Convex Recoloring Revisited: Complexity and Exact Algorithms

Iyad A. Kanj* Dieter Kratsch†

Abstract

We take a new look at the CONVEX PATH RECOLORING (CPR), CONVEX TREE RECOLORING (CTR), and CONVEX LEAF RECOLORING (CLR) problems through the eyes of the INDEPENDENT SET problem. This connection allows us to give a complete characterization of the complexity of all these problems in terms of the number of occurrences of each color in the input instance, and consequently, to present simpler NP-hardness proofs for them than those given earlier. For example, we show that the CLR problem on instances in which the number of leaves of each color is at most 3, is solvable in polynomial time, by reducing it to the INDEPENDENT SET problem on chordal graphs, and becomes NP-complete on instances in which the number of leaves of each color is at most 4.

This connection also allows us to develop improved exact algorithms for the problems under consideration. For instance, we show that the CPR problem on instances in which the number of vertices of each color is at most 2, denoted 2-CPR, proved to be NP-complete in the current paper, is solvable in time $2^{n/4}n^{O(1)}$ (n is the number of vertices on the path) by reducing it after $2^{n/4}$ enumerations to the weighted independent set problem on interval graphs, which is solvable in polynomial time. Then, using an exponential-time reduction from CPR to 2-CPR, we show that CPR is solvable in time $2^{4n/9}n^{O(1)}$. We also present exact algorithms for CTR and CLR running in time $2^{0.454n}n^{O(1)}$ and $2^{n/3}n^{O(1)}$, respectively. Our algorithms improve the previous algorithms for these problems.

1 Introduction

Given a tree T and a color function $\mathcal{C}$ assigning each vertex in T a color, the coloring $\mathcal{C}$ is said to be *convex* if, for each color $c \in \mathcal{C}$, the vertices of color c induce a subtree of T. The CONVEX TREE RECOLORING (CTR) problem is: given a tree T and a coloring $\mathcal{C}$—which is not necessarily convex, recolor the minimum number of vertices in T so that the resulting coloring is convex. The CTR problem has received considerable attention in the last few years [2, 3, 4, 5, 12, 13, 14, 15].

The CTR problem was first studied by Moran and Snir in [13] (journal version), and was motivated by applications in computational biology (see [13] for an extensive discussion on these applications). Moran and Snir [13] studied the problem from within a general model in which recoloring a vertex is associated with a nonnegative cost, and a convex recoloring that minimizes the total cost is sought. This model is referred to as the *weighted model*, whereas the model in which no weights are assigned (or all weights are the same) is referred to as the *unweighted model*. For the unweighted model, it was shown in [13] that the problem is NP-hard even for the simpler case when the tree is a path, referred to as the CONVEX STRING RECOLORING problem in [13], and as the CONVEX PATH RECOLORING (CPR) problem in this paper. Moran and Snir [13] also considered the CONVEX LEAF RECOLORING (CLR) problem, where only the leaves of T are assigned colors by $\mathcal{C}$, and a convex recoloring of T that minimizes the number of recolored leaves is sought. It was also shown in [13] that CLR is NP-hard under the unweighted model. The paper [13] also studied exact algorithms for solving CPR and CTR under the weighted model. The authors of [13] gave an algorithm that solves CPR in time $O(n \cdot n_c \cdot 2^{n_c})$, where n_c is the number of colors and n is the number of vertices on the path. They showed that this algorithm can be extended to trees to solve CTR in time $O(n \cdot n_c \cdot \Delta^{n_c})$, where Δ is the maximum degree of the tree. Moreover, they showed that the number of colors n_c in the previously mentioned algorithms, can be replaced with the number β

*School of Computing, DePaul University, 243. S. Wabash Avenue, Chicago, IL 60604, USA. ikanj@cs.depaul.edu.
†LITA, Université Paul Verlaine Metz, 57045 Metz Cedex 01, France. kratsch@univ-metz.fr.

of bad colors: those are the colors in $\mathcal{C}$ that do not induce subtrees (or subpaths in the case of a path) in T.

In a later paper [12], Moran and Snir studied the approximability of CPR and CTR under the weighted model. For CPR, they presented an algorithm of approximation ratio 2, running in time $O(n_c \cdot n)$, whereas for CTR, they presented an algorithm of approximation ratio 3, running in time $O(n_c \cdot n^2)$. Bar-Yehuda et al. [2] improved the approximation ratio for CTR to $(2+\epsilon)$; their algorithm runs in time $O(n^2+n/\epsilon^2+1/4^\epsilon)$.

From the parameterized complexity perspective, the CTR problem received a lot of attention [2, 3, 4, 5, 13, 14, 15] under both the weighted and the unweighted models. It was studied with respect to different parameters, including the number of colors n_c, the number of bad colors β, and the number of recolored vertices k, and was shown to be fixed-parameter tractable with respect to all these parameters [2, 4, 13, 14]. Among the notable FPT results for CTR we mention the $O(2^k kn + n^2)$ time algorithm in [2], the $2^\beta n^{O(1)}$ time algorithm in [14], and the $O(k^2)$ kernel upper bound (on the number of vertices) in [3].

In this paper we consider the CPR, CTR, and CLR problems under the unweighted model. We take a new look at these problems through the eyes of the INDEPENDENT SET problem. This connection to INDEPENDENT SET, together with the structural results developed in this paper, allow us to give a complete characterization of the complexity of each of these three problems with respect to the maximum number of occurrences of each color in the input instance. For example, via a simple (polynomial-time) reduction from INDEPENDENT SET, we show that the CPR problem on instances in which the number of vertices of each color is at most 2, denoted 2-CPR, remains NP-complete. (Note that the CPR problem is obviously solvable in polynomial time on instances in which the number of occurrences of each color is 1.) This provides a simpler NP-hardness proof for CPR than that in [13], and characterizes the complexity of the problem modulo the number of occurrences of each color. We note that the reduction in [13] from 3-SAT to CPR produces instances in which the number of vertices of a given color is unbounded. Note also that the complexity results for CPR carry over to CTR (every path is a tree). For CLR, we show that it is solvable in polynomial time when the number of leaves of each color is at most 3, denoted 3-CLR, by reducing it to the weighted INDEPENDENT SET problem on chordal graphs, which is solvable in polynomial time [7]. On the other hand, we show that CLR becomes NP-complete when the number of vertices of each color is at most 4, by a reduction from INDEPENDENT SET.

