



HAL
open science

Fractional Path Coloring in Bounded Degree Trees with Applications

I. Caragiannis, Afonso Ferreira, C. Kaklamanis, Stéphane Pérennes, Hervé Rivano

► **To cite this version:**

I. Caragiannis, Afonso Ferreira, C. Kaklamanis, Stéphane Pérennes, Hervé Rivano. Fractional Path Coloring in Bounded Degree Trees with Applications. *Algorithmica*, 2010, 58 (2), pp.516-540. 10.1007/s00453-009-9278-3 . hal-00371052

HAL Id: hal-00371052

<https://hal.science/hal-00371052>

Submitted on 26 Mar 2009

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Fractional Path Coloring in Bounded Degree Trees with Applications*

I. Caragiannis[†] A. Ferreira[‡] C. Kaklamanis[†] S. Pérennes[‡] H. Rivano[‡]

January 12, 2009

Abstract

This paper studies the natural linear programming relaxation of the path coloring problem. We prove constructively that finding an optimal fractional path coloring is Fixed Parameter Tractable (FPT), with the degree of the tree as parameter: the fractional coloring of paths in a bounded degree trees can be done in a time which is linear in the size of the tree, quadratic in the load of the set of paths, while exponential in the degree of the tree. We give an algorithm based on the generation of an efficient polynomial size linear program. Our algorithm is able to explore in polynomial time the exponential number of different fractional colorings, thanks to the notion of *trace* of a coloring that we introduce. We further give an upper bound on the cost of such a coloring in binary trees and extend this algorithm to bounded degree graphs with bounded treewidth. Finally, we also show some relationships between the integral and fractional problems, and derive a $1 + 5/3e \approx 1.61$ -approximation algorithm for the path coloring problem in bounded degree trees, improving on existing results. This classic combinatorial problem finds applications in the minimization of the number of wavelengths in wavelength division multiplexing (WDM) optical networks.

Keywords: Fractional coloring, path coloring, linear relaxation, approximation algorithms, wavelength division multiplexing, optical networks, fixed parameter tractable problem

1 Introduction

Given a graph, the fundamental graph coloring problem (also known as vertex coloring) is to assign colors to vertices in such a way that adjacent vertices are assigned different colors and the number

*A preliminary version of this paper appears in *Proceedings of the 28th International Colloquium on Automata, Languages and Programming*, Crete, Greece, 2001. This work was supported in part by the European Union under IST FET Project AEOLUS.

[†]Computer Technology Institute, 61 Riga Feraiou str., Patras, Greece.

[‡]MASCOTTE Project, CNRS-I3S-INRIA, B.P. 93, 06902 Sophia Antipolis, France. A.Ferreira is currently on leave as Head of Science Operations at the COST Office, Brussels, <http://www.cost.esf.org>. Corresponding author: Herve.Rivano@sophia.inria.fr.

of colors used is minimized. In general, graph coloring is hard to solve to optimality or even to approximate [19].

A related problem, the *path coloring* problem, consists in coloring a set \mathcal{P} of paths on a graph \mathcal{G} so that two paths sharing an arc (if the graph is directed) or an edge of \mathcal{G} have different colors. This problem is equivalent to coloring the corresponding *conflict graph*, which is the undirected graph in which the vertices correspond to the paths of \mathcal{P} and are joined by an edge if the corresponding paths share an arc (or edge) of \mathcal{G} . Notice, however, that this problem was proven to be the same as standard graph coloring in terms of complexity or difficulty to approximate, since any n -vertex graph is the conflict graph of a set of paths of an $n \times n$ grid [8]. Furthermore, path coloring was proved to be \mathcal{NP} -hard even when \mathcal{G} is a tree [26].

This problem has many applications in wavelength division multiplexing (WDM) optical networks and call scheduling, which have recently triggered a renewed interest in path coloring on special classes of graphs. In such applications, one is given an optical network with n nodes and a multiset of point-to-point communication requests, and must assign to each request a lightpath and to each lightpath a color (wavelength) so that conflicting lightpaths (i.e., lightpaths using the same link) are assigned different colors. The goal is to minimize the number of colors used. This problem (known as the wavelength routing problem) has been widely studied in the literature [1, 3, 9, 11, 27]. It has been proved to be difficult (\mathcal{NP} -hard even for rings [27]), and one can show that there exist networks with $O(n^2)$ vertices and n requests on which it is hard to decide if the optimal number of colors is either 1 or n . Thus, in general, the problem is also hard to approximate. In the case where the lightpaths have already been assigned to requests, the problem is known as the wavelength assignment problem (WAP).

This problem can be reduced to coloring directed paths of a directed graph (since optical transmissions are one-way) usually symmetric. Note that there are important differences between the directed and the undirected version of the problem; for instance, the problem can be solved in polynomial time in symmetric directed stars, but is in \mathcal{NP} -hard for undirected stars [16].

Our work focuses on *fractional directed path coloring* on symmetric directed graphs, a natural linear programming relaxation of directed path coloring. We first remark that a coloring of directed paths is a covering of the paths by *independent sets of directed paths*. The relaxation arises when the independent sets may have fractional weights [17]. In this paper we prove exact and approximation results for directed path coloring and fractional directed path coloring in directed symmetric graphs. Unless otherwise specified, we shall use the terms paths and graphs to denote directed paths and symmetric directed graphs, respectively.

1.1 Previous work

Recently, several papers studied path coloring on simple networks like meshes, rings and trees [1, 3, 9, 15, 23, 25]. Most of the results study the relationship between the load of the set of paths (i.e., the maximum number of paths crossing an arc) and the number of colors of an optimal coloring. Notice that the load is a lower bound for the optimal number of colors.

For rings (where the problem is indeed the classical circular arc coloring problem), a $\frac{3}{2}$ approximation was proposed in [21]; this approximation ratio was recently improved to $1 + \frac{1}{e} \approx 1.37$ [23]. This latter result exploits a reduction of the circular arc coloring problem to a special instance of integral multicommodity flow problem [27]. The coloring is obtained by first solving the multicommodity flow problem with fractional numbers and then performing a randomized rounding of this solution.

For trees, a polynomial time deterministic algorithm which colors any set of paths of load l has been designed and proved to use at most $5l/3$ colors [9]. This algorithm is greedy and proceeds in phases, one per each node v of the tree. The nodes are considered following their breadth-first numbering. The phase associated with node v assumes that there is already a partial proper coloring where all paths that *touch* (i.e., start, end, or go through) nodes with numbers strictly smaller than v 's have been colored and no other path has been colored. It has also been proved that this algorithm is optimal within the class of greedy deterministic algorithms [9].

Recently, a different approach has been followed for binary trees giving rise to a randomized algorithm [1]. Instead of greedily computing a solution from top to bottom, this algorithm actually keeps a distribution of solutions (it computes from top to bottom one random element of the distribution). With high probability, an element of the distribution have the particularity to look locally random that is when considering the restriction of the distribution to each star composing the tree. This implies a kind of “average case” of the greedy deterministic algorithm and, hence, an improvement of the approximation ratio. This algorithm colors any set of paths of load l on a binary tree using at most $7l/5 + o(l)$ colors, with high probability. The hidden constants in the lower order terms are huge and are due to the integrality constraints of the problem and the random choices of the algorithm.

Note that these two results approximate the optimal number of colors within $5/3$ and $7/5 + o(1)$, respectively.

A recent survey of results on the path coloring problem in trees has recently been published [6].

1.2 Our results

Our approach, detailed in Section 2, is based on the fact that the graph coloring problem is equivalent to assigning unit cost to some independent sets of the graph such that all vertices are covered (i.e., are contained in an independent set of unit cost) and the total cost (i.e., the number of independent sets of unit cost) is minimized.

We then observe that Kumar’s fractional solution [23] gives in fact an optimal fractional coloring of paths in the ring. Notice, however, that this relaxation is generally also hard to approximate in general graphs.

The positive side of our observation is that it allows us to prove several results related to integral and fractional path coloring, as follows.

- In Section 3 we show that fractional path coloring in bounded degree trees can be solved in

polynomial time. This result is constructive, i.e., our algorithm recursively builds in polynomial time a polynomial size linear program whose final solution is such an optimal fractional coloring. For this we introduce the notion of *trace* of a coloring, which encodes a *local and aggregate* snapshot of the coloring of the paths crossing each edge (or pair of opposite arcs). Using the traces, we are able to explore the exponential number of different fractional colorings in polynomial time.

- This fractional path coloring algorithm for trees can be easily adapted to any bounded-degree graph with bounded-treewidth, as described in Section 4.1.
- We show an upper bound of $7l/5$ for the fractional path chromatic number in binary trees in terms of the load l of the set of paths (see Sections 4.2 and 4.3). This is somewhat related to the results in [1], since their randomized algorithm can be seen as an attempt to emulate a balanced fractional coloring and, conversely, a balanced fractional coloring can be interpreted as a perfect random sample of their algorithm. However, the analysis of our algorithm is easier, our algorithm is deterministic, and our bound is tighter. Note that a tighter upper-bound of $1.336l$ has been shown for the specific case of locally symmetric set of paths on binary trees [7].
- With respect to integral path coloring, in Section 5 we provide a randomized approximation algorithm for path coloring in bounded-degree trees with approximation ratio $1.61+o(1)$. This is done by applying randomized rounding to the fractional solution presented in Section 3. A subsequent work has recently tighten the approximation ratio to $1.511 + o(1)$ [5].

