Math. Program., Ser. A (2017) 165:529–548
DOI 10.1007/s10107-016-1094-3

FULL LENGTH PAPER

An extended formulation of the convex recoloring problem on a tree

Sunil Chopra[1] · Bartosz Filipecki[2] · Kangbok Lee[3] · Minseok Ryu[4] · Sangho Shim[5] · Mathieu Van Vyve[2]

Received: 12 April 2016 / Accepted: 17 November 2016 / Published online: 30 November 2016

Abstract We introduce a strong extended formulation of the convex recoloring problem on a tree, which has an application in analyzing phylogenetic trees. The extended formulation has only a polynomial number of constraints, but dominates the conventional formulation and the exponentially many valid inequalities introduced by Campêlo et al. (Math Progr 156:303–330, 2016). We show that all valid inequali-

Electronic supplementary material The online version of this article (doi:10.1007/s10107-016-1094-3) contains supplementary material, which is available to authorized users.

✉ Sangho Shim
shim@rmu.edu

Sunil Chopra
s-chopra@kellogg.northwestern.edu

Bartosz Filipecki
bartosz.filipecki@uclouvain.be

Kangbok Lee
kblee@postech.ac.kr

Minseok Ryu
msryu@umich.edu

Mathieu Van Vyve
mathieu.vanvyve@uclouvain.be

1 Kellogg School of Management, Northwestern University, Evanston, IL 60208, USA

2 Center for Operations Research and Econometric, Université catholique de Louvain, Louvain, Belgium

3 Department of Industrial and Management Engineering, Pohang University of Science and Technology, Pohang, Republic of Korea

4 Department of Industrial and Operations Engineering, University of Michigan, Ann Arbor, MI 48109, USA

5 Department of Engineering, Robert Morris University, Moon Twp, PA 15108, USA

ties introduced by Campêlo et al. can be derived from the extended formulation. We also show that the natural restriction of the extended formulation provides a complete inequality description of the polytope of subtrees of a tree. The solution time using the extended formulation is much smaller than that with the conventional formulation. Moreover the extended formulation solves all the problem instances attempted in Campêlo et al. (2016) and larger sized instances at the root node of the branch-and-bound tree without branching.

Mathematics Subject Classification 92B05 General biology and biomathematics · 90C05 Linear programming · 90C10 Integer programming · 90C27 Combinatorial optimization · 90C57 Polyhedral combinatorics, branch-and-bound, branch-and-cut

1 Introduction

The convex recoloring problem on a tree was first investigated by Moran and Snir [16]. Campêlo et al. [4] have studied the associated integer programming formulation and provided several classes of facet defining inequalities. Since our work builds on Campêlo et al. [4], we introduce the problem with the notation used by them.

Let $\mathcal{C} = \{1, \ldots, k\}$ be a set of colors and $G = (V, E)$ be a graph with node set V and edge set E. A *partial coloring* of a graph G is a function $C : V \to \mathcal{C} \cup \{\emptyset\}$, where $\emptyset$ indicates absence of color. A node $v \in V$ is said to be *uncolored* if $C(v) = \emptyset$. The coloring C is called *total* if there is no uncolored node; i.e., $\emptyset \notin C(V)$ where $C(V)$ is the image of the function C.

A colored graph is a pair (G, C) consisting of a graph G and a coloring C of its nodes. A total coloring C is said to be *convex* if, for each $t \in \mathcal{C}$, the set of nodes with color t induces a connected subgraph of G. A *convex partial coloring* is a partial coloring that can be extended to a convex total coloring by solely assigning a color in $\mathcal{C}$ to each uncolored node. A *good coloring* is a partial coloring in which each color induces a connected subgraph. Campêlo et al. [4] point out that every good coloring of a graph $G = (V, E)$ can be extended to a convex total coloring in $O(|V| + |E|)$ time.

Given a non-convex coloring of a graph, the *recoloring distance* is defined as the minimum number of color changes at the colored nodes needed to obtain a convex partial coloring [18]. This measure can be generalized to a weighted model, where changing the color of node v costs a nonnegative weight $c(v)$ depending on v. This problem can be stated as follows.

Problem 0.1 [Convex Recoloring (CR)] Given a partially colored graph (G, C), an available color set $\mathcal{C}$ and a cost function $w : V \to \mathbb{Q}_{\geq 0}$, find a convex partial recoloring C' that minimizes $\sum_{v \in R_C(C')} w(v)$ where $R_C(C') = \{v \in V : C(v) \neq \emptyset \text{ and } C(v) \neq C'(v)\}$ is the set of nodes recolored by C'.

We note that, corresponding to any convex partial coloring that is not good there is a good coloring with the same weight. Thus, we will consider that in the CR problem we are interested only in finding good recolorings of the graph.

The study of the CR problem on a tree by Moran and Snir [16] was motivated by its application in the study of phylogenetic trees. A phylogenetic tree shows relationships among various biological species or other entities based upon similarities and differences in their physical or genetic characteristics. If each biological characteristic is associated with a color, natural biological constraints require that the subgraph induced by each color be connected [18]. In addition to several computational biology applications such as protein interaction networks, more applications on routing problems and transportation networks have also been mentioned in [7,12].

The CR problem has been shown to be NP-hard in many settings. Moran and Snir [18] showed the problem to be NP-hard on paths. Kanj and Kratsch [13] proved that it is NP-hard on paths even if each color appears at most twice. Moran et al. [19] showed that computing the convex recoloring cost of a 2-colored graph is NP-hard. Campêlo et al. [2] improved this result by showing that the unweighted uniform CR problem is NP-hard even on 2-colored grids. Given the intractability of the problem, there have been three different solution approaches: (i) dynamic programming algorithms, (ii) approximation algorithms and (iii) integer programming algorithms.

Moran and Snir [16] first presented dynamic programming algorithms for optimal convex recoloring of totally colored strings (paths) or trees. Afterwards, Bar-Yehuda et al. [1] and Ponta et al. [21] provided improved dynamic programming algorithms.

Approximation algorithms have been designed with the ratio of 2 for paths by Moran and Snir [17], with the ratio of $(2 + \epsilon)$ for bounded treewidth graphs by Kammer and Tholey [12], and with the ratio of 3/2 for paths in which each color appears at most twice by Lima and Wakabayashi [15]. Campêlo et al. [2] have identified limits of approximation algorithms by showing that the uniform CR problem cannot be approximated in polynomial time by $O(\log n)$ on n-node 2-colored bipartite graphs, unless P = NP.

More recently, Campêlo et al. [4] introduced a mathematical programming approach for the CR problem on a tree. They presented an integer linear programming (ILP) formulation and studied the facial structure of the associated integer polytope. They also presented several classes of facet defining inequalities and designed the corresponding separation algorithms.

Our paper builds on their work and introduces an extended formulation of the convex recoloring problem on a tree described by $O\left((|V| + |E|)|\mathcal{C}|\right)$ variables and constraints. For extended formulations in graph theory using both node and edge variables, we refer to Chopra and Owen [5], and for a survey on extended formulations in general see Conforti, Cornuéjols and Zambelli [9] and Vanderbeck and Wolsey [22]. For constructing extended formulations as well as lower bounds on their sizes, the reader may read Kaibel [11]. Recoloring is equivalent to partitioning, and we employ additional edge variables introduced in Chopra and Rao [6].