In addition, this connection to INDEPENDENT SET allows us to develop improved exact algorithms for CPR, CTR, and CLR, by reducing them—after some enumerations—to the INDEPENDENT SET problem on restricted classes of graphs. For example, we show that the 2-CPR problem is solvable in time $2^{n/4}n^{O(1)}$, where n is the number of vertices in the path, by reducing the problem after $O(2^{n/4})$ enumerations to the weighted INDEPENDENT SET problem on interval graphs, which is polynomial-time solvable [9]. This algorithm for 2-CPR, coupled with an exponential-time reduction from CPR to 2-CPR, give an exact algorithm for CPR running in time $2^{4n/9}n^{O(1)}$. Similarly for CTR, we can show that it can be solved in time $2^{0.454n}n^{O(1)}$. For CLR, the polynomial-time solvability for 3-CLR proved in this paper, together with an exponential time reduction from CLR to 3-CLR, give an algorithm running in time $2^{n/3}n^{O(1)}$.

We note that all the above algorithms improve on the algorithms in [13] for the corresponding problems, and on the $2^{n/2}n^{O(1)}$ algorithm of Ponta et al. [14], implied by their $2^\beta n^{O(1)}$ algorithm, after observing that the number of bad colors β is bounded by $n/2$.

2 Preliminaries

We refer the reader to [16] for the basic graph terminologies and notations, and to [10] for some of the facts and results on special graph classes that are used in this paper.

For an asymptotically positive integer-function $t(n)$, we will use the asymptotic notation $O^*(t(n))$ to denote time complexity of the form $O(t(n).p(n))$, where $p(n)$ is a polynomial.

Let T be a tree, and let $\mathcal{C}$ be a function assigning each vertex in T a color in $\{c_1, \ldots, c_p\}$. The coloring $\mathcal{C}$ is said to be *convex*, if for every color $c \in \{c_1, \ldots, c_p\}$, the set of vertices in T of color c induces a subtree of T. If T is given with a coloring $\mathcal{C}$ of its vertices, we can view any other coloring $\mathcal{C}'$ of $V(T)$ as a recoloring of $\mathcal{C}$. For a vertex $v \in V(T)$, we say that $\mathcal{C}'$ *retains* the color of v if $\mathcal{C}'(v) = \mathcal{C}(v)$; otherwise, we say that $\mathcal{C}'$ *recolors* v. If c is a color assigned by $\mathcal{C}$, we say that the recoloring $\mathcal{C}'$ *retains* the color c if there exists at least one vertex $v \in T$ such that $\mathcal{C}'(v) = c$.

The CONVEX TREE RECOLORING problem, abbreviated CTR, is defined as follows: Given a tree T on n vertices and a function $\mathcal{C}$ assigning each vertex in T a color in $\{c_1, \ldots, c_p\}$, compute a convex recoloring of T that recolors the minimum number of vertices.

The CONVEX PATH RECOLORING problem, abbreviated CPR, is defined as follows: Given a path P on n vertices and a function $\mathcal{C}$ assigning each vertex in P a color in $\{c_1, \ldots, c_p\}$, compute a convex recoloring of P that recolors the minimum number of vertices.

The CONVEX LEAF RECOLORING problem, abbreviated CLR, is defined as follows: Given a tree T with n leaves and a function $\mathcal{C}$ assigning each leaf in T a color in $\{c_1, \ldots, c_p\}$, compute a convex recoloring of T that recolors the minimum number of leaves.

3 The CPR problem

In this section we give a simple proof showing that the CPR problem, even when restricted to instances having at most two vertices from each color, denoted by the 2-CPR problem, remains NP-complete. This provides an alternative—yet simple—proof of the NP-completeness of CPR. We also give an exact algorithm for the 2-CPR problem. This algorithm for 2-CPR, together with an exponential-time reduction from CPR to 2-CPR, yield an exact algorithm for CPR with the best running time.

Let $(P, \mathcal{C})$ be an instance of 2-CPR. Call a color c a *singleton color* if $\mathcal{C}(v) = c$ for exactly one vertex $v \in P$; otherwise, call c an *interval color*. If c is an interval color, we call the two vertices on P with color c *mates*. For two mate vertices u and v on P, we call the subpath between u and v on P an *interval*, and refer to it by $[u, v]$. The results in this section rely heavily on the following key structural result:

Lemma 3.1 (The Exchange Lemma) *Let $(P, \mathcal{C})$ be an instance of 2-CPR, and suppose that there exists a convex recoloring of P that recolors at most ℓ vertices. Then there exists a convex recoloring of P that recolors at most ℓ vertices and that retains every color assigned by $\mathcal{C}$.*

PROOF. Let $\mathcal{C}'$ be a convex recoloring of P that recolors at most ℓ vertices such that the number of colors retained by $\mathcal{C}'$ (with respect to $\mathcal{C}$) is maximum, over all convex recolorings of P that recolor at most ℓ vertices. We will show that $\mathcal{C}'$ retains every color assigned by $\mathcal{C}$.

Suppose that there exists a vertex v on P such that $\mathcal{C}(v)$ is not retained by $\mathcal{C}'$. Then there must exist two vertices u and w such that $\mathcal{C}'(u) = \mathcal{C}'(w)$, and such that v is in the interval $[u, w]$; otherwise, the two vertices adjacent to v on P are assigned two different colors by $\mathcal{C}'$, and by assigning v the color $\mathcal{C}(v)$, we would still obtain a convex recoloring of P that recolors at most ℓ vertices but retains more colors than $\mathcal{C}'$, contradicting the maximality of $\mathcal{C}'$. Without loss of generality, let u and w be the two such vertices that are farthest apart (with the maximum distance on P). Note that, by the convexity of $\mathcal{C}'$, all the vertices on $[u, w]$ are assigned by $\mathcal{C}'$ the same color as u and w. We can assume, without loss of generality, that at least one vertex y in $[u, w]$ satisfies $\mathcal{C}'(y) = \mathcal{C}(y)$. Assume that $y \in [u, v]$; the case when $y \in [v, w]$ can be handled in a similar fashion. Let $\langle z_1, \ldots, z_r = w\rangle$ be $[u, w]$ excluding u. Modify $\mathcal{C}'$ to obtain a recoloring $\mathcal{C}''$ by applying the following recoloring procedure to $\langle z_1, \ldots, z_r\rangle$: Initialize $\mathcal{C}''$ to $\mathcal{C}'$; for $i = 1, \ldots, r$: if the color $\mathcal{C}(z_i)$ is not retained by $\mathcal{C}''$, modify $\mathcal{C}''$ by recoloring z_i with $\mathcal{C}(z_i)$, otherwise, set $\mathcal{C}''(z_i) = \mathcal{C}''(z_{i-1})$.

