A Kernel for Convex Recoloring of Weighted Forests

Hans L. Bodlaender[1] and Marc Comas[2]

[1] Department of Information and Computing Sciences
Utrecht University, Utrecht, The Netherlands
[2] Departament de Llenguatges i Sistemes Informàtics
Universitat Politècnica de Catalunya, Barcelona, Spain

Abstract. In this paper, we show that the following problem has a kernel of quadratic size: given is a tree T whose vertices have been assigned colors and a non-negative integer weight, and given is an integer k. In a recoloring, the color of some vertices is changed. We are looking for a recoloring such that each color class induces a subtree of T and such that the total weight of all recolored vertices is at most k. Our result generalizes a result by Bodlaender et al. [3] who give quadratic size kernel for the case that all vertices have unit weight.

1 Introduction

In this paper, we consider the following problem. Given is a tree T with for each vertex a color from some given set of colors, and a non-negative integer weight. In a recoloring of T, the color of some vertices is changed. The cost of a recoloring is the total weight of all vertices with a changed color. A coloring is convex, if for each color, the set of vertices with that color forms a (connected) subtree of T. We consider the decision version of the problem: given an integer k, we ask if there is a convex recoloring with cost at most k.

In this paper, w1e look at the parameterized variant of the problem, and show that the problem has a quadratic kernel, i.e., we give a polynomial time algorithm, that given an instance of the problem, transforms it to an equivalent instance with $O(k^2)$ vertices and edges. Our result generalizes an earlier result by Bodlaender et al. [3] who give a quadratic kernel for the unweighted version of the problem, i.e., for the case that all vertices have unit weight. We call the problem Weighted Convex Tree Recoloring. A generalization with only positive weights appears to be relatively simple, by reducing it to the unweighted case; allowing zero weight vertices asks for a different set of rules and analysis. These zero weight vertices are interesting, also from application point of view, as they also model vertices that initially do not have a color assigned to them.

The convex recoloring problem for trees is motivated from applications in phylogenetic and other areas from bio-informatics and linguistics. We refer the reader to [6,8,9] for more background and motivation of the problem.

Finding kernels of small size for combinatorial problems is a topic of much current research, and an important topic in the area of parameterized complexity and

J. van Leeuwen et al. (Eds.): SOFSEM 2010, LNCS 5901, pp. 212–223, 2010.

algorithms. We assume the reader to be familiar with standard notions from parameterized complexity and kernelization; for an introduction, see e.g., [7], [5], [4] or [10].

In [8], Moran and Snir showed that the Convex Tree Recoloring problem is NP-complete. Also, several special cases remain NP-complete, even when the tree is restricted to be a path. Improving some of the previous results in [8,9,12], Bar-Yehuda et al. [2] gave a polynomial $(2 + \epsilon)$-approximation algorithm and an exact algorithm whose parameterized complexity, parameterized by the number of recolorings k, is $O(n^2 + nk2^k)$. Further improvements on the running time for an exact algorithm can be found in [11].

In [8,9], different variants of the problem are presented. In [1], an algorithm with a running time of $O(4^k n)$ is given for the case that only the leafs are colored, and the problem asks if it is possible to recolor the leafs in such a way that the resulting coloring can be extended to a convex one. In [3], a quadratic kernel is given for the unweighted version of the problem. One step of the algorithm in [3] is to generalize the problem to the case where some vertices can have a fixed color, and to forests instead of trees. Two final steps can transform the problem back to an instance without fixed color vertices and with a tree instead of a forest. In [3], it is asked as an open problem if it is possible to find small kernels for other convex recoloring variants. In this paper, we extend the result of [3] by allowing zero weights (thus allowing leaf and other partial colorings) and positive weights to the vertices (which generalizes the case when we allow vertices with a fixed color), and we show that indeed, in this situation we are also able to obtain a quadratic kernel.

2 Convex Recoloring Problem

We denote the set of non-negative integers by $\mathbb{N}_0$. Let $F=(V,E,\mu)$ be a weighted forest, with $\mu : V \to \mathbb{N}_0$ and let $\mathcal{C}$ be a set of colors. A *coloring* of F is a function defined from the set of vertices V to the set of colors $\mathcal{C}$. Given a coloring Γ, any different coloring will be called a *recoloring* of Γ. We define the *cost* of a recoloring Γ' of Γ, denoted $cost_\Gamma(\Gamma')$, to be the sum of the weights of recolored vertices:

$$cost_\Gamma(\Gamma') = \sum_{v\in V, \Gamma(v)\neq\Gamma'(v)} \mu(v)$$

For any forest F' contained in F, we denote by $\mu_{\Gamma,c}(F')$, the sum of $\mu(v)$ over all vertices v in F' colored c by Γ. We also use the previous notation for any subset of vertices of V, i.e., for $W \subseteq V$, $\mu_{\Gamma,c}(W) = \sum_{v\in\Gamma^{-1}(c)\cap W} \mu(v)$, where $\Gamma^{-1}(c) = \{v \mid \Gamma(v) = c\}$.