In this paper, we show that our extended formulation is superior to Campêlo et al. [4]'s solution approach in three aspects.

- From a mathematical perspective, we show that our extended formulation strictly dominates the formulation by Campêlo et al. [4].

- From an experimental perspective, our extended formulation can be solved faster than Campêlo et al. [4]'s solution approach in the tested instances. Moreover, our extended formulation can handle much larger problem instances.
- From an implementation perspective, our extended formulation is easy to understand and implement. It has $O((|V|+|E|)|\mathcal{C}|)$ variables and constraints and its LP-relaxation gives an integer optimum solution in every tested case in our experiments. This implies that the LP-relaxation of the extended formulation is a very good approximation of the integer polytope.

Furthermore, we show that our extended formulation (restricted to one color) is a complete inequality description of the polytope of subtrees of a tree (in the node-and-edge space). We also characterize its projection onto the edge-variable space and exhibit facet-defining inequalities that are missing in the work of Campêlo et al. [4]. These theoretical results are not new (see among others [8,10,14,23]), but our proof based on dynamic programming is new, short, simple and self-contained. In particular, it does not rely on the theory of greedoids.

In Sect. 2, we introduce the conventional formulation given by Campêlo et al. [4] along with our extended formulation. In Sect. 3, we show that the natural restriction of our extended formulation provides a complete inequality description of the polytope of subtrees of a tree. In Sect. 4, we prove that our model is strictly stronger than the conventional model and implies all the exponentially many valid inequalities identified by Campêlo et al. [4]. In Sect. 5, we present computational experiments with problem instances from phylogentic trees in [4] and larger trees from TreeBASE.org. In Sect. 6, we summarize the current results and suggest future work.

2 Integer linear programming formulations for the convex recoloring problem on a tree

In this section we provide a couple of integer linear programming formulations for the convex recoloring problem on a tree. Section 2.1 contains the formulation provided by Campêlo et al. [4]. Section 2.2 contains an extended formulation that we show to be stronger than the formulation of Campêlo et al. [4]. The convex recoloring problem is usually to minimize the number of color changes at the colored nodes to obtain a good coloring. This is equivalent to maximizing the number of colored nodes that do not change their color when obtaining a good coloring. We now define the set of node variables that are used in both formulations. Given a set of colors $\mathcal{C} = \{1, \ldots, k\}$, a tree $T = (V, E)$ and a partial coloring C, define the *node variables* $x = (x_{ut} : u \in V, t \in \mathcal{C})$ where $x_{ut} = 1$ if node u is assigned to color $t \in \mathcal{C}$, and 0 otherwise. To express the objective function, for each $u \in V$ and $t \in \mathcal{C}$, we define a constant $w(u, t)$, which is $w(u)$ if $C(u) = t$, and is 0 otherwise.

2.1 The conventional formulation by Campêlo et al. [4]

In this section, we introduce the integer linear programming formulation given by Campêlo et al. [4] for the convex recoloring problem on a tree. Let Path$[u, v]$ denote the path connecting u to v in tree T. For the CR problem restricted to trees, Campêlo et al. [4] provided the following formulation:

$$\max \sum_{t=1}^{k} \sum_{u \in V} w(u,t) x_{ut} \tag{1}$$

$$s.t. \sum_{t=1}^{k} x_{ut} \leq 1 \text{ for } u \in V \tag{2}$$

$$x_{ut} - x_{ht} + x_{vt} \leq 1 \text{ for } u, v \in V, h \in V(\text{Path}[u,v]) \setminus \{u,v\} \text{ and } t \in \mathcal{C} \tag{3}$$

$$x_{ut} \in \{0,1\} \text{ for } u \in V \text{ and } t \in \mathcal{C}. \tag{4}$$

For a partial coloring $C : V \to \mathcal{C} \cup \{\emptyset\}$ and a color $t \in \mathcal{C}$, the objective function (1) is to maximize the value of the recoloring. The *packing inequalities* (2) and the *convexity constraints* (3) restrict a feasible recoloring C' to be partial and to be good, respectively. We denote by $P_x(T, \mathcal{C})$ the convex hull of the solutions x to (2), (3) and (4). Throughout the paper, we refer to the above formulation by Campêlo et al. [4] as the *conventional formulation*.

2.2 A strong extended formulation

In this section, we present a new formulation for the CR problem with a polynomial number of additional variables compared to the formulation of Campêlo et al. [4]. This formulation is similar in spirit to the one given by Goemans [10] for the steiner arborescence problem. We employ additional variables y_{et} which we call *edge variables*. An edge variable y_{et} for each edge $e \in E$ and for each color $t = 1, \ldots, k$ takes the value 1 if both end nodes of edge e are colored by t, and 0 otherwise. Observe that the number of variables in our extended formulation is about twice the number of variables used by Campêlo et al. [4]. With edge variables, we use a basic property of a tree that the number of edges equals the number of nodes minus one.

An extended formulation for the CR problem can be written as

$$\max \sum_{t=1}^{k} \sum_{u \in V} w(u,t) x_{ut}$$

$$s.t. \sum_{t=1}^{k} x_{ut} \leq 1 \text{ for } u \in V \tag{5}$$

$$\sum_{u \in V} x_{ut} - \sum_{e \in E} y_{et} \leq 1 \text{ for } t \in \mathcal{C} \tag{6}$$

$$\left.\begin{aligned} -x_{ut} + y_{uvt} &\leq 0 \\ -x_{vt} + y_{uvt} &\leq 0 \end{aligned}\right\} \text{for edge } uv \in E \text{ and } \quad \text{for } t \in \mathcal{C} \tag{7}$$

$$x_{ut} \in \{0,1\} \quad \text{for } u \in V \text{ and } t \in \mathcal{C} \tag{8}$$

$$y_{et} \in \{0,1\} \quad \text{for } e \in E \text{ and } t \in \mathcal{C} \tag{9}$$

In the extended formulation, (5) is the same as the packing inequalities (2) in the conventional formulation. We refer to (6) as *subtree inequalities* and to (7) as *node-*

edge inequalities. Inequalities (6) ensure that the nodes of each color $t \in \mathcal{C}$ induce a connected subgraph (subtree) of T. Inequalities (7) ensure that if an edge $e = (u, v)$ is assigned color t, both nodes u and v must also be assigned to color t. The LP-relaxation of our extended formulation has $2|E||\mathcal{C}| + |V| + |\mathcal{C}|$ constraints, which is about the same as the number of variables in the formulation. We show that the coloring corresponding to a feasible solution of this formulation is a good coloring.

Theorem 2.1 *x is a projection of an integer solution (x, y) to the extended formulation (5)–(9) if and only if it defines a good coloring.*

Proof If x defines a good coloring, we construct y as follows for each edge uv,

$$y_{uvt} = \begin{cases} 1 & \text{if and only if } x_{ut} = x_{vt} = 1, \\ 0 & \text{otherwise.} \end{cases}$$

It is easy to verify that (x, y) satisfies all constraints (5)–(9) in the extended formulation.