In the following section we recall the formal definitions of (fractional) coloring and (fractional) path coloring.

2 Fractional coloring

The graph coloring decision problem is usually considered as deciding on input of a graph \mathcal{G} and an integer k , whether it is possible to find a mapping from the vertices to a given set of k colors such that no two adjacent vertices are mapped to the same color. The related optimization problem consists of minimizing k , the size of the color set. In other words, the coloring problem is a constrained packing of the vertices into the set of colors. One can then express it as a polynomial size integer linear program (denoted ILP), but unfortunately its linear relaxation gives no practical information for building good integral solutions.

On the other hand, the coloring problem can also be seen as a *covering* problem. Recall that a covering of the vertices by a collection of covering sets is a subset of the collection such that each vertex belongs to at least one of the selected covering set. In the following we explain that the coloring problem is the problem of finding a minimum cost covering of the vertices by independent sets.

2.1 Vertex coloring as a covering problem

In a graph coloring problem, a set of vertices that can be mapped to the same color does not include any pair of adjacent vertices. We remark that this is the definition of an independent set of the graph. Therefore, a coloring of the vertices is a family of independent sets (the colors or, equivalently, the sets of vertices mapped to a common color) such that any vertex belongs to exactly one of these independent sets. In other words and defining the cost of a family of sets as the sum of the costs of each set, finding an optimal coloring of the graph is equivalent to finding a minimum cost integral *partition* of the vertices into independent sets of unit cost (i.e. each vertex belongs to exactly one independent set).

Furthermore, a covering which is not a partition (i.e. at least one vertex belongs to more than one independent set) induces a partition of cost at most the cost of the covering, since any subset of an independent set is still an independent set. Therefore, the graph coloring problem is also equivalent to finding a minimal cost integral *covering* of the vertices by independent sets of unit cost.

Given a graph $\mathcal{G} = (V, E)$, this means solving the following integer linear program:

$$\begin{aligned} & \text{minimize} && \sum_{I \in \mathcal{I}} x(I) \\ & \text{subject to} && \sum_{I \in \mathcal{I}: v \in I} x(I) \geq 1 && \forall v \in V \\ & && x(I) \in \{0, 1\} && \forall I \in \mathcal{I} \end{aligned}$$

where \mathcal{I} denotes the collection of the independent sets of \mathcal{G} and variable $x(I)$ is set to 1 if the independent set I belongs to the coloring (i.e. is a color), 0 otherwise.

Although this ILP may have an exponential number of variables (one per independent set), its linear relaxation will be useful in the following. When the weights over the independent sets are no longer constrained to integrality, the problem can be written as the following linear program:

$$\begin{aligned} & \text{minimize} && \sum_{I \in \mathcal{I}} \bar{x}(I) \\ & \text{subject to} && \sum_{I \in \mathcal{I}: v \in I} \bar{x}(I) \geq 1 && \forall v \in V \\ & && 0 \leq \bar{x}(I) \leq 1 && \forall I \in \mathcal{I} \end{aligned}$$

The corresponding combinatorial problem is called the *fractional coloring problem* [17], and the cost of an optimal solution is called *fractional chromatic number*, denoted $\chi_f(\mathcal{G})$.

If \bar{x} is a cost function over the independent sets of the graph \mathcal{G} satisfying the constraints of the above linear program, we call it a *fractional coloring* of \mathcal{G} . We use the symbol $\bar{x}(\mathcal{G})$ to denote the cost of the solution \bar{x} .

In general, the fractional chromatic number is as hard to approximate as the chromatic number up to a logarithmic factor since Lovász's theorem on fractional covers [24] states that any ρ -approximation of the fractional chromatic number leads to a $\rho \cdot \log(n)$ -approximation of the chromatic number. Note that the logarithmic factor is negligible from a complexity point of view since the chromatic number is non-approximable within $n^{1-\epsilon}$ for all $\epsilon > 0$ [10].

Notwithstanding, this approach is very useful if we use duality results, as follows.

Lemma 1 *Computing the fractional chromatic number is polynomially equivalent to solving the maximum weighted independent set problem.*

Proof: It is well-known that the dual of the linear program of a covering problem is the linear program of the packing problem where there is a constraint per variable of the primal and a variable per primal constraint. Therefore, the dual of the linear program of the fractional coloring problem is the following [17]:

$$\begin{aligned} & \text{maximize} && \sum_{v \in V} y(v) \\ & \text{subject to} && \sum_{v \in I} y(v) \leq 1 \quad I \in \mathcal{I} \\ & && y(v) \geq 0 \end{aligned}$$

Let us define the weight of an independent set $I \in \mathcal{I}$ as the sum of the weights of its vertices: $y(I) = \sum_{v \in I} y(v)$.

In this dual problem, a nontrivial constraint is violated if and only if the weight of one independent set I is greater than 1. Hence, the *maximum weighted independent set problem* (MWIS) is a separation oracle for this last problem. According to the separation and optimization equivalence ([18], Th. 4.2.7), the dual of fractional coloring and the MWIS are equivalent up to a polynomial reduction. Moreover, by duality, computing the fractional chromatic number and the optimum of its dual are polynomially equivalent. Hence the lemma. □

Note that this duality argument does not provide any effective fractional coloring algorithm, but rather a way to compute the fractional chromatic number in specific cases.

On the positive side, we can notice that in the dual linear program, some constraints are obviously useless. If I is a maximal independent set for inclusion, any independent set $J \subseteq I$ will induce a constraint parallel to (considered as a cut hyperplane of the linear program's polytope), and less restrictive than the one induced by I . This means that for a maximal independent set of size s , $2^s - 1$ constraints are useless and could be dropped from the linear program, hence removing an exponential number of constraints. The intuition is then that the huge number of constraints of the dual (or, equivalently, variables of the primal) may not be a drawback when solving some classes of this problem.

The extension of the definition of fractional coloring to *fractional path coloring* is given below. We also show that the fractional path chromatic number of paths in bounded-degree trees can be computed in polynomial time using Lemma 1.

2.2 Fractional path coloring

As stated in the Introduction, the vertex coloring and path coloring are equivalent problems, since coloring a set of paths in a graph is equivalent to coloring the corresponding *conflict graph*, defined below.

Definition 1 Given \mathcal{G} a symmetric directed graph, and \mathcal{P} a set of paths in \mathcal{G} , the conflict graph of \mathcal{P} in \mathcal{G} is the graph whose vertices represent the paths of \mathcal{P} and where there is an edge between vertices representing paths sharing an arc of \mathcal{G} .

We are now able to extend some terms of graph coloring to path coloring.

Given a set of paths \mathcal{P} in a graph \mathcal{G} , we define an *independent set of paths* as a set of pairwise arc-disjoint paths, i.e., a set of paths whose corresponding vertices form an independent set of the conflict graph¹. We can hence define the *fractional path coloring problem* analogously to the vertex coloring problem.

Definition 2 Given \mathcal{G} a graph and \mathcal{P} a set of paths in \mathcal{G} , an *optimal fractional coloring of \mathcal{P}* is a minimum cost fractional covering of the paths by independent sets of unit cost.

The corresponding linear program is the same as the fractional vertex coloring, and Lemma 1 still holds. We will denote by $\chi(\mathcal{G}, \mathcal{P})$ ($\chi_f(\mathcal{G}, \mathcal{P})$) the (fractional) path chromatic number of \mathcal{P} on \mathcal{G} . We will also denote by $l(\mathcal{G}, \mathcal{P})$ the load of \mathcal{P} on \mathcal{G} , i.e. the maximum number of paths of \mathcal{P} using any given arc of \mathcal{G} .

As far as the relationship between χ and χ_f is concerned, the following result is implicit in the work of Kumar [23], which addresses the case of the n -nodes ring, denoted by C_n :

$$\chi(C_n, \mathcal{P}) \leq \chi_f(C_n, \mathcal{P}) + \frac{l(C_n, \mathcal{P})}{e} + o(l(C_n, \mathcal{P})).$$

In this context, e is the well-known Neper constant.

In the very specific case where the conflict graph is a proper circular arc graph, Niessen and Kind [25] proved that $\chi = \lceil \chi_f \rceil$.

In the following we prove that computing the fractional chromatic number of a set of paths in a bounded-degree tree can be done in polynomial time.

Theorem 2 Given \mathcal{T} a bounded-degree tree and \mathcal{P} a set of paths in \mathcal{T} , the fractional chromatic number of \mathcal{P} in \mathcal{T} , $\chi_f(\mathcal{T}, \mathcal{P})$, can be computed in polynomial time.

Proof: Lemma 1 claims the polynomiality equivalence between the computation of the fractional chromatic number and the maximum weighted independent set.

