algorithm uses at most k colors. This leads to a polynomial-time algorithm for Problem 1.1 for F-trees. Next, in Sect. 3, assuming that $F = P_3$, we prove analogous results for the class of all trees. For proper coloring of trees (recall $F = P_2$) the problem was solved in [7], while an intriguing problem for trees, when F is a tree of order greater than 3 constitutes an open problem presented in the last section of this paper.

2 Dynamic F-free Coloring, when F is 2-Connected

In this section, under the assumption that F is 2-connected, we consider GREEDY DYNAMIC F- FREE k- COLORING for F-trees.

In order to define F-trees we need some additional notions. Namely, by $G - v$ we denote the graph obtained from G by removal of vertex v. Given two disjoint graphs G_1, G_2 and vertices $v_1 \in V(G_1), v_2 \in V(G_2)$, we say that a graph G is obtained from G_1 and G_2 by the *identification* of v_1 and v_2 if the following hold: (i) $V(G) = (V(G_1)\setminus\{v_1\}) \cup V(G_2)$, and (ii) $E(G) = E(G_2) \cup E(G_1 - v_1) \cup \{v_2u \mid u$ was a neighbor of v_1 in $G_1\}$.

Definition 2.1 A single-vertex graph K_1 and the graph F are *F-trees*. Moreover, if G_1, G_2 are disjoint F-trees with vertices $v_1 \in V(G_1)$ and $v_2 \in V(G_2)$, then the graph obtained by the identification of v_1 and v_2 is an *F-tree*.

Clearly, if F is a tree (in the classical sense), then any F-tree is a tree. Concerning simple properties of F-trees note that if F is 2-connected, then every block (i.e., maximal 2-connected subgraph) of an F-tree is isomorphic to F. It is also not hard to see that independently of F, every F-tree admits an F-free coloring with at most two colors.

In what follows, since F is always clear from the context, it is usually omitted to simplify notation. In particular in this section we write k-forcing tree instead of k-forcing F-tree (this resembles omitting prefix hyper-, which is quite common practice in papers dealing with hypergraphs). Let $\eta = |V(F)|$.

Definition 2.2 Let $k \geq 1$. A *k-forcing tree* T_k is a rooted graph defined as follows:

(a) $T_1 = K_1$, while $T_2 = F$ with the root in any vertex,
(b) for $k \geq 3$, let T_{k-2} be a $(k-2)$-forcing tree and let $T^1_{k-1}, \dots, T^{\eta-1}_{k-1}$ denote arbitrary $(k-1)$-forcing trees such that $T^1_{k-1}, \dots, T^{\eta-1}_{k-1}, T_{k-2}$ are pairwise disjoint. A k-forcing tree T_k can be obtained by adding the edges between the roots of $T^1_{k-1}, \dots, T^{\eta-1}_{k-1}, T_{k-2}$ such that the graph H induced by the roots is isomorphic to F, and setting one of the roots of $T^1_{k-1}, \dots, T^{\eta-1}_{k-1}$ as the root of T_k.

For an illustration of a k-forcing tree T_k see Fig. 2a, where appropriate forcing trees are marked with solid lines. In order to prove minimality of forcing trees with respect to the dynamic $\mathcal{F}$-Grundy number we introduce the notion of a k-branch, which, despite significant differences between the cases in which F is 1- and 2-connected, allows a unified description of structural properties of k-forcing trees.

Definition 2.3 Let $k \geq 1$. A *k-branch* B_k is a rooted graph defined as follows:

(a) $B_1 = F$ with the root in any vertex,
(b) for $k \geq 2$, let T_1 be the 1-forcing tree and let T_{k-1} be a $(k-1)$-forcing tree. Moreover, let $T^2_k, \dots, T^{\eta-1}_k$ denote arbitrary k-forcing trees such that $T_1, T^2_k, \dots, T^{\eta-1}_k, T_{k-1}$ are pairwise disjoint. A k-branch B_k can be obtained by adding the edges between the roots of $T_1, T^2_k, \dots, T^{\eta-1}_k, T_{k-1}$ such that the graph H induced by the roots is isomorphic to F, and setting the root of T_1 as the root of B_k.

In Definition 2.2 (Definition 2.3) the subgraph H, induced by the roots, is called the *base* of k-forcing tree T_k (of k-branch B_k). Note that the root of any branch belongs to exactly one block of this branch. For an illustration see Fig. 2a, where B_{k-1} is marked with a dashed line.

Lemma 2.1 *Let $k \geq 2$ and for each $i < k$ let B_i be an i-branch. If we identify the roots of the branches $B_1, \dots, B_{k-1}$, then we obtain a k-forcing tree T_k.*

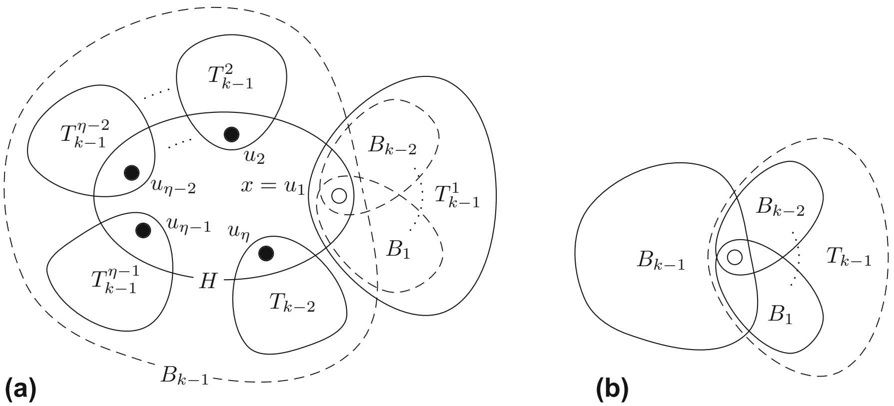

Fig. 2 The structure and branches of k-forcing tree T_k (the white vertex is the root)

Proof For the base step of the induction, observe that the assertion holds for $k = 2$. Now, assume $k > 2$ and that for all $i \leq k-1$ the identification of the roots of $B_1, \ldots, B_{i-1}$, results in an i-forcing tree T_i.