Now consider $(\bar{x}, \bar{y})$, an integer feasible solution to the extended formulation. For each color t define $\bar{V}_t = \{u : \bar{x}_{ut} = 1\}$ and $\bar{E}_t = \{e : \bar{y}_{et} = 1\}$. By (7), $\bar{y}_{et} = 1$ for $e = uv$ implies $\bar{x}_{ut} = \bar{x}_{vt} = 1$. The subgraph $\bar{T}_t = (\bar{V}_t, \bar{E}_t)$ is thus well-defined for each color t that has been assigned to at least one node.

In order to prove that this is a good coloring we need to show that each subgraph $\bar{T}_t$ is connected. Let us assume that $\bar{T}_t$ is not connected. $\bar{T}_t$ then consists of at least two subtrees of T. Thus,

$$|\bar{V}_t| \geq |\bar{E}_t| + 2.$$

But then $(\bar{x}, \bar{y})$ violates inequality (6) for the color t. This contradiction proves that each subgraph $\bar{T}_t$ is connected and thus $(\bar{x}, \bar{y})$ defines a good coloring. □

We denote by $P_{(x,y)}(T, \mathcal{C})$ the convex hull of the binary solutions (x, y) to (5)–(9).

Interestingly, edge variables can be relaxed to non-negative continuous variables. In the proof of Theorem 2.1, if we let $\bar{E}_t = \{e : \bar{y}_{et} > 0\}$, then by (7) and (8), $\bar{y}_{et} > 0$ for $e = (u, v)$ implies $\bar{x}_{ut} = \bar{x}_{vt} = 1$. Thus, by $|\bar{V}_t| \geq |\bar{E}_t| + 1$ and the subtree inequality (6),

$$\sum_{u \in V} \bar{x}_{ut} = |\bar{V}_t| \geq |\bar{E}_t| + 1 \geq \sum_{e \in E} \bar{y}_{et} + 1 \geq \sum_{u \in V} \bar{x}_{ut},$$

implying that $\bar{T}_t$ is a connected subtree. Therefore, we can say that in the extended formulation the number of binary variables is nk and the number of continuous variables is $(n-1)k$ for $n = |V|$.

3 A subproblem of the CR problem—finding a maximum weight subtree of a tree

The maximum weight subtree problem can be defined as follows: given a graph $G = (V, E)$, weights a_i for each node $i \in V$ and b_e for each edge $e \in E$ find a subtree that maximizes the sum of node and edge weights which may be negative. Observe that a convex total coloring of a tree is a collection of subtrees with each subtree corresponding to a color. Thus, finding the subtree of a tree is a subproblem when finding a convex total coloring. In this section we focus on the case where G is a tree T and we show that the restriction of the LP-relaxation of the extended formulation to a single color (defined by inequalities (6) and (7) for a fixed value of t) is actually tight and has only integer extreme points. This result provides support for our observation (in Sect. 5) that the LP-relaxation of the extended formulation often solves the CR problem without any need for branching. This result is also important because we later show that the projection of inequalities (6) and (7) contain all the valid inequalities identified by Campêlo et al. [4].

We first show that we can obtain the optimal subtree of a tree in linear time by dynamic programming. Let us fix a root $r \in V$ arbitrarily. Such a root implicitly defines an orientation of each edge away from the root. Let S_i be the set of children of node i and p_i its parent node (undefined for the root). Let us define $T_i = (V_i, E_i)$ for each $i \in V$ as the subtree of T rooted in i, and containing all descendant nodes of i in T and corresponding edges.

We let $b_{p_r,r} = 0$. The weight of an empty subtree is assumed to be zero. For each node $i \in V$, let us define $H(i)$ as the optimal solution value of the following problem: Find a subtree of T_i that is either empty or is rooted at i and maximizes the sum of node and edge weights in the subtree plus $b_{p_i,i}$. Furthermore, let K be the optimal solution value of the maximum subtree of a tree problem (i.e. maximum sum of node and edge weights of a subtree of T_r). We can write the following recursions:

$$H(i) = \max\left(0, a_i + b_{p_i,i} + \sum_{j \in S_i} H(j)\right)$$

$$K = \max\left(0, \max_{i \in V}\left(a_i + \sum_{j \in S_i} H(j)\right)\right)$$

For $H(i)$, the first term corresponds to the case of the empty solution and the second term corresponds to the non-empty case (node i, plus parent edge (p_i, i), plus profitable downstream descendant trees). K corresponds to the maximum of values of each node of T, or the empty tree.

This recursion can be written as the following LP

$$\begin{aligned}
&\min K \\
&\text{s.t. } H(i) \geq a_i + b_{p_i,i} + \sum_{j \in S_i} H(j) \quad \forall i \in V && (10)\\
&\qquad K \geq a_i + \sum_{j \in S_i} H(j) \quad \forall i \in V && (11)\\
&\qquad H, K \geq 0. && (12)
\end{aligned}$$

with dual

$$\begin{aligned}
&\max \sum_{i \in V} a_i(u_i + v_i) + \sum_{i \in V \setminus \{r\}} b_{p_i,i} u_i \\
&\text{s.t. } u_i \leq u_{p_i} + v_{p_i} \quad \forall i \in V && (13)\\
&\qquad \sum_i v_i \leq 1 && (14)\\
&\qquad u, v \geq 0, && (15)
\end{aligned}$$

where we use the convention $u_{p_r} = v_{p_r} = 0$ and therefore $u_r = 0$. Assuming a (u, v)-binary solution to the above system, the variables can be interpreted as follows: $v_i = 1$ if the node i is the root of the optimal solution (0 otherwise), and $u_i = 1$ if node i is part of the solution without being the root and 0 otherwise. Constraint (14) allows only one node to be selected as a root. Constraints (13) model the construction of the solution rooted at node i. More specifically if node i is part of the solution and is not the root ($u_i = 1$), then its parent is part of the solution, being the root ($v_{p_i} = 1$) or not ($u_{p_i} = 1$).

Operating the change of variable $x_i = u_i + v_i$ and $y_{i,j} = u_j$ where $i = p_j$, one finally obtains the following formulation

$$\begin{aligned}
&\max \sum_{i \in V} a_i x_i + \sum_{e \in E} b_e y_e \\
&\text{s.t. } \sum_{i \in V} x_i - \sum_{e \in E} y_e \leq 1 && (16)\\
&\qquad y_{i,j} \leq x_i \quad \forall (i, j) \in E && (17)\\
&\qquad y_{i,j} \leq x_j \quad \forall (i, j) \in E && (18)\\
&\qquad x, y \geq 0, && (19)
\end{aligned}$$

where (16) corresponds to (14), constraint (17) corresponds to (13) and (18) corresponds to $v_i \geq 0$. This formulation has much more natural variables with x_i and y_e taking value 1 if the corresponding node or edge is part of the optimal tree and 0 otherwise. Also no arbitrary special node plays the role of a root. But more importantly, constraints (16), (17) and (18) are exactly the inequalities (6) and (7) in the extended formulation when restricted to a specific value of t.

