

The Complexity of Minimum Convex Coloring

Frank Kammer and Torsten Tholey

Institut für Informatik, Universität Augsburg, D-86135 Augsburg, Germany
{kammer, tholey}@informatik.uni-augsburg.de

Abstract. A coloring of the vertices of a graph is called convex if each subgraph induced by all vertices of the same color is connected. We consider three variants of recoloring a colored graph with minimal cost such that the resulting coloring is convex. Two variants of the problem are shown to be \mathcal{NP} -hard on trees even if in the initial coloring each color is used to color only a bounded number of vertices. For graphs of bounded treewidth, we present a polynomial-time $(2+\epsilon)$ -approximation algorithm for these two variants and a polynomial-time algorithm for the third variant. Our results also show that, unless $\mathcal{NP} \subseteq \text{DTIME}(n^{O(\log \log n)})$, there is no polynomial-time approximation algorithm with a ratio of size $(1 - o(1)) \ln \ln n$ for the following problem: Given pairs of vertices in an undirected graph of bounded treewidth, determine the minimal possible number l for which all except l pairs can be connected by disjoint paths.

Key words: Convex Coloring, Maximum Disjoint Paths Problem

1 Introduction

A colored graph (G, C) is a tuple consisting of a graph G and a *coloring* C of G , i.e., a function assigning each vertex v a color that is either 0 or a so-called *real color*. A vertex colored with 0 is also called *uncolored*. A coloring is an (a, b) -coloring if the color set used for coloring the vertices contains at most a real colors and if each real color is used to color at most b vertices. Two equal-colored vertices v and w in a colored graph (G, C) are *C -connected* if there is a path from v to w whose vertices are all colored with the color of v and w . A coloring C is called *convex* if all pairs of vertices colored with the same real color are C -connected. For a colored graph (G, C_1) , another arbitrary coloring C_2 of G is also called a *recoloring* of (G, C_1) . We then say that C_1 is the *initial coloring* of G and that C_2 *recolors* or *uncolors* a vertex v of G if $C_2(v) \neq C_1(v)$ and $C_2(v) = 0$, respectively. The *cost* of a recoloring C_2 of a colored graph (G, C_1) with $G = (V, E)$ is $\sum_{v \in V: 0 \neq C_1(v) \neq C_2(v)} w(v)$, where $w(v)$ denotes the weight of v with $w(v) = 1$ in the case of an unweighted graph. This means that we have to pay for recoloring or uncoloring a real-colored vertex, but not for recoloring an uncolored vertex. In the *minimum convex recoloring problem* (MCRP) we are given a colored graph and search for a convex recoloring with minimal cost.

The MCRP describes a fundamental problem in graph theory with different applications in practice: a first systematic study of the MCRP on trees is from Moran and Snir [10] and was motivated by applications in biology. Further applications are so-called multicast communications in optical wavelength division

multiplexing networks; see, e.g., [6] for a short discussion of these applications. Here we focus on the MCRP as a special kind of routing problem. Suppose we are given a telecommunication or transportation network modeled by a graph whose vertices represent routers. Moreover, assume that each router can establish a connection between itself and an arbitrary set of adjacent routers. Then routers of the same initial color could represent clients that want to be connected by the other routers to communicate with each other or to exchange data or commodities. More precisely, connecting clients of the same color means finding a connected subgraph of the network containing all the clients, where the subgraphs for clients of different colors should be disjoint. If we cannot establish a connection between all the clients, we want to give up connecting as few clients to the other clients of the same color as possible in the unweighted case ($w(v) = 1$ for all $v \in V$) or to give up a set of clients with minimal total weight in the weighted case. Hence, our problem reduces to the MCRP, where a recoloring colors all those vertices with color c that represent routers used to connect clients of color c . The case in which routers can connect a constant number of disjoint sets of adjacent routers can be handled by copying vertices representing a router.

We also introduce a new relaxed version of the problem that we call the *minimum restricted convex recoloring problem* (MRRP). In this problem we ask for a convex recoloring C' that does not recolor any real-colored vertex with a different real color. In practice clients often cannot be used for routing connections for other clients so that a clear distinction between clients and routers should be made. This can be modeled by the MRRP, where a client that cannot be connected to the other clients of the same color may only be uncolored.

Finally, we consider a variant of the MCRP where we search for a convex recoloring, but assign costs to each color c . We have to pay the cost for color c if at least one vertex of color c is recolored. We call this coloring problem the *minimum block recoloring problem* or MBRP. In an unweighted version we assign cost 1 to each color. The MBRP is useful if in an application it is not useful to connect only a proper subset of clients that want to be connected.

The MCRP, the MRRP, and the MBRP can also be considered as generalizations of the *maximum disjoint paths problem* (MDPP) and the *disjoint paths problem* (DPP), where in the first case a maximum number and in the second case all pairs of given pairs of vertices of a graph are to be connected by vertex-disjoint paths, if possible. Indeed any algorithm solving one of our recoloring problems on $(\infty, 2)$ -colorings to optimality also can solve the DPP. Given an algorithm for the MBRP one can also solve the problem of connecting a subset of a given set of weighted node pairs $(s_1, t_1), \dots, (s_l, t_l)$ by disjoint paths such that the sum of the weights of the connected pairs is maximized.