First, it is easy to see that $\mathcal{C}''$ is a convex recoloring of P. Second, the number of colors retained by $\mathcal{C}''$ (with respect to $\mathcal{C}$) is more than that retained by $\mathcal{C}'$. This is true because vertex v retains its color by $\mathcal{C}''$ but not by $\mathcal{C}'$, and every color that is retained by $\mathcal{C}'$ is also retained by $\mathcal{C}''$, by the nature of the recoloring procedure. Finally, the number of vertices recolored by $\mathcal{C}''$ is not more than that recolored by $\mathcal{C}'$. This follows from the facts that at most two vertices on $[u, w]$ retained their color by $\mathcal{C}'$, vertex v retains its color by $\mathcal{C}''$ but not by $\mathcal{C}'$, and either vertex y retains its color by $\mathcal{C}''$, or a vertex in $[u, y]$ retains its color by $\mathcal{C}''$ (but not by $\mathcal{C}'$). It follows that $\mathcal{C}''$ is a convex recoloring of P that recolors at most ℓ vertices and retains more colors than $\mathcal{C}'$. This contradicts the maximality of $\mathcal{C}'$, and completes the proof. $\square$

3.1 2-CPR is NP-complete

The decision version of 2-CPR is formulated as follows. Given an instance $(P, \mathcal{C})$ of 2-CPR and a nonnegative integer-parameter ℓ, decide if P has a convex recoloring that recolors at most ℓ vertices on P.

Theorem 3.2 *The* 2-CPR *problem is NP-complete.*

Proof. The proof that 2-CPR is in NP is straightforward, and is omitted.

To show that 2-CPR is NP-hard, we show that the NP-hard problem independent set [8] is polynomial-time reducible to 2-CPR.

Given an instance (G, k) of independent set, where G has $n(G)$ vertices and $m(G)$ edges, we construct an instance $(P, \mathcal{C})$ of 2-CPR as follows.

For every vertex $v \in G$, we associate two vertices v_1 and v_2 on P; we will refer to the interval $[v_1, v_2]$ on P by the *vertex-interval* corresponding to v. We place the vertex-intervals corresponding to the vertices in G in such a way that, for any two distinct vertices in G, their two corresponding vertex-intervals do not intersect. For every vertex-interval on P, we color its two endpoints with the same unique color, that is, this color will be distinct from the color of any other vertex on P. For an edge $e = (u, v) \in G$, we create two vertices u_e and v_e on P, color u_e and v_e with the same unique color, place u_e arbitrarily in the vertex-interval (i.e., between the two endpoints of the interval) corresponding to u on P, and place v_e arbitrarily in that corresponding to v. We will refer to the subpath of P between the two vertices corresponding to an edge in G by an *edge-interval*. For any two vertex-intervals that are consecutive on P, we insert between them a large number N of vertices, each of a unique color on P, that we call *buffer vertices*. The number N can be chosen to be any number larger than $m(G) + n(G) - k$. (The reason behind this choice will become clear later in the proof.) The role of the buffer vertices will be to force any desired convex recoloring of P from retaining the color of both endpoints of any edge-interval. Note that there will be a total of $(n(G)-1) \cdot N$ buffer vertices, each of which has a unique color on P. Finally, we set the parameter $\ell := m(G) + n(G) - k$. This completes the construction of the instance $(P, \mathcal{C})$. Clearly, the above transformation can be carried out in polynomial time. We will show that the above transformation is a valid reduction from independent set to 2-CPR.

Suppose first that G has an independent set I of size at least k. We construct the following recoloring $\mathcal{C}_I$ of P. For every vertex $v \in I$, $\mathcal{C}_I$ retains the color of both endpoints of its corresponding vertex-interval on P, and recolors all the vertices in that interval with the same color as its endpoints. Moreover, the coloring $\mathcal{C}_I$ will retain the color of every buffer vertex on P. Finally, for each vertex-interval corresponding to a vertex in $V(G) \setminus I$, and for each edge-interval whose both endpoints still retain their color after the recoloring of the vertex-intervals corresponding to the vertices in I, we color exactly one of its endpoints—and hence retain the color of the other endpoint—in such a way that the resulting recoloring is convex. This can be done, for example, by scanning P from one extremity to the other, and retaining the color of the first encountered endpoint of each such interval, and recoloring the second endpoint of the interval with the color of the vertex preceding it (with respect to the order in which we scanned the vertices). It is not difficult to see that the resulting recoloring is convex. Now to count how many vertices were recolored by $\mathcal{C}_I$, observe first that every buffer vertex retains its color by $\mathcal{C}_I$. Second, at most $n(G) - k$ vertices corresponding to endpoints of vertex-intervals have been recolored (during the scanning of P): one endpoint for each vertex-interval corresponding to a vertex in $V(G) \setminus I$. Moreover, since no edge exists between any two vertices in I, at most one endpoint of any edge-interval was recolored during the coloring of the endpoints of vertex-intervals that correspond to vertices in I. Since when the path P is scanned at the end we only recolor exactly one endpoint from every edge-interval such that none of its endpoints was recolored before, during the whole process, we recolored exactly one endpoint from each edge-interval. Consequently, the total number of endpoints of edge-intervals that were recolored by $\mathcal{C}_I$ is exactly $m(G)$. We conclude that the coloring $\mathcal{C}_I$ totally recolors at most $m(G) + n(G) - k$ vertices on P.

Conversely, suppose that $(P, \mathcal{C})$ admits a convex recoloring $\mathcal{C}'$ that recolors at most $m(G) + n(G) - k$ vertices on P. By **The Exchange Lemma** (Lemma 3.1), we can assume that every color in $\mathcal{C}$ is retained by $\mathcal{C}'$. In particular, the color of every buffer vertex on P is retained. Since the endpoints of any edge-interval are separated by $N > m(G) + n(G) - k$ buffer vertices, at least one endpoint from every edge-interval is recolored by $\mathcal{C}'$. Therefore, at least m endpoints of edge-intervals are recolored by $\mathcal{C}'$. Since the total number of vertices recolored by $\mathcal{C}'$ is at most $m(G) + n(G) - k$, the color of at least k vertex-intervals are retained by $\mathcal{C}'$; let I be the set of vertices in G corresponding to these vertex-intervals in P, and note that $|I| \geq k$. Since at least one endpoint from every edge-interval was recolored by $\mathcal{C}'$, no edge-interval on P has its endpoints situated in two vertex-intervals corresponding to two vertices in I (otherwise, both endpoints of such an edge-interval would be recolored by $\mathcal{C}'$). Consequently, no edge in G exists between any two vertices in I, and I is an independent set in G containing at least k vertices. □

Corollary 3.3 *The* CPR *problem is NP-complete.*

3.2 An exact algorithm for 2-CPR

Let $(P, \mathcal{C})$ be an instance of 2-CPR. Let u and v be two vertices on P that are mates. We call an interval $[x, y]$ a *long interval* if there exists an interval $[u, v]$ such that $[u, v]$ is contained in $[x, y]$ (i.e., the path from u to v is a subpath of that from x to y), or if there exists a vertex w contained in $[x, y]$ such that $\mathcal{C}(w)$ is a singleton color; otherwise, we call $[x, y]$ a *short interval*. Let N_{short} be the number of short intervals on P, N_{long} that of long intervals, and note that $N_{short} + N_{long} \leq n/2$, since the total number of intervals on P is at most $n/2$.