Let T be an F-tree obtained by the identification of the roots of $B_1, \ldots, B_{k-1}$. By the induction hypothesis, $B_1, \ldots, B_{k-2}$ form a $(k-1)$-forcing tree, say T' (see Fig. 2b). Now, consider the base H of B_{k-1} and observe that the root x of B_{k-1} is the root of T'. By the definition of $(k-1)$-branch there is a vertex of H (different from x) that is the root of a $(k-2)$-forcing tree, say T'', while the remaining $\eta - 2$ vertices of H are the roots of $(k-1)$-forcing trees $T^2_{k-1}, \ldots, T^{\eta-1}_{k-1}$ (see Fig. 2a). Clearly, the trees $T', T^2_{k-1}, \ldots, T^{\eta-1}_{k-1}, T''$ are disjoint, and since their roots form the vertex set of H, by Definition 2.2 we conclude that T is a k-forcing tree T_k. □

Lemma 2.2 *Let $k \geq 2$ and for each $i < k$ let B_i be an i-branch. Every k-forcing tree T_k can be obtained by the identification of the roots of the branches $B_1, \ldots, B_{k-1}$.*

Proof For the base step of the induction, observe that if $k = 2$, then the assertion follows from the definitions of 2-forcing tree and 1-branch. Assume $k > 2$ and that the assertion holds for all $i \leq k-1$.

Let T be a k-forcing tree with the root x and the base H. By the definition of k-forcing tree, x is the root of the $(k-1)$-forcing tree T' that does not contain H. Using the same definition, let $H_1, \ldots, H_{\eta-1}$ denote $\eta - 1$ forcing trees obtained from T by removal of T' and all edges of H. By the induction hypothesis, we see that T' can be obtained by the identification of the roots of the branches $B_1, \ldots, B_{k-2}$. Now, the assertion follows from the observation that H and $H_1, \ldots, H_{\eta-1}$ form B_{k-1}, and that by Lemma 2.1 the branches $B_1, \ldots, B_{k-1}$ form T_k. □

In what follows we need a slightly deeper insight into the structure of k-forcing trees and their branches.

Property 2.1 *If T_k is a k-forcing tree with the root x, then for each $i \in \{1, \ldots, k\}$ the k-forcing tree T_k contains some i-forcing tree T_i with the vertex x as its root.* □

Property 2.2 *If B_k is a k-branch with the root x, then for each $i \in \{1, \ldots, k\}$ the branch B_k contains some i-branch B_i with the vertex x as its root.* □

Let k be a positive integer and let $F_1, \ldots, F_k$ be disjoint rooted copies of a given graph F and let $x_1, \ldots, x_k$ be their roots, respectively. An (F, k)-*star* with the *center* x and *rays* $F_1, \ldots, F_k$ is a graph obtained from the graphs $F_1, \ldots, F_k$ by the identification of their roots and setting the resulting common vertex as the center x. Similarly as for k-forcing trees, F is always clear from the context and hence it is omitted. Consequently, an (F, k)-star is briefly called a k-star, and it is denoted by $S_k(x)$.

For an uncolored vertex v we say that colored vertex u is *fixed colored with respect to* v if Presenter will never discolor u before v becomes colored. To express this fact we write $u \prec v$. We also say that $S_k(v)$ is k-*precolored* if its center v is not colored but for each ray F_i all vertices in $V(F_i)\backslash\{v\}$ are already fixed colored i with respect to v. Observe that the greedy algorithm uses color k for a vertex v if and only if v is the center of a $(k-1)$-precolored star $S_{k-1}(v)$. Thus we have the following property.

Property 2.3 *For every $k > 1$ Presenter has a strategy that forces the greedy algorithm to use color k for vertex v of a graph G if and only if G contains $S_{k-1}(v)$ and Presenter has a strategy (realized on G) that forces the greedy algorithm to produce a $(k-1)$-precoloring of $S_{k-1}(v)$.* □

Lemma 2.3 *Let F be 2-connected. For every k-forcing tree T_k, we have*

$$\Gamma_d(T_k, F) \geq k \ .$$

Proof The proof is by induction on k. It is easy to check that the assertion holds for $k \leq 2$. Assume $k > 2$ and that our lemma holds for all T_i with $i \leq k-1$.

Following the notation introduced in the definition of k-forcing trees, let $u_1, \ldots, u_{\eta-1}$ denote the roots of $T_{k-1}^1, \ldots, T_{k-1}^{\eta-1}$ and let u_η stand for the root of T_{k-2}. Moreover, set $x = u_1$, and let x be the root and H the base of T_k (for an illustration see Fig. 2a).

First, we prove that T_k contains a $(k-1)$-forcing tree (we denote it by T') with the root u_η. From Property 2.1 it follows that $T_{k-1}^1, \ldots, T_{k-1}^{\eta-2}$ contain $T_{k-2}^1, \ldots, T_{k-2}^{\eta-2}$ with the roots $u_1, \ldots, u_{\eta-2}$, respectively, and that $T_{k-1}^{\eta-1}$ contains $(k-3)$-forcing tree T'' with the root $u_{\eta-1}$. Naturally, the trees $T_{k-2}^1, \ldots, T_{k-2}^{\eta-2}, T_{k-2}, T''$ are disjoint and since their roots are the vertices of H, a $(k-1)$-forcing tree T' with the root u_η is formed. Hence, by the induction hypothesis and Property 2.3 Presenter has a strategy that forces the greedy algorithm to use color $k-1$ for u_η. Now, without loss of generality we may assume that Presenter discolors all vertices, except the vertex u_η. Let u_η be fixed colored $k-1$ with respect to x, i.e., $u_\eta \prec x$ (see Fig. 3a for an example with $F = K_3$, $k = 4$ and vertex u_η colored 3).

Next, since $u_2, \ldots, u_{\eta-1}$ are the roots of disjoint $(k-1)$-forcing trees $T_{k-1}^2, \ldots, T_{k-1}^{\eta-1}$, by the induction hypothesis it follows that for each of them Presenter has a strategy of forcing color $k-1$ in the root (note that as long as x remains uncolored, colors of the vertices of the base H cannot influence each other). Hence,

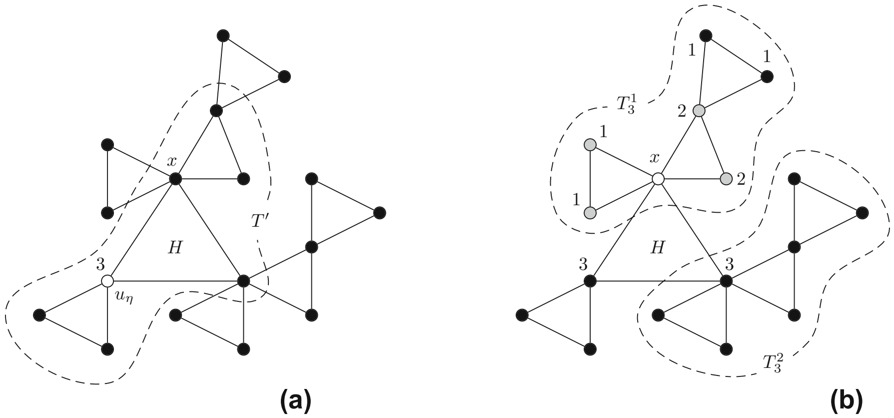

Fig. 3 An example of forcing color 4 in the root x of T_k, where $F = K_3$, $k = 4$ and $\eta = 3$

assume that $u_2, \ldots, u_{\eta-1}$ are fixed colored $k-1$ with respect to x, i.e., $u_j \prec x$ for all $j \in \{2, \ldots, \eta-1\}$ (see Fig. 3b for an illustration of T_3^2 with the root colored 3).