Previous results. The \mathcal{NP} -hardness of the unweighted MCRP, MRRP, and MBRP follows directly from the \mathcal{NP} -hardness of the MDPP [8, 9]. However, non-approximability results for our recoloring problems do not follow from corresponding results for the MDPP since the latter problem is a maximization and not a minimization problem. Moran and Snir [10] showed that the MCRP on

(∞, ∞) -colorings remains \mathcal{NP} -hard on trees, and the same is true for the MRRP, as follows implicitly from the results in [10] concerning leaf colored trees.

Snir [13] presented a polynomial-time 2-approximation algorithm for the weighted MCRP on strings and a polynomial-time 3-approximation algorithm for the weighted MCRP on trees also published in [11]. Bar-Yehuda, Feldman, and Rawitz [3] could improve the approximation ratio on trees to $2 + \epsilon$.

New results. In contrast to the work of Bar-Yehuda et. al. and of Snir here we consider initial (a, b) -colorings with a and b different from ∞ . In addition, we also consider graphs of bounded treewidth instead of only trees.

Surprisingly, the variants of the three coloring problems all have different complexities on graphs of bounded treewidth, as we prove in Section 2 and 3. We show that the MCRP is NP-hard even on trees initially colored with $(\infty, 2)$ -colorings whereas the MRRP can be solved in polynomial time for the more general $(\infty, 3)$ -colorings as input colorings even on weighted graphs of bounded treewidth. We also observe the NP-hardness of the MRRP on trees colored with $(\infty, 4)$ -colorings. Moreover, we present a polynomial-time algorithm for the MBRP on weighted graphs of bounded treewidth for general colorings.

Extending the result of Bar-Yehuda et. al., we present a polynomial-time $(2 + \epsilon)$ -approximation algorithm for the MCRP and the MRRP on weighted graphs of bounded treewidth. However, if we follow their approach in a straight forward way, we would have to store too much information at each node of a so-called tree decomposition tree. Therefore, we would obtain a running time of size $\Omega(n^k)$ with k being the treewidth of the graph considered. Additional ideas allows us to guarantee a quadratic running time.

Beside our results on graphs of bounded treewidth we show that the unweighted versions of our recoloring problems cannot be approximated within an approximation ratio of $(1 - o(1)) \ln \ln n$ in polynomial time unless $\mathcal{NP} \subseteq \text{DTIME}(n^{O(\log \log n)})$ even if the initial coloring is restricted to be an $(\infty, 2)$ -coloring. As a consequence of this result, if we are given pairs of vertices, there is no good approximation possible for approximating in polynomial-time the minimal l such that all except l pairs are connected by disjoint paths, unless $\mathcal{NP} \subseteq \text{DTIME}(n^{O(\log \log n)})$. Determining l can be considered in some kind as the inverse of the MDPP problem. Due to space limitations some proofs in this article are omitted. They can be found in the full version of this paper.

2 Hardness Results

Theorem 1. *Given an unweighted n -vertex graph with an $(\infty, 2)$ -coloring, no polynomial-time algorithm for the MCRP, the MRRP or the MBRP has an approximation ratio of $(1 - o(1)) \ln \ln n$ unless $\mathcal{NP} \subseteq \text{DTIME}(n^{O(\log \log n)})$.*

Theorem 2. *The MCRP on unweighted graphs is \mathcal{NP} -hard even if the problem is restricted to trees colored by an initial $(\infty, 2)$ -coloring.*

Proof. The theorem can be proven by a reduction from 3-SAT. Let F be an instance of 3-SAT, i.e., F is a Boolean formula in 3-CNF. W.l.o.g. we assume

that each clause in F has exactly three literals. Let n be the number of literals in F , let m be the number of clauses of F and let r be the minimal number such that each literal in F appears at most r times in F . For the time being let us construct a forest G which later can be easily connected to a tree. We construct G by introducing for each variable x a so-called *gadget* G_x consisting of an uncolored vertex v_x , leaves $v_x^{L,i}, v_x^{R,i}$ colored with a color c_x^i and an edge $\{v_x, v_x^{L,i}\}$ for each $i = \{1, 2\}$, and two internally disjoint paths of length $r+1$, one from v_x to $v_x^{R,1}$, and the other from v_x to $v_x^{R,2}$. Let us call the internal vertices of the path connecting v_x and $v_x^{R,i}$ for $i = 1$ the *positive* and for $i = 2$ the *negative vertices* in the gadget G_x . For each clause K , we introduce a similar *gadget* G_K consisting of an uncolored vertex v_K , leaves $v_K^{L,j}, v_K^{R,j}$ colored with a color c_K^j and an edge $\{v_K, v_K^{L,j}\}$ for each $j \in \{1, 2, 3\}$, and three internally disjoint paths of length 2, all starting in v_K but ending in different endpoints, $v_K^{R,1}, v_K^{R,2}$, and $v_K^{R,3}$, respectively. In addition, we also introduce $2nr$ extra vertices without any incident edges called the *free* vertices of G . From this forest we obtain a tree T if we simply connect all gadgets and all free vertices by the following two steps. First, add two adjacent vertices v^1 and v^2 into G that both are colored with the same new color. Second, for each variable x , connect v_x to v^1 , for each clause K , connect v_K to v^2 and finally also all free vertices to v^2 .