Considering that \mathcal{T} is rooted at any given vertex r , the maximum weighted independent set of \mathcal{P} in \mathcal{T} is obtained by a call to an algorithm proposed and proved polynomial by Garg [13, 14]. This algorithm is reproduced as Algorithm 1.

□

This result proves that it is possible to compute the fractional chromatic number of a set of paths in a bounded-degree tree in polynomial time, but does not give an algorithm to obtain a

¹In particular, such independent sets may include path crossing opposite arcs, the graph being directed.

Algorithm 1 Garg’s algorithm: MWIS algorithm for paths \mathcal{P} in a tree \mathcal{T}

Input: \mathcal{T}_v subtree of \mathcal{T} rooted at node v , d_v number of children of v , r root of \mathcal{T}

Input: On each arc, a “void” path of length 1 is defined and denoted \emptyset

Output: MWIS of paths \mathcal{P} in a \mathcal{T} and its weight

Data: $\forall v, p, q$, $I[v, p, q]$ MWIS of paths in \mathcal{T}_v with paths p, q touching v oppositely

Data: $w[v, p, q]$ weight of $I[v, p, q]$ *not counting* $w(p)$ *nor* $w(q)$

for all node v such that its children have been processed **do**

for all paths p, q touching v oppositely **do**

 let c_j , $j = 1 \dots d_v$, be the j^{th} child of v

$I[v, p, q] \leftarrow \emptyset$, $w[v, p, q] \leftarrow 0$

for all $P = \{p_j, q_j, j = 1 \dots d_v\}$ s. t. $p, q \in P$ and $\forall j$, p_j, q_j touch both v and c_j **do**

$w(P) \leftarrow \sum_{\rho \in P \setminus \{p, q\}} w(\rho)$ { p and q are not counted}

if $w(P) + \sum_j w[c_j, p_j, q_j] > w[v, p, q]$ **then**

$I[v, p, q] \leftarrow \cup_j I[c_j, p_j, q_j]$, $w[v, p, q] \leftarrow w(P) + \sum_j w[c_j, p_j, q_j]$

 mark v as processed

Return $I[r, \emptyset, \emptyset]$, $w[r, \emptyset, \emptyset]$

corresponding fractional coloring. The main argument of the proof is the fact that the dynamic programming tables generated by the algorithm have size $O(l^{2d})$, where l is the load of the set of paths and d the maximum degree of the tree.

Moreover, due to the exponential number of variables involved in the fractional coloring linear program, it is not straightforward to design a strategy relying on the fractional chromatic number to compute an optimal fractional coloring. Nevertheless, we give in the following section an algorithm which builds a small linear program in order to effectively produce solutions of the fractional path coloring problem in bounded-degree trees. The strategy of our algorithm is closely related to Garg’s algorithm: our linear program is based on subproblems that are exponential in the degree of the tree.

3 An algorithm for fractional path coloring in bounded-degree trees

In this section we study the fractional path coloring problem in trees. We denote a path going from node u to v by $(u \rightsquigarrow v)$, and a path going from u to v using the arc (a, b) by $(u \rightsquigarrow a \rightarrow b \rightsquigarrow v)$. The special path \emptyset denotes a non-existent or void path on any arc.

The main idea of our algorithm is to recursively build a linear program of polynomial size. In the following we will prove that two fractional path colorings of two subtrees can be *merged* into one for the whole tree if and only if they coincide on the arc where they overlap. This may still yield an exponential algorithm since there may exist an exponential number of different fractional path colorings on an arc. In order to cope with this difficulty, we define an aggregate coding of the

“behavior” of a coloring on a pair of opposite arcs, or edge, called the *trace* of this coloring.

3.1 Trace of a fractional coloring

Given a tree \mathcal{T} , and a solution for path coloring that is a weight function \bar{x} on the independent sets of paths in the tree \mathcal{T} , we define the *trace of the fractional coloring \bar{x} on an edge e* as the function $X^e : \mathcal{P}^2 \rightarrow \mathbb{R}$, defined as follows.

- For any path p , let $\mathcal{I}(p)$ be the set of independent sets containing p , and let $\mathcal{I}(p, q) = \mathcal{I}(p) \cap \mathcal{I}(q)$.
- Let $\mathcal{I}^e(p, \emptyset)$ be the set of independent sets where only p crosses e . Let $\mathcal{I}^e(\emptyset, \emptyset)$ be the set of independent sets not using e .
- For any pair of paths p, q crossing e in opposite directions, let the trace variable for p and q on e be $X^e(p, q) = \sum_{I \in \mathcal{I}(p, q)} \bar{x}(I)$.

The trace function is a *local* and *aggregate* snapshot of the fractional coloring. Local, because only the paths going through the edge e are considered and two different colorings may have the same trace if their difference does not affect the coloring of these paths. Aggregate, because the trace function does not code all the weights of the independent sets, only sums of these variables. In the following we will show that this loss of information is not critical, even if it allows to code an exponential number of different colorings with polynomial size objects.

The *trace \mathbb{X} of the fractional coloring \bar{x}* is the set of the trace functions on all edges of \mathcal{T} : $\mathbb{X} = \{X^e, \forall e \in \mathcal{T}\}$.

Given an instance of fractional path coloring $(\mathcal{T}, \mathcal{P})$ and a constant c , we denote by $Tr_c(\mathcal{T}, \mathcal{P})$ the set of all the fractional colorings of $(\mathcal{T}, \mathcal{P})$ with cost less than c .

3.2 Split and merge

Our algorithm proceeds by recursively constructing a polynomial size linear program whose solution provides a fractional coloring. Our induction is based on *merge* and *split* procedures which allow to build any instance of fractional path coloring starting from instances on stars (i.e. a tree where only one vertex has degree larger than 1) and merging them step by step.

Consider a fractional path coloring of $(\mathcal{T}, \mathcal{P})$, where \mathcal{T} is not a star and such that $e = \{u, v\}$ is a non-terminal edge of \mathcal{T} . The splitting of \mathcal{T} on edge e yields two smaller instances of fractional path coloring, denoted $(\mathcal{T}_0, \mathcal{P}_0)$ and $(\mathcal{T}_1, \mathcal{P}_1)$, as described by Algorithm 2 (cf. Figure 1 for an illustration). Note that the split procedure of any fractional coloring of \mathcal{T} with cost c induces fractional colorings on $(\mathcal{T}_0, \mathcal{P}_0)$ and $(\mathcal{T}_1, \mathcal{P}_1)$ with cost less than c . From construction of the split procedure, both have the same trace on e .

The split procedure is reverted by the *merge* procedure, described in Algorithm 3 and working as follows. Once two subproblems are generated by the split of $(\mathcal{T}, \mathcal{P})$ into $(\mathcal{T}_0, \mathcal{P}_0)$ and $(\mathcal{T}_1, \mathcal{P}_1)$,

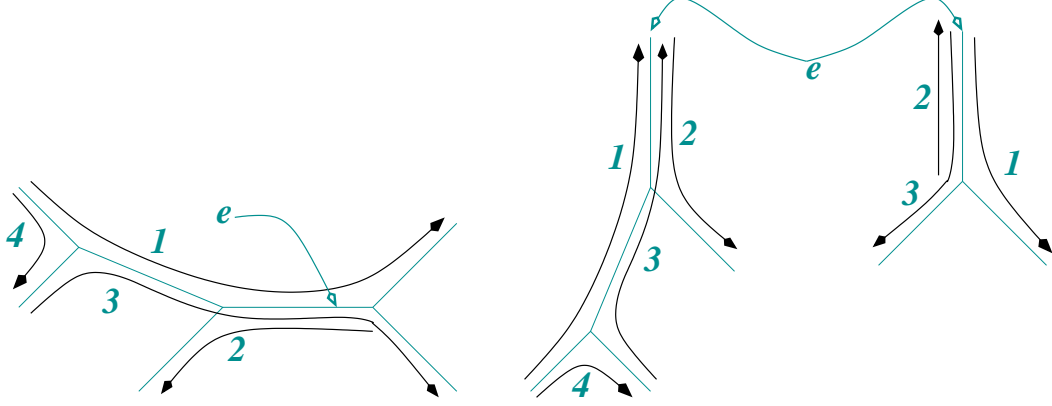


Figure 1: Splitting a tree on e .

if a fractional coloring is computed on each subproblem such that they have the same trace on the splitting edge e , the merge builds a fractional coloring of $(\mathcal{T}, \mathcal{P})$, as proved in proposition 3.

Now, we prove that any element of $Tr_c(\mathcal{T}, \mathcal{P})$ can be obtained from one element of $Tr_c(\mathcal{T}_0, \mathcal{P}_0)$ and one of $Tr_c(\mathcal{T}_1, \mathcal{P}_1)$.