By the induction hypothesis, since x is the root of T_{k-1}^1, Presenter has a strategy forcing the greedy algorithm to use color $k-1$ for x. By Property 2.3 this is equivalent to the existence of a $(k-2)$-precolored star $S_{k-2}(x)$, contained in T_{k-1}^1, and sharing a single vertex with H whose vertices (except x) are already colored $k-1$ (see Fig. 3b for an illustration of T_3^1 with gray circles denoting precolored vertices of $S_2(x)$; recall $k = 4$). Thus, the base H and $S_{k-2}(x)$ form a $(k-1)$-precolored star $S_{k-1}(x)$. Consequently, by the induction hypothesis and Property 2.3, this implies the existence of a strategy that forces the greedy algorithm to use color k for x. ◻

Theorem 2.1 *Let F be 2-connected and let $k \geq 1$. If T is an F-tree, then $\Gamma_d(T, F) \geq k$ if and only if T contains a k-forcing tree T_k.*

Proof By Lemma 2.3 we have $\Gamma_d(T_k, F) \geq k$, and since $T_k \leq T$, we get $\Gamma_d(T, F) \geq k$. It is not hard to see that the theorem holds for $k \leq 2$. Assume $k > 2$ and that the assertion holds for all T_i with $i \leq k-1$.

If $\Gamma_d(T, F) \geq k$, then Presenter has a dynamic presentation that forces the greedy algorithm to assign color k to some vertex x of T. Hence, by Property 2.3, just before x gets colored k the tree T contains a $(k-1)$-precolored star $S_{k-1}(x)$. Let $H_1, \ldots, H_{k-1}$ be the rays of $S_{k-1}(x)$ and for all $i \leq k-1$ let $u_1^i, \ldots, u_\eta^i$ denote the vertices of H_i with $u_1^i = x$. Recall that because $S_{k-1}(x)$ is $(k-1)$-precolored, for each ray H_i all vertices in $V(H_i) \setminus \{x\}$ are fixed colored i with respect to x (in particular $u_j^i \prec x$, $j \in \{2, \ldots, \eta\}$).

Consider an arbitrary ray H_i and without loss of generality assume that u_η^i was fixed colored with respect to all other vertices of H_i, i.e., $u_\eta^i \prec u_j^i$ for $j \in \{1, \ldots, \eta-1\}$. Since u_η^i is colored i, by the induction hypothesis T contains an i-forcing tree T_i' rooted at u_η^i. Moreover, from Lemma 2.2 it follows that T_i' contains all branches $B_1, \ldots, B_{i-1}$ rooted at u_η^i. Recall that the root of each branch belongs to exactly one block (the base of branch). Let $R_1, \ldots R_{i-1}$ be the bases of the branches $B_1, \ldots, B_{i-1}$, respectively. We claim that either $V(B_j) \cap V(H_i) = \{u_\eta^i\}$ or $V(B_j) \cap V(H_i) = V(H_i)$ for all $j \in$

$\{1, \ldots, i-1\}$. Suppose that there is R_j such that $2 \leq |V(R_j) \cap V(H_i)| \leq \eta - 1$. Now, observe that no η-element superset of $V(R_j) \cap V(H_i)$ distinct from $V(H_i)$ can induce a 2-connected graph. Since R_j is isomorphic to F, this contradicts 2-connectedness of F. Thus, at most one of the bases, and consequently at most one of the branches $B_1, \ldots, B_{i-1}$ contains H_i. Assume that this is the branch B_{i-1}. Consequently, T_i' contains branches $B_1, \ldots, B_{i-2}$ all of which share only vertex u_η^i with H_i. Thus, by Lemma 2.2 vertex u_η^i is the root of an $(i-1)$-forcing tree T_{i-1}' that contains no vertex of $H_i - u_\eta^i$. Now, consider vertices u_j^i with $j \in \{2, \ldots, \eta - 1\}$ and recall that they are fixed colored i with respect to u_1^i. Hence, by the induction hypothesis, each u_j^i is the root of an i-forcing tree T_i^j. Moreover, since among all vertices of H_i, the vertex u_η^i was colored first, Presenter cannot use u_η^i, when forcing Painter to use color i for u_j^i with $j \neq \eta$. Consequently, for each $j \in \{2, \ldots, \eta - 1\}$ the i-forcing tree T_i^j must be disjoint from $H_i - u_j^i$. Hence $T_i^2, \ldots, T_i^{\eta-1}$ and T_{i-1}', rooted at $u_2^i, \ldots, u_{\eta-1}^i$ and u_η^i, respectively, are pairwise disjoint and their roots together with u_1^i induce H_i. Thus, by Definition 2.3 we get an i-branch B_i with the root $u_1^i = x$ and the base H_i.

Since i was selected arbitrarily, we conclude that for each $i \in \{1, \ldots, k-1\}$ the tree T contains an i-branch with the root x and base H_i, and since by Lemma 2.1 a subgraph obtained by the identification of the roots of the branches $B_1, \ldots, B_{k-1}$ is a k-forcing tree T_k, the F-tree T contains T_k. □

3 Dynamic Subcoloring

In this section we consider GREEDY DYNAMIC F-FREE k-COLORING when $F = P_3$. The results in this section are of more general character than those of Sect. 2, in the sense that instead of considering our problem for F-trees we solve it for the class of all trees. As previously, with a small abuse of notation we usually omit F.