Concerning the coloring C of T , we want to color further vertices of T . For each literal x or \bar{x} part of clause K , color in the gadget for x one positive vertex (in case of literal x) or one negative vertex (in case of literal \bar{x}) as well as one of the non-leaves adjacent to v_K with a new color $c_{x,K}$. If after these colorings there is at least one uncolored positive or negative vertex, we take for each such vertex y a new color c_y and assign it to y as well as to exactly one uncolored free vertex. One can show that F is satisfiable if and only if (T, C) has a convex recoloring C' with cost $\leq 2nr + (n + 2m)$. The proof of this equivalence is omitted. \square

Although the MCRP is \mathcal{NP} -complete when being restricted to initial $(\infty, 2)$ -colorings, this is not the fact for the MRRP as we show in Theorem 6. However, a slight modification of the reduction above shows that the MRRP on weighted graphs with an initial $(\infty, 4)$ -coloring is also \mathcal{NP} -hard even for trees. The idea is, for each colored non-leaf x , to add two new vertices x_1, x_2 , and edges $(x, x_1), (x, x_2)$, to color x_1, x_2 with the color of x , and finally to uncolor x .

3 Exact algorithms

In this section we present algorithms on graphs with bounded treewidth. For defining graphs of bounded treewidth we have to define tree decompositions. Tree decompositions and treewidth were introduced by Robertson and Seymour [12] and a survey for both is given by Bodlaender [4].

Definition 3. A tree decomposition of treewidth k for a graph $G = (V, E)$ is a pair (T, B) , where $T = (V_T, E_T)$ is a tree and B is a mapping that maps each node w of V_T to a subset $B(w)$ of V such that (1) $\bigcup_{w \in V_T} B(w) = V$, (2) for

each edge $(u, v) \in E$, there exists a node $w \in V_T$ such that $\{u, v\} \subseteq B(w)$, (3) $B(x) \cap B(y) \subseteq B(w)$ for all $w, x, y \in V_T$ with w being a vertex on the path from x to y in T , (4) $|B(w)| \leq k+1$ for all $w \in V_T$. Moreover, a tree decomposition is called nice if (5) T is a rooted and binary tree, (6) $B(\bar{w}) = B(\bar{w}_1) = B(\bar{w}_2)$ holds for each node \bar{w} of T with two children \bar{w}_1 and \bar{w}_2 , (7) either $|B(w) \setminus B(w_1)| = 1$ and $B(w) \supset B(w_1)$ or $|B(w_1) \setminus B(w)| = 1$ and $B(w_1) \supset B(w)$ holds for all nodes w of T with exactly one child w_1 .

The treewidth of a graph G is the smallest number k for which a tree decomposition of G with treewidth k exists. If $k = O(1)$, G has *bounded treewidth*. For an n -vertex graph of constant treewidth k , one can determine a nice tree decomposition (T, B) with T consisting of $O(n)$ nodes in linear time [5].

In this section we therefore will assume that we are given an n -vertex graph $G = (V, E)$ and a nice tree decomposition (T, B) of G of treewidth $k-1$ ($k \in \mathbb{N}$) with T having $O(n)$ nodes. Before presenting our algorithm we introduce some further notations and definitions. For clarity, overlined vertices—as for example \bar{v} —should always denote nodes of T . Moreover, we will refer to nodes and arcs instead of vertices and edges if we mean the vertices or edges of T . By \bar{v}_l and \bar{v}_r we denote the left and the right child of \bar{v} in T , respectively. If \bar{v} has only one child, we define it to be a left child. We also introduce a new set consisting of k gray colors—in the following always denoted by Y —and we allow for each recoloring additionally to use the gray colors. A gray colored vertex w intuitively means that w is uncolored and will later be colored with a real color. We therefore define the cost for recoloring a gray colored vertex to be 0 and do not consider the gray colors as real colors. A convex coloring from now on should denote a coloring C where all pairs of vertices of the same gray or real color are C -connected. For each node \bar{v} in T , each subset S of vertices of G , each subgraph H of G , and each coloring C of G we let

- $G(\bar{v})$ be the subgraph of G induced by all vertices contained in at least one set $B(\bar{w})$ of a node \bar{w} contained in the subtree of T rooted in \bar{v} .
- $C(S), C(H)$ be the set of colors used by a coloring C for coloring the vertices of S and of H , respectively.
- $\text{SEP}(C, \bar{v})$ be the set of real colors used to color vertices except from $B(\bar{v})$ in more than one of the subgraphs $G(\bar{v}_l), G(\bar{v}_r)$ and $G - G(\bar{v})$.

Finally, for each subgraph H of G , a *legal recoloring* of (H, C) is a recoloring C' of (H, C) such that for each real color c assigned by C' there is a vertex u of H with $c = C(u) = C'(u)$. Observe that, if there is a convex recoloring C'' of a colored graph (H, C) of cost k , there is also a legal convex recoloring C' of (H, C) with cost k . C' can be obtained from C'' without increasing the cost by uncoloring all vertices colored with a color c for which no vertex u with $C''(u) = C(u) = c$ exists. Hence for solving the MCRP, the MRRP, and the MBRP we only need to search for legal recolorings solving the problem.