Proposition 3 *The set of all possible merging of elements of $Tr_c(\mathcal{T}_0, \mathcal{P}_0)$ and $Tr_c(\mathcal{T}_1, \mathcal{P}_1)$ is included in $Tr_c(\mathcal{T}, \mathcal{P})$ and is empty if and only if $Tr_c(\mathcal{T}, \mathcal{P})$ is empty.*

Proof: Given two fractional colorings $\bar{x}_0 \in Tr_c(\mathcal{T}_0, \mathcal{P}_0)$ and $\bar{x}_1 \in Tr_c(\mathcal{T}_1, \mathcal{P}_1)$ having the same trace X_0^e and X_1^e on e , the merge procedure (Algorithm 3) applied to \bar{x}_0 and \bar{x}_1 produces an element of $Tr_c(\mathcal{T}, \mathcal{P})$. To verify this claim, just note that this procedure preserves the following invariants at each step:

- Paths are covered either by \bar{x}_0 , \bar{x}_1 or \bar{x} ;
- $\bar{x}(\mathcal{P}) + \bar{x}_0(\mathcal{P}_0) = \bar{x}(\mathcal{T}, \mathcal{P}) + \bar{x}_1(\mathcal{P}_1) = c$.

Algorithm 2 SPLIT $(\mathcal{T}, \mathcal{P})$ on e into $(\mathcal{T}_0, \mathcal{P}_0)$ and $(\mathcal{T}_1, \mathcal{P}_1)$

Input: $e = \{u_0, u_1\}$ non-terminal edge of \mathcal{T}

$(V_i, E_i) \leftarrow$ connected components of $\mathcal{T} \setminus e \ni u_i, i = 0, 1$

$\mathcal{T}_i \leftarrow (V_i \cup \{u_{1-i}\}, E_i \cup \{e\}), i = 0, 1$

for all $p \in \mathcal{P}$ **do**

if $\exists i, p \in U_i$ **then**

$\mathcal{P}_i \leftarrow p$

else if $\exists i, p = (a \rightsquigarrow u_i \rightarrow u_{1-i} \rightsquigarrow b), a \in U_i, b \in U_{1-i}$ **then**

$\mathcal{P}_i \leftarrow p = (a \rightsquigarrow u_{1-i})$

$\mathcal{P}_{1-i} \leftarrow p = (u_i \rightsquigarrow b)$

Algorithm 3 MERGE $\bar{x}_0 \in Tr_c(\mathcal{T}_0, \mathcal{P}_0)$, $\bar{x}_1 \in Tr_c(\mathcal{T}_1, \mathcal{P}_1)$ into $\bar{x} \in Tr_c(\mathcal{T}, \mathcal{P})$

Input: Trace equality on edge $e = \{u_0, u_1\}$: $X_0^e = X_1^e$, and $\mathcal{P}_i = \{p_{\mathcal{T}_i}, p \in \mathcal{P}\}$

Ouput: $X^e = X_0^e = X_1^e$

$\forall I \in \mathcal{I}$, $\bar{x}(I)$ is implicitly initialized with value 0.

while $\exists p_0, q_0 \in \mathcal{P}_0 \cup \{\emptyset\} | X_0^e(p_0, q_0) \neq 0$ **do**

 {Trace equality ensures $\exists p, q \in \mathcal{P}, p_0 = p|_{\mathcal{T}_0}, q_0 = q|_{\mathcal{T}_0}$ and $X_1^e(p|_{\mathcal{T}_1}, q|_{\mathcal{T}_1}) > 0$ }

 select $I_0 \in \mathcal{I}_0(p_0, q_0)$ such that $\bar{x}_0(I_0) > 0$

 select $I_1 \in \mathcal{I}_1(p|_{\mathcal{T}_1}, q|_{\mathcal{T}_1})$ such that $\bar{x}_1(I_1) > 0$

$\bar{x}_{min} \leftarrow \min(\bar{x}_0(I_0), \bar{x}_1(I_1))$

$I \leftarrow I_0 \setminus \{p_0, q_0\} \cup I_1 \setminus \{p|_{\mathcal{T}_1}, q|_{\mathcal{T}_1}\} \cup \{p, q\}$ {By construction $I \in \mathcal{I}$ }

 increase $\bar{x}(I)$ by \bar{x}_{min}

 decrease $\bar{x}_i(I_i)$ by \bar{x}_{min} {Ensures trace equality}

It follows that, at the end of the main loop, \bar{x} is a fractional coloring of $(\mathcal{T}, \mathcal{P})$ with cost c . Moreover, for $i = 1, 2$, the trace of \bar{x} on any edge of \mathcal{T} equals the trace of \bar{x}_i in \mathcal{T}_i . Therefore, the set of all possible merging of elements of $Tr_c(\mathcal{T}_0, \mathcal{P}_0)$ and $Tr_c(\mathcal{T}_1, \mathcal{P}_1)$ is included in $Tr_c(\mathcal{T}, \mathcal{P})$.

Moreover, as stated before, the split of any element of $Tr_c(\mathcal{T}, \mathcal{P})$ produces two elements of $Tr_c(\mathcal{T}_0, \mathcal{P}_0)$ and $Tr_c(\mathcal{T}_1, \mathcal{P}_1)$ that can be merged together into an element of $Tr_c(\mathcal{T}, \mathcal{P})$. \square

Corollary 4 *We can compute a polynomial size linear program whose solutions are valid traces of elements of $Tr_c(\mathcal{T}, \mathcal{P})$.*

Proof: We first assume that such a program exists for bounded degree d stars. We then use a recursive algorithm to generate a linear program whose solutions are traces of $Tr_c(\mathcal{T}, \mathcal{P})$. Assume that \mathcal{T} can be split on edge e into $(\mathcal{T}_0, \mathcal{P}_0)$ and $(\mathcal{T}_1, \mathcal{P}_1)$ and let S_0 (S_1) be the linear program for traces of $Tr_c(\mathcal{T}_0, \mathcal{P}_0)$ ($Tr_c(\mathcal{T}_1, \mathcal{P}_1)$), where we assume that the trace on e is associated to the variables $X_0^e(p, q)$ and $X_1^e(p, q)$. Then, a system for $(\mathcal{T}, \mathcal{P})$ is:

$$S = S_0 \cup S_1 \cup \{X_0^e(p, q) = X_1^e(p, q) \mid \forall p, q\}$$

Therefore, if l denotes the load of the set of paths, there are $O(l^2)$ constraints per edge of the tree (the trace equality constraints) and one system per star, i.e. per non-terminal node of the tree. If s is a bound on the size of a linear program for a star, the size of our linear program is then $O(n(l^2 + s))$.

A naïve way to get a polynomial size linear program for the bounded-degree d stars case is to set one variable per independent set, which are polynomial in number, as explained below.

The number of paths composing an independent set is obviously at most $2d$ and there are at most $2dl$ paths in a bounded degree d star whose load is l . Therefore, the number of independent sets composed of i paths is at most $\binom{2dl}{i} \leq (2dl)^{2d}$.

Consequently, there are less than $2d(2dl)^{2d} = O(l^{2d})$ independent sets on a bounded degree d star of load l , and as many variables in the linear program.

Fractional coloring is trivially described from these variables, and each trace variable is obtained, by definition, as the addition of some of these variables. \square

Proposition 5 *Fractional path coloring in bounded-degree d trees can be reduced to solving a polynomial size linear program.*

Proof: We simply have to show how to compute an element of $Tr_c(\mathcal{T}, \mathcal{P})$ from a trace \mathbb{X} of an element of $Tr_c(\mathcal{T}, \mathcal{P})$ (obtained from Corollary 4). Again we proceed inductively: we start from stars and find for each one a fractional coloring having the trace that \mathbb{X} induces on it. Then, if $(\mathcal{T}, \mathcal{P})$ can be split into $(\mathcal{T}_0, \mathcal{P}_0)$ and $(\mathcal{T}_1, \mathcal{P}_1)$, Proposition 3 provides a way to merge fractional colorings of $(\mathcal{T}_0, \mathcal{P}_0)$ and $(\mathcal{T}_1, \mathcal{P}_1)$, when their traces are equal, and this is the case since they are subsets of \mathbb{X} . \square

3.3 Fractional coloring is Fixed Parameter Tractable

Note that the model above induces very large systems, since we could get $O(nl^6)$ variables for binary trees, and $O(nl^8)$ for ternary ones. In this section we show that fractional path coloring on trees is FPT [12]. Indeed, by identifying the independent sets of a the paths on a star to a perfect matching of the complete bipartite graph, one can describe the trace variable on a star with flow equations.

It is therefore possible to reduce the system size to $O(dM(d)l^2n)$, that is polynomial in the problem size, while exponential only in the parameter d .

Proposition 6 *Fractional path coloring in bounded-degree d trees can be reduced to solving a linear program of size $O(dM(d)l^2n)$.*

Proof: First, and without loss of generality, we assume that the load is uniform. Corollary 4 shows that the size of the linear program of the fractional path coloring of $(\mathcal{T}, \mathcal{P})$ of cost less than c is $O(nl^2)$ plus the sum of the sizes of the linear programs for each star of the tree \mathcal{T} , which is less than n times the size for a star of degree d .

Then, remark that if each possible path is present exactly once (what we will call a *canonical d -star*), then the set of all independent sets loading each arc exactly once (*canonical independent sets*) is isomorphic to the set of the perfect matchings in $K_{d,d}$. Let $M(d)$ denote the size of this set and note that $M(d)$ depends only on d .