Definition 3.1 Let $k \geq 1$. A k-forcing tree T_k is a rooted graph defined as follows:

(a) $T_1 = K_1$, while $T_2 = P_3$ with the root in any vertex,
(b) for $k \geq 3$, let T_{k-1} be an arbitrary $(k-1)$-forcing tree and let T_{k-2}^1, T_{k-2}^2 denote arbitrary $(k-2)$-forcing trees such that $T_{k-2}^1, T_{k-2}^2, T_{k-1}$ are pairwise disjoint. Moreover, let x_1, x_2, x_3 denote the roots of T_{k-2}^1, T_{k-2}^2 and T_{k-1}, respectively. A k-forcing tree T_k can be obtained by adding the edges between x_1, x_2, x_3 such that the graph H induced by x_1, x_2, x_3 is isomorphic to P_3, and setting x_3 as the root of T_k,
(c) for $k \geq 4$, let T_{k-3} be an arbitrary $(k-3)$-forcing tree and let T_{k-1}^1, T_{k-1}^2 denote arbitrary $(k-1)$-forcing trees such that $T_{k-1}^1, T_{k-1}^2, T_{k-3}$ are pairwise disjoint. Moreover, let x_1, x_2, x_3 denote the roots of T_{k-1}^1, T_{k-3} and T_{k-1}^2, respectively. A k-forcing tree T_k can be obtained by adding the edges x_1x_2, x_2x_3 and setting either x_1 or x_3 as the root of T_k.

Note that Definition 3.1(b), in contrast to (c) allows x_3 in the middle as well as at the end of P_3 (see the upper row in Fig. 4).

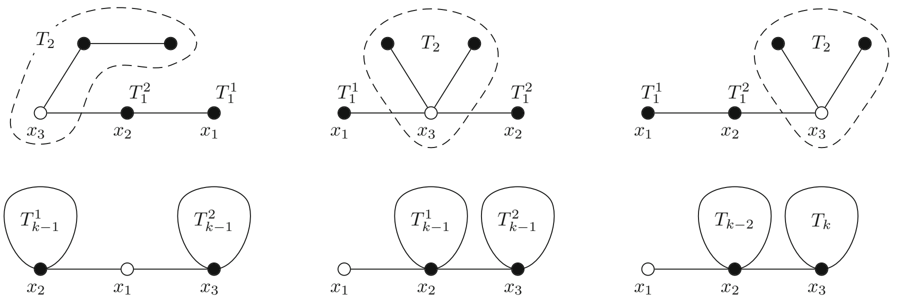

Fig. 4 Dynamic subcoloring: all k-forcing trees for $k = 3$ (upper row), and k-branches for $k \geq 3$ (bottom row)

Definition 3.2 Let $k \geq 1$. A k-branch B_k is a graph defined as follows:

(a) $B_1 = B_2 = P_3$ with the root in any vertex,

(b) if $k \geq 3$, then B_k can be obtained by adding the edges between the roots x_1, x_2, x_3 of three pairwise disjoint forcing trees:

  (b1) $T_1, T_{k-1}^1, T_{k-1}^2$ (where T_1 is the 1-forcing tree and T_{k-1}^1, T_{k-1}^2 denote arbitrary $(k-1)$-forcing trees) such that the graph H induced by the roots is isomorphic to P_3, or

  (b2) T_1, T_{k-2}, T_k (where T_1 is the 1-forcing tree, T_{k-2} is an arbitrary $(k-2)$-forcing tree and T_k is an arbitrary k-forcing tree) such that the graph H induced by the roots is isomorphic to P_3 with x_2 as the middle vertex,

then setting x_1 as the root.

Note that in the above definition, in contrast to Definition 2.3, the root of a branch may belong to more than one subgraph isomorphic to F and contained in the branch. Analogously as in previous section, in Definition 3.1 (Definition 3.2) the subgraph H, induced by the roots, is called the *base* of k-forcing tree (of k-branch).

To keep our paper concise, instead of proving basic lemmas for $F = P_3$ we shortly remark on similarities to their counterparts in Sect. 2. To see that an analogue of Lemma 2.1 holds for $F = P_3$ it is enough to observe that if H is the base, then by Definition 3.2, the vertices in $V(H)\setminus\{x\}$ are either the roots of two $(k-1)$-forcing trees or one k-forcing tree and one $(k-2)$-forcing tree, instead of $\eta - 1$ forcing trees as in Definition 2.3. Also note that we need no changes in the proof of Lemma 2.2. Similarly, it is not hard to see that if $F = P_3$, then Properties 2.1, 2.2 and 2.3 can be applied directly. On the other hand, the assumption on 2-connectivity of F in the proof of Theorem 2.1 is crucial. Similarly, Presenter's strategy given in the proof of Lemma 2.3 has to be modified. Therefore, our main results for subcoloring, stated as Lemma 3.1 and Theorem 3.1, need their own proofs.

Lemma 3.1 *For every k-forcing tree T_k, we have*

$$\Gamma_d(T_k, P_3) \geq k\ .$$

Proof It is not hard to see that our lemma holds for $k \leq 3$. Recall that by Lemma 2.2 every k-forcing tree T_k with $k \geq 2$ can be obtained by the identification of the roots of branches $B_1, \dots, B_{k-1}$. Let x be the root of T_k and let u_1, u_2 and x denote the vertices of the base of B_{k-1}. We prove that Presenter has a strategy that forces the greedy algorithm to use color k for the vertex x. In fact, we prove even more, i.e., we show that there exists a strategy resulting in a coloring such that for each branch B_i with $i \in \{1, \dots, k-1\}$ all vertices of the base H_i of B_i (except x) are colored i, which clearly implies the existence of a $(k-1)$-forcing star $S_{k-1}(x)$ with the rays $H_1, \dots, H_{k-1}$ that allows forcing color k in x.

The above assertion easily holds for small values of k. Now, let us assume that it holds for all B_i with $i < k-1$ and that our lemma holds for all T_i with $i \le k-1$. In what follows we argue that Presenter can force color $k-1$ in u_1 and u_2 of B_{k-1} and that both vertices can be set as fixed colored with respect to the vertex x in which Presenter is going to force color k.

Case 1 Let T_k be a k-forcing tree as in Definition 3.1(c), and let u_1, u_2 and x be the roots of forcing trees T_{k-3}, T^1_{k-1} and T^2_{k-1}, respectively (see Fig. 5a for an example with $k = 5$). Clearly, according to the definition $u_1u_2, u_1x \in E(T_k)$ and $k \ge 4$ while from Lemmas 2.1 and 2.2 we see that T_{k-3} and T^1_{k-1} are contained in B_{k-1}.

First, we argue that Presenter can force the greedy algorithm to use color $k-1$ for u_1 (see Fig. 5b for an illustration). Recall that $B_1, \dots, B_{k-2}$ are the branches that form T^2_{k-1}. By the induction hypothesis, Presenter has a strategy that can be played on T^2_{k-1} to force color $k-2$ in non-root vertices of the base H_{k-2} of B_{k-2}. Without loss of generality let v be a vertex of H_{k-2} that is adjacent to x and assume that after coloring v with color $k-2$ Presenter discolors all vertices of T^2_{k-1} except v.