By **The Exchange Lemma** (Lemma 3.1), we may assume that an optimal convex recoloring of P (i.e., a convex recoloring of P that recolors the minimum number of vertices), $\mathcal{C}_{opt}$, retains every color assigned by $\mathcal{C}$. In particular, for every vertex v on P whose color is a singleton color, we have $\mathcal{C}_{opt}(v) = \mathcal{C}(v)$. Moreover, for every long interval $[x, y]$, exactly one of its endpoints will retain its color by $\mathcal{C}_{opt}$. This is true because at least one of the endpoints of the interval $[x, y]$ retains its color by $\mathcal{C}_{opt}$, and at most one endpoint of $[x, y]$ retains its color (otherwise, there exists an interval $[u, v]$ inside $[x, y]$ such that none of the vertices in $\{u, v\}$ retains its color by $\mathcal{C}_{opt}$). Therefore, in the remainder of this subsection, when enumerating recolorings of P, we are only interested in those that retain every color on P, and we may work under this assumption.

Lemma 3.4 *An optimal convex recoloring for $(P, \mathcal{C})$ can be computed in time $O^*(2^{N_{short}})$.*

PROOF. For every short interval $[u, v]$, we enumerate whether or not the colors of both its endpoints u and v are retained by an optimal convex recoloring $\mathcal{C}_{opt}$; if the colors of both endpoints are retained, we assign the interval $[u, v]$ the status *kept*, otherwise, we assign it the status *unkept*. After enumerating the status of every short interval, we check the validity of this enumeration by checking that no interval on P has each of its endpoints inside an interval whose status is kept (otherwise, the color of such an interval cannot be retained). If the enumeration is valid, we compute a convex recoloring of P as follows. First, for every short interval $[u, v]$ whose status is kept, we color all the vertices in $[u, v]$ with the color $\mathcal{C}(u) = \mathcal{C}(v)$, and we shrink the whole interval to a single vertex whose color ($\mathcal{C}(u) = \mathcal{C}(v)$) becomes a singleton color. After this process, we have reduced the instance to an instance consisting of vertices whose colors are singleton colors, and hence need to be retained, and intervals from which exactly one endpoint must retain its color. Then we scan the resulting path from one extremity to the other. If we encounter a vertex whose color is a singleton color, we retain its color. If we encounter the first endpoint of an interval, we retain its color, and if we encounter the second endpoint of an interval, we recolor it with the color of the vertex preceding it (in the scanning order). It is not difficult to see that the resulting recoloring is convex. Note that the number of vertices recolored by the above process is precisely equal to the number of short intervals that are unkept, plus the number of long intervals. For each valid enumeration, we keep track of the number of vertices recolored. Eventually, we output the valid enumeration, and hence the convex recoloring, corresponding to the minimum number of recolored vertices.

It is clear that the above algorithm yields an optimal solution. Since for each short interval we enumerate two possibilities (kept or unkept), the total number of enumerations is $O(2^{N_{short}})$. Checking the validity of each enumeration, and extending a valid enumeration to a convex recoloring, take polynomial time. It follows that an optimal convex recoloring for $(P, \mathcal{C})$ can be computed in time $O^*(2^{N_{short}})$. $\square$

Lemma 3.5 *An optimal convex recoloring for $(P, \mathcal{C})$ can be computed in time $O^*(2^{N_{long}})$.*

Proof. For each long interval $[x, y]$, we enumerate which vertex in $\{x, y\}$ retains its color in an optimal convex recoloring of P. Note that exactly one vertex in $\{x, y\}$ retains its color, under the assumption that every color on P is retained by the desired enumerated recoloring. If x (resp. y) retains its color, we keep x (resp. y) and remove y (resp. x), because y (resp. x) needs to be recolored, and we can always recolor it with the color of a neighboring vertex, once a convex recoloring of the resulting path has been computed. Note that this procedure might result in a vertex whose color becomes a singleton color (because its color is retained, while its mate in a long interval needs to be recolored and has been removed). For each vertex whose color becomes a singleton color after the above enumeration, and for each interval containing it, we change the status of this interval and make it a long interval. We note that no long interval at this point contains another interval; this property will be crucial for the remaining part of the proof to go through.

Let P' be the resulting path at the end of this process. Then P' consists of: (1) vertices whose colors are singleton colors, (2) long intervals which contain at least one vertex of a singleton color but no nested intervals, and (3) short intervals. Let N'_{short} be the number of short intervals on P', and N'_{long} that of long intervals. Note that if a convex recoloring of P' (that retains every color) retains the color of both endpoints of exactly k short intervals, then the total number of vertices on P' that need to be recolored by this recoloring is exactly $N'_{long} + N'_{short} - k$. Note also that no two short intervals whose colors are retained by a convex recoloring can contain both endpoints of another interval (otherwise, its color would not be retained), and no two such short intervals can intersect. This implies that the set of short intervals whose colors are retained by a convex recoloring of P' (that retains every color) corresponds to an independent set of the same size in the square of the interval graph $G_{P'}$, defined naturally as follows: For every short interval on P' associate a vertex in $G_{P'}$ of weight 1. For every long interval on P' associate a vertex of weight 0. Two vertices in $G_{P'}$ are adjacent if and only if their corresponding intervals on P' intersect.