It is possible to describe the polytope of the matching of this bipartite graph with simple flow equations [22], as follows.

Let us consider a star $\mathcal{S} = \{v_0, v_1, \dots, v_d\} \subseteq \mathcal{T}$, where v_0 is the center of \mathcal{S} and d its degree. Arcs of \mathcal{S} are $\{(v_0, v_i), (v_i, v_0), \forall i \in [1, \dots, d]\}$. Let \mathcal{M} be the set of independent sets of load 1 of \mathcal{S} . For all $M \in \mathcal{M}$, we build an auxiliary flow network F_M , as follows.

- For each path $p \in T$ including a path of M , we add a vertex $V(p)$ to F_M .
- For each pair of vertices $V(p), V(q)$ such that $\exists i \in [1, \dots, d] \mid (v_0, v_i) \subseteq p$ and $(v_i, v_0) \subseteq q$, we add the arc $(V(p), V(q))$ to F_M .
- We define a flow function² f_M on the arcs of F_M which induces a flow function f'_M on its vertices.
- For each leaf v_i of the star we add to F_M the constraint on f'_M

$$\sum_{p|(v_0, v_i) \subseteq p} f'_M(V(p)) = c_M.$$

Any solution to the previous system induces a covering of the paths with weight $f'(p)$ and a subpart of the trace of value $f(p, q)$.

The total system for a star is obtained as follows.

- The flow equations for each matching $M \in \mathcal{M}$: define f_M variables;
- $X(p, q) = \sum_{M \in \mathcal{M}} f_M(p, q)$: computation of the trace variables;
- $\sum_{M \in \mathcal{M}} f'_M(p) \geq 1$: covering of the paths;
- $\sum_{M \in \mathcal{M}} c_M \leq c$: cost constraint.

Due to the above mentioned property of the matching polytope, this system describes the fractional coloring problem for a star with an encoding of the trace variables.

It follows that the fractional coloring of each star of \mathcal{T} can be described with a system of size at most $M(d)(= |\mathcal{M}|)$ times l^2d . Hence, the system for \mathcal{T} is of size $O(dM(d)l^2n)$. \square

Notice that an optimal fractional coloring can be computed in polynomial time even if the degree of the tree is $d = O(\max\{\log l / \log \log l, \log n / \log \log n\})$, since $M(d) = O(d!)$.

4 Extensions and applications

In this section we extend the technique described in the previous section to bounded degree graphs with bounded treewidth.

Furthermore, we obtain an algorithm for fractional path coloring on binary trees with cost at most $7l/5$. Using the results for the fractional path coloring, we achieve improved approximation algorithms for the path coloring problem in bounded-degree trees and bounded-degree trees of rings, and upper bounds for the path chromatic number in terms of the fractional path chromatic number and the load.

²By flow function we mean conservative for vertices. One can note that this function is a circulation since no sink and no source are present.

4.1 Some polynomial instances

In the case of bounded degree graphs with bounded treewidth, we obtain the following.

Proposition 7 *Fractional path coloring can be solved in polynomial time in bounded degree graphs with bounded treewidth.*

Proof: The proof follows similar lines with Section 3. Consider a tree-decomposition of the graph of width bounded by a given constant k' . An “edge” of the tree-decomposition is a cut of the graph that is also upper-bounded by a given $k = \Omega(k'^2)$ (see [4] for details). A node of the tree is a set of $\Omega(k')$ vertices of the initial graph.

One can then define trace variables on each edge of the decomposition of size bounded by $O(l^{2k})$ and build a polynomial size linear program. \square

4.2 Fractional path coloring on binary trees

Fractional path coloring on binary trees can be performed using a particular coloring, called *balanced coloring*, where the traces depend only on the number of paths going through an arc.

Definition 3 *For any given constant $w \in [0, 1]$, a w -balanced coloring of a set of paths is a fractional path coloring such that the trace function over any arc e is $\frac{w}{l(e)}$ for all paths using this arc. I.e.,*

$$X^e(p, q) = \frac{w}{l(e)}, \quad X^e(p, \emptyset) = 1 - w, \quad \forall e, \quad \forall p, q \text{ paths using } e \text{ in opposite directions.}$$

This particular kind of fractional path coloring will help us to exhibit an upper bound on the cost of a fractional path coloring on binary trees.

Corollary 8 *For any given constant $w \in [0, 1]$, any set of paths \mathcal{P} in any binary tree \mathcal{T} can be w -balanced colored by independently coloring all the restrictions of \mathcal{P} to each of the ternary stars composing \mathcal{T} .*

Proof: If one performs independent local colorings of all the ternary stars composing \mathcal{T} , then, by definition of the w -balanced coloring, the trace equality equations of the linear program built in Section 3 are respected. Therefore, as stated in the proof of Proposition 5, one can find a global fractional coloring consistent with the local colorings using the merge procedure.

Moreover, the cost of such a w -balanced coloring is the maximum, over all the ternary stars composing \mathcal{T} , of the local w -balanced colorings. \square

A simple but exhaustive analysis shows that ternary stars can be colored in a balanced way with at most $\frac{7}{5}l$ colors – detailed in the next section –, this yields the following Proposition.

Proposition 9 *For any set of paths \mathcal{P} of load $l(T, \mathcal{P})$ on a binary tree T , there exists a fractional coloring of cost at most $\frac{7l(T, \mathcal{P})}{5}$. Moreover, such a fractional coloring can be computed in polynomial time.*

This result is proved in Section 4.3, by showing that there exists a $\frac{4}{5}$ -balanced coloring of \mathcal{P} with cost at most $\frac{7l(T,\mathcal{P})}{5}$ for any set of path \mathcal{P} on any ternary star T .

As a corollary, we obtain that given a set of paths \mathcal{P} on a binary tree T , the maximum independent set of paths of \mathcal{P} has size at least $\frac{5|\mathcal{P}|}{7l(T,\mathcal{P})}$.

4.3 Balanced fractional coloring of ternary stars

This section is dedicated to the case analysis of the w -balanced coloring of paths in ternary stars. In order to get an upper bound of the cost of any balanced coloring, we first define *reduced 3-stars*.

4.3.1 Reduced 3-stars

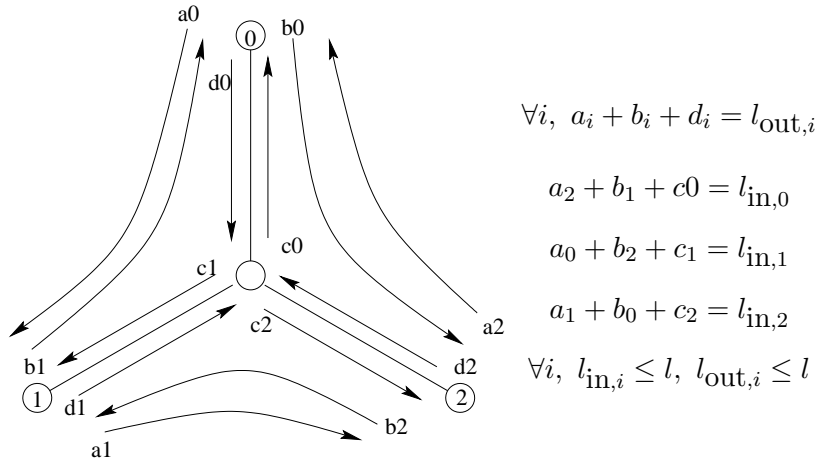


Figure 2: The generic 3-star and its parameters.

A generic set of paths in a ternary star can be described by 12 parameters as depicted in Figure 2. This is too large to permit an exhaustive case study. In order to make such an analysis tractable, we use a standard reduction [1], Algorithm 4, which transforms any ternary star into a *reduced 3-star* which is described by only 3 parameters.

Algorithm 4 Reduction of a set of paths \mathcal{P} in a ternary star

Input: Central vertex = n_c , leaves = $n_i, i = 0, 1, 2$

Let l be the load of \mathcal{P}

for all under-loaded arc e **do**

add paths of length 1 on e to reach load l

repeat

select p , path of length 1 from n_i to n_c

select q , path of length 1 from n_c to $n_j, j \neq i$

concatenate p and q into a path ($n_i \rightarrow n_c \rightarrow n_j$)

until there are no more paths to concatenate

The resulting reduced 3–star, depicted in Figure 3, has a more compact description than the generic one, allowing an analysis which will give an upper bound on the cost of the fractional coloring of any ternary star.

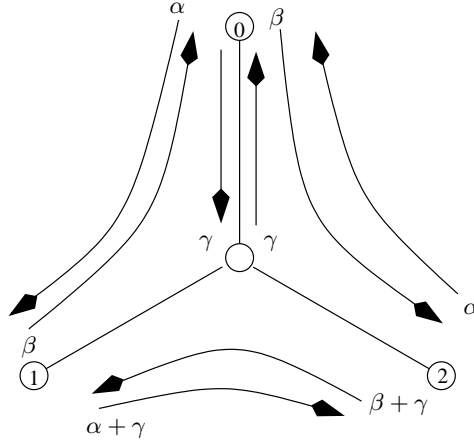


Figure 3: The generic reduced 3-star

Lemma 10 *The reduced 3–star can be described with 3 parameters α, β and γ with $\alpha + \beta + \gamma = l$. Furthermore, the reduction can only increase the cost of the fractional coloring.*

Proof: Algorithm 4 first adds paths of length 1 in order to equalize the load. This step can only increase the cost of the coloring. Once the load is uniform there is a flow conservation law at the central vertex of the star since each edge supports the same amount of paths in both directions.

