Alcom-FT Technical Report Series
ALCOMFT-TR-03-167

Fractional and Integral Coloring of Locally-Symmetric Sets of Paths on Binary Trees *

Ioannis Caragiannis

Computer Technology Institute and
Dept. of Computer Engineering and Informatics,
University of Patras, 26500 Rio, Greece

Christos Kaklamanis

Computer Technology Institute and
Dept. of Computer Engineering and Informatics,
University of Patras, 26500 Rio, Greece

Pino Persiano

Dipartimento di Informatica ed Applicazioni,
Università di Salerno,
84081 Baronissi, Italy

Anastasios Sidiropoulos

Computer Science and Artificial Intelligence Laboratory,
Massachusetts Institute of Technology
200 Technology Square,
Cambridge, MA 02139 USA

October 2, 2003

Abstract

Motivated by the problem of allocating optical bandwidth in tree–shaped WDM networks, we study the fractional path coloring problem in trees. We consider the class of locally-symmetric sets of paths on binary trees and prove that any such set of paths has a fractional coloring of cost at most $1.367L$, where L denotes the load of the set of paths. Using this result, we obtain a randomized algorithm that colors any locally-symmetric set of paths of load L on a binary tree (with reasonable restrictions on its depth) using at most $1.367L + o(L)$ colors, with high probability.

1 Introduction

Let $T(V, E)$ be a bidirected tree, i.e., a tree with each edge consisting of two opposite directed edges. Let $\mathcal{P}$ be a set of directed paths on T. The *path coloring problem* (or integral path

*This work was partially funded by the European Union under IST-1999-14186 FET Project ALCOM–FT, IST-2001-33135 FET Project CRESCCO and RTN Project ARACNE.

coloring problem) is to assign colors to paths in $\mathcal{P}$ so that no two paths that share a directed edge of T are assigned the same color and the total number of colors used is minimized. The problem has applications to WDM (Wavelength Division Multiplexing) routing in tree–shaped all–optical networks. In such networks, communication requests are considered as ordered transmitter–receiver pairs of network nodes. WDM technology establishes communication by finding transmitter–receiver paths and assigning a wavelength to each path, so that no two paths going through the same fiber are assigned the same wavelength. Since state–of–the–art technology [11] allows for a limited number of wavelengths, the important engineering question to be solved is to establish communication so that the total number of wavelengths used is minimized.

The path coloring problem in trees has been proved to be NP–hard in [6], thus the work on the topic mainly focuses on the design and analysis of approximation algorithms. Known results are expressed in terms of the load L of $\mathcal{P}$, i.e., the maximum number of paths that share a directed edge of T. Note that, for any set of paths, its load is a lower bound on the optimal number of colors. An algorithm that assigns at most $2L$ colors to any set of paths is implicit in the work of Raghavan and Upfal [10]. The best known upper bound for arbitrary trees is $5L/3$ [7]. A randomized $(1.613 + o(1))$–approximation algorithm for trees of bounded degree is presented in [2]. This bound has been recently improved to $1.511 + o(1)$ in [3].

The path coloring problem is still NP–hard when the input instances are restricted to sets of paths on binary trees [6]. A randomized path coloring algorithm for binary trees is presented in [1]. This algorithm colors any set of paths of load L on a binary tree (with some restrictions on its depth) using at most $7L/5 + o(L)$ colors. In [9], it is proved that there are sets of paths on binary trees whose optimal path colorings require at least $5L/4$ colors.

The path coloring problem is still NP–hard when the input instances are restricted to symmetric sets of paths on binary trees [4]. A set of paths $\mathcal{P}$ is called symmetric if it can be partitioned into disjoint pairs of symmetric paths, i.e., (u, v) and (v, u). Symmetric sets of paths are important since many services which are supported by WDM networks (or are expected to be supported in the future) require bidirectional reservation of bandwidth. Algorithms that color any symmetric set of paths of load L on a tree (not necessarily binary) with $3L/2$ colors can be obtained by considering each pair of symmetric paths as an undirected one and then applying an algorithm for coloring undirected paths on undirected trees [6, 10]. For binary trees, the randomized algorithm presented in [1] gives the best known upper bound in this case. In [4] it is also proved that there are symmetric sets of paths on binary trees whose optimal path colorings require at least $5L/4$ colors.