The above derivation proves that the above formulation is tight (i.e. the LP always gives the right optimal solution value). But note also that since the last LP has no additional variables and solves the problem for any value of a and b, this implies that all extreme points of the polytope described by the constraint system (16)–(19) are characteristic vectors of subtrees (otherwise the LP would not be tight for any objective function, see prop. 1.3 of Nemhauser and Wolsey [20], p. 536). We have therefore obtained the following result.

Theorem 3.1 *The extreme points of the polytope described by* (16)–(19) *are the characteristic vectors of the subtrees of the tree* (V, E).

4 A comparison of the extended formulation with a strengthened conventional formulation

Our goal of this section is to show that the LP-relaxation of our extended formulation is stronger than the LP-relaxation obtained by adding all valid inequalities identified by Campêlo et al. [4] (star, path and tree inequalities) to the LP-relaxation of the conventional formulation. We do this in three steps. In Sect. 4.1 we characterize all facet defining inequalities in the space of node (x) variables obtained by projecting out the edge (y) variables from the inequalities (16)–(19) (which are precisely the inequalities (6) and (7) for a specific value of t). In Sect. 4.2 we then show that the inequalities obtained as a result of this projection include all the inequalities identified by Campêlo et al. [4] (star, path, and tree inequalities). We then show the existence of new inequalities obtained as a result of this projection that are not from the inequalities identified by Campelo et al. [4].

4.1 A characterization of all facet defining inequalities for the subtree polytope with only *x* variables

In Sect. 3 we have characterized all valid inequalities for the subtree polytope [(16)–(19)] with a formulation that uses both node (x) and edge (y) variables. In this section we characterize all facet defining inequalities in the space of node (x) variables obtained by projecting out the edge (y) variables from the inequalities (16)–(19). The results of this section are then used in Sect. 4.2 to show that the projection of inequalities (6) and (7) to the space of x variables includes all the valid inequalities identified by Campêlo et al. [4].

Let P_{xy} be the feasible set of (16)–(19) and $P_x = \text{proj}(P_{xy})$ be the projection of P_{xy} onto the space of x variables. Formally, testing if for a given vector x, there exist y such that $(x, y) \in P_{xy}$ is equivalent to testing whether the following LP has optimal value 0 (which is equivalent to the LP having a feasible solution)

$$\begin{aligned} &\max\ 0 \\ &\text{s.t.}\ -\sum_{e \in E} y_e \leq 1 - \sum_{i \in V} x_i \\ &\qquad y_{i,j} \leq x_i \quad \forall (i, j) \in E \\ &\qquad y_{i,j} \leq x_j \quad \forall (i, j) \in E \\ &\qquad y \geq 0, \end{aligned}$$

or equivalently if its dual

$$\begin{aligned}\min \ & \sum_{(i,j)\in E} (\alpha_{i,j}x_i + \beta_{i,j}x_j) + \lambda\left(1 - \sum_{i\in V} x_i\right)\\ \text{s.t. } & \alpha_{i,j} + \beta_{i,j} \geq \lambda \quad \forall (i,j)\in E\\ & \alpha, \beta, \lambda \geq 0,\end{aligned}$$

has optimal value 0. Observe that setting $\lambda = 0$ trivially provides a feasible solution to the dual with objective function value 0 (set $\alpha_{i,j} = 0, \beta_{i,j} = 0\ \forall i, j$). Thus, normalizing $\lambda = 1$ captures all other relevant solutions. Normalizing $\lambda = 1$ and rearranging, we obtain:

$$\begin{aligned}\min & \sum_{i\in V}\left(\sum_{j:(i,j)\in E} \alpha_{i,j} + \sum_{j:(j,i)\in E} \beta_{j,i} - 1\right) x_i + 1\\ \text{s.t. } & \alpha_{i,j} + \beta_{i,j} \geq 1 \quad \forall (i,j)\in E\\ & \alpha, \beta \geq 0,\end{aligned}$$

Observe that the constraint matrix of the LP is totally unimodular. Thus all extreme points must be 0-1 vectors. As a result, the last LP is completely separable by edges: we just have to choose for each edge (i, j) whether $\alpha_{i,j}$ or $\beta_{i,j}$ is 1. The optimal choice solely depends on the relative value of x_i and x_j. In graph theoretical terms, we have obtained the following characterization of P_x. Each facet-defining inequality of P_x corresponds to an orientation of the edges of T. The inequality associated to such an orientation is

$$\sum_{i\in V} (1 - \delta_i)x_i \leq 1 \tag{20}$$

where δ_i is the number of arcs pointing into node i. Each such orientation leads to a valid inequality. But, as we see next there is more. Our next result (Theorem 4.1) was first obtained by Korte, Lovasz, and Schrader [14] using greedoids. A similar result has been proved independently by Wang, Buchanan, and Butenko [23]. We provide our detailed proof because it is simpler and more accessible.

Theorem 4.1 (Korte, Lovász and Schrader [14]) *Each orientation leads to a facet-defining inequality for P_x.*

Proof Let us consider an orientation A whose corresponding inequality (20) is not facet defining. If this is so, there must exist a family of facet defining inequalities of P_x whose weighted sum implies the A-inequality. Since each facet defining inequality corresponds to an orientation of the edges of T, there exists a family of (distinct) orientations $(B_1, \ldots, B_k)$ giving rise to inequalities whose weighted sum implies the A-inequality. Observe that since all orientations yield inequalities with right-hand sides equal to 1, we can assume that the nonnegative weights sum up to 1.

Because the graph is a tree and has no cycle, there must exist a node j with all incident arcs oriented out of node j in the orientation A. This implies that all the edges incident to that node j must have the same orientation in all orientations $B_1, \ldots, B_k$ (otherwise the weighted sum of the inequalities associated to $B_1, \ldots, B_k$ will have a coefficient for node j strictly smaller than that of the inequality A). Let us remove these edges and the subsequently isolated nodes from the graph. Iterating the same argument until the graph is empty proves that the orientations $B_1, \ldots, B_k$ must be identical to A. □

For separation, it suffices to orient each edge toward the incident node i with the smallest value x_i. We use this characterization to fully explain all valid inequalities ((3), star, path, and tree) identified by Campêlo et al. [4] in the remaining part of this section and the online appendix.

4.2 Dominance of the extended formulation

In this section and the online appendix we use the results of Sect. 4.1 to show that all valid inequalities introduced by Campêlo et al. [4] [(3), star, path, and tree] are of the form (20) and therefore dominated by inequalities (6) and (7) in the extended formulation.[1] As discussed in Sect. 4.1, obtaining each of the inequalities (3), star, path, and tree from (6) and (7) simply requires the definition of a suitable orientation of the edges.

We define an *orientation* $\sigma : E \to V$ to be a function which maps an edge $uv \in E$ into one of its end nodes u or v. In other words, for $e = uv$, if $\sigma(e) = v$, then e is oriented from u to v. For $w \in V$, let $\sigma^{-1}(w)$ denote the set of edges e with $\sigma(e) = w$; i.e., $\sigma^{-1}(w) = \{e : \sigma(e) = w\}$. Given a color t and an orientation σ, we consider the subtree inequality (6) corresponding to color t, and node-edge inequalities (7) corresponding to color t and orientation σ,

$$\sum_{w \in V} x_{wt} - \sum_{e \in E} y_{et} \leq 1, \text{ and}$$

$$-x_{\sigma(e)t} + y_{et} \leq 0 \text{ for all } e \in E.$$

Adding these inequalities cancels out all edge variables and gives us the inequality

$$\sum_{w \in V} \left(1 - \left|\sigma^{-1}(w)\right|\right) x_{wt} \leq 1, \tag{21}$$

where $|\sigma^{-1}(w)|$ represents the cardinality of $\sigma^{-1}(w)$; i.e., the number of edges oriented into w in the orientation σ. Recall that in (20) in Sect. 4.1, we use $\delta_w = |\sigma^{-1}(w)|$ making (20) and (21) identical.