We say that a coloring Γ is *convex* if for every color c, $\Gamma^{-1}(c)$ induces a connected component. In this paper, we deal with the following parameterized problem.

Convex recoloring of a Weighted Colored Forest (crp)
Instance: A weighted forest $F = (V,E,\mu)$, a set of colors $\mathcal{C}$, a coloring Γ of F and a positive integer k.

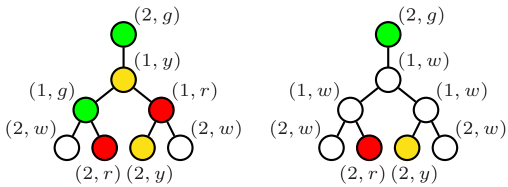

Fig. 1. Γ' is the only recoloring of Γ with cost at most 3 and convex. In every node we represent the pair *weight-color*.

Parameter: k
Question: Is there a convex recoloring Γ' of Γ with $cost_\Gamma(\Gamma') \leq k$?

While the case that F is a tree is most interesting from an application point of view, the version with forests helps to design our algorithms. As in [3], an instance with a forest can be transformed to an instance with a tree by adding one new vertex with cost $k+1$ and making it incident to a vertex in each tree in the forest.

In Figure 1, we can see a coloring Γ in a tree T and a convex recoloring with cost at most 3. In fact, it is not difficult to check that Γ' is the only convex recoloring with cost at most three, all other convex recolorings have larger cost.

We use the consideration introduced in [9], of adding a set of new special colors. Formally, for each vertex v, we add a new color c_v, and we allow that a vertex v can be recolored to this color c_v. Using the previous consideration, we are going to assume that any instance with a vertex v such that $\mu(v) = 0$, has v colored with $\Gamma(v) = c_v$. The motivation of such assumption, comes from the fact that any convex recoloring Γ' of Γ will have $cost_\Gamma(\Gamma') = cost_{\Gamma_{v=c_v}}(\Gamma')$ for the coloring $\Gamma_{v=c_v}$ obtained from Γ by changing the color of v to c_v. We make this assumption for each vertex v with $\mu(v) = 0$.

3 Definitions

Given a forest $F = (V_F, E_F)$ and a coloring Γ, we denote by $sub_\Gamma(F, c)$ the set of vertices in V_F colored c by Γ, i.e., $sub_\Gamma(F, c) = V_F \cap \Gamma^{-1}(c)$. For a forest F and a color c, $Bag_c(F)$ is defined as the subset $sub_\Gamma(T^*, c)$ for a component T^* of F with maximum $\mu_{\Gamma,c}(T^*)$. In other words, $Bag_c(F)$ is the set of vertices of color c in the connected component of F in which the total weight of such vertices is maximum.

Consider a path s between two vertices with color c, and consider the forest $F - s$ obtained after removing s from F. Let $Tag(s)$ be the set consisting of all vertices with color c' different from c not belonging to a $Bag_{c'}(F - s)$, i.e.,

$$Tag(s) = \bigcup_{c' \in \mathcal{C} \setminus \{c\}} sub_\Gamma(F - Bag_{c'}(F - s), c'). \tag{1}$$

Let

$$tag(s) = \sum_{v \in Tag(s)} \mu(v). \quad (2)$$

From (1), it is not difficult to see that (2) can also be written as

$$tag(s) = \sum_{c' \in \mathcal{C} \setminus \{c\}} \mu_{\Gamma,c'}(F) - \mu_{\Gamma,c'}(Bag_{c'}(F - s)).$$

Proposition 1. *Let s be a path between two vertices with color c. For any convex recoloring Γ' such that for all vertices v in s, $\Gamma'(v) = c$,*

$$tag(s) \leq cost_{\Gamma}(\Gamma').$$

Proof. If all vertices in s receive color c, then for any color $c' \neq c$, at most one component of $F - s$ can have vertices of color c'. So, we need to recolor all vertices with color c' in all except maybe one component of $F - s$. Because for any component T in $F - s$ we have $\mu_{\Gamma,c'}(T) \leq \mu_{\Gamma,c'}(Bag_{c'}(F - s))$, the cost for recoloring vertices with color c' is at least $\mu_{\Gamma,c'}(F) - \mu_{\Gamma,c}(Bag_{c'}(F - s))$. Summation over all $c' \neq c$ gives:

$$\sum_{c' \in \mathcal{C} \setminus \{c\}} \mu_{\Gamma,c'}(F) - \mu_{\Gamma,c'}(Bag_{c'}(F - s)). \qquad \square$$