The *fractional path coloring problem* is a natural relaxation of path coloring. Given a set of paths on a tree, a solution of the path coloring problem is also a solution for the fractional path coloring problem with cost equal to the number of colors in the solution of path coloring. Furthermore, the lower bound proofs in [4] and [9] still hold for fractional path coloring. Finding a fractional path coloring of optimal cost can be solved in polynomial time in trees of bounded degree [2]. Fractional path colorings are important since they can be used to obtain path colorings using a number of colors which is provably close to the cost of the fractional solution by applying randomized rounding techniques [2, 3, 8]. Moreover, it is interesting to prove upper bounds on the cost of optimal fractional path colorings in terms of the load, since they give insight to the path coloring problem. In [2], the result of [1] was extended to prove that any set of paths of load L on a binary tree has a fractional path coloring of cost $7L/5$.

In this paper, we consider locally-symmetric sets of paths on binary trees. A set of paths

$\mathcal{P}$ on a tree is called locally-symmetric if, for any two nodes u and v of the tree with distance at most 2, the number of paths coming from v and going to u equals the number of paths coming from u and going to v. Clearly, the class of locally-symmetric sets of paths on a tree T contain the class of symmetric sets of paths on T. We prove that any locally-symmetric set of paths of load L on a binary tree has a fractional path coloring of cost at most $1.367L$. This fractional path coloring has some additional nice properties, and, using this result and techniques of [1], we obtain a randomized algorithm that colors any locally-symmetric set of paths of load L on a binary tree (with some restrictions on its depth) using $1.367L + o(L)$ colors, with high probability. Since the load of a set of paths is a lower bound on the minimum number of colors sufficient for coloring it, our algorithm is an $(1.367 + o(1))$-approximation algorithm.

We remark that an $(1.336 + o(1))$-approximation algorithm has been recently obtained in [3] using different techniques. This algorithm applies to sets of paths (not necessarily locally-symmetric) on binary trees with similar restrictions on their depth. However, the two results are different. The result in [3] is based on randomized rounding of the optimal fractional coloring solution and the analysis actually upper-bounds the number of colors by $1.336 + o(1)$ times the cost of the optimal fractional solution which is also a lower bound for the optimal number of colors. This may not give a good upper bound on the number of colors in terms of the load since the cost of the optimal fractional path coloring can be much higher than the load.

The rest of the paper is structured as follows. In Section 2 we give formal definitions for the (fractional) path coloring problem. In Section 3 we discuss some properties of locally-symmetric sets of paths which are useful in the analysis of our algorithms. We describe the fractional path coloring algorithm in Section 4 and present its analysis in Section 5. We discuss the extension of the result for fractional path coloring to path coloring in Section 6.

2 Fractional path colorings

The graph coloring problem can be considered as finding an integral covering of the vertices of a graph by independent sets of unit weight so that the total weight (or cost) is minimized. Given a graph $G = (V, E)$, this means solving the following integer linear program:

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

where $\mathcal{I}$ denotes the set of the independent sets of G.

This formulation has a natural relaxation into the following linear program:

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

The corresponding combinatorial problem is called the *fractional coloring problem.* If $\bar{x}$ is a valid weight assignment over the independent sets of the graph G, we call it a *fractional coloring* of G.

Given a set of paths $\mathcal{P}$ on a graph G, we define an *independent set of paths* as a set of pairwise edge–disjoint paths. Equivalently, we may think of the conflict graph of $\mathcal{P}$ which is the graph having a node for each path of $\mathcal{P}$ and edges between any two nodes corresponding to conflicting (i.e., not edge–disjoint) paths; an independent set of paths corresponds to an independent set on the conflict graph. The fractional path coloring problem is defined as the fractional coloring problem on the conflict graph.

In [2] it is proved that optimal fractional path colorings in bounded–degree trees can be computed by solving a linear program of polynomial (in terms of the load of the set of paths and the size of the tree) size. These techniques can be extended to graphs of bounded degree and bounded treewidth. A polynomial time algorithm for computing optimal fractional path colorings in rings is implicit in [8].