For the rest of this section we assume that our given graph G is colored by an initial coloring C not using gray colors. We first present an algorithm for the MCRP. This algorithm considers the nodes of T in a bottom-up strategy

and computes for each node \bar{v} a set of so-called *characteristics*. Intuitively, each characteristic represents a recoloring C' of $G(\bar{v})$ that will be stepwise extended to a convex recoloring of the whole graph G . *Extending a (re-)coloring* C_1 for a graph H_1 means replacing C_1 by a new color function C_2 for a graph $H_2 \supseteq H_1$ with $C_2(w) = C_1(w)$ for all vertices w of H_1 with the following exception: A vertex colored with a gray color c_1 may be recolored with a real color c_2 if all vertices of color c_1 are recolored with c_2 . We next define a characteristic for a node \bar{v} precisely as a tuple $(P, \{P_S \mid S \in P\}, \{c_S \mid S \in P\}, Z)$, where

- P is a partition of $B(\bar{v})$, i.e., a family of nonempty pairwise disjoint sets S_1, \dots, S_j with $\bigcup_{1 \leq i \leq j} S_i = B(\bar{v})$. These sets are called *macro sets*.
- P_S is a partition of the macro set S , where the subsets of S contained in P_S are called *micro sets*.
- c_S for each macro set S is a value in $\text{SEP}(C, \bar{v}) \cup Y \cup \{0, b\}$, where b is an extra value different from 0 and the real and gray colors.
- $Z \subseteq \text{SEP}(C, \bar{v})$. The colors in Z are called the *forbidden colors*.

In the following for a characteristic \mathcal{Q} and a macro set S of \mathcal{Q} we denote the values P, P_S, c_S and Z above by $P^{\mathcal{Q}}, P_S^{\mathcal{Q}}, c_S^{\mathcal{Q}}$ and $Z^{\mathcal{Q}}$. We next describe a first intuitive approach of solving the MCRP extending the ideas of Bar-Yehuda et. al. [3] from trees to graphs of bounded treewidth by introducing macro and micro sets but not using gray colors or the extra value b .

A characteristic \mathcal{Q} for a node \bar{v} should represent a coloring C' of $G(\bar{v})$ such that the following holds: A macro set S of \mathcal{Q} denotes a maximal subset of vertices in $B(\bar{v})$ that are colored by C' with the same unique color equal to the value $c_S^{\mathcal{Q}}$ stored with the macro set—maximal means that there is no further vertex in $B(\bar{v}) \setminus S$ colored with $c_S^{\mathcal{Q}}$. A micro set is a maximal subset of a macro set that is C' -connected in $G(\bar{v})$. When later extending the recoloring C' we need to know which of the colors not in $C'(B(\bar{v}))$ are used by C' to color vertices of $G(\bar{v})$ since these colors may not be used any more to color a vertex outside $G(\bar{v})$. These colors are exactly the forbidden colors of the characteristic. Note that there can be more than one recoloring of $G(\bar{v})$ leading to the same characteristic for \bar{v} . Hence, a characteristic does not really represent one recoloring, but an equivalence class of recolorings. The main idea of our algorithm is the following:

Given all characteristics for the children of a node \bar{v} and, for each equivalence class \mathcal{E} described by one of these characteristics, the minimal cost among all costs of recolorings in \mathcal{E} , our algorithm uses a bottom-up strategy to compute the same information also for \bar{v} and its ancestors. Since we only want to compute convex recolorings, at the root of T we have to remove all characteristics having a real colored macro set that consists of at least two micro sets. The cost of an optimal convex recoloring is the minimal cost among all costs computed for the remaining characteristics. An additional top-down traversal of T can also determine a recoloring having optimal cost. Unfortunately, the number of characteristics to be considered by the approach above would be too high for an efficient algorithm. The problem is that for graphs of bounded treewidth, in contrary to what is the case for trees, a path connecting two vertices outside $G(\bar{v})$ may use vertices in $G(\bar{v})$ and vice versa. Hence, as a further change compared

to the algorithm of Bar Yehuda et al., we use gray colors and the extra value b . These colors are intuitively used as follows:

If a color c is used by C only to color vertices outside $G(\bar{v})$, a recoloring C' of G may possibly also want to recolor a set S of vertices in $G(\bar{v})$ with c in order to C' -connect some vertices with color c . The cost for recoloring vertices of $G(\bar{v})$ with c are independent from the exact value of c , and can be computed as the costs of uncoloring all vertices of S (since $C(\bar{w}) \neq c$ for all \bar{w} in $G(\bar{v})$) and of recoloring it (without any cost) with color c . Therefore, when considering recolorings of the graph $G(\bar{v})$, we do not allow to color it with a real color $c \notin C(G(\bar{v}))$. Instead of c we use a gray color, since coloring a vertex gray has the same costs as of uncoloring the vertex but allows us to distinguish the vertex from vertices colored with another gray color or being uncolored. Note that our definition of extending a recoloring allows us with zero costs to recolor gray and uncolored vertices in a later step with a real color, whereas recoloring real-colored vertices is forbidden when extending a recoloring.

If a recoloring C' of $G(\bar{v})$ colors a macro set S with a color c that is only used by C to color vertices of $G(\bar{v})$, then for extending the recoloring C' to a recoloring C'' , we do not need to know the exact color of S . The reason for this is that, for any vertex w outside $G(\bar{v})$, the cost for setting $C''(w) = c$ can be computed again independently from the color of S : We have to pay the weight of w as costs if w is real-colored by C and zero costs otherwise. Therefore, we use the extra value b to denote that a macro set S is real-colored with a color c that with respect to C only appears in $G(\bar{v})$ and, in this case, we will set $c_S^{\mathcal{Q}} = b$ instead of setting $c_S^{\mathcal{Q}} = c$.