The previous proposition is the motivation for the following definitions. A *k-string* of color c is a path s consisting of two vertices u and v with color c and positive weight, called endpoints, and *interior* vertices (vertices different of u and v in s) with color different of c, in such a way that $tag(s) \leq k$. Note that we allow $u = v$. In this case, we denote the path with only the vertex v indistinctly by s_v or $\{v\}$. Let $\mathcal{S}tr_k$ be the set of all k-strings of any color and let $\mathcal{S}tr_k^c$ be the set with all k-strings of color c. If $\mathcal{S}$ is a subset of $\mathcal{S}tr_k$, we denote by $F_{\mathcal{S}}$ the forest obtained by the union of all k-strings contained in $\mathcal{S}$.

Note that from Proposition 1, if two vertices with the same color are not forming a k-string, one of them have to be recolored. Concretely, if for a vertex v $\{v\}$ is not a k-string, in a convex recoloring v is recolored.

Similar to [2,3], for every vertex v, we define a subset of $\mathcal{C}$, defined by

$$S_k(v) = \{c \in \mathcal{C} \mid \ v \in s \text{ for some } s \in \mathcal{S}tr_k^c \ \}.$$

Let $S_k^*(v) = S_k(v) \cup \{c_v\}$. A recoloring Γ' of Γ is *k-normalized* if for every vertex v, $\Gamma'(v) \in S_k^*(v)$. This means, that in a k-normalized recoloring, any vertex v receives a color of some k-strings containing it or a color c_v.

Lemma 1. *If there is a convex recoloring Γ' of Γ with $cost_{\Gamma}(\Gamma') \leq k$, there is a convex recoloring Γ'' of Γ with $cost_{\Gamma}(\Gamma'') \leq k$ which is k-normalized.*

Proof. Consider a convex recoloring Γ' of Γ with $cost_{\Gamma}(\Gamma') \leq k$. Moreover, assume Γ' has the maximum number of vertices colored c_v. Under the last assumption (being maximum in the number of vertices colored c_v), we claim that the recoloring Γ' is exactly the recoloring Γ'' of Γ defined by

$$\Gamma''(v) = \begin{cases} \Gamma'(v) & \text{if } \Gamma'(v) \in S_k(v) \\ c_v & \text{if } \Gamma'(v) \notin S_k(v), \end{cases}$$

which is clearly k-normalized. Suppose not, this is $\Gamma'' \neq \Gamma'$. Then, there is a vertex v with $\Gamma'(v) \notin S_k(v)$ such that $\Gamma'(v) \neq c_v$. By definition, in this situation, $\Gamma''(v) = c_v$. Also note that $\Gamma(v) \neq \Gamma'(v)$, this is because if v maintain the color, then v is forming a k-string of color $\Gamma(v)$ and therefore $\Gamma(v) \in S_k(v)$. So, let $\Gamma'(v) = c$ $(\neq \Gamma(v))$ and let T_c be the tree induced by all the vertices colored c by Γ'. Let x and y be two leaves in T_c having v in the path joining them. If such a vertices don't exists, it means that v is a leaf in T_c and therefore can be recolored to c_v maintaining the convexity and contradicting the optimality of Γ' on the number of vertices colored c_v. At last, note that $\Gamma(x) = \Gamma(y) = c$, otherwise, if one of them (for example x) has $\Gamma(x) \neq c$, then we can recolor x to c_v maintaining again the convexity and contradicting the optimality. Finally, rest to point out that if x and y have $\Gamma(x) = \Gamma(y) = \Gamma'(x) = \Gamma'(y) = c$, the path between x and y is forming a k-string and therefore, $c \in S_k(v)$ which contradicts the first assumption. □

4 Kernelization Rules and Analysis

The next two rules allow us to have an instance holding some desirable properties, by recoloring some vertices or eliminating some edges in the instance.

Rule 1. *Consider a vertex v such that $\{s\}$ is not a k-string, i.e., $tag(\{v\}) > k$. Suppose $|S_k(v)| \leq 1$. Then,*

- *if $S_k(v) = \emptyset$, return NO,*
- *if $S_k(v) = \{c\}$, recolor vertex v to c and reduce k by $\mu(v)$.*

Rule 2. *If Rule 1 cannot be applied, set $F = F_{\mathcal{S}tr_k}$.*

Lemma 2. *In any instance reduced with respect to Rule 1 and Rule 2, the forest*

$$F - \bigcup_{c' \in \mathcal{C} \setminus c} F_{\mathcal{S}tr_k^{c'}}$$

contains only vertices v with color c and c_v.