If a convex recoloring $\mathcal{C}'$ of P' retains the colors of both endpoints of k short intervals on P', and hence recolors exactly $N'_{long} + N'_{short} - k$ vertices on P', then it is easy to verify that these k short intervals correspond to an independent set of weight k in the square of $G_{P'}$. On the other hand, if I is an independent set in G'_P of weight k, then I contains exactly k vertices, each of weight 1, whose corresponding short intervals on P' are of pairwise distance at least 2 (i.e., no two short intervals on P' whose corresponding vertices are in I' intersect, and no interval on P' intersects with two intervals whose corresponding vertices are in I). Therefore, by retaining the color of every short interval corresponding to a vertex in I, and recoloring exactly one endpoint from every other interval on P', we obtain a convex recoloring of P' that recolors $N'_{long} + N'_{short} - k$ vertices on P'. This shows that an optimal convex recoloring of P' corresponds to a maximum-weight independent set in the square of $G_{P'}$, and vice versa. Since the square of an interval graph is also an interval graph [1, 11], and since the WEIGHTED INDEPENDENT SET problem is solvable in polynomial time on interval graphs [9], computing an optimal convex recoloring after guessing the status of the long intervals on P can be done in polynomial time.

The analysis of the number of enumerations is analogous to that of Lemma 3.4. We conclude that an optimal convex recoloring for $(P, \mathcal{C})$ can be computed in time $O^*(2^{N_{long}})$. This completes the proof. $\square$

Theorem 3.6 *The* 2-CPR *problem can be solved in time $O^*(2^{n/4})$.*

Proof. This follows from Lemma 3.4 and Lemma 3.5, and the fact that $N_{short} + N_{long} \leq n/2$ (hence, either N_{short} or N_{long} is at most $n/4$). $\square$

3.3 An exact algorithm for CPR

Let $(P, \mathcal{C})$ be an instance of CPR. Recall that a color c is *bad* [13, 14] if the vertices on P of color c do not form a subpath of P. Clearly, no singleton color is a bad color, and hence, the number of bad colors N_{bad} is at most $n/2$.

The notion of a bad color was defined for the CTR problem (by replacing subpath with subtree). The results in [14] showed that the CTR problem can be solved in time $O^*(2^{N_{bad}}) = O^*(2^{n/2})$. This shows that the CPR problem can be solved in $O^*(2^{n/2})$ time as well. In this subsection, we shall improve on this upper bound by reducing the CPR problem to the 2-CPR problem, and then using the results of Subsection 3.2.

Let N_1 be the number of singleton colors on P, N_2 that of interval colors, $N_{>2}$ that of colors that appear at least three times on P (i.e., appear on at least 3 vertices on P). For each color c that appears at least three times on P, we fix any two vertices of color c on P and call them *stationary* vertices, and we call each other vertex of color c an *excess* vertex. Let N_e be the number of excess vertices on P, and note that $N_{>2} \leq N_e$. We have the following equality:

$$N_1 + 2N_2 + 2N_{>2} + N_e = n. \tag{1}$$

Consider the following algorithm $\mathcal{A}$ that solves the CPR problem. For every excess vertex, enumerate whether the color of this vertex is retained by an optimal convex recoloring of P or not. In addition, for each color that appears at least 3 times on P, pick one stationary vertex of that color (arbitrarily) and enumerate whether the color of that vertex is retained or not by an optimal convex recoloring of P. Note that, if, for a color c, two vertices of color c that are retained by this enumeration are separated by a vertex of color c whose status is not retained, then we can reject this enumeration. Moreover, no two vertices of the same color, whose color is retained, can be separated by a vertex of different color whose color is either retained or is a singleton color; otherwise, we reject the enumeration.

For each vertex whose color is not retained, we remove the vertex because its color can be determined once an optimal convex recoloring has been computed (such a vertex can be recolored with the color of one of its neighbors). Afterwards, for each color c such that at least two vertices of color c retain their color under the enumeration, we pick the two vertices of color c that are farthest apart on P, color all the vertices between them with color c, and shrink all these vertices to a single vertex whose color becomes a singleton color. After this operation, no color appears more than twice on P, and we end up with an instance $(P', \mathcal{C}')$ of the 2-CPR problem; let n' be the number of vertices on P'.

The number of vertices enumerated by the above procedure is $N_e + N_{>2}$. Therefore, the total number of enumerations in the above procedure is bounded by $2^{N_e+N_{>2}} \leq 2^{2N_e}$. After that, we apply the algorithm in Subsection 3.2 to the instance $(P', \mathcal{C}')$, whose number of vertices n' is at most $n - N_e$ (at most 2 vertices from each color remain), which takes time $O^*(2^{(n-N_e)/4})$. Therefore, algorithm $\mathcal{A}$ solves the CPR problem in time $O^*(2^{(n+7N_e)/4})$.

On the other hand, the number of bad colors is $N_2 + N_{>2} \leq (n - N_e)/2$ by Equation (1). Therefore, using the results in [14], we can solve the CPR problem by an algorithm $\mathcal{A}'$ in time $O^*(2^{(n-N_e)/2})$. This suggests the following algorithm. If $N_e > n/9$, we apply the algorithm $\mathcal{A}'$ that solves the CPR problem in time $O^*(2^{4n/9})$, and if $N_e \leq n/9$, we apply the algorithm $\mathcal{A}$ which solves the CPR problem in time $O^*(2^{4n/9})$. Therefore, we have the following theorem:

Theorem 3.7 *The* CPR *problem is solvable in time* $O^*(2^{4n/9})$.

4 The CTR problem

Similarly to the definition of the 2-CPR problem, we can define the 2-CTR problem to be the set of instances $(T, \mathcal{C})$ of CTR such that, for every color c, the number of vertices in T of color c is at most 2.

The NP-completeness of 2-CTR, and hence of CTR, follows from the NP-completeness of 2-CPR established by Theorem 3.2, since a path is a special case of a tree.

4.1 An exact algorithm for 2-CTR

The general approach is very similar to that of 2-CPR. Therefore, to avoid repetition and redundancy, we will only explain the differences between the two algorithms. Let $(T, \mathcal{C})$ be an instance of 2-CTR.

First, **The Exchange Lemma** (Lemma 3.1) for 2-CPR carries over to 2-CTR: for every convex recoloring of T that recolors at most ℓ vertices, there exists a convex recoloring of T that recolors at most ℓ vertices and that retains every color in T. Second, we can define the notions of short and long intervals similarly: an interval is *long* if it either contains a vertex whose color is a singleton color, or if it contains another interval; otherwise, the interval is *short*. Here the interval determined by two vertices on T that are mates is defined to be the unique path in T between the two vertices. The statement of Lemma 3.4 carries over: If N_{short} is the number of short intervals on T, then a similar proof to that of Lemma 3.4 shows that 2-CTR is solvable in time $O^*(2^{N_{short}})$.