Since $B_1, \dots, B_{k-3}$ form a $(k-2)$-forcing tree (we denote it by T') with the root x, by the induction hypothesis Presenter has a strategy that can be played on T' to force color $k-2$ in the vertex x (note that $v \notin V(T')$). From now on, assume that v and x are fixed colored $k-2$ with respect to u_1, i.e., $v \prec u_1$ and $x \prec u_1$ (see gray vertices in the figure).

Next, let $B'_1, \dots, B'_{k-2}$ be the branches that form T^1_{k-1} and let v' be a neighbor of u_2 in the base of B'_{k-3}. Similarly as above, by the induction hypothesis Presenter can easily use the $(k-3)$-forcing tree formed by $B'_1, \dots, B'_{k-2}$ (we denote it by T'') to force color $k-3$ in v' and u_2. Let $v' \prec u_1$ and $u_2 \prec u_1$.

Assume first that $k \ge 5$. Since by the induction hypothesis Presenter has a strategy that can be played on T_{k-3} to force color $k-3$ in u_1, we know that just before coloring u_1 there is a $(k-4)$-precolored star $S_{k-4}(u_1)$. Moreover, $S_{k-4}(u_1)$ and the subgraphs induced by the already colored vertices v, x and v', u_2 form a $(k-2)$-precolored star $S_{k-2}(u_1)$ (see Fig. 5b for $S_3(u_1)$ with precolored vertices represented by gray circles). On the other hand if $k = 4$ it is enough to observe that u_1 together with the precolored vertices v, x and v', u_2 form a 2-precolored star $S_2(u_1)$. This implies the existence of a strategy of forcing color $k-1$ in u_1. Let u_1 be fixed colored $k-1$ with respect to x.

Now, assume that u_1 is the only colored vertex of T_k (recall that u_1 does not belong to $V(T^1_{k-1})$). Since u_2 is the root of T^1_{k-1}, by the induction hypothesis it follows that u_2 can be colored $k-1$. Thus, u_1 and u_2 can be fixed colored $k-1$ with respect to x, i.e., $u_1 \prec x$ and $u_2 \prec x$.

Case 2 Let T_k be a k-forcing tree as in Definition 3.1(b), and let let u_1, u_2 and x be the roots of forcing trees T^1_{k-2}, T^2_{k-2} and T_{k-1}, respectively (see Fig. 6a for an

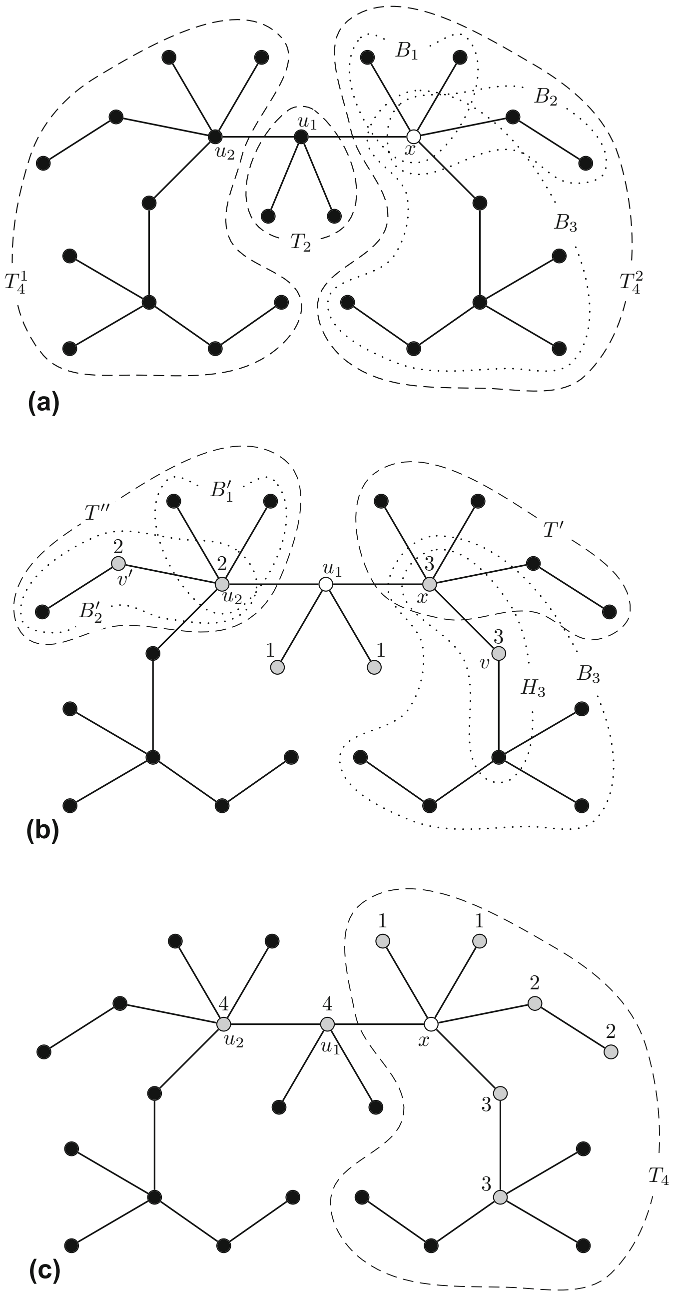

Fig. 5 An illustration for Case 1 in the proof of Lemma 3.1: **a** a 5-forcing tree T_5 [see Definition 3.1(c)]; **b** forcing color 4 in u_1; **c** forcing color 5 in x

example with $k = 4$). Since our lemma holds for small k, we assume that $k \geq 4$. By Lemmas 2.1 and 2.2 it follows that T^1_{k-2} and T^2_{k-2} are contained in B_{k-1}. Without loss of generality we can assume that u_1 is a pendant vertex of the base H_{k-1} of B_{k-1}. Hence, according to the definition of T_k, we have $u_2 x \in E(T_k)$. Consequently, either x or u_2 is the second pending vertex of H_{k-1}.

First, we argue that Presenter has a strategy of forcing the greedy algorithm to use color $k-1$ for u_1 (see Fig. 6b for an illustration). By Property 2.1 both T_{k-1} and T^2_{k-2} contain $(k-3)$-forcing tree T' rooted at x and T'' with the root u_2, respectively. Since the roots of T', T'' and T^1_{k-2} are exactly the vertices of H_{k-1}, by Definition 3.1(b) the base H_{k-1} and the above trees form a $(k-1)$-forcing tree, say T'_{k-1}, with root u_1. Hence, by the induction hypothesis there exists Presenter's strategy forcing color $k-1$ in u_1. Let u_1 be fixed colored $k-1$ with respect to x.