Proof. By Rule 2, we have that $F = F_{\mathcal{S}tr_k}$ and then,

$$F_c = F - \bigcup_{c' \in \mathcal{C} \setminus c} F_{\mathcal{S}tr_k^{c'}}$$

only contains vertices belonging to k-strings of color c. If there exists a vertex v in F_c with color c' different from c and c_v, then s_v is not a k-string and $S_k(v) = \{c\}$. So, vertex v is recolored to c by Rule 1. □

From now on, we assume that Rule 1 and Rule 2 cannot be applied.

4.1 Pieces of a Color

Consider the forest $F_c = F - \bigcup_{c' \in \mathcal{C} \setminus \{c\}} F_{\mathcal{S}tr_k^{c'}}$. Every component of F_c is called a *piece of color c*. By Lemma 2, F_c contains only vertices v with color c or c_v, and moreover, by Lemma 1, we can assume that the vertices in F_c can only receive color c or c_v. We have the following lemma,

Lemma 3. *There is always an optimum recoloring such that for each piece of color c, every vertex v in the piece is colored c or c_v.*

Proof. Clearly, if a piece of color c has some vertex v recolored to c_v (c_v or c are the only colors in $S_k(v)$) by a recoloring Γ', then the recoloring Γ'' with all vertices in the pieces colored c has at most the same cost as Γ', and it is not difficult to see that this recoloring is still convex. □

Suppose that a piece W of color c has at least half of the total weight of the sum of vertices of color c. Then, by Lemma 3, if a recoloring Γ' recolors some vertex in W from c to c_v, the recoloring Γ'' not recoloring any vertex in W from c to c_v has at most the same cost as Γ'. So we can assume that the piece of color is not recolored in an optimum recoloring. This argument and Lemma 3 are captured by the following rule.

Rule 3. *For every piece W of color c, contract W to a single vertex w with color c and $\mu(w)$ defined as follows,*

- *if $\mu_{\Gamma,c}(W) > \mu_{\Gamma,c}(F - W)$, set $\mu(w) = k + 1$,*
- *otherwise set $\mu(w) = \mu_{\Gamma,c}(W)$.*

4.2 Irrelevant Colors

We say that a color c is *irrelevant*, if all the vertices of color c are contained in some piece of color c, and for any vertex v with color different of c and c_v, $c \notin S_k(v)$. The *cost of removing* an irrelevant color c is defined by

$$\Delta_c = \mu_{\Gamma,c}(F_{\mathcal{S}tr_k^c}) - \mu_{\Gamma,c}(Bag_c(F_{\mathcal{S}tr_k^c})).$$

Intuitively, when a color c is irrelevant, forests $F_{\mathcal{S}tr_k^c}$ and $\bigcup_{c' \in \mathcal{C} \setminus \{c\}} F_{\mathcal{S}tr_k^{c'}}$ are disjoint. Moreover, by Lemma 2, all the colors in one forest do not appear in the other one. So, we can solve both forests separately. Because $F_{\mathcal{S}tr_k^c}$ is easy to solve (it only contains color c and c_v), we can solve $F_{\mathcal{S}tr_k^c}$ and reduce F to $\bigcup_{c' \in \mathcal{C} \setminus \{c\}} F_{\mathcal{S}tr_k^{c'}}$ and decrease k by Δ_c which is the cost of making $F_{\mathcal{S}tr_k^c}$ convex. We have the following rule,

Rule 4. *Suppose c is an irrelevant color in F. Then, set $F = \bigcup_{c' \in \mathcal{C} \setminus \{c\}} F_{\mathcal{S}tr_k^{c'}}$ and decrease k by Δ_c.*

Suppose a vertex v with color c has weight greater than k, then in any recoloring with cost at most k, v cannot be recolored. In this situation, the vertices recolored in one component of $F - v$ do not affect the vertices recolored in another component of $F - v$. In other words, there are no k-strings, containing v, with color different from v's color. So, we can study the problem with respect to v and each component in $F - v$ independently. These independencies can be carried out by splitting v into a number of copies: as many as there are components in $F - v$ and recolor all vertices with color c in each of these components by a new color, unique for the component.