In the remainder of this paper, we provide edge orientations as described in Sect. 4.1 to show that (6) and (7) dominate the inequalities (3), and the star inequalities. Detailed

[1] Thanks to one of the referees, we identified that Wang, Buchanan and Butenko [23] have independently proved a similar result. They showed that all inequalities in (20) are dominated by inequalities (6) and (7).

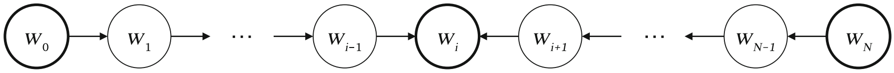

Fig. 1 The orientation σ implies $x_{w_0t} - x_{w_it} + x_{w_Nt} = \sum_{j=0}^{N}\left(1 - \left|\sigma^{-1}(w_j)\right|\right) x_{w_jt} \le 1$

proofs for the path and tree inequalities are provided in an online appendix. We first show that (6) and (7) dominate convexity inequalities (3).

Theorem 4.2 *In a tree T, subtree inequalities* (6) *and node-edge inequalities* (7) *dominate convexity constraints* (3).

Proof Consider an inequality (3) and the corresponding nodes u, v and h. We define an orientation σ that results in the left-hand side of (21) to be the same as that of (3).

Let $\text{Path}[u, v] = (w_0 = u, w_1, \ldots, w_i = h, w_{i+1}, \ldots, w_N = v)$ be the path connecting u to v. For all edges $e_j = w_{j-1}w_j$ of the path, define the orientation as shown in Fig. 1.

$$\sigma(e_j) = \begin{cases} w_j & \text{for } j \le i, \\ w_{j-1} & \text{for } j > i, \end{cases}$$

Then, we can define the orientation of all other edges in the tree that are not on the path $\text{Path}[u, v]$ away from $\text{Path}[u, v]$: For every $w \in V \backslash V(\text{Path}[u, v])$, there is one and only one node w_j in $\text{Path}[u, v]$ such that the path $\text{Path}[w_j, w]$ connecting w_j to w intersects with $\text{Path}[u, v]$ at only w_j; i.e.,

$$V(\text{Path}[w_j, w]) \cap V(\text{Path}[u, v]) = \{w_j\}.$$

All edges e of $\text{Path}[w_j, w]$ are oriented from w_j to w.

Now, the orientation σ is defined over all the edges of the tree T. It satisfies

$$\delta_w = \left|\sigma^{-1}(w)\right| = \begin{cases} 0 & \text{for } w \in \{u, v\}, \\ 2 & \text{for } w = h, \text{ and} \\ 1 & \text{for the other nodes } w \in V \backslash \{u, h, v\}. \end{cases}$$

Thus, the left-hand side of (21) is the same as (3), completing the proof. □

Now, we show that (6) and (7) dominate the star inequalities described in [4].

4.2.1 Star inequalities

For $S \subseteq V$, let $N(S) = \{h \in V \backslash S : hv \in E \text{ for some } v \in S\}$. Campêlo et al. [4] described the following star inequalities.

Theorem 4.3 (STAR INEQUALITIES in Theorem 3.1 of [4]) *Let $v \in S \subset V$ such that the subgraph of T induced by S is connected and each connected component of $T - S$ has more than one node. Let $\{V_1, \ldots, V_p\}$ denote the partition of $V \backslash S$ and let*

$\{h_i\} = V_i \cap N(S)$ for $i = 1, \ldots, p$. For every color $t \in \mathcal{C}$ and $u_i \in V_i \backslash N(S)$ for $i = 1, \ldots, p$, the following inequality defines a facet of $P_x(T, \mathcal{C})$:

$$x_{vt} + \sum_{i=1}^{p} x_{u_i t} - \sum_{i=1}^{p} x_{h_i t} \leq 1. \tag{22}$$

An example of a star inequality (22) is shown in Fig. 2.

We show that inequalities (6) and (7) together imply the star inequalities. As in the proof of Theorem 4.2, we provide a suitable orientation of the edges of T such that the left-hand side of the resulting inequality (21) is the same as (22).

Theorem 4.4 *The star inequalities* (22) *are dominated by* (6) *and* (7)*.*

Proof For inequality (22), we are given $v \in S \subseteq V$ such that the induced graph by S is connected. The set $N(S)$ represents the set of immediate neighbors of S in the tree.

The partition $\{V_1, V_2, \ldots, V_p\}$ of $V \backslash S$ is induced by the connected components of $T[V_i]$ of $T[V \backslash S]$. Recall that $\{h_i\} = V_i \cap N(S)$ for $i = 1, \ldots, p$. The inequality (22) is then defined for nodes $u_i \in V_i \backslash N(S)$ for $i \in \{1, 2, \ldots, p\}$. Figure 2 illustrates v, S, $N(S)$, V_i, h_i and $u_i \in V_i \backslash N(S)$ for $i = 1, \ldots, p$.

We define the orientation σ as follows: For all edges of the subgraph of T induced by $S \cup N(S)$, orient all edges away from the node v. Then, all the other edges are partitioned into the subsets $E(T[V_i])$ of the edges of the subtrees $T[V_i]$ induced by V_i. For each i there is a unique path connecting u_i to h_i in $T[V_i]$. Orient all edges in this path from u_i to h_i. All other edges of $T[V_i]$ are oriented away from the path connecting u_i to h_i as in the proof of Theorem 4.2.

Now, the orientation σ is defined over all the edges of the tree T. It satisfies

$$\delta_w = \left|\sigma^{-1}(w)\right| = \begin{cases} 0 & \text{for } w \in \{v\} \cup \{u_1, \ldots, u_p\}, \\ 2 & \text{for } w \in N(S) = \{h_1, \ldots, h_p\}, \text{ and} \\ 1 & \text{for the other nodes.} \end{cases}$$

Thus, the left-hand side of (21) is the same as (22), completing the proof. □

4.2.2 New facet-defining inequalities

In addition to the path, star and tree inequalities, new facet-defining inequalities can be derived from (20). The smallest graph for which this happens has 7 nodes (see Fig. 3). The depicted orientation of the edges gives rise to the facet-defining inequality

$$x_1 + x_2 + x_4 + x_6 + x_7 - 2x_3 - 2x_5 \leq 1$$

that is not a path, star or tree inequality. In particular, Nodes 3 and 5 are incident to inward-pointing arcs only. For a tree inequality only the single root satisfies this property.