Now, assume that u_1 is the only colored vertex of T_k. We argue that Presenter can force color $k-1$ in vertex u_2 (see Fig. 6c). Similarly as in Case 1, by the induction hypothesis Presenter can easily use the tree T_{k-1} formed by $B_1, \ldots, B_{k-2}$ to force color $k-2$ in the root x and its neighbor v in B_{k-2}. Let x and v be fixed colored $k-2$ with respect to u_2, i.e., $x \prec u_2$ and $v \prec u_2$. Naturally, $u_2 \notin V(T_{k-1})$ and since u_2 is the root of T^2_{k-2}, by the induction hypothesis Presenter can force color $k-2$ in u_2. Following Property 2.3, this implies that T^2_{k-2} contains a $(k-3)$-precolored star $S_{k-3}(u_2)$. Observe that $S_{k-3}(u_2)$ together with the subgraph induced by v, x, u_2 forms a $(k-2)$-forcing star $S_{k-2}(u_2)$, which by Property 2.3 implies the possibility of forcing color $k-1$ in u_2 (see Fig. 6c for $S_2(u_2)$ with precolored vertices represented by gray circles). Thus u_1 and u_2 can be fixed colored $k-1$ with respect to x, i.e., $u_1 \prec x$ and $u_2 \prec x$.

Finally, observe that in both cases x is the root of a $(k-1)$-forcing tree T_{k-1} containing neither u_1 nor u_2 (see Figs. 5c and 6d) and hence by the induction hypothesis Presenter can use this tree to realize his strategy of forcing color $k-1$ in x. This, in turn, implies the existence of a $(k-2)$-precolored star $S_{k-2}(x)$ such that $u_1, u_2 \notin V(S_{k-2}(x))$. The star $S_{k-2}(x)$, together with the subgraph induced by u_1, u_2 and x, form a $(k-1)$-precolored star $S_{k-1}(x)$, which clearly implies $\Gamma_d(T_k, P_3) \geq k$. □

Theorem 3.1 *If T is a tree, then $\Gamma_d(T, P_3) \geq k$ if and only if T contains a k-forcing tree T_k.*

Proof By Lemma 3.1 we have $\Gamma_d(T_k, P_3) \geq k$, and since $T_k \leq T$, we get $\Gamma_d(T, P_3) \geq k$. If $\Gamma_d(T, P_3) \geq k$, then Presenter has a dynamic presentation that forces the greedy algorithm to assign color k to some vertex x of T. Hence, by Property 2.3 before x gets colored k the tree T contains a $(k-1)$-precolored star $S_{k-1}(x)$. Let $H_1, \ldots, H_{k-1}$ be the rays of $S_{k-1}(x)$ and for all $i \leq k-1$ let u^i_1, u^i_2, u^i_3 denote the vertices of H_i with $u^i_1 = x$. Since $S_{k-1}(x)$ is $(k-1)$-precolored, for each ray H_i the vertices u^i_2 and u^i_3 are fixed colored i with respect to x, i.e., $u^i_j \prec x$, $j \in \{2, 3\}$. In what follows we assume that u^i_2 becomes fixed colored before u^i_3, i.e., $u^i_2 \prec u^i_3$.

We continue by proving that if Presenter has a strategy to force color k in x, then T contains a k-forcing tree T_k with the root x, and T_k can be obtained by the identification

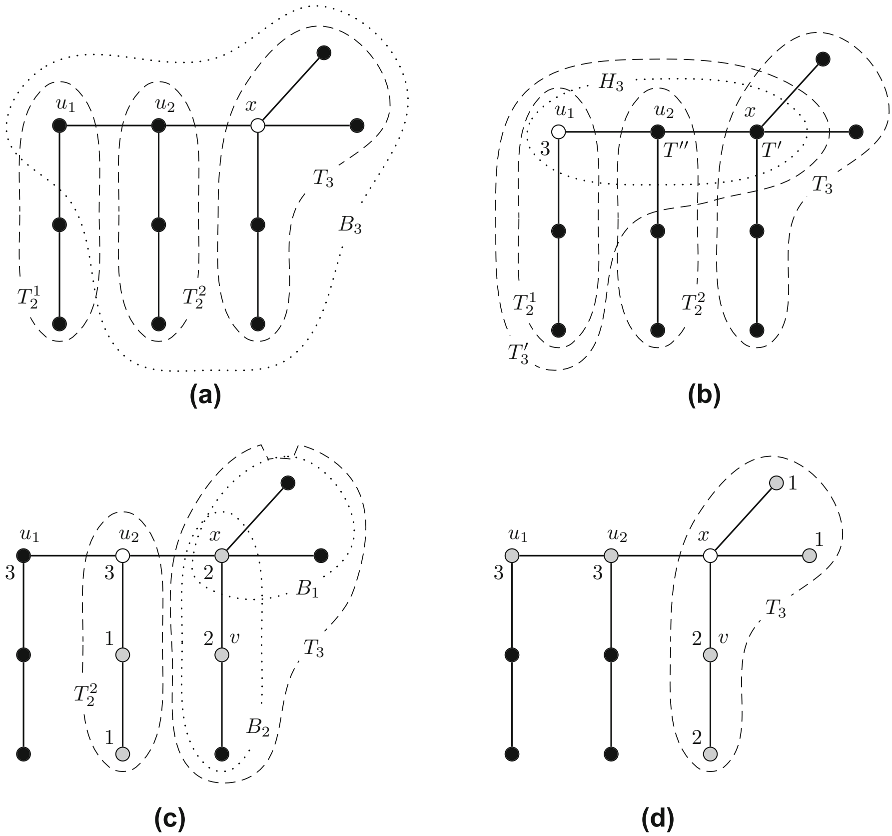

Fig. 6 An illustration for Case 2 in the proof of Lemma 3.1: **a** a 4-forcing tree T_4 [Definition 3.1(b)]; **b** forcing color 3 in u_1; **c** forcing color 3 in u_2; **d** forcing color 4 in x

of the roots of i-branches $B_i, i \in \{1, \ldots, k-1\}$ performed in such a way that the rays of $S_{k-1}(x)$ are the bases of the appropriate branches. It is not hard to see that the theorem holds for $k \leq 3$ (see, e.g., Fig. 4). Assume $k > 3$ and that the above assertion holds for all T_i with $i \leq k-1$.

Considering an arbitrary ray H_i of $S_{k-1}(x)$ we prove that H_i is the base of an i-branch. Clearly, H_1 and H_2 are 1- and 2-branches, respectively. Let $i \geq 3$.