Following the ideas described above we let our algorithm consider only a restricted class of characteristics. For a node \bar{v} of T , we define $C|_{G(\bar{v})}$ to be the coloring C restricted to $G(\bar{v})$. We call a characteristic \mathcal{Q} a *good characteristic* if there exists a legal recoloring C' of $(G(\bar{v}), C|_{G(\bar{v})})$ with the properties (A1)-(A7). C' is then said to be *consistent* with \mathcal{Q} .

- (A1) $C'(G(\bar{v})) \subseteq C(G(\bar{v})) \cup Y \cup \{0\}$.
- (A2) For each macro set S of \mathcal{Q} , C' colors all vertices of S with one color, namely with $c_S^{\mathcal{Q}}$ if $c_S^{\mathcal{Q}} \neq b$, and with a real color not in $C(G - G(\bar{v}))$ if $c_S^{\mathcal{Q}} = b$.
- (A3) C' colors two different macro sets of \mathcal{Q} with different colors.
- (A4) A micro set is a maximal subset of $B(\bar{v})$ that is C' -connected in $G(\bar{v})$.
- (A5) C' is a convex recoloring for the graph obtained from $G(\bar{v})$ by adding, for each macro set S , edges of an arbitrary simple path visiting exactly one vertex of each micro set of S .
- (A6) Every gray colored vertex in $G(\bar{v})$ is C' -connected to a vertex in $B(\bar{v})$.
- (A7) $Z^{\mathcal{Q}} = \text{SEP}(C, \bar{v}) \cap (C'(G(\bar{v})) \setminus C'(B(\bar{v})))$.

Note that each convex legal recoloring C' of the initial colored graph (G, C) is consistent with a good characteristic \mathcal{Q} for the root \bar{r} of T . More explicitly, we obtain \mathcal{Q} by dividing $B(\bar{r})$ into macro sets each consisting of all vertices of one color with respect to C' , by defining the partition of each macro set to consist of only one micro set, by setting $Z^{\mathcal{Q}} = \emptyset$ and by defining, for each macro set S , $c_S^{\mathcal{Q}} = b$ if $C'(S)$ is a real color, or $c_S^{\mathcal{Q}} = 0$ otherwise.

Our algorithm computes in a bottom-up process for each node \bar{v} of T all good characteristics of \bar{v} from the good characteristics of the children of \bar{v} . However not all pairs of good characteristics of the children can be combined to good characteristics of \bar{v} . Therefore we call a characteristic \mathcal{Q}_l of \bar{v}_l and a characteristic \mathcal{Q}_r of \bar{v}_r *compatible* if they satisfy the following three conditions:

- Two vertices $v_1, v_2 \in B(\bar{v}_1) = B(\bar{v}_r)$ belong to the same macro set in \mathcal{Q}_l if and only if this is true for \mathcal{Q}_r .
- Let S be a macro set of \mathcal{Q} and hence also of \mathcal{Q}_l and \mathcal{Q}_r . Then either $c_S^{\mathcal{Q}_l} = c_S^{\mathcal{Q}_r} \neq b$ or exactly one of $c_S^{\mathcal{Q}_l}$ and $c_S^{\mathcal{Q}_r}$ is a gray color.
- The sets of forbidden colors of \mathcal{Q}_r and of \mathcal{Q}_l are disjoint.

The following algorithm computes for each node \bar{v} of T a set $\mathcal{M}_{\bar{v}}$ of characteristics from which we will show in the full version of this paper that it is exactly the set of good characteristics of \bar{v} . First of all, in a preprocessing phase compute by a bottom-up and a top-down traversal of T , for each node \bar{v} of T , the set $\text{SEP}(C, \bar{v})$ as well as the set of colors that are used by C to color vertices in $G(\bar{v})$ but no vertex outside $G(\bar{v})$. The latter set is in the following denoted by $U(C, \bar{v})$. Next, for each leaf \bar{v} of T , $\mathcal{M}_{\bar{v}}$ is obtained by taking into account all possible divisions of the vertices of $B(\bar{v})$ into macro sets and all possible colorings of the macro sets with different colors of $C(B(\bar{v})) \cup Y \cup \{0\}$. More precisely, for each choice, a characteristic \mathcal{Q} is obtained and added to $\mathcal{M}_{\bar{v}}$ by defining, for each macro set S colored with c , the micro sets of S to be the connected components of the subgraph of G induced by the vertices of S , and by setting $c_S^{\mathcal{Q}} = b$ if c is a real color in $U(C, \bar{v})$, and $c_S^{\mathcal{Q}} = c$ otherwise. The set $Z^{\mathcal{Q}}$ of forbidden colors is set to \emptyset .