Rule 5. *Suppose there is a vertex v with color c and $\mu(v) > k$. Let T_v be the component containing v in F, let $F_0 = F - T_v$ and let T_1, ..., T_ℓ be the components in $T_v - v$ with w_1, ..., w_ℓ the neighbors of v in T_1, ..., T_ℓ respectively. Then, remove the vertex v from F, connect every w_i to a new vertex v_i with a new color c_i and weight $\mu_{\Gamma,c}(T_i)$, and recolor all vertices in T_i with color c to c_i (for each i, $1 \leq i \leq \ell$), add an isolated vertex v_0 with a new color c_0 and weight $\mu_{\Gamma,c}(F_0)$, and recolor all vertices in F_0 with color c to c_0.*

Lemma 4. *When Rules 1-5 cannot be applied, the total weight of vertices with a color c is at least two times the total weight of vertices in any piece of color c. I.e., for any piece W of color c,*

$$2\mu_{\Gamma,c}(W) \leq \mu_{\Gamma,c}(F).$$

Proof. When Rule 3 cannot be applied, for any piece W of color c, either $\mu_{\Gamma,c}(W) \leq \mu_{\Gamma,c}(F - W)$ or W consists of a single vertex w with $\mu_{\Gamma,c}(w) = k + 1$. In the first case, we get directly that $2\mu_{\Gamma,c}(W) \leq \mu_{\Gamma,c}(F)$. In the second case, Rule 5 applies, which gives again that $\mu_{\Gamma,c}(w) \leq \mu_{\Gamma,c}(F - w)$ □

At this point, the kernelization is almost completed. In the rest of this section, we assume that the next rule (Rule 6) is safe. Its safeness will be proved in the next section.

Rule 6. *If Rules 1-5 cannot be applied and $\sum_{c \in \mathcal{C}} \mu_{\Gamma,c}(F) > 6k^2$, return NO.*

From Rule 6, we know that there are at most $6k^2$ vertices with positive weight. It remains to show that the number of vertices with zero weight are also bounded. The following rules are clearly safe.

Rule 7. *If a vertex v has $\mu(v) = 0$ and deg$(v) \leq 1$, remove v.*

Rule 8. *If a vertex v has $\mu(v) = 0$ and deg$(v) = 2$, add an edge between its neighbors and remove v.*

It is well known that the number of vertices of degree greater than two in a tree is bounded by the number of leaves. So, because every vertex with zero weight has degree greater than 2 and every leaf has positive weight, the next result follows.

Theorem 1. *In a reduced instance, there are at most $12k^2$ vertices and the sum of its weights is at most $6k^2$.*

4.3 Removing k-Strings and Counting Them

A *skeleton* of $\mathcal{S}tr_k$ is a subset $\mathcal{R}$ of $\mathcal{S}tr_k$ containing a minimal number of k-strings of $\mathcal{S}tr_k$ in such a way that, if we denote the k-strings in $\mathcal{R}$ of color c by $\mathcal{R}^c$, for every $c \in \mathcal{C}$, $F_{\mathcal{R}^c} = F_{\mathcal{S}tr_k^c}$. A possible procedure for generating a skeleton of $\mathcal{S}tr_k$ can be the following: Initially, let $\mathcal{R} = \mathcal{S}tr_k$ and then, apply the following operation while possible, remove from $\mathcal{R}$ a k-string s of color c if it is contained in the rest of k-strings of color c in $\mathcal{R}$. Clearly, when the procedure ends, we have $F_{\mathcal{R}^c} = F_{\mathcal{S}tr_k^c}$ for every $c \in \mathcal{C}$. Let $Tag_{\mathcal{R}} = \bigcup_{s \in \mathcal{R}} Tag(s)$. We say that a vertex v is *tagged* by $\mathcal{R}$ if it is contained in $Tag_{\mathcal{R}}$.

Lemma 5. *In an instance reduced by Rule 1-5, for any color c in $\mathcal{C}$ and a skeleton $\mathcal{R}$ of $\mathcal{S}tr_k$,*

$$\mu_{\Gamma,c}(F - Tag_{\mathcal{R}}) \leq \mu_{\Gamma,c}(Tag_{\mathcal{R}}).$$

Proof. First, we prove that all the vertices of color c not belonging to $Tag_{\mathcal{R}}$ are contained in *one* piece of color c.

Note that a vertex of color c not belonging to $Tag_{\mathcal{R}}$ has to be in some piece of color c. Suppose two vertices x and y of color c are in different components of

$$F_c = F_{\mathcal{S}tr_k^c} - \bigcup_{c' \in \mathcal{C} \setminus \{c\}} F_{\mathcal{S}tr_k^{c'}} = F_{\mathcal{R}^c} - \bigcup_{c' \in \mathcal{C} \setminus \{c\}} F_{\mathcal{R}^{c'}}.$$

To be in different components of F_c, either there is a k-string s' in $\mathcal{R}^{c'}$ between them or x and y are in different components of $F_{\mathcal{S}tr_k^c} = F_{\mathcal{R}^c}$. In the first case, $Tag(s')$ contains x or y. In the second case, because by Rule 2, $F = F_{\mathcal{S}tr_k} = F_{\mathcal{R}}$, and therefore, either x and y have a k-string in $\mathcal{R}^{c'}$ for some c' different of c between them like in the first case or they are in different component of F. In such a case, any k-string with color different from c tags x or y. In any case, there is always a k-string in $\mathcal{R}$ tagging x or y and then, x and y must be in the same piece of color c.