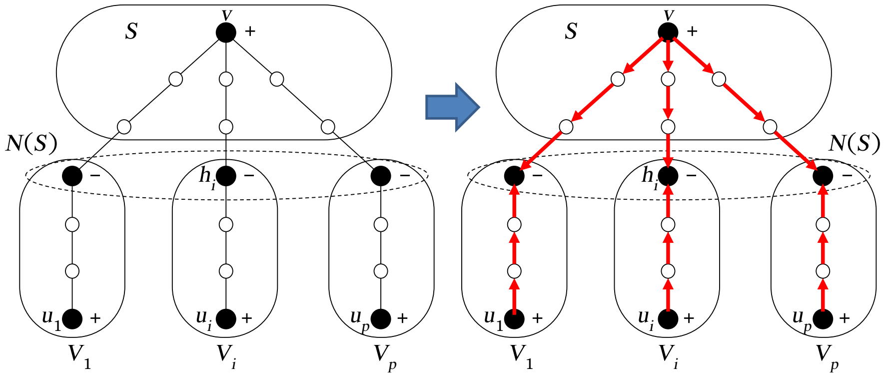

Fig. 2 The orientation σ implies the coefficients of a star inequalities

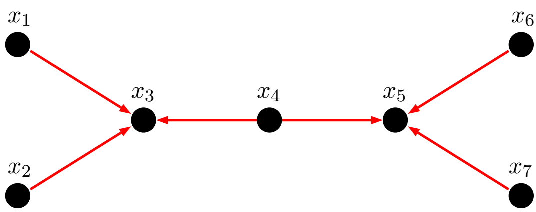

Fig. 3 The orientation σ implies a new type of facet-defining inequality

5 Computational results

We conducted computational experiments to see the performance of the proposed extended formulation. In our computational experiments we use only the inequalities (5)–(8) with the non-negativity constraints for y-variables. We do not include any other inequalities and our implementation of GUROBI does not add any other inequalities either. No presolve feature is used in GUROBI. Tested problem instances are constructed with trees from the phylogenetic tree repository, TreeBASE.org. As in Campêlo et al. [4], we solved CR problems with objective $w(u, t) = \mathbf{1}[C(u) = t]$ where $\mathbf{1}[C(u) = t] = 1$ if $C(u) = t$ and 0 otherwise.

Campêlo et al. [4] provided us with the 90 instances of colored phylogenetic trees which they tried to solve. They had selected ten phylogenetic trees from TreeBASE.org and randomly generated a coloring according to two probability parameters: $p_c = 0.005$, 0.05 or 0.5, the probability of changing the color, and $p_n = 0.25$, 0.5 or 0.75, the probability of noise. The root of the tree is assigned to the color 1, and in a recursive manner, the child node has the same color as its immediate forefather with probability $1 - p_c$ or has the next unused color with probability p_c. After generating the coloring, each node keeps its color with probability $1 - p_n$, or changes its color with probability p_n. If a color change occurs, a color is selected with equal probability across the available colors. Thus, for each tree, they generated nine instances with three p_c's and three p_n's. Among ten provided trees, one tree with $|V| = 1577$ in their paper includes two trees: ID of Tr9414 with $|V| = 900$ and ID of Tr6664 with $|V| = 678$. We perform a computational experiment on the 81 instances from the other

Table 1 Data set from TreeBASE.org

TB-ID	n	Note
Tr60729	213	Campêlo et al. [4]
Tr60079	301	
Tr6287	404	
Tr4755	567	
Tr2400	636	
Tr4756	710	
Tr6038	813	
Tr53777	981	
Tr25470	1441	
Tr69195	1838	Trees larger than those considered in Campêlo et al. [4]
Tr60915	2025	
Tr57261	2387	
Tr46272	2409	
Tr73427	2632	
Tr47159	4586	
Tr48025	5743	

nine trees except the one with $|V| = 1577$. Table 1 summarizes information of the nine phylogenetic trees from TreeBASE.org; the ID and the number of nodes in each of them. The computational results over the 81 instances are summarized in Sect. 5.1.

As shown in Table 1, we also attempted larger instances from TreeBASE than those attempted by Campêlo et al. [4]. One of the larger instances Tr48025 is the largest one among all trees of TreeBASE. The results on the larger instances are summarized in Sect. 5.2. The strength of the extended formulation is evident from the fact that GUROBI solved every instance we attempted at the root node without branching.

5.1 The instances used by Campêlo et al. [4]

We used *Python 2.7* as the programming language and *Gurobi 6.0* as the MIP Solver turning off all presolve features while Campêlo et al. [4] used *Java 1.6* as the language and *Gurobi 5.0* as the solver. Our computational experiments were carried out on a machine with 16 GB of RAM running a single thread on a 2.3 GHz processor while the experiments performed by Campêlo et al. [4] were carried out with 65 GB of RAM and 1.6 GHz processor.

Campêlo et al. [4]'s branch-and-cut algorithm is a two-phase algorithm. In the first phase, the conventional formulation without convexity constraints is solved and violated constraints are dynamically generated as cutting-planes and added as Lazy Constraints. In the second phase, the branch-and-cut algorithm is applied by generating most violated star and tree inequalities for each color with a certain stopping criterion to

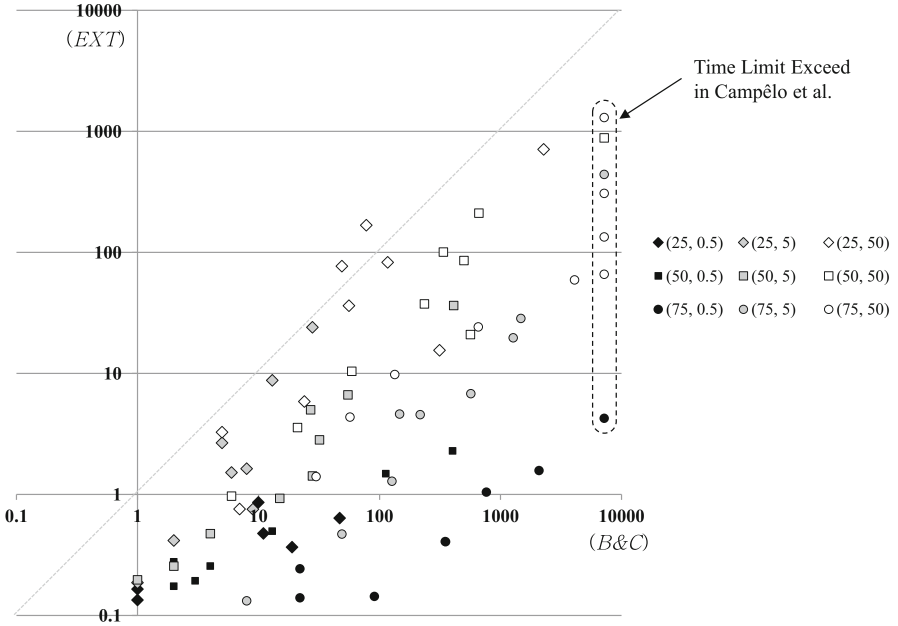

Fig. 4 Computation times of Campêlo et al. [4] ($B\&C$) versus the extended formulation (EXT)

generate cutting-planes. Each phase has one hour time limit. Some detailed parameters are determined with best setting from their experiments.