Since u_2^i is fixed colored i, by our assumptions we know that Presenter has a strategy of forcing color i in u_2^i which results in $(i-1)$-precolored star $S_{i-1}(u_2^i)$. Moreover, by the induction hypothesis T contains an i-forcing tree T_i' with the root u_2^i, and T_i' can be obtained by the identification of the roots of j-branches B_j, $j \in \{1, \ldots, i-1\}$ such that the rays of $S_{i-1}(u_2^i)$ are the bases of appropriate branches. An analogous observation holds for vertex u_3^i for which there is a tree T_i'' rooted at u_3^i and obtained by the identification of the roots of appropriate j-branches with the rays of $S_{i-1}(u_3^i)$ as their bases. In what follows we analyze the relationships between the branches that form T_i' and T_i''. Recall that $u_2^i \prec u_3^i$.

Case 1 First, assume that u_2^i is a pendant vertex of H_i. Note that because all branches $B_1, \ldots, B_{i-1}$ that form T_i' are rooted at u_2^i, at most one of them may contain vertex u_3^i. Following Property 2.2 assume that this is the branch B_{i-1}. However, even under such assumption, T_i' still contains branches $B_1, \ldots, B_{i-2}$ that do not contain u_3^i. Hence, by Lemma 2.2 the vertex u_2^i is the root of the $(i-1)$-forcing tree T_{i-1}' such that $T_{i-1}' \leq T_i'$ and $u_3^i \notin V(T_{i-1}')$. Now, consider u_3^i. If u_3^i is a pendant vertex of H_i, i.e., $u_3^i x, u_2^i x \in E(T)$, then similarly as above we can argue that u_3^i is the root of an $(i-1)$-forcing tree T_{i-1}'' such that $T_{i-1}'' \leq T_i''$ and $u_2^i \notin V(T_{i-1}'')$. Assume that u_3^i is not pendant, i.e., $xu_3^i, u_3^i u_2^i \in E(T)$. Since $u_2^i \prec u_3^i$, Presenter cannot use u_2^i when forcing color i in u_3^i. Therefore, the $(i-1)$-precolored star $S_{i-1}(u_3^i)$ does not contain u_2^i and hence at most one of the branches that form T_i'' may contain x. Again, by Property 2.2 the tree T_i'' contains branches $B_1, \ldots, B_{i-2}$ that do not contain u_2^i. Hence, by Lemma 2.2 vertex u_3^i is the root of an $(i-1)$-forcing tree T_{i-1}'' such that $T_{i-1}'' \leq T_i''$ and $u_2^i \notin V(T_{i-1}'')$.

Thus, independently of whether u_3^i is a pendant vertex of H_i or not, the trees T_{i-1}' and T_{i-1}'' are disjoint, and it follows that H_i is the base of an i-branch of the type described in Definition 3.2(b1).

Case 2 Next, assume that u_2^i is not a pendant vertex of H_i, i.e. $u_2^i x, u_2^i u_3^i \in E(T)$. Note that since u_2^i has two independent neighbors in H_i and all branches $B_1, \ldots, B_{i-1}$ that form T_i' are rooted at u_2^i, at most one of the branches may contain x and at most one of them may contain u_3^i. Hence, from Property 2.2 it follows that T_i' contains branches $B_1, \ldots, B_{i-3}$ containing neither u_3^i nor x. Thus, by Lemma 2.2 the vertex u_2^i is the root of an $(i-2)$-forcing tree T_{i-2}' such that $T_{i-2}' \leq T_i'$, $u_3^i \notin V(T_{i-2}')$ and $x \notin V(T_{i-2}')$. It remains to consider the vertex u_3^i. Since u_3^i is a pendant vertex of H_i and $u_2^i \prec u_3^i$, the i-precolored star $S_i(u_3^i)$ contains neither u_2^i nor x. Hence, none of the two vertices belongs to $V(T_i'')$.

Thus, the trees T_{i-2}' and T_i'' are disjoint, and it follows that H_i is the base of the i-branch of the type described in Definition 3.2(b2).

Since i was selected in an arbitrary manner, we conclude that for each $i \in \{1, \ldots, k-1\}$ the tree T contains an i-branch with the root x and base H_i. Clearly, by Lemma 2.1 a subgraph obtained by the identification of the roots of the branches $B_1, \ldots, B_{k-1}$ is a k-forcing tree. Hence, T contains T_k. □

4 Conclusions

In the preceding sections we considered greedy dynamic F-free coloring under the assumption that F is either 2-connected or P_3. In both cases, from the constructions of k-forcing F-trees it easily follows that for the fixed F and $k \geq 1$ the number of k-forcing F-trees is finite. Putting this together with Theorems 2.1 and 3.1 leads to the following result.

Theorem 4.1 *Let $k \geq 1$ be a fixed integer and let F be a fixed graph. If F is 2-connected, then the problem* GREEDY DYNAMIC F-FREE k-COLORING *admits a* *polynomial-time algorithm for F-trees, and it admits such an algorithm for trees when $F = P_3$.*

Proof From the definitions of k-forcing trees it easily follows that the order of any k-forcing tree is at most $|V(F)|^{k-1}$. Since for a given graph G of order n checking if a graph H of order p is an induced subgraph of G can be done by a simple brute force algorithm in time $O(n^p)$ and the number of k-forcing F-trees that we have to consider is finite, it follows that checking if $\Gamma_d(T, F) \geq k$ can be done in time $O(n^{|V(F)|^{k-1}})$. □

Given two graphs G_1 and G_2 (not necessarily disjoint) by $G_1 \cup G_2$ we denote the graph with the vertex set $V(G_1) \cup V(G_2)$ and edge set $E(G_1) \cup E(G_2)$.

Definition 4.1 Let G be a graph and let $\{G_1, \ldots, G_p\}$ be the set of induced subgraphs of G such that $G_i = F$ for all $i \in \{1, \ldots, p\}$. The graph $B = G_1 \cup \cdots \cup G_p$ is called the *backbone* of G.

Theorem 4.2 *If G is a graph with backbone B, then $\Gamma_d(G, F) = \Gamma_d(B, F)$.*

Proof We first observe that every vertex $v \in V(G)\setminus V(B)$ is F-isolated, i.e., G does not contain an induced subgraph H such that $H = F$ and $v \in V(H)$. Hence, for every vertex in $V(G)\setminus V(B)$, independently of when it is colored, the greedy algorithm will always use color 1. Moreover, the color of an F-isolated vertex cannot influence the colors of other vertices. Thus $\Gamma_d(G, F) \leq \Gamma_d(B, F)$. On the other hand, since $B \leq G$, we have $\Gamma_d(G, F) \geq \Gamma_d(B, F)$. □

Let $\mathcal{B}_F$ be an arbitrary class of graphs for which Problem 1.1 admits a polynomial-time solution, and let $\mathcal{G}_F$ be the class of graphs with backbones in $\mathcal{B}_F$. By arguments similar to those used in the proof of Theorem 4.1, for any fixed F and k appropriate backbone can be determined in polynomial-time. Together with Theorems 4.1 and 4.2 this leads to the following corollary.