Because all vertices with color c in $F - Tag(\mathcal{R})$ are contained in a piece of color c, by Lemma 4,

$$2\mu_{\Gamma,c}(F - Tag_{\mathcal{R}}) \leq \mu_{\Gamma,c}(F).$$

Using that

$$\mu_{\Gamma,c}(F) = \mu_{\Gamma,c}(F - Tag_{\mathcal{R}}) + \mu_{\Gamma,c}(Tag_{\mathcal{R}}),$$

we get

$$\mu_{\Gamma,c}(F - Tag_{\mathcal{R}}) \leq \mu_{\Gamma,c}(Tag_{\mathcal{R}}).$$ □

Lemma 6. *Rule 6 is safe.*

Proof. Suppose Γ' is a convex recoloring with $cost_{\Gamma}(\Gamma') \leq k$, we want to show that in this situation $\sum_{c \in \mathcal{C}} \mu_{\Gamma,c}(F) \leq 6k^2$. For this, we construct a skeleton $\mathcal{R}_{\Gamma'}$ of $\mathcal{S}tr_k$ in the following way, let $W_c = \{v \mid \Gamma(v) = \Gamma'(v) = c\}$:

– For every color c, add to $\mathcal{R}_{\Gamma'}$ a minimal number of k-strings of color c in such a way that the vertices in W_c are connected in $F_{\mathcal{R}_{\Gamma'}}$.
– For every color c, add to $\mathcal{R}_{\Gamma'}$ a minimal number of k-strings of color c in such a way that $F_{\mathcal{R}^c_{\Gamma'}} = F_{\mathcal{S}tr^c_k}$.

We separate the set $\mathcal{R}_{\Gamma'}$ into two parts: a subset $\mathcal{R}^{\mathcal{N}}_{\Gamma'}$ containing all the k-strings added in the first step and a subset $\mathcal{R}^{\mathcal{Y}}_{\Gamma'}$ containing the vertices added in the second step. In other words, $\mathcal{R}^{\mathcal{N}}_{\Gamma'}$ contains k-strings whose endpoints are not recolored, and $\mathcal{R}^{\mathcal{Y}}_{\Gamma'}$ contains k-strings with some endpoint recolored.

Claim. For every convex recoloring Γ' with $cost_\Gamma(\Gamma') \leq k$,

$$\sum_{c\in\mathcal{C}} \mu_{\Gamma,c}(Tag_{\mathcal{R}^{\mathcal{Y}}_{\Gamma'}}) \leq k^2$$

Proof of claim. In the way $Tag_{\mathcal{R}^{\mathcal{Y}}_{\Gamma'}}$ is constructed, for every recolored vertex at most one k-string is added to $Tag_{\mathcal{R}^{\mathcal{Y}}_{\Gamma'}}$. Because, $cost_\Gamma(\Gamma') \leq k$, in the second step we add at most k k-strings to $\mathcal{R}^{\mathcal{Y}}_{\Gamma'}$, i.e., $|\mathcal{R}^{\mathcal{Y}}_{\Gamma'}| \leq k$. From k-string definition we have $\sum_{c\in\mathcal{C}} \mu_{\Gamma,c}(Tag(s)) \leq k$. Putting all together,

$$\begin{aligned}\sum_{c\in\mathcal{C}} \mu_{\Gamma,c}(Tag_{\mathcal{R}^{\mathcal{Y}}_{\Gamma'}}) &= \sum_{c\in\mathcal{C}} \mu_{\Gamma,c}\Big(\bigcup_{s\in\mathcal{R}^{\mathcal{Y}}_{\Gamma'}} Tag(s)\Big) \\ &\leq \sum_{c\in\mathcal{C}} \sum_{s\in\mathcal{R}^{\mathcal{Y}}_{\Gamma'}} \mu_{\Gamma,c}(Tag(s)) \\ &= \sum_{s\in\mathcal{R}^{\mathcal{Y}}_{\Gamma'}} \sum_{c\in\mathcal{C}} \mu_{\Gamma,c}(Tag(s)) \\ &\leq |\mathcal{R}^{\mathcal{Y}}_{\Gamma'}|k \leq k^2. \end{aligned}$$ □

Claim. For every convex recoloring Γ' with $cost_\Gamma(\Gamma') \leq k$,

$$\sum_{c\in\mathcal{C}} \mu_{\Gamma,c}(Tag_{\mathcal{R}^{\mathcal{N}}_{\Gamma'}}) \leq 2k^2$$