Next start a bottom-up traversal of T . At a non-leaf \bar{v} all already computed characteristics of the children are considered. In detail, for each characteristic \mathcal{Q}_l of $\mathcal{M}_{\bar{v}_l}$ and—if \bar{v} has two children—for each compatible good characteristic \mathcal{Q}_r of $\mathcal{M}_{\bar{v}_r}$, we add to $\mathcal{M}_{\bar{v}}$ the set of characteristics that also could be obtained as output by the following non-deterministic algorithm:

- Take for \mathcal{Q} and the vertices in $B(\bar{v}) \cap B(\bar{v}_1)$ the same division into macro sets as for \mathcal{Q}_l . If \bar{v} has only one child and there is also a vertex $w \in B(\bar{v}) \setminus B(\bar{v}_1)$, choose one of the $\leq k$ possibilities of assigning w to one of the macro sets of $B(\bar{v}) \cap B(\bar{v}_1)$ or choose $\{w\}$ to be its own new macro set.
- For dividing the vertices of $B(\bar{v})$ into micro sets, construct the graph H consisting of the vertices in $B(\bar{v})$ and having an edge between two vertices if and only if both vertices belong to the same macro set and either this edge exists in G or both vertices belong to the same micro set in \mathcal{Q}_l or \mathcal{Q}_r . Define the vertices of each connected component in H to be a micro set of \mathcal{Q} .
- For each macro set S obtained by the construction above, distinguish between three cases.
 - $S \subseteq S'$ for a macro set S' of \mathcal{Q}_l : If \bar{v}_r does not exist or if $c_{S'}^{\mathcal{Q}_l} = c_{S'}^{\mathcal{Q}_r}$, set $c_S^{\mathcal{Q}} = c_{S'}^{\mathcal{Q}_l}$. Otherwise, set $c_S^{\mathcal{Q}}$ to the non-gray value in $\{c_{S'}^{\mathcal{Q}_l}, c_{S'}^{\mathcal{Q}_r}\}$.
 - S has a vertex $w \in B(\bar{v}) \setminus B(\bar{v}_1)$ and $|S| > 1$: Choose $c_S^{\mathcal{Q}} \in \{c_{S \setminus \{w\}}^{\mathcal{Q}_l}, C(w)\}$ if $c_{S \setminus \{w\}}^{\mathcal{Q}_l}$ is a gray color, otherwise, $c_S^{\mathcal{Q}} = c_{S \setminus \{w\}}^{\mathcal{Q}_l}$.

- $S = \{w\}$ with $w \in B(\bar{v}) \setminus B(\bar{v}_1)$: Choose for $c_S^{\mathcal{Q}}$ a value of $Y \cup \{0, C(w)\}$. After defining $c_S^{\mathcal{Q}}$ as described above, if $c_S^{\mathcal{Q}}$ is a real color and $c_S^{\mathcal{Q}} \in U(c, \bar{v})$, redefine $c_S^{\mathcal{Q}} = b$.
- Reject the computation if there is a micro set S' part of a macro set S in \mathcal{Q}_1 with $S' \cap B(\bar{v}) = \emptyset$ and either $c_{S'}^{\mathcal{Q}_1}$ is a gray color or $S \setminus S' \neq \emptyset$.
- If there is macro set $S = B(\bar{v}_1) \setminus B(\bar{v})$ of \mathcal{Q}_1 and if $c_S^{\mathcal{Q}_1}$ is a real color, set $Z' = \{c_S^{\mathcal{Q}_1}\}$ and $Z' = \emptyset$ otherwise. Finally, set $Z^{\mathcal{Q}} = \text{SEP}(C, \bar{v}) \cap (Z' \cup Z^{\mathcal{Q}_1} \cup Z^{\mathcal{Q}_2})$.

As mentioned before, one can show that our algorithm correctly computes for each node \bar{v} the set of all good characteristics of \bar{v} and that our algorithm can be extended such that it computes with each good characteristic \mathcal{Q} the costs of a recoloring consistent with \mathcal{Q} that among all such recolorings has minimal costs. One can also show that our algorithm has a running time of $O(n^2 + 4^s(k + s + 2)^{6k+1}(k^2 + s)n)$, where $s = \max_{\bar{v} \text{ node of } T} |\text{SEP}(C, \bar{v})|$.

After the removal of all characteristics having a real colored macro set that consists of at least two micro sets or having a gray colored macro set we obtain the cost of an optimal legal convex recoloring as the minimal costs among all costs stored with the remaining characteristics constructed for the root of T . Finally by an additional top-down traversal our algorithm can—beside the minimal costs of a legal recoloring—also determine the coloring itself within the same time bound. We obtain the following theorem.

Theorem 4. *Given a colored graph (G, C) and a nice tree decomposition (T, B) of width $k - 1$ as input the MCRP can be solved in $O(n^2 + 4^s(k + s + 2)^{6k+1}(k^2 + s)n)$ time, where $s = \max_{\bar{v} \text{ node of } T} |\text{SEP}(C, \bar{v})|$.*

It is easy to modify the algorithm above such that it solves the MRRP within the same time bound. In each bottom-up step we only have to exclude recolorings that recolor a real colored vertex with a gray or another real color.

Unfortunately, the algorithms above for the MCRP and the MRRP are exponential in s since there are 2^s different possible lists of forbidden colors. The good news concerning the MBRP on general initial colorings and the MRRP with its initial coloring being an $(\infty, 3)$ -coloring is that we can omit to store the forbidden colors explicitly. We next describe the necessary modifications.