Corollary 4.1 *If k is a fixed positive integer and F is a fixed graph, then the problem* GREEDY DYNAMIC F-FREE k-COLORING *is polynomial-time solvable for graphs in $\mathcal{G}_F$.*

Consider Problem 1.1 with inputs in the class of classical trees. In this context we already know polynomial-time algorithms when $F \in \{K_2, P_3\}$ (see [7] and Sect. 3, respectively). Hence, a natural question to ask is about similar results for F being a tree of order at least 4. One of the possibilities is to forbid paths P_p with $p \geq 4$. Another natural extension is to forbid certain monochromatic stars, which for trees comes down to an *m-improper coloring*, i.e., a partition of the vertex set in which each color class induces a graph of maximum degree at most m (in other words, no vertex x has more than m neighbors colored with the same color as x). In the class of trees, an m-improper coloring is equivalent to $K_{1,m+1}$-free coloring, where $K_{1,m+1}$ denotes the star with $m + 1$ rays.

Problem 4.1 Let $F \in \{P_p, K_{1,m+1}\}$ where $p \geq 4$ and $m \geq 2$, and let $k \geq 1$ be a fixed integer. For a given tree T decide whether $\Gamma_d(T, F) \geq k$.

Acknowledgements We would like to acknowledge the anonymous referees for their valuable comments that helped us to improve the readability of the paper.

References

1. Achlioptas, D.: The complexity of G-free colourability. Discret. Math. **165**(166), 21–30 (1997)
2. Albertson, M.O., Jamison, R.E., Hedetniemi, S.T., Locke, S.C.: The subchromatic number of a graph. Discret. Math. **74**, 33–49 (1989)
3. Araujo, J., Linhares-Sales, C.: On the Grundy number of graphs with few P_4's. Discret. Appl. Math. **160**, 2514–2522 (2012)
4. Bermond, J.-C., Havet, F., Huc, F., Linhares-Sales, C.: Improper coloring of weighted grid and hexagonal graphs. Discret. Math. Algorithm Appl. **2**, 395–412 (2010)
5. Borowiecki, P., Rautenbach, D.: New potential functions for greedy independence and coloring. Discret. Appl. Math. **182**, 61–72 (2015)
6. Borowiecki, P.: Computational aspects of greedy partitioning of graphs. J. Comb. Optim. **35**(2), 641–665 (2018). https://doi.org/10.1007/s10878-017-0185-2
7. Borowiecki, P., Sidorowicz, E.: Dynamic coloring of graphs. Fund. Inf. **114**, 105–128 (2012)
8. Broere, I., Mynhardt, C.M.: Generalized Colorings of Graphs. Graph Theory with Applications to Algorithms and Computer Science, (Kalamazoo, 1984). Wiley, New York (1985)
9. Broersma, H., Fomin, F., Kratochvíl, J., Woeginger, G.J.: Planar graph coloring avoiding monochromatic subgraphs: trees and paths make it difficult. Algorithmica **44**, 343–361 (2006)
10. Caramia, M., Dell'Olmo, P.: Iterative coloring extension of a maximum clique. Naval Res. Log. **48**, 518–549 (2001)
11. Chartrand, G., Geller, D., Hedetniemi, S.: A generalization of chromatic number. Proc. Camb. Philos. Soc. **64**, 265–271 (1968)
12. Chartrand, G., Zhang, P.: Chromatic Graph Theory. Discrete Mathematics and its Applications. Chapman and Hall/CRC Press, Boca Raton (2008)
13. Christen, C.A., Selkow, S.M.: Some perfect coloring properties of graphs. J. Combin. Theory Ser. B **27**, 49–59 (1979)
14. Chudnovsky, M., Goedgebeur, J., Schaudt, O., Zhong, M.: Obstructions for three-coloring graphs with one forbidden induced subgraph. In: Proceedings of 27th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA'16, pp. 1774–1783 (2016)
15. Cowen, L., Cowen, R.H., Woodall, D.R.: Defective colorings of graphs in surfaces: partitions into subgraphs of bounded valency. J. Graph Theory **10**, 187–195 (1986)
16. Fiala, J., Jansen, K., Le, V.B., Seidel, E.: Graph subcolorings: complexity and algorithms. SIAM J. Discret. Math. **16**, 636–650 (2003)
17. Gimbel, J., Hartman, C.: Subcolorings and the subchromatic number of a graph. Discret. Math. **272**, 139–154 (2003)
18. Gimbel, J., Nešetřil, J.: On partitions of graphs into cographs. Discret. Math. **310**, 3437–3445 (2010)
19. Goedgebeur, J., Schaudt, O.: Exhaustive generation of k-critical $\mathcal{H}$-free graphs. In: Proceedings of Graph-Theoretic Concepts in Computer Science—42nd International Workshop, WG'16, pp. 109–120 (2016)
20. Goyal, N., Vishwanathan, S.: NP-completeness of undirected Grundy numberings and related problems. Unpublished manuscript (1998)
21. Havet, F., Kang, R.J., Sereni, J.-S.: Improper coloring of unit disk graphs. Networks **54**, 150–164 (2009)
22. Hedetniemi, S.M., Hedetniemi, S.T., Beyer, T.: A linear algorithm for the Grundy (coloring) number of a tree. Congr. Numer. **36**, 351–363 (1982)
23. Hoàng, C.T., Le, V.B.: P_4-free colorings and P_4-bipartite graphs. Discret. Math. Theor. Comput. Sci. **4**, 109–122 (2001)
24. Hoàng, C.T., Moore, B., Recoskie, D., Sawada, J., Vatshelle, M.: Constructions of k-critical P_5-free graphs. Discret. Appl. Math. **182**, 91–98 (2015)

25. Jensen, T.R., Toft, B.: Graph Coloring Problems. Willey, New Jersey (1995)
26. Molloy, M.S., Reed, B.: Graph Colouring and the Probabilistic Method. Algorithms and Combinatorics, vol. 23. Springer, New York (2002)
27. Newman, M.E.J.: Networks: An introduction. Oxford University Press, Oxford (2010)
28. Zaker, M.: Results on the Grundy chromatic number of graphs. Discret. Math. **306**, 3166–3173 (2006)