Unfortunately, the statement of Lemma 3.5 does not carry over to 2-CTR. The reason being that the auxiliary graph $G_{T'}$ ($G_{P'}$ in the case of 2-CPR), where T' is the resulting tree from T after enumerating every long interval (in a similar fashion to that of 2-CPR) is no longer an interval graph, and using a polynomial-time algorithm for computing a maximum weighted independent set on the square of $G_{T'}$ is no longer an option[1]. We deal with this issue by changing the definition of $G_{T'}$, and reducing the problem to that of computing a maximum (in terms of cardinality) independent set on the newly-defined auxiliary graph $G_{T'}$. To compute a maximum independent set in $G_{T'}$, we now use one of the exact (exponential-time) algorithms for computing a maximum independent set on a general graph, namely the polynomial-space algorithm by Fomin et al. [6], which computes a maximum independent set in a graph of N vertices in time $O^*(2^{0.288N})$. We present the modified version of Lemma 3.5 next.

Lemma 4.1 *An optimal convex recoloring for $(T, \mathcal{C})$ can be computed in time $O^*(2^{0.712N_{long}+0.144n})$.*

Proof. For each long interval, we start by enumerating which of its endpoints retains its color. The endpoint that does not retain its color is removed. The endpoint retaining its color remains, and becomes a vertex whose color is a singleton color; note again that every short interval containing such a vertex becomes a long interval.

Let T' be the tree resulting form T at the end of this enumeration. We define the auxiliary graph $G_{T'}$ as follows. For each short interval in T', we correspond a vertex in $G_{T'}$. Two vertices u and v in $G_{T'}$ are adjacent if and only if: (1) their corresponding short intervals in T' intersect, or (2) there exists an interval $[x, y]$ in T' such that each of the intervals corresponding to u and v in T' contains one endpoint of $[x, y]$. By a similar argument to that made in Lemma 3.5, it can be shown that a maximum independent set in $G_{T'}$ corresponds to an optimal convex recoloring of T'.

Enumerating all long intervals takes $O^*(2^{N_{long}})$ time. Computing a maximum independent set in $G_{T'}$ takes time $O^*(2^{0.288N_{short}})$ using the algorithm in [6], after noting that the number of vertices in $G_{T'}$ is at most N_{short}.

Therefore, an optimal convex recoloring for $(T, \mathcal{C})$ can be computed in time $O^*(2^{N_{long}+0.288N_{short}})$. The statement of the lemma follows after noting that $N_{long} + N_{short} \leq n/2$. $\square$

Theorem 4.2 *The* 2-CTR *problem can be solved in time $O^*(2^{0.293n})$.*

Proof. If $N_{short} \leq 0.293n$, then the statement follows from the fact that 2-CTR is solvable in time $O^*(2^{N_{short}})$ (see the discussion at the beginning of this subsection). Otherwise, from the fact that $N_{short} + N_{long} \leq n/2$, we derive $N_{long} \leq 0.207n$. Now the statement follows from Lemma 4.1. $\square$

4.2 An exact algorithm for CTR

The idea (again) is to reduce CTR to 2-CTR, and then to use Theorem 4.2. The reduction is very similar to that from CPR to 2-CPR, described in Subsection 3.3. We highlight the differences.

For each color c, we consider the excess vertices of color c, in addition to exactly one stationary vertex of color c. For each considered vertex, we enumerate whether it retains its color or not by an optimal convex recoloring. If a vertex does not retain its color, we remove it. If two vertices u and v of color c both retain their colors, we contract the unique path P_{uv} between the two vertices on the tree. This contraction

[1]It can be proved that independent set is NP-complete on squares of chordal graphs.

is justified by the fact that, assuming that the enumeration is correct, all vertices on P_{uv} must be colored with the same color as that of u and v in a convex recoloring. Note that this contraction transforms the path P_{uv} into a single vertex whose neighbors are the vertices in T that do not appear on P_{uv}, but are adjacent to some vertex on P_{uv}. After repeatedly contracting every path between two vertices of the same color that retain their colors by the enumeration, we end up with an instance of CTR in which each color appears on at most two vertices, and hence, we end up with an instance of 2-CTR. To this instance we apply the algorithm described in Subsection 4.1 (Theorem 4.2). By performing an analysis similar to that in Subsection 3.3, we can derive the following result:

Theorem 4.3 *The* CTR *problem is solvable in time* $O^*(2^{0.454n})$.

5 The CLR problem

In this section we show that 3-CLR (each color appears on at most 3 leaves) is solvable in polynomial time, and use this fact to develop an exact algorithm for the CLR problem by reducing it to 3-CLR. We can also show that the 4-CLR problem is NP-complete, where 4-CLR consists of the set of instances of CLR in which each color appears on at most 4 leaves; the proof is given in the appendix (Theorem 7.1). This implies the NP-hardness of CLR, a result already proved in [13]. Not only do our results provide a simpler proof than that in [13] for the NP-completeness of CLR, they also provide a complete characterization of the complexity of the problem with respect to the maximum number of occurrences of each color.

Let $(T, \mathcal{C})$ be an instance of CLR. We first observe a stronger version of **The Exchange Lemma** for CLR, than those for 2-CPR and 2-CTR.

Observation 1 *Let $\mathcal{C}_{opt}$ be a convex recoloring of $\mathcal{C}$ that recolors the minimum number of leaves in T. Then $\mathcal{C}_{opt}$ retains every color in $\mathcal{C}$.*

PROOF. If c is a color in $\mathcal{C}$ that is not retained by $\mathcal{C}_{opt}$, let v be any leaf in T such that $\mathcal{C}(v) = c$. Let $\mathcal{C}'$ be the recoloring of $\mathcal{C}$ that is identical to $\mathcal{C}_{opt}$, except at v, where $\mathcal{C}'(v) = \mathcal{C}(v) = c$. Clearly, $\mathcal{C}'$ is a convex recoloring of $\mathcal{C}$ that recolors fewer leaves than $\mathcal{C}_{opt}$, contradicting the maximality of $\mathcal{C}_{opt}$. □

5.1 A polynomial-time algorithm for 3-CLR

In this subsection we show that the 3-CLR problem is solvable in polynomial time by reducing it (in polynomial time) to the problem of computing a maximum weighted independent set in a chordal graph. The latter problem is known to be solvable in polynomial time [7].

Theorem 5.1 *The* 3-CLR *problem is solvable in polynomial time.*

PROOF. Let $(T, \mathcal{C})$ be an instance of 3-CLR, and let n_c be the number of colors that appear in T. Call a color that appears on 3 leaves in T a *tripodal* color. We construct a graph G_T as follows.