For the MBRP we use the same basic algorithm as for the MCRP. However, we compute as a solution for the MBRP w.l.o.g. only recolorings that, for each real color c , either recolor all or none of the vertices initially colored with c . Following this approach, a characteristic of a node \bar{v} should only represent recolorings that, for each real color c , either recolor all or none of the vertices u in $G(\bar{v})$ for which $C(u) = c$ holds. If in the latter case there is a vertex in $G(\bar{v})$ and also a vertex outside $G(\bar{v})$ initially colored with c , we therefore claim that a vertex of $B(\bar{v})$ is also colored with c since otherwise the recoloring can not be extended to a legal convex recoloring not recoloring any vertex of c . This implies an additional rule for constructing characteristics:

Assume that—as in our basic algorithm—we want to construct a characteristic \mathcal{Q} of a non-leaf \bar{v} from a characteristic $\mathcal{Q}_{\bar{u}}$ of a child \bar{u} of \bar{v} . Then we are

only allowed to color a macro set S of \mathcal{Q} with c if (1) all vertices in $B(\bar{v})$ that are initially colored with c are contained in S and (2) either $\mathcal{Q}_{\bar{v}}$ also contains a macro set S' with $c_{S'}^{\mathcal{Q}_{\bar{v}}} = c$ or $c \notin C(G(\bar{u}))$.

For efficiently testing condition (2), we construct in a preprocessing phase for each node \bar{v} an array $A_{\bar{v}}$, with the following entries: For each color $c \in C(G)$, $A_{\bar{v}}[c] = 1$ if $G(\bar{v})$ contains a vertex u with $C(u) = c$. Otherwise $A_{\bar{v}}[c]$ is defined to be 0. If the array is computed by a bottom-up traversal of T , the preprocessing phase takes $O(n^2)$ time. After the preprocessing phase we can test for each characteristic \mathcal{Q} of a node \bar{v} and each color c in $O(k)$ time whether (1) and (2) hold. Hence, the asymptotic running time of our algorithm does not increase. Moreover, our additional rules enables us to find out, for each color $c \in \text{SEP}(C, \bar{v})$, whether a vertex of $G(\bar{v}_l)$ or $G(\bar{v}_r)$ is colored with c by considering $A_v[c]$ and by testing whether a macro set S of \mathcal{Q}_l or \mathcal{Q}_r , respectively, is colored with c . Hence, there is no need to store the forbidden colors.

Theorem 5. *On graphs of bounded treewidth the MBRP is solvable in polynomial time.*

More complicated modifications are necessary for the MRRP. We assume w.l.o.g. that, for each color c , there are either no or at least two vertices colored with c by C . The main idea of our algorithm is the following: For improving the running time at a node \bar{v} of T we only want to consider recolorings C' of $G(\bar{v})$ such that for each color $c \in C(G)$ the following condition (D, c) holds. The correctness of this step will be discussed later.

(D, c) If u is a vertex in $G(\bar{v})$ with $C'(u) = C(u) = c$, either there exists a vertex w outside $G(\bar{v})$ with $C(w) = c$ and a vertex $w' \in B(\bar{v})$ with $C'(w') = c$, or there exists another vertex $w \in G(\bar{v})$ with $C'(w) = C(w) = c$.

This property guarantees that, for a node \bar{v} of highest depths with $G(\bar{v}_l)$ containing a vertex u_l initially colored with c and $G(\bar{v}_r)$ containing a vertex u_r initially colored with c , a recoloring C' with property (D, c) colors u_l or u_r with c if and only if $B(\bar{v}_l)$ and $B(\bar{v}_r)$, respectively, also contains a vertex colored with c . Therefore, there is no need to store c explicitly as a forbidden color in a characteristic of \bar{v}_l and of \bar{v}_r any more. With similar arguments one can show that for no node its characteristic has to store c explicitly as a forbidden color.

The problem is that some legal recolorings are permitted by (D, c). However, each convex recoloring C_{opt} of optimal cost either is a recoloring with property (D, c) or it colors w.l.o.g. exactly one vertex u with c . In the latter case a coloring with the same cost as C_{opt} can be obtained from a recoloring with property (D, c) not coloring any vertex with c by undoing the recoloring of the vertex originally colored with c that among all such vertices has a maximal weight. Therefore, for computing the costs of an optimal convex recoloring, we only have to consider the costs of recolorings with property (D, c) and eventually to subtract the maximal weight over all vertices originally colored with c . Let us call such a subtraction a *c-cost adaption*. Our goal now is to describe an algorithm that runs the *c-cost-adaption* during the bottom-up traversal of T at a certain node \bar{v} —called the *c-decision node*—having the following property:

For each characteristic \mathcal{Q} of \bar{v} , either each recoloring C' extending a recoloring consistent with \mathcal{Q} and having property (D,c) C' -connects at least two vertices initially colored with c (and we therefore must not run a c -cost-adaption) or all such recolorings uncolor all vertices initially colored with c (and therefore we have to run a c -cost-adaption).

If, for each color c , we know the c -decision node, our algorithm runs as follows: For each node \bar{v} (also above the c -decision node), we only compute characteristics representing recolorings for which property (D,c) holds for each color c . If we reach the c -decision node \bar{v} , for each characteristic \mathcal{Q} of \bar{v} , we test whether all recolorings extending \mathcal{Q} do not use color c and if so, we run a c -cost adaption for \mathcal{Q} . One can show that, for all colors c , a c -decision node exists and that one can efficiently determine the characteristics representing the recoloring with property (D,c) .

Theorem 6. *On graphs of bounded treewidth the MRRP restricted to initial $(\infty, 3)$ -colorings is solvable in polynomial time.*

Note that the running times for the MCRP and the MRRP on arbitrary initial colorings are also polynomial if s —defined as in Theorem 4—is of size $O(\log n)$. This is the case if an (a, b) -coloring with $a = O(\log n)$ is given.