Proof of claim. To prove the claim, we first reduce the set $\mathcal{R}^{\mathcal{N}}$ to a subset $\mathcal{R}^*$ of $\mathcal{R}^{\mathcal{N}}$ in the following way: Initially, let $\mathcal{R}^* = \mathcal{R}^{\mathcal{N}}$ and while possible, remove from $\mathcal{R}^*$ any k-string s such that $Tag(s) \subseteq \bigcup_{s'\in\mathcal{R}^*\setminus\{s\}} Tag(s')$. After the procedure is applied, the following two properties are held by $\mathcal{R}^*$, (1) $Tag_{\mathcal{R}^{\mathcal{N}}_{\Gamma'}} = Tag_{\mathcal{R}^*}$, and (2) for every k-string s in $\mathcal{R}^*$ there is a vertex ν_s such that $\nu_s \in Tag(s)$ and for any $s' \in \mathcal{R}^*$ different of s, $\nu_s \notin Tag(s')$. From the last property, we can associate to every string s of $\mathcal{R}^*$ a vertex ν_s not tagged by any other k-string in $\mathcal{R}^*$.

Let $\mathcal{C}^+ = \{c \in \mathcal{C} \mid \exists s \in \mathcal{R}^*, \Gamma(\nu_s) = c\}$. Note if c is the color of a vertex ν_s associated to a k-string s, at least one vertex of color c should be recolored, otherwise s (that is a k-string not recolored) is separating ν_s from $Bag_c(F - s)$

and then, Γ' cannot be convex. From previous argument, $\mathcal{C}^+$ has at most k colors, i.e., $|\mathcal{C}^+| \leq k$.

Let n_c be the number of k-strings in $\mathcal{R}^*$ with an associated vertex of color c. We show that at least $n_c - 1$ vertices of color c are recolored by Γ'. Suppose not. Then, there are two vertices ν_{s_1} and ν_{s_2} with color c for two k-strings s_1 and s_2 in $\mathcal{R}^*$. Because Γ' maintains ν_{s_1} and ν_{s_2} with color c, ν_{s_1} and ν_{s_2} are in the same component of $F_{\mathcal{S}tr_k^c}$. An then, the only way s_1 tags ν_{s_1} but not tag ν_{s_2} is because ν_{s_2} is in $Bag_c(F - s_1)$. Implying that s_1 and $p(\nu_{s_1}, \nu_{s_2})$, the path going from ν_{s_1} to ν_{s_2}, intersect in some vertex, contradicting the convexity of Γ'. Consequently, at least $n_c - 1$ vertices of color c are recolored by Γ'. So, $\sum_{c \in \mathcal{C}^+} n_c - 1 \leq k$. Because $|\mathcal{C}^+| \leq k$, we get $|\mathcal{R}^*| = \sum_{c \in \mathcal{C}^+} n_c \leq 2k$. Finally, putting all together,

$$\begin{aligned}
\sum_{c \in \mathcal{C}} \mu_{\Gamma,c}(Tag_{\mathcal{R}^{\mathcal{N}}_{\Gamma'}}) &= \sum_{c \in \mathcal{C}} \mu_{\Gamma,c}(Tag_{\mathcal{R}^*}) \\
&= \sum_{c \in \mathcal{C}} \mu_{\Gamma,c}(\bigcup_{s \in \mathcal{R}^*} Tag(s)) \\
&\leq \sum_{c \in \mathcal{C}} \sum_{s \in \mathcal{R}^*} \mu_{\Gamma,c}(Tag(s)) \\
&\leq \sum_{s \in \mathcal{R}^*} \sum_{c \in \mathcal{C}} \mu_{\Gamma,c}(Tag(s)) \\
&\leq |\mathcal{R}^*| k \leq 2k^2.
\end{aligned}$$

□

Using Lemma 5 with the skeleton $\mathcal{R}_{\Gamma'}$, we have that

$$\mu_{\Gamma,c}(F) = \mu_{\Gamma,c}(Tag_{\mathcal{R}_{\Gamma'}}) + \mu_{\Gamma,c}(F - Tag_{\mathcal{R}_{\Gamma'}}) \leq 2\mu_{\Gamma,c}(Tag_{\mathcal{R}_{\Gamma'}}).$$

Finally, by the previous claims,

$$\mu_{\Gamma,c}(F) \leq 2(\sum_{c \in \mathcal{C}} \mu_{\Gamma,c}(Tag_{\mathcal{R}^{\mathcal{Y}}_{\Gamma'}}) + \sum_{c \in \mathcal{C}} \mu_{\Gamma,c}(Tag_{\mathcal{R}^{\mathcal{N}}_{\Gamma'}})) \leq 6k^2.$$