For each interval color c that appears on two leaves x and y in T, we associate a vertex v_{xy} of weight 1 in G_T that we call a *path vertex*. For each tripodal color c that appears on three leaves x, y, and z in T, we associate 4 vertices in G_T: three path vertices v_{xy}, v_{xz}, and v_{yz} each of weight 1, and a vertex v_{xyz} of weight 2 that we call a *tripodal* vertex. Two vertices in G_T are adjacent if and only if their corresponding subtrees intersect (for a path vertex, the corresponding subtree is the path in T between the two corresponding leaves, and for a tripodal vertex, the corresponding subtree is the subtree of T that is the union of the three paths corresponding the three possible combinations of the corresponding 3 leaves). This completes the construction of G_T.

Observe that the vertices corresponding to the same tripodal color form a clique in G_T. Observe also that G_T, which is the intersection graph of some subtrees of T, is a chordal graph.

Let $\mathcal{C}_{opt}$ be an optimal convex recoloring of the leaves in T. By Observation 1, $\mathcal{C}_{opt}$ retains every color. Therefore, the number of leaves retaining their colors by $\mathcal{C}_{opt}$ can be expressed as $n_c + k$, for some non-negative integer k, and equivalently, the number of leaves that are recolored by $\mathcal{C}_{opt}$ can be expressed as $n - n_c - k$. From each color c appearing in T, fix one leaf of color c that retains its color by $\mathcal{C}_{opt}$. Therefore, there are precisely k leaves other than the fixed ones, whose colors are retained by $\mathcal{C}_{opt}$; call each of these leaves a *floating* leaf. We construct a set of vertices I in G_T as follows. For each color c such that there is exactly one floating leaf of color c, this floating leaf, together with the fixed leaf of color c correspond to a path vertex in G_T of weight 1: we place this vertex in I. For each floating color c such that there are exactly two floating leaves of color c, these two floating leaves, together with the fixed leaf of color c, correspond to a tripodal vertex in G_T of weight 2: we place this vertex in I. Observe that the total weight of the vertices in I is k. Moreover, the set of vertices I is an independent set in G_T, since each vertex in I corresponds to a subtree of T whose vertices all have the same unique color in $\mathcal{C}_{opt}$ (by the convexity of $\mathcal{C}_{opt}$). It follows that G_T has an independent set of weight k.

Conversely, if I is an independent set in G_T of weight k, then the recoloring that retains the colors of the leaves corresponding to the vertices in I, and retains the color of exactly one leaf from every other color in T, recolors exactly $n - n_c - k$ leaves, and can be extended to a convex recoloring of T.

This shows that an optimal convex recoloring of $(T, \mathcal{C})$ can be computed by computing a maximum weighted independent set in the chordal graph G_T. Since computing a maximum weighted independent set in a chordal graph can be done in polynomial time [7], the theorem follows. □

5.2 An exact algorithm for CLR

Although the approach is similar to that used in Subsection 4.2, there are some subtle differences, especially in the analysis.

Let $(T, \mathcal{C})$ be an instance of CLR. Let N_1 be the number of singleton colors in T, N_2 that of interval colors, N_3 that of colors that appear on exactly three leaves in T, and $N_{>3}$ that of colors that appear on at least 4 leaves in T. For each color c that appears on at least 4 leaves in T, we fix (arbitrarily) three leaves of color c in T and call them *stationary* leaves, and we call each other leaf of color c an *excess* leaf. Let N_e be the number of excess leaves in T. As in Subsection 4.2, the following equality holds:

$$N_1 + 2N_2 + 3N_3 + 3N_{>3} + N_e = n. \tag{2}$$

Consider the following algorithm $\mathcal{A}$, which is very similar to that in Subsection 4.2: enumerate every excess leaf, in addition to one stationary leaf from each color that appear on at least 4 leaves. The instance then becomes an instance of 3-CLR (after the path contractions), and we can solve it in polynomial time.

The number of leaves enumerated by $\mathcal{A}$ is $N_e + N_{>3}$. Therefore, the total number of enumerations is bounded by $2^{N_e+N_{>3}}$. After that, we apply the algorithm in Subsection 5.1 to the resulting instance, which runs in polynomial time. Therefore, algorithm $\mathcal{A}$ solves the instance $(T, \mathcal{C})$ in time $O^*(2^{N_e+N_{>3}})$.

On the other hand, the number of bad colors is $N_2+N_3+N_{>3} = (n-N_e-N_1+N_2)/3 \leq (n-N_e+N_2)/3$ by Equation (2). Therefore, using the algorithm in [14] for the weighted case[2], we can solve the CLR problem by an algorithm $\mathcal{A}'$ in time $O^*(2^{(n-N_e+N_2)/3})$.

Let $0 < \mu \leq 1$ be a constant to be determined later. If $N_e + N_{>3} \leq \mu n$, then algorithm $\mathcal{A}$ solves the instance in time $O^*(2^{\mu n})$. Suppose now that $N_e+N_3 > \mu n$. Equation (2) implies that $2N_2 \leq n-N_e-3N_{>3}$. Therefore, $N_2 \leq (n - N_e - 3N_{>3})/2$, and hence $(n - N_e + N_2)/3 \leq (n - N_e - N_{>3})/2 < (1 - \mu n)/2$ (since $N_e + N_{>3} > \mu n$). Since the instance can be solved in time $O^*(2^{(n-N_e+N_2)/3})$ by algorithm $\mathcal{A}'$, it follows that when $N_e + N_3 > \mu n$ the instance can be solved in time $O^*(2^{(1-\mu)n/2})$. By choosing $\mu = 1/3$, we conclude that, in both cases, the instance can be solved in time $O^*(2^{n/3})$. We have the following theorem:

Theorem 5.2 *The* CLR *problem is solvable in time $O^*(2^{n/3})$.*

6 Acknowledgment

We would like to thank Andreas Brandstädt, Haiko Müller, and Oriana Ponta for the helpful discussions related to some of the problems in this paper.

[2]The algorithm for the weighted case in [14] can be used to solve CLR after assigning weight 1 to the leaves, and weight 0 to the interior vertices.

References

[1] G. Agnarsson, R. Greenlaw, and M. Halldórsson. On powers of chordal graphs and their colorings. *Congressus Numerantium*, 144:41–65, 2000.

[2] R. Bar-Yehuda, I. Feldman, and D. Rawitz. Improved approximation algorithm for convex recoloring of trees. *Theory of Computing Systems*, 43(1):3–18, 2008.