Theorem 7. *On graphs of bounded treewidth the MCRP and the MRRP, both restricted to initial (a, b) -colorings with $a = O(\log n)$, are solvable in polynomial time.*

4 Approximation algorithms

Since the MCRP is \mathcal{NP} -hard even on trees, we can not hope for a polynomial-time algorithm that solves the problem to optimality—even if we consider graphs of bounded treewidth. Using the algorithm of the last section we now present for graphs of bounded treewidth a $(2 + \epsilon)$ -approximation algorithm for the MCRP and the MRRP given an arbitrary (∞, ∞) -coloring. The following algorithm is inspired by the algorithm of Bar-Yehuda et al. [3]. We extend the algorithm from trees to graphs of bounded treewidth and present a slightly different description for proving the correctness of the algorithm.

Given a graph G with a coloring C and a nice tree decomposition (T, B) of width $k - 1$ for G our results can be obtained by iteratively modifying the coloring C and the weights of the vertices such that finally $|\text{SEP}(C, \bar{v})| < s$ for all nodes \bar{v} of T and a fixed $s \in \mathbb{N}$ with $s > k$. Let \bar{v} be a node of T such that there is a set $R' \subseteq \text{SEP}(C, \bar{v})$ containing exactly s colors and let V' be a set consisting of two vertices of color c for all $c \in R'$ such that for each pair of vertices $x, y \in V'$ of the same color the vertices x and y are in different components in $G - B(\bar{v})$. Moreover, let α be the minimal weight of a vertex in V' . The size of $\text{SEP}(C, \bar{v})$ is decremented by reducing the weight of all vertices in V' by α and subsequently uncoloring the vertices of zero-weight.

On the one hand, this weight reduction decreases the cost of an optimal convex (re-)coloring C' of G by at least $(s - k)\alpha$ since the at most k vertices in $B(\bar{v})$ allow to color connect only k of the s colors in R' , i.e., $s - k$ vertices in V' can not be C' -connected. On the other hand, if we have a solution for the MCRP (or the MRRP) with the reduced weight function, we can simply take this solution as a solution for the MCRP (or the MRRP) with the original weights and our costs increase by at most $2s\alpha$. Thus, in each iteration our costs decrease at most by a factor of $2s/(s - k)$ more than the decrease of the costs of an optimal solution. If at the end no further reduction is possible, we can use the exact algorithms from the previous section, i.e., we can solve the instance obtained by this weight reduction as good as an optimal algorithm. Altogether, we have only recoloring costs that are a factor of $2s/(s - k)$ bigger than the costs of an optimal solution. Choosing s large enough, we obtain the following.

Corollary 8. *For graphs of bounded treewidth a $(2 + \epsilon)$ -approximation algorithms exist for the MCRP and the MRRP with quadratic running time.*

References

1. N. Alon, D. Moshkovitz, and S. Safra. Algorithmic construction of sets for k -restrictions, *ACM Transactions on Algorithms* **2** (2006), pp. 153-177.
2. M. Andrews and L. Zhang, Hardness of the undirected edge-disjoint paths problem, Proc. 37th Annual ACM Symposium on Theory of Computing (STOC 2005), pp. 276-283.
3. R. Bar-Yehuda, I. Feldman, and D. Rawitz, Improved approximation algorithm for convex recoloring of trees, Workshop on Approximation and Online Algorithms (WAOA '05), LNCS, Vol. 3879, Springer, Berlin, 2006, pp. 55-68.
4. H. L. Bodlaender, A partial k -arboretum of graphs with bounded treewidth, *Theoret. Comput. Sci.* **209** (1998), pp. 1-45.
5. H. L. Bodlaender and T. Kloks, Efficient and constructive algorithms for the path-width and treewidth of graphs, *J. Algorithms* **21** (1996), pp. 358-402.
6. X. Chen, X. Hu, and T. Shuai, Inapproximability and approximability of maximal tree routing and coloring, *J. Comb. Optim.* **11** (2006), pp. 219-229.
7. U. Feige, A threshold of $\ln n$ for approximating set cover, *J. ACM* **45** (1998), pp. 634-652.
8. R. M. Karp, On the computational complexity of combinatorial problems, *Networks* **5** (1975), pp. 45-68.
9. J. F. Lynch, The equivalence of theorem proving and the interconnection problem, *(ACM) SIGDA Newsletter*, **5** (1975), pp. 31-36.
10. S. Moran and S. Snir, Convex recolorings of strings and trees: definitions, hardness results and algorithms, Workshop on Algorithms and Data Structures (WADS '05), LNCS, Vol. 3608, Springer, Berlin, 2005, pp. 218-232.
11. S. Moran and S. Snir, Efficient approximation of convex recoloring, *J. Comput. System Sci.* **73** (2007), pp. 1078-1089.
12. N. Robertson and P. D. Seymour, Graph minors. I. Excluding a forest, *J. Comb. Theory Ser. B* **35** (1983), pp. 39-61.
13. S. Snir, *Computational Issues in Phylogenetic Reconstruction: Analytic Maximum Likelihood Solutions, and Convex Recoloring*. Ph.D. Thesis, Department of Computer Science, Technion, Haifa, Israel, 2004.