Our computational experiments use the extended formulation given by (5)–(8) and the non-negativity constraints for y-variables. We do not add any other valid inequalities in our computational experiments and GUROBI does not add any constraints either.

Figure 4 and Table 2 show the comparison between our computational results and those of Campêlo et al. [4]. In Fig. 4, the horizontal axis marked by ($B\&C$) represents the computational time (in log scale) of the branch-and-cut algorithm performed by Campêlo et al. [4] in seconds. The vertical axis marked by (EXT) represents the computational time of our extended formulation. Each mark corresponds to a problem instance. The marks under the diagonal line indicate the instances where the extended formulation outperformed Campêlo et al. [4]. The details of Fig. 4 are elaborated in Table 2. In Fig. 4, Time Limit Exceeded (TLE) of Table 2 are replaced by the time limit (2 h = 7200 s.)

The extended formulation was able to solve in under 22 min all TLE cases in Campêlo et al. [4]. For 79 instances out of total 81, the extended formulation beats Campêlo et al. [4] in terms of running time. The strength of the extended formulation is highlighted by the fact that every instance is solved by GUROBI at the root node without any need for branching. In fact, the LP-relaxation of the extended formulation using inequalities (5)–(7) and non-negativity provides integer solutions for all instances solved by us. This fact further validates the strength of the LP-relaxation of the extended formulation.

Table 2 Computation times of Campêlo et al. ($B\&C$) in [4] versus the extended formulation (EXT)

$(p_n, p_c) = (25\%, 0.5\%)$				$(p_n, p_c) = (25\%, 5\%)$				$(p_n, p_c) = (25\%, 50\%)$			
$\|V\|$	k	$B\&C$	EXT	$\|V\|$	k	$B\&C$	EXT	$\|V\|$	k	$B\&C$	EXT
213	3	1	0.0	213	14	0	0.2	213	103	7	0.8
301	6	0	0.1	301	20	1	0.2	301	148	5	3.3
404	6	1	0.1	404	29	2	0.4	404	197	24	5.9
567	4	3	0.1	567	29	9	0.8	567	277	314	15.5
636	6	1	0.2	636	35	6	1.5	636	333	49	76.8
710	10	19	0.4	710	35	8	1.6	710	361	56	36.2
813	9	11	0.5	813	42	5	2.7	813	430	117	82.6
981	13	10	0.9	981	65	13	8.8	981	469	78	167.7
1441	7	47	0.6	1441	75	28	24.0	1441	699	2275	709.1

$(p_n, p_c) = (50\%, 0.5\%)$				$(p_n, p_c) = (50\%, 5\%)$				$(p_n, p_c) = (50\%, 50\%)$			
$\|V\|$	k	$B\&C$	EXT	$\|V\|$	k	$B\&C$	EXT	$\|V\|$	k	$B\&C$	EXT
213	4	0	0.0	213	14	1	0.2	213	98	6	1.0
301	3	0	0.1	301	20	2	0.3	301	153	21	3.6
404	4	2	0.2	404	22	4	0.5	404	206	59	10.4
567	5	3	0.2	567	28	15	0.9	567	273	564	20.9
636	7	4	0.3	636	28	28	1.4	636	317	235	37.4
710	4	2	0.3	710	41	32	2.8	710	338	499	85.6
813	7	13	0.5	813	49	27	5.0	813	393	337	100.6
981	9	113	1.5	981	45	55	6.7	981	476	664	211.1
1441	10	403	2.3	1441	80	411	36.4	1441	681	TLE	881.8

$(p_n, p_c) = (75\%, 0.5\%)$				$(p_n, p_c) = (75\%, 5\%)$				$(p_n, p_c) = (75\%, 50\%)$			
$\|V\|$	k	$B\&C$	EXT	$\|V\|$	k	$B\&C$	EXT	$\|V\|$	k	$B\&C$	EXT
213	3	1	0.0	213	12	8	0.1	213	110	30	1.4
301	7	22	0.1	301	21	49	0.5	301	165	57	4.4
404	5	91	0.1	404	22	127	1.3	404	207	134	9.8
567	5	22	0.2	567	34	147	4.6	567	278	657	24.2
636	5	352	0.4	636	28	217	4.6	636	309	4102	59.3
710	6	349	0.4	710	36	570	6.8	710	361	TLE	65.8
813	9	2085	1.6	813	45	1273	19.7	813	425	TLE	133.8
981	6	765	1.0	981	50	1476	28.4	981	492	TLE	306.6
1441	9	TLE	4.3	1441	65	TLE	440.4	1441	750	TLE	1299.5

TLE time limit (2 h) exceeded

Table 3 Computation times of the extended formulation for larger instances

$(p_n, p_c) = (25\%, 0.5\%)$			$(p_n, p_c) = (25\%, 5\%)$			$(p_n, p_c) = (25\%, 50\%)$		
$\|V\|$	k	Time	$\|V\|$	k	Time	$\|V\|$	k	Time
1838	11	1.8	1838	114	57.2	1838	934	1127.4
2025	16	4.2	2025	91	48.2	2025	985	1541.2
2387	16	6.7	2387	129	173.4	2387	1221	4394.6
2409	7	2.7	2409	137	157.4	2409	1145	2855.8
2632	13	6.2	2632	126	133.5	2632	1314	6172.6
4586	24	36.9	4586	236	1537.9	4586	2275	OOM
5743	28	103.5	5743	312	4740.4	5743	2854	OOM
$(p_n, p_c) = (50\%, 0.5\%)$			$(p_n, p_c) = (50\%, 5\%)$			$(p_n, p_c) = (50\%, 50\%)$		
$\|V\|$	k	Time	$\|V\|$	k	Time	$\|V\|$	k	Time
1838	8	4.6	1838	87	55.5	1838	945	2599.1
2025	6	1.9	2025	101	78.1	2025	1007	2593.8
2387	22	37.0	2387	128	429.5	2387	1191	4899.7
2409	13	10.7	2409	112	251.0	2409	1231	5711.0
2632	18	34.8	2632	133	438.1	2632	1300	9427.2
4586	16	114.3	4586	228	1973.5	4586	2248	OOM
5743	29	1619.4	5743	316	39,324.2	5743	2895	OOM
$(p_n, p_c) = (75\%, 0.5\%)$			$(p_n, p_c) = (75\%, 5\%)$			$(p_n, p_c) = (75\%, 50\%)$		
$\|V\|$	k	Time	$\|V\|$	k	Time	$\|V\|$	k	Time
1838	8	5.0	1838	106	713.6	1838	919	3060.7
2025	12	19.5	2025	110	520.4	2025	1008	3542.6
2387	13	64.7	2387	122	3231.6	2387	1214	6753.3
2409	12	44.7	2409	115	2669.3	2409	1177	8070.1
2632	14	94.1	2632	136	4287.5	2632	1309	25,198.2
4586	22	945.3	4586	228	85,137.5	4586	2273	OOM
5743	27	3585.9	5743	302	LTLE	5743	2867	OOM

LTLE longer time (24 h) limit exceeded, *OOM* out-of-memory

Moreover, Campêlo et al. [4] made comparison in terms of the computation times of the cases with $p_c = 0.5\%$ between their algorithm and dynamic programming algorithm developed by Moran and Snir [18] and implemented by Campêlo et al [4]. Our running times with the extended formulation are significantly smaller than those of both algorithms.