Although computing an optimal fractional path coloring of a locally-symmetric set of paths $\mathcal{P}$ on a binary tree can be done in polynomial time, in this paper we are interested in exploring the relation of the cost of the optimal fractional path coloring with the load of $\mathcal{P}$.

3 Properties of locally-symmetric sets of paths

In this section we give some useful properties of locally-symmetric sets of paths on binary trees. Let T be a binary tree. Without loss of generality, we assume that all nodes of T have degree either 1 or 3. We denote by r (root) a leaf node of T. Starting from r, we assign labels to the nodes of the tree by performing a breadth-first-search (r is assigned 0). For each non-leaf node v, we will denote by $p(v)$ the parent of v, and by $l(v)$ and $r(v)$ the left and right child of v, respectively.

Let $\mathcal{P}$ be a locally-symmetric set of paths. We may assume that $\mathcal{P}$ is *normal*, i.e., it satisfies the following properties:

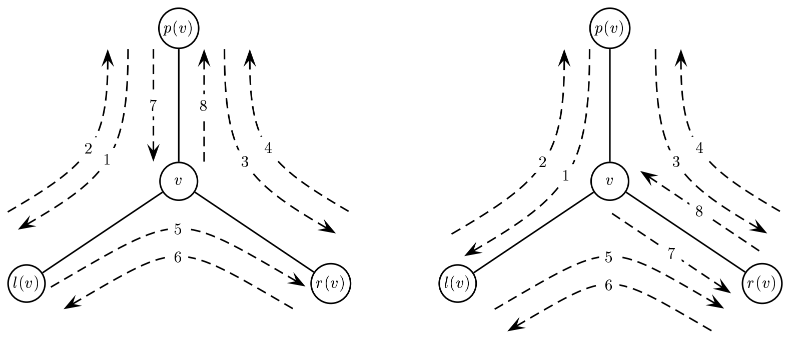
- It has full load L at each directed edge.
- For every node u, the paths that originate or terminate at u, appear on only one of the three edges adjacent to u.

If the initial set of paths is not normal, we can transform it to a normal one by first adding pairs of symmetric single–hop paths on the edges of the tree which are not fully–loaded, and then, for each non-leaf node v, by merging paths terminating at v with paths originating at v that do not traverse the same edge in opposite directions. It can be easily verified that this can be done in such a way that the resulting set of paths is locally-symmetric.

Claim 1 *Let $\mathcal{P}$ be a locally-symmetric set of paths on a binary tree T with load L. There exists a normal locally-symmetric set of paths $\mathcal{P}'$ such that if $\mathcal{P}'$ has a (fractional) coloring of cost c, then $\mathcal{P}$ has a (fractional) coloring of cost at most c.*

We partition the set of paths of $\mathcal{P}$ that go through v to the following disjoint subsets: the set M_v^1 of the paths that come from $p(v)$ and go to $l(v)$, the set M_v^2 of the paths that come from $l(v)$ and go to $p(v)$, the set M_v^3 of the paths that come from $p(v)$ and go to $r(v)$, the set M_v^4 of the paths that come from $r(v)$ and go to $p(v)$, the set M_v^5 of the paths that come from $l(v)$ and go to $r(v)$, and the set M_v^6 of the paths that come from $r(v)$ and go to $l(v)$. Since $\mathcal{P}$ is normal, we only need to consider two cases for the set P_v of paths of $\mathcal{P}$ touching node v. These cases are depicted in Figure 1.

- **Scenario I:** The paths of P_v originating from or terminating at v touch node $p(v)$. We denote by M_v^7 the set of paths that come from $p(v)$ and stop at v and by M_v^8 the set of paths that originate from v and go to $p(v)$.



(a) Scenario I (b) Scenario II

Figure 1: The two cases for paths touching a non-leaf node v. Numbers represent groups of paths (number 1 implies the set of paths M_v^1, etc.).

- **Scenario II:** The paths of P_v originating from or terminating at v touch a child node of v (wlog $r(v)$). We denote by M_v^7 the set of paths that originate from v and go to $r(v)$ and by M_v^8 the set of paths that come from $r(v)$ and stop at v.