[3] H. Bodlaender, M. Fellows, M. Langston, M. Ragan, F. Rosamond, and M. Weyer. Quadratic kernelization for convex recoloring of trees. In *COCOON*, volume 4598 of *LNCS*, pages 86–96. Springer, 2007.

[4] H. Bodlaender and M. Weyer. Convex and connected recoloring of trees and graphs. Unpublished manuscript, (2005).

[5] B. Chor, M. Fellows, M. Ragan, I. Razgon, F. Rosamond, and S. Snir. Connected coloring completion for general graphs: Algorithms and complexity. In *COCOON*, volume 4598 of *LNCS*, pages 75–85. Springer, 2007.

[6] F. Fomin, F. Grandoni, and D. Kratsch. Measure and conquer: a simple $O(2^{0.288})$ independent set algorithm. In *Proceedings of SODA*, pages 18–25, 2006.

[7] A. Frank. Some polynomial algorithms for certain graphs and hypergraphs. In *Proceedings of the Fifth British Combinatorial Conference*, pages 211–226, 1975.

[8] M. Garey and D. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman & Co., New York, NY, USA, 1990.

[9] F. Gavril. Maximum weight independent sets and cliques in intersection graphs of filaments. *Information Processing Letters*, 73(5-6):181–188, 2000.

[10] M. Golumbic. *Algorithmic Graph Theory and Perfect Graphs*, volume 57 of *Annals of Discrete Mathematics.* North-Holland Publishing Co., 2004.

[11] R. Laskar and D. Shier. On chordal graphs. *Congressus Numerantium*, 29:579–588, 1980.

[12] S. Moran and S. Snir. Efficient approximation of convex recolorings. *Journal of Computer and System Sciences*, 73(7):1078–1089, 2007.

[13] S. Moran and S. Snir. Convex recolorings of strings and trees: Definitions, hardness results and algorithms. *Journal of Computer and System Sciences*, 74(5):850–869, 2008.

[14] O. Ponta, F. Hüffner, and R. Niedermeier. Speeding up dynamic programming for some NP-hard graph recoloring problems. In *TAMC*, volume 4978 of *LNCS*, pages 490–501. Springer, 2008.

[15] I. Razgon. A $2^{O(k)}\mathrm{poly}(n)$ algorithm for the parameterized convex recoloring problem. *Information Processing Letters*, 104(2):53–58, 2007.

[16] D. West. *Introduction to graph theory.* Prentice Hall Inc., Upper Saddle River, NJ, 1996.

7 Appendix

Theorem 7.1 *The* 4-CLR *problem is NP-complete.*

Proof. The proof is similar to that of the NP-completeness of 2-CPR, given in Theorem 3.2.

The reduction is from the NP-complete problem independent set [8]. Let (G, k) be an instance of independent set, where G has $n(G)$ vertices and $m(G)$ edges. We construct an instance $(T, \mathcal{C})$ of 4-CLR as follows.

First, for each vertex $v \in G$, we associate a unique color c_v. Similarly, for each edge e in G we associate a unique color c_e. For each vertex v, we construct a caterpillar: a path with $deg(v) + 2$ many vertices that we call the *body* of the caterpillar, and leaves attached to the body as follows. Designate one of the extremities of the body of the caterpillar as the *top*, and the other one as the *bottom*. The top and bottom vertices of the body are each attached with a single leaf colored with c_v. Each other vertex in the body corresponds in a one-to-one fashion to an incident edge on v (the order does not matter), and is attached with two leaves colored by the color of the corresponding incident edge. Finally, we attach all the top vertices of the caterpillars to a new vertex r, and attach 4 leaves of a new color c_r to r. Let T be the resulting tree from this construction. Note that: (1) every colored vertex in T is a leaf, (2) for each vertex $v \in G$ there are exactly two leaves of color c_v in T, (3) for each edge $e \in G$ there are exactly 4 leaves of color c_e in T, and (4) there are exactly 4 leaves of color c_r attached to vertex r in T. Therefore, $(T, \mathcal{C})$, where $\mathcal{C}$ is the coloring induced by the colors of the leaves in T, is an instance of 4-CLR. Clearly, the above construction can be carried out in polynomial time.

If I is an independent set in G of size at least k, then we can construct a convex recoloring $\mathcal{C}_I$ for T as follows. The recoloring $\mathcal{C}_I$ retains the color of the two leaves in T colored with c_v, for each $v \in I$, and recolors the caterpillar corresponding to v (all vertices and leaves) with the color c_v. The vertex r is colored with c_r. Finally, for every edge $e \in G$, not both vertices incident on e belong to I (I is an independent set). Thus, there is a caterpillar in T where the vertex with the two leaves of color c_e attached to it, can be colored c_e. All other vertices and leaves of T can be colored/recolored in such a way that the recoloring C_I is convex (e.g., color of a neighbor already colored/recolored).

Observe that, for every edge $e \in G$ exactly two leaves of color c_e in T are recolored. Furthermore, for every vertex $v \in G$, exactly one leaf of color c_v is recolored if $v \notin I$, and no leaf of color c_v is recolored if $v \in I$. Therefore, the number of recolored leaves in T is $2m(G) + n(G) - |I| \leq 2m(G) + n(G) - k$.

Now suppose that there is a recoloring of T such that the number of recolored leaves is at most $2m(G) + n(G) - k$. Let I be the set of vertices v in G whose corresponding color c_v is retained. We may assume that no two leaves colored c_e in two different caterpillars of T in which c_e appears, are retained since this would imply that the root is colored c_e, and hence at least three leaves of color c_r have been recolored (c_e). In such case we can produce a new recoloring by coloring the root with color c_r and recoloring two nodes of color c_e (appropriately). It follows from the preceding statement that $|I| \geq k$. Further, we can assume that from each color c_e, exactly two leaves colored c_e retain their color (and hence they must be in the same caterpillar). If not, then for some edge $e = \{u, v\}$ all leaves colored c_e except one must have been recolored. In this case, it can be verified that we can produce a new convex recoloring that recolors at most $2m(G) + n(G) - k$ leaves, and in which two leaves of color c_e in the same caterpillar are retained, while one leaf of color c_u (or c_v) is recolored.

Now we claim that I is an independent set in G. Suppose not, and assume that there are two vertices $u, v \in I$, such that both colors c_u and c_v are retained in T, but u and v are adjacent in G. Then, for color c_e, where $e = \{u, v\}$, all leaves except one must be recolored. This contradicts the above assumption.

It follows that I is an independent set in G of size at least k. This completes the proof. $\square$