5.2 Larger instances from TreeBASE.org

Given the computational effectiveness of the LP-relaxation of the extended formulation, we also considered larger phylogenetic trees from TreeBASE.org than those attempted by Campêlo et al [4]. In particular, we considered trees ranging from $|V| = 1838$ to $|V| = 5743$ as shown in Table 3. From seven larger trees, we generated 63 problem instances for different values of p_n and p_c as shown in Table 3. 51 out of the 63 problem instances were solved within the time limit of 2 h. Another five problems were solved but required more than 2 h, which are underlined in Table 3. One instance exceeded longer time limit (24 h) marked by LTLE. Six of the largest problems could not be solved because of insufficient memory.

From our computational experiments, we want to highlight the fact that every problem instance solved by GUROBI was solved at the root node without any need for branching. This experimental observation provides support for the strength of our extended formulation. This strength of the extended formulation has allowed us to solve much larger problems than were solved before.

6 Concluding remarks

We introduced an extended formulation for the convex recoloring problem on a tree using $O(nk)$ variables. The number of variables in the extended formulation is about twice that of the conventional formulation. The extended formulation also has $O(nk)$ constraints. We showed that the LP-relaxation of the extended formulation dominates the LP-relaxation of the conventional formulation along with three classes of (exponentially many) valid inequalities identified by Campêlo et al. [4].

Besides the quick run time, the LP-relaxation of the extended formulation provides an integer optimum for all problem instances we considered. This indicates that the LP-relaxation of the extended formulation is a very good approximation of the integer polytope. We also showed that the natural restriction of the extended formulation completely defines the subtree polytope on a tree. This provides some theoretical support to the strength of our LP-relaxation.

There exist, however, fractional extreme points to the LP-relaxation of the extended formulation even for path graphs. For example, the convex recoloring problem on a path graph T with $V(T) = \{1, 2, 3\}$ and $\mathcal{C} = \{1, 2\}$ has a fractional extreme point to the LP-relaxation of the extended formulation [(5)–(7) and non-negativity],

$$\left(x_{11} = x_{12} = x_{21} = x_{22} = x_{31} = x_{32} = y_{12,1} = y_{23,2} = 1/2;\, y_{12,2} = y_{23,1} = 0\right).$$

The extended formulation provided in this paper uses structural properties of a tree to obtain the subtree inequalities (6). These inequalities do not naturally extend to a general graph. A natural follow up to our work would be to obtain strong extended formulation for the CR problem on general graphs.

Acknowledgements We thank Professor Yoshiko Wakabayashi and the other authors of [4] for sharing their data. We are of course grateful for their paper that inspired our work. Mathieu Van Vyve and Bartosz Filipecki were supported by the Interuniversity Attraction Poles Programme P7/36 COMEX of the Belgian Science Policy Office and the Marie Curie ITN “MINO” from the European Commission.

References

1. Bar-Yehuda, R., Feldman, I., Rawitz, D.: Improved approximation algorithm for convex recoloring of trees. Theory Comput. Syst. **43**, 3–18 (2008)
2. Campêlo, M., Huiban, C.G., Sampaio, R.M., Wakabayashi, Y.: On the complexity of solving or approximating convex recoloring problems. Lect. Notes Comput. Sci. **7936**, 614–625 (2013)
3. Campêlo, M., Lima, K.R., Moura, P., Wakabayashi, Y.: Polyhedral studies on the CR problem. Electron. Notes Discret. Math. **44**, 233–238 (2013)
4. Campêlo, M., Freire, A.S., Lima, K.R., Moura, P., Wakabayashi, Y.: The convex recoloring problem: polyhedra, facets and computational experiments. Math. Progr. **156**, 303–330 (2016)
5. Chopra, S., Owen, J.L.: Extended formulations for the A-cut problem. Math. Progr. **73**, 7–30 (1996)
6. Chopra, S., Rao, M.R.: The partition problem. Math. Progr. **59**, 87–115 (1993)
7. Chor, B., Fellows, M., Ragan, M., Razgon, I., Rosamond, F., Snir, S.: Connected coloring completion for general graphs: algorithms and complexity. Lect. Comput. Sci. **4598**, 75–85 (2007)
8. Conforti, M., Kaibel, V., Walter, M., Weltge, S.: Subgraph polytopes and independence polytopes of count matroids. Oper. Res. Lett. **43**, 457–460 (2015)
9. Conforti, M., Cornuéjols, G., Zambelli, G.: Extended formulations in combinatorial optimization. 4OR **8**, 1–48 (2010)
10. Goemans, M.X.: The Steiner tree polytope and related polyhedra. Math. Progr. **63**, 157–182 (1994)
11. Kaibel, V.: Extended Formulations in Combinatorial Optimization, arXiv:1104.1023v1 [math.CO] 6 Apr (2011), manuscript
12. Kammer, F., Tholey, T.: The complexity of minimum convex coloring. Discret. Appl. Math. **160**, 810–833 (2012)
13. Kanj, I.A., Kratsch, D.: Convex recoloring revisited: complexity and exact algorithms. Lect. Notes Comput. Sci. **5609**, 388–397 (2009)
14. Korte, B., Lovász, L., Schrader, R.: Greedoids, Algorithms and Combinatorics, vol. 4. Springer, Berlin (1991)
15. Lima, K.R., Wakabayashi, Y.: Convex recoloring of paths. Discret. Appl. Math. **164**, 450–459 (2014)
16. Moran, S., Snir, S.: Convex recolorings of strings and trees: definitions, hardness results and algorithms, In: Proceedings WADS 2005: 9th International Workshop on Algorithms and Data Structures, pp. 218–232 (2005)
17. Moran, S., Snir, S.: Efficient approximation of convex recolorings. J. Comput. Syst. Sci. **73**, 1078–1089 (2007)
18. Moran, S., Snir, S.: Convex recolorings of strings and trees: definitions, hardness results and algorithms. J. Comput. Syst. Sci. **74**, 850–869 (2008)
19. Moran, S., Snir, S., Sung, W.K.: Partial convex recolorings of trees and galled networks: tight upper and lower bounds. ACM Trans. Algorithms **7**, 42 (2011)
20. Nemhauser, G.L., Wolsey, L.A.: Integer and Combinatorial Optimization. Wiley-Interscience, New York (1988)
21. Ponta, O., Hüffner, F., Niedermeier, R.: Speeding up dynamic programming for some NP-hard graph recoloring problems, In: Proceedings of the 5th International Conference on Theory and Applications of Models of Computation, TAMC08, pp. 490–501. Springer (2008)
22. Vanderbeck, F., Wolsey, L.A.: Reformulation and decomposition of integer programs. In: Michael Jünger, T., Liebling, D.N., George, N., William, P., Gerhard, R., Giovanni, R., Laurence, W. (eds.) Fifty Years of Integer Programming 1958–2008, pp. 431–502. Springer, Berlin (2010)
23. Wang, Y., Buchanan, A., Butenko, S.: On Imposing Connectivity Constraints in Integer Programs. www.optimization-online.org/DB_HTML/2015/02/4768.html