Claim 2 *Let $\mathcal{P}$ be a normal locally-symmetric set of paths on a binary tree T with load L. Let v be a non-leaf node of T and P_v be the subset of P that contains the paths touching v.*

- *If P_v belongs to Scenario I, then $|M_v^1| = |M_v^2| = |M_v^3| = |M_v^4| \leq L/2$, $|M_v^5| = |M_v^6| = L - |M_v^1|$, and $|M_v^7| = |M_v^8| = L - 2|M_v^1|$.*

- *If P_v belongs to Scenario II, then $|M_v^1| = |M_v^2| \geq L/2$, $|M_v^3| = |M_v^4| = |M_v^5| = |M_v^6| = L - |M_v^1|$, and $|M_v^7| = |M_v^8| = 2|M_v^1| - L$.*

4 The algorithm

In this section we present our fractional path coloring algorithm. It uses a parameter $D \in [2/3, \frac{2+\sqrt{2}}{4}]$.

Let T be a binary tree and $\mathcal{P}$ a locally-symmetric set of paths on T. We denote by $\mathcal{I}$ the set of all independent sets of paths in $\mathcal{P}$. Given an edge e, we denote by $\mathcal{I}_e^S$ the set of independent sets of $\mathcal{I}$ that contain exactly one path traversing e in some direction and by $\mathcal{I}_e^D$ the set of independent sets of $\mathcal{I}$ that contain two paths traversing e (in opposite directions).

Definition 3 (Property 1) *A weight assignment x on the independent sets of $\mathcal{I}$ satisfies Property 1 at a node v if*

- *Property 1 is satisfied on each node u which has label smaller than the label of v, and*

- *either v is a leaf different from the root or for any path p traversing an edge e between v and a child of v in some direction, it is*

$$\sum_{I \in I_e^S : p \in I} x(I) = 1 - D.$$

Definition 4 (Property 2) *A weight assignment x on the independent sets of $\mathcal{I}$ satisfies Property 2 at a node v if*

- *Property 2 is satisfied on each node u which has label smaller than the label of v, and*
- *either v is a leaf different from the root or for any two paths p, q traversing an edge e between v and a child of v in opposite directions, it is*

$$\sum_{I \in I_e^D : p,q \in I} x(I) = \frac{D}{L}.$$

Definition 5 (Property 3) *A weight assignment x on the independent sets of $\mathcal{I}$ satisfies Property 3 at a node v if for any independent set I of $\mathcal{I}$ that contains at least one path touching neither v nor nodes of label smaller than v's, it is $x(I) = 0$.*

Definition 6 (Property 4) *A weight assignment x on the independent sets of $\mathcal{I}$ satisfies Property 4 if*

$$\sum_{I \in \mathcal{I}} x(I) \leq \frac{4D^2 - 4D + 4}{3D} L.$$

Our algorithm assigns weights x to the independent sets of $\mathcal{I}$ such that x is a valid fractional coloring of small cost.

First, the algorithm performs the following initialization step:

- For each independent set $I \in \mathcal{I}$ consisting only of one path that terminates at or originates from r, it sets $x_r(I) = 1 - D$.
- For each independent set $I \in \mathcal{I}$ consisting only of two opposite directed paths one terminating at and the other originating from r, it sets $x_r(I) = D/L$.
- For any other independent set $I \in \mathcal{I}$, it sets $x_r(I) = 0$.

Then, for $i = 1, \ldots, n-1$ (where n is the number of nodes of the tree) , the algorithm executes the procedure Fract-Color on node v with label i.

The procedure Fract-Color at a node v takes as input the set P_v of paths touching node v and a non-negative weight assignment $x_{p(v)}$ which satisfies Properties 1, 2 and 3 at node $p(v)$ and Property 4. If v is a leaf, the procedure Fract-Color sets the weight assignment x_v equal to $x_{p(v)}$ and stops. Otherwise, it computes a new non-negative weight assignment x_v which satisfies Properties 1, 2 and 3 at node v and Property 4. It also sets the weight assignments x_u for all the nodes u with label smaller than v's equal to x_v.

Finally, the algorithm outputs a weight assignment $x = x_u$ where u is the node with label $n - 1$.

Fract-Color will be precisely described in the next section. For now, we use its input-output specification to show inductively that the algorithm produces a weight assignment x which satisfies Properties 1, 2 and 3 at each node of T and Property 4. Clearly, after the initialization step, the assignment x_r is non-negative and satisfies Properties 1