The **repeat-until** loop terminates in at most $3l$ steps since there are at most $3l$ paths arriving at the central vertex and one is processed at each step. The loop terminates when there are no more paths to concatenate, which means that all remaining paths of length 1 are found on only one link, say the link between n_0 and n_c without loss of generality. Moreover, the loop maintains the flow conservation since each step concatenates one path arriving at the central vertex with one leaving it. Therefore, it is obvious that there are exactly the same amount, γ , of paths of length 1 in each direction (from n_0 to n_c and from n_c to n_0).

The resulting set of paths is then a convex combination of 3 kinds of independent sets: the *counter-clockwise* ($(n_0 \rightarrow n_1 \rightarrow n_2)$, α of them) and *clockwise* ($(n_0 \rightarrow n_2 \rightarrow n_1)$, β of them) circuits, and the set with 2 paths of length 1, $(n_0 \rightarrow n_c)$ and $(n_c \rightarrow n_0)$, and 2 paths of lengths 2, $(n_1 \rightarrow n_2)$ and $(n_2 \rightarrow n_1)$ (γ of them), as depicted in Figure 3. Therefore, the 3 parameters α, β , and γ are enough to describe the resulting set.

The cost of the coloring of the resulting set is greater or equal to the original one since the concatenation cannot decrease it. \square

We can identify the parameters of the reduced star with the one of the generic star, as follows.

$$a_0 = a_2 = \alpha, b_0 = b_1 = \beta, d_0 = c_0 = \gamma, a_1 = \alpha + \gamma, b_1 = \beta + \gamma.$$

In the following, a pair of paths crossing an edge on opposite direction is called a *doublon*. We say that a path *touches* a vertex v if it starts, ends, or goes through v .

Definition 4 *At a vertex n_i , a doublon is a couple of non-void paths touching n_i in opposite directions. A half-doublon is a non-void path touching n_i associated with the void path \emptyset in the opposite direction.*

The definition of a w -balanced fractional coloring requires that on each link the trace variable for any pair of paths using the link oppositely equals $\frac{w}{l}$. In other words, any doublon has to be covered with weight $\frac{w}{l}$. Consequently, the half-doublon have to be covered with weight $1 - w$.

The actual list of the doublons and half-doublons of a reduced 3-star is described in Appendix A. Based on this list, Appendix B describe the numbers of independent sets of paths that are used in a balanced coloring of such a reduced 3-star. The cost of such a coloring implies the result stated as proposition 9 : *For any set of paths \mathcal{P} of load $l(T, \mathcal{P})$ on a binary tree T , there exists a fractional coloring of cost at most $\frac{7l(T, \mathcal{P})}{5}$.*

4.3.2 Proof of proposition 9

The proof consists in showing that any set of paths on any binary tree admits a balanced coloring with cost at most $\frac{7l}{5}$.

A detailed construction of such a coloring for paths on a reduced 3-star is given in Appendix B. Corollary 8 and Lemma 10 complete the proof.

□

Note that this result seems highly related to the randomized algorithm in [1], which colors paths in binary trees with $\frac{7l}{5} + o(l)$ colors, but at the time of this writing no formal equivalence has been proven.

5 Integral path coloring in trees

In this section we present an important application of our methods to the path coloring problem in trees.

Note that the result of [9] states that there exists a polynomial time algorithm which colors any set of paths \mathcal{P} of load $l(T, \mathcal{P})$ on a tree T with at most $5l(T, \mathcal{P})/3$ colors. Since the load $l(T, \mathcal{P})$ is a lower bound for the optimal number of colors, this gives a $5/3$ -approximation algorithm. In the following we exploit the optimal solution for the fractional path coloring which can be obtained in polynomial time for bounded degree trees to design a randomized algorithm with better approximation ratio.

Given a solution \bar{x} of the fractional path coloring of the set of paths \mathcal{P} on a tree T , the idea is to perform a randomized rounding to \bar{x} and obtain an integral solution x . After rounding, x may not be a feasible solution to the path coloring problem since some of the constraints of the form $\sum_{I \in \mathcal{I}: p \in I} x(I) \geq 1$ may be violated. However, this is a feasible solution to the path coloring problem on the set of paths $\mathcal{P}' \subseteq \mathcal{P}$, defined as the set of paths contained in independent sets I such that $x(I) = 1$. This means that we have properly colored some of the paths of \mathcal{P} with $\chi_f(\mathcal{T}, \mathcal{P})$ colors.

After this rounding procedure, a set of paths $\mathcal{P} \setminus \mathcal{P}'$ remains to be colored. The analysis presented in [23] for a similar procedure in circular arc graphs holds. The procedure is a *balls and bins*-like algorithm that defines a martingale sequence for the set of colored paths. Following Azuma's inequality [2], we therefore obtain that if $l(\mathcal{T}, \mathcal{P}) = \Omega(\log n)$, where n is the number of vertices in T , then after the rounding procedure the load of paths in $\mathcal{P} \setminus \mathcal{P}'$ is

$$l(\mathcal{T}, \mathcal{P} \setminus \mathcal{P}') \leq \frac{l(\mathcal{T}, \mathcal{P})}{e} + o(l(\mathcal{T}, \mathcal{P})),$$

with high probability. Now, we can apply the algorithm of [9] to color the paths in $\mathcal{P} \setminus \mathcal{P}'$ with $\frac{5l(\mathcal{T}, \mathcal{P})}{3e} + o(l(\mathcal{T}, \mathcal{P}))$ additional colors. In total, we use at most

$$\chi_f(\mathcal{T}, \mathcal{P}) + \frac{5l(\mathcal{T}, \mathcal{P})}{3e} + o(l(\mathcal{T}, \mathcal{P}))$$

colors. Since $l(\mathcal{T}, \mathcal{P}) \leq \chi_f(\mathcal{T}, \mathcal{P}) \leq \chi(\mathcal{T}, \mathcal{P})$, we obtain the following results.

Proposition 11 *There exists a randomized $1.61 + o(1)$ -approximation algorithm to the path coloring problem in bounded-degree trees.*

Corollary 12 *For any set of paths \mathcal{P} on a tree T , it holds:*

$$\chi(\mathcal{T}, \mathcal{P}) \leq \chi_f(\mathcal{T}, \mathcal{P}) + \frac{5l(\mathcal{T}, \mathcal{P})}{3e} + o(l(\mathcal{T}, \mathcal{P})).$$

Note that using a much more sophisticated parameterized randomized rounding, a $(1.511 + o(1))$ -approximation randomized algorithm has been proposed in [5].

We can also apply similar arguments to bounded degree trees of rings to obtain a 2.22-approximation algorithm. For that, an optimal fractional coloring is computed and randomly rounded as above. The remaining set of paths is colored as if it was a set of paths on the tree obtained by contracting each ring into an bi-directed edge. This transformation obviously doubles the load of the set of path and over-constrained the coloring. Following the same proof as above, such a coloring is bounded by $\chi_f + \frac{5.2l}{3e} + o(l)$.

6 Further application and research

Our research in this paper was motivated by questions related to the design of wavelength division multiplexing optical networks. Many applications of our techniques to WDM networks can be foreseen, like in branch and bound methods, or in the design of multifiber networks.

A recent paper exploits similar techniques in the more general settings of the approximate strong separation problem [20]. In this paper, applications to task and call scheduling issues are discussed. Since then, more sophisticated rounding techniques have improved the approximation ratios of integral coloring [5], but the main ideas are the same.

The approximation results presented in this paper, as well as the best known in the literature, show a gap between the cost of integral and fractional path coloring in trees while there are no tight negative results.

We conjecture that this gap is small and that a $1 + o(1)$ randomized approximation may even exist. Indeed, randomized rounding strategies like those developed in [1] or [5] but based on an optimal fractional coloring might be quite efficient. The main challenge is then to characterize the distribution defined by the traces of optimal fractional colorings, which is still open.

References

- [1] V. Auletta, I. Caragiannis, C. Kaklamanis, and P. Persiano, “Randomized path coloring on binary trees,” in *APPROX’00*, Lecture Notes in Computer Science, Ed. 2000, vol. 1913, pp. 60–71, Springer-Verlag.
- [2] K. Azuma. Weighted sum of certain dependent random variables. *Tohoku Math. Journal*, 19:357–367, 1967.
- [3] J-C. Bermond, L. Gargano, S. Pérennes, A. A. Rescigno, and U. Vaccaro, “Efficient collective communication in optical networks,” in *ICALP’96*. July 1996, vol. 1099 of *Lecture Notes in Computer Science*, pp. 574–585, Springer-Verlag.
- [4] J.A. Bondy, and U.S.R. Murty, *Graph Theory*, Springer *Graduate Texts in Mathematics* series, vol. 244, 2008.
- [5] I. Caragiannis, and C. Kaklamanis. Approximate Path Coloring with Applications to Wavelength Assignment in WDM Optical Networks. In *Proceedings of the 21st International Symposium on Theoretical Aspects of Computer Science (STACS ’04)*, LNCS 2996, Springer, pages 258–269, 2004.
- [6] I. Caragiannis, C. Kaklamanis, and P. Persiano. Approximation Algorithms for Path Coloring in Trees. Chapter in *Efficient Approximation and Online Algorithms*, LNCS 3484, Springer, pp. 74-96, 2006.
- [7] I. Caragiannis, C. Kaklamanis, P. Persiano, and A. Sidiropoulos. *Fractional and Integral Coloring of Locally-Symmetric Sets of Paths on Binary Trees*, In Proceedings of the 1st Workshop on Approximation and On-line Algorithms (WAOA ’03), LNCS 2909, Springer, pp. 81-94, 2003.
- [8] I. Chlamtac, A. Ganz, and G. Karmi, “Lightpath communications: An approach to high bandwidth optical WAN’s,” *IEEE Transactions on Communications*, vol. 40, no. 7, pp. 1171–1182, July 1992.
- [9] T. Erlebach, K. Jansen, C. Kaklamanis, M. Mihail, and P. Persiano, “Optimal wavelength routing on directed fiber trees,” *Theoretical Computer Science*, vol. 221, no. 1–2, pp. 119–137, 1999.
- [10] U. Feige, and J. Kilian, “Zero Knowledge and the Chromatic Number,” *Journal on Computer Systems Science*. vol. 57, no. 2, pp. 187–199, 1998.
- [11] A. Ferreira, S. Pérennes, A. Richa, H. Rivano, and N. Stier, “On the design of multifiber WDM networks,” in *AlgoTel’02*, Mèze, France, May 2002, pp. 25–32.
- [12] J. Flum, and M. Grohe, *Parameterized Complexity Theory*, Springer, 2006.

- [13] N. Garg, *Multicommodity Flows and Approximation Algorithms*, Ph.D. thesis, Indian Institute of Technology, Delhi, April 1994.
- [14] N. Garg, V. V. Vazirani, and M. Yannakakis, “Primal-Dual Approximation Algorithms for Integral Flow and Multicut in Trees”, *Algorithmica*, vol 18, no 1, pp 3–20, 1997.
- [15] L. Gargano, P. Hell, and S. Pérennes, “Colouring paths in directed symmetric trees with applications to WDM routing,” in *ICALP’97*, vol. 1256 of *Lecture Notes in Computer Science*, pp. 505–515. Springer-Verlag, Bologna, Italy, July 1997.
- [16] M.C. Golumbic and R.E. Jamison, “The edge intersection graphs of paths in a tree,” *Journal of Comb. Theory*, vol. 38, pp. 8–22, 1985.
- [17] M. Grötschel, L. Lovász, and A. Schrijver, “The ellipsoid method and its consequences in combinatorial optimization,” *Combinatorica*, vol. 1, pp. 169–197, 1981.
- [18] M. Grötschel, L. Lovász, and A. Schrijver, *Geometric Algorithms and Combinatorial Optimization*, vol. 2, Springer-Verlag, second corrected edition, 1993.
- [19] D. S. Hochbaum, Ed., *Approximation Algorithms for NP-Hard Problems*, PWS Publishing Company, 1997.
- [20] Klaus Jansen. Approximate strong separation with application in fractional graph coloring and preemptive scheduling. In *Proceedings of the 19th International Symposium on Theoretical Aspects of Computer Science (STACS ’02)*, LNCS 2285, Springer, pages 100–111, 2002.
- [21] I. A. Karapetyan, “On coloring of arc graphs,” *Doklady Akad. Nauk Armianskoi CCP*, vol. 70, no. 5, pp. 306–311, 1980, In Russian.
- [22] D. König, *Graphok és matrixok*, pp. 116–119, Mat. Fiz. Lapok, 1931.
- [23] V. Kumar, “Approximating arc circular colouring and bandwidth allocation in all-optical ring networks,” in *APPROX’98*, 1998.
- [24] L. Lovász, “On the ratio of optimal integral and fractional covers,” *Discrete Mathematics*, vol. 13, pp. 383–390, 1975.
- [25] T. Niessen and J. Kind, “The round-up property of the fractional chromatic number for proper circular arc graphs,” *Journal of Graph Theory*, 1998.
- [26] R. E. Tarjan, “Decomposition by clique separators,” *Discrete Mathematics*, vol. 55, no. 2, pp. 221–232, 1985.
- [27] A. Tucker, “Coloring a family of circular arcs,” *SIAM Journal of Applied Mathematics*, vol. 29, no. 3, pp. 493–502, Nov. 1975.

A Description of the doublons of a reduced 3–star

This appendix is dedicated to the description and counting of all the doublons and half–doublons that are to be covered by a balanced coloring of reduced 3–star.

We define the *type of (half–)doublon* depending on the paths composing it and give the number present on each vertex.

- A doublon of type:
 1. A is 2 paths in a counter–clockwise circuit; $n_0 : \alpha^2$, $n_{1,2} : \alpha(\alpha + \gamma)$
 2. B is 2 paths in a clockwise circuit; $n_0 : \beta^2$, $n_{1,2} : \beta(\beta + \gamma)$
 3. C is two paths of length 1; $n_0 : \gamma^2$
 4. D_i is two parallel paths of length 2 not touching vertex n_i ;

D_0	$n_{1,2} : (\alpha + \gamma)(\beta + \gamma)$
D_1	$n_{0,2} : \alpha\beta$
D_2	$n_{0,2} : \alpha\beta$
 5. E_i , $i = 1, 2$, is a path of length 1 and a path touching vertex n_i in a counter–clockwise circuit; $n_0 : \alpha\gamma$ each
 6. F_i , $i = 1, 2$, is a path of length 1 and a path touching vertex n_i in a clockwise circuit; $n_0 : \beta\gamma$ each.
- A half–doublon of type:
 1. A_i^\emptyset is a path not touching vertex n_i in a counter–clockwise circuit;

A_0^\emptyset	$n_{1,2} : \alpha + \gamma$
A_1^\emptyset	$n_{0,2} : \alpha$
A_2^\emptyset	$n_{0,1} : \alpha$
 2. B_i^\emptyset is a path not touching vertex n_i in a clockwise circuit ;

B_0^\emptyset	$n_{1,2} : \beta + \gamma$
B_1^\emptyset	$n_{0,2} : \beta$
B_2^\emptyset	$n_{0,1} : \beta$
 3. C_+^\emptyset is a path of length 1 from n_0 to n_c ; $n_0 : \gamma$
 4. C_-^\emptyset is a path of length 1 from n_c to n_0 . $n_0 : \gamma$.

B Balanced coloring of a reduced 3–star

This Appendix is dedicated to the main step in the proof of proposition 9. We show that any reduced 3–star admits a path coloring with cost at most $\frac{7l}{5}$, where $l = \alpha + \beta + \gamma$ is the load of the set of paths. In the following we build a w –balanced coloring where $w = \frac{4}{5}$ and compute its cost.

B.1 Description of the independent sets

This coloring uses independent sets in order to cover the doublons defined and enumerated in Appendix A with weight $\frac{w}{l}$ ($1 - w$ for half–doublons). In particular, one can note that building a w –balanced coloring requires the use of some non–maximal sets.

The shape of the independent sets that we use is closely related to the characterization of the doublons they cover. Hence, their names are used interchangeably. The actual weight that has to be put on each shape of independent set is described in the following tables, as well as the type of (half–)doublons that they cover at each vertex of the star. One can check that, given the choice of $w = \frac{4}{5}$, all the doublons and half-doublons described in Appendix A are covered with the relevant weight.

The cost of the coloring is then the sum of the weights of all the independent sets in these tables.










For the sake of readability, let us define

$$G_A = \min \left\{ (1 - w)\alpha, (1 - w)(\alpha + \gamma) - \alpha\gamma\frac{w}{l} \right\}$$

and

$$G_B = \min \left\{ (1 - w)\beta, (1 - w)(\beta + \gamma) - \beta\gamma\frac{w}{l} \right\}$$

1) Maximal independent sets

IS	n_0	n_1	n_2	Quantity	IS	n_0	n_1	n_2	Quantity
	A			$\alpha^2\frac{w}{l} - G_A$		B			$\beta^2\frac{w}{l} - G_B$
	D_1		D_1	$\alpha\beta\frac{w}{l}$		D_2			$\alpha\beta\frac{w}{l}$
	E_1	A	A_0^\emptyset	$\alpha\gamma\frac{w}{l}$		E_2	A_0^\emptyset	A	$\alpha\gamma\frac{w}{l}$
	F_1	B_0^\emptyset	B	$\beta\gamma\frac{w}{l}$		F_2	B	B_0^\emptyset	$\beta\gamma\frac{w}{l}$
	C	D_0		$\gamma^2\frac{w}{l}$					

2) Maximal independent sets obtained by combining these shapes

IS	n_0	n_1	n_2	Quantity	
				A_0^\emptyset	$(1-w)(\alpha+\gamma) - \alpha\gamma\frac{w}{l} - G_A$
				B	$(1-w)(\beta+\gamma) - \beta\gamma\frac{w}{l} - G_B$
				D_0	$(\alpha\gamma + \alpha\beta + \beta\gamma)\frac{w}{l}$

IS	n_0	n_1	n_2	Quantity	
				C_+^\emptyset	$(1-w)\gamma$
				C_-^\emptyset	$(1-w)\gamma$

3) Non-maximal independent sets

IS	n_0	n_1	n_2	Quantity			
				A_i^\emptyset	A	G_A	
				A_i^\emptyset	A	A_i^\emptyset	G_A
				A	A_i^\emptyset	G_A	
				A_1^\emptyset		A_1^\emptyset	$(1-w)\alpha - G_A$
				A_2^\emptyset			$(1-w)\alpha - G_A$

IS	n_0	n_1	n_2	Quantity			
				B_i^\emptyset	B	G_B	
				B_i^\emptyset	B	B_i^\emptyset	G_B
				B	B_i^\emptyset	G_B	
				B_1^\emptyset		B_1^\emptyset	$(1-w)\beta - G_B$
				B_2^\emptyset			$(1-w)\beta - G_B$

B.2 Setting w

The independent sets covering E_i and F_i doublons impose the two following constraints on w .

$$\alpha\gamma\frac{w}{l} \leq (1-w)(\alpha+\gamma) \quad \text{and} \quad \beta\gamma\frac{w}{l} \leq (1-w)(\beta+\gamma)$$

These constraints and $\alpha+\gamma=l-\beta$ imply $w\left(1+\frac{\alpha\gamma}{4l(l-\beta)}\right) \leq 1$.

Moreover, $\alpha\beta \leq \frac{(l-\beta)^2}{4}$, which gives $w\left(1+\frac{l-\beta}{4l}\right) \leq w\left(1+\frac{1}{4}\right) \leq 1$.

Therefore, setting w to $\frac{4}{5}$ is always a valid solution.

B.3 Cost of a $\frac{4}{5}$ -balanced coloring

Using the quantities of independent sets given in the tables above (uniformly divided between all possible instantiation of each independent set), all the doublons and half-doublons are covered

with the relevant weight. This gives a complete w -balanced coloring of the star. The cost of this coloring is the total quantity of independent sets that have been used.

- The cost of the maximal independent sets of table 1) is

$$\begin{aligned} Cost_1 &= \frac{w}{l} (\alpha^2 + \beta^2 + \gamma^2 + 2\alpha\beta + 2\alpha\gamma + 2\beta\gamma) - G_A - G_B \\ &= \frac{w}{l} l^2 - G_A - G_B \end{aligned}$$

- The cost of the combination of independent sets in table 2) is

$$\begin{aligned} Cost_3 &= \max \left\{ 2(1-w)\gamma, \right. \\ &\quad \left. \frac{w}{l}(\alpha\gamma + \alpha\beta + \beta\gamma) + (1-w)(\alpha + \gamma) - \alpha\gamma\frac{w}{l} - G_A + (1-w)(\beta + \gamma) - \beta\gamma\frac{w}{l} - G_B \right\} \\ &= \max \left\{ 2(1-w)\gamma, \frac{w}{l}\alpha\beta + (1-w)(\alpha + \beta + 2\gamma) - G_A - G_B \right\} \end{aligned}$$

- The cost of the non-maximal independent sets of table 3) is

$$\begin{aligned} Cost_2 &= 3G_A + 2((1-w)\alpha - G_A) + 3G_B + 2((1-w)\beta - G_B) \\ &= 2(1-w)(l - \gamma) + G_A + G_B \end{aligned}$$

- Therefore, using $\max\{a, a+b\} = a + \max\{0, b\}$ with $a = 2(1-w)\gamma$, the total cost is

$$\begin{aligned} Cost &= Cost_1 + Cost_2 + Cost_3 \\ &= \frac{w}{l}l^2 - G_A - G_B + 2(1-w)(l - \gamma) + G_A + G_B + 2(1-w)\gamma \\ &\quad + \max \left\{ 0, \frac{w}{l}\alpha\beta + (1-w)(\alpha + \beta) - G_A - G_B \right\} \\ &= wl + 2(1-w)l + \max \left\{ 0, \frac{w}{l}\alpha\beta + (1-w)(\alpha + \beta) \right. \\ &\quad \left. - \min \left\{ (1-w)\alpha, (1-w)(\alpha + \gamma) - \alpha\gamma\frac{w}{l} \right\} \right. \\ &\quad \left. - \min \left\{ (1-w)\beta, (1-w)(\beta + \gamma) - \beta\gamma\frac{w}{l} \right\} \right\} \\ &= (2-w)l + \max \left\{ 0, \right. \\ &\quad \left. \frac{w}{l}\alpha\beta - \min \left\{ 0, (1-w)\gamma - \alpha\gamma\frac{w}{l} \right\} - \min \left\{ 0, (1-w)\gamma - \beta\gamma\frac{w}{l} \right\} \right\} \\ &= (2-w)l + \max \left\{ 0, \right. \\ &\quad \left. \frac{w}{l}\alpha\beta + \max \left\{ 0, \alpha\gamma\frac{w}{l} - (1-w)\gamma \right\} + \max \left\{ 0, \beta\gamma\frac{w}{l} - (1-w)\gamma \right\} \right\} \end{aligned}$$

- Using $-\min\{0, a\} = +\max\{0, -a\}$ we get

$$\begin{aligned} Cost &= (2-w)l + \frac{w}{l}\alpha\beta + \max \left\{ 0, \alpha\gamma\frac{w}{l} - (1-w)\gamma \right\} + \max \left\{ 0, \beta\gamma\frac{w}{l} - (1-w)\gamma \right\} \end{aligned}$$

The cost of the coloring depends on the values of the two max. Three cases have to be considered: none, one or both max is null.

1. If $\alpha\gamma\frac{w}{l} - (1-w)\gamma > 0$ and $\beta\gamma\frac{w}{l} - (1-w)\gamma > 0$, using $ab \leq \frac{(a+b)^2}{4}$, we get

$$\begin{aligned}
& \text{Cost} \\
&= (2-w)l + \frac{w}{l}\alpha\beta + \alpha\gamma\frac{w}{l} - (1-w)\gamma + \beta\gamma\frac{w}{l} - (1-w)\gamma \\
&\leq (2-w)l + \frac{w(\alpha+\beta)^2}{4l} + \gamma(\alpha+\beta)\frac{w}{l} - 2(1-w)\gamma \\
&\leq (2-w)l + \frac{w(l-\gamma)^2}{4l} + \gamma(l-\gamma)\frac{w}{l} - 2(1-w)\gamma \\
&\leq (2-w)l + \frac{w}{4l}(l^2 - 2l\gamma + \gamma^2) + w\gamma - \gamma^2\frac{w}{l} - 2\gamma + w\gamma \\
&\leq (2-w)l + \frac{wl}{4} - \frac{w\gamma}{2} + \frac{w\gamma^2}{4l} - \frac{w\gamma^2}{l} - 2\gamma + 2w\gamma \\
&\leq (2 - \frac{w}{4})l - \gamma(2 - \frac{5w}{2} + \frac{3w\gamma}{4l})
\end{aligned}$$

As we choose $w = \frac{4}{5}$, we obtain

$$\text{Cost} \leq \frac{7}{5}l - \frac{3w\gamma^2}{4l} \leq \frac{7}{5}l$$

2. If $\alpha\gamma\frac{w}{l} - (1-w)\gamma > 0$ and $\beta\gamma\frac{w}{l} - (1-w)\gamma < 0$, we get

$$\begin{aligned}
& \text{Cost} \\
&= (2-w)l + \frac{w}{l}\alpha\beta + \alpha\gamma\frac{w}{l} - (1-w)\gamma \\
&= (2-w)l + \frac{w}{l}\alpha(\beta+\gamma) - \gamma(1-w) \\
&\leq (2-w)l + \frac{w}{l}\alpha(l-\alpha) \leq (2-w)l + \frac{wl}{4} \\
&\leq \frac{7}{5}l
\end{aligned}$$

3. If $\alpha\gamma\frac{w}{l} - (1-w)\gamma < 0$ and $\beta\gamma\frac{w}{l} - (1-w)\gamma < 0$, we get

$$\begin{aligned}
& \text{Cost} \\
&= (2-w)l + \frac{w}{l}\alpha\beta \\
&\leq (2-w)l + \frac{w(l-\gamma)^2}{4l} \\
&\leq (2-w)l + \frac{wl}{4} \\
&\leq \frac{7}{5}l
\end{aligned}$$

(1), (2) and (3) put together show that any reduced 3-star admits a coloring of cost upper-bounded by $\frac{7l}{5}$. This coloring is a $\frac{4}{5}$ -balanced coloring.