□

We have shown that all rules are safe, and thus, by Theorem 1 and the fact that we can test in polynomial time if a rule can be carried out, and if so, apply it, we obtain:

Theorem 2. *There is a polynomial time kernelization algorithm for* Weighted Convex Tree Recoloring *that yields a kernel with at most* $12k^2$ *vertices whose total weight is at most* $6k^2$*.*

5 Conclusions

In this paper, we gave a kernel of quadratic size for the Weighted Convex Tree Recoloring problem. As we also allow weights that are zero, our result also implies a kernel for the case where we have initially some uncolored vertices. We have fewer rules than the result that we generalize from [3]. In particular,

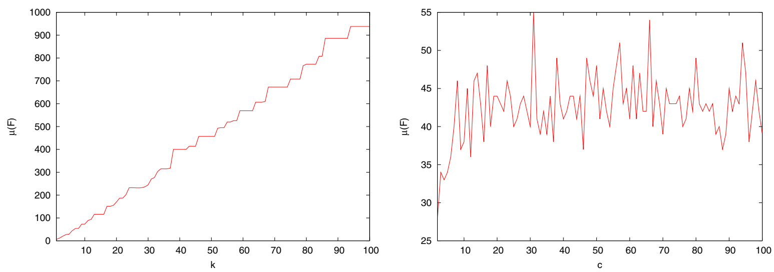

Fig. 2. On the left, we represent the maximum over 100 simulations, taking k until 100 and $n = 10k^2$. On the right, fixed $k = 5$ and $n = 500$, the maximum after 1000 simulations modifying c between 2 and 100.

we did not need most of the rules that we used in [3] to ensure that each color is given to a linear number of vertices. In a practical setting, it is not hard to generalize these rules from [3] to the weighted case, and add these as additional preprocessing heuristics to our rules. An intriguing open problem is whether a linear kernel exists for our problem, or, at least, for the unweighted case.

We implemented Rules 1-4 and applied these to randomly generated instances; in each case, we took a randomly generated tree and then recolored k randomly chosen vertices. The results of this experiment are shown in Figure 2. In these random instances, it seems that in a practically point of view, the size of the kernel is linear. Although, it is not difficult to construct instances such that its reduced instance grows quadratically on k. We leave such an analysis (or a different set of rules with a linear kernel) as open problem for further research.

References

1. Bachoore, E.H., Bodlaender, H.L.: Convex recoloring of leaf-colored trees. Technical Report UU-CS-2006-010, Department of Information and Computing Sciences, Utrecht University (2006)
2. Bar-Yehuda, R., Feldman, I., Rawitz, D.: Improved Approximation Algorithm for Convex Recoloring of Trees. Theory Comput. Syst. 43(1), 3–18 (2008)
3. Bodlaender, H.L., Fellows, M.R., Langston, M.A., Ragan, M.A., Rosamond, F.A., Weyer, M.: Quadratic Kernelization for Convex Recoloring of Trees. In: Lin, G. (ed.) COCOON 2007. LNCS, vol. 4598, pp. 86–96. Springer, Heidelberg (2007)
4. Downey, R.G., Fellows, M.R.: Parameterized Complexity. Springer, Heidelberg (1999)
5. Flum, J., Grohe, M.: Parameterized Complexity Theory. Springer, Secaucus (2006)
6. Gramm, J., Nickelsen, A., Tantau, T.: Fixed-Parameter Algorithms in Phylogenetics. The Computer Journal 51(1), 79–101 (2008)
7. Guo, J., Niedermeier, R.: Invitation to Data Reduction and Problem Kernelization. SIGACT News 38(1), 31–45 (2007)
8. Moran, S., Snir, S.: Efficient Approximation of Convex Recolorings. Journal of Computer and System Sciences 73(7), 1078–1089 (2007)

9. Moran, S., Snir, S.: Convex Recolorings of Strings and Trees: Definitions, Hardness Results and Algorithms. Journal of Computer and System Sciences 74(5), 850–869 (2008)
10. Niedermeier, R.: Invitation to Fixed Parameter Algorithms. Oxford Lecture Series in Mathematics and Its Applications. Oxford University Press, USA (2006)
11. Ponta, O., Hüffner, F., Niedermeier, R.: Speeding up Dynamic Programming for Some NP-Hard Graph Recoloring Problems. In: Agrawal, M., Du, D.-Z., Duan, Z., Li, A. (eds.) TAMC 2008. LNCS, vol. 4978, pp. 490–501. Springer, Heidelberg (2008)
12. Razgon, I.: A $2^{O(k)}$Poly(n) Algorithm for the Parameterized Convex Recoloring Problem. Information Processing Letters 104(2), 53–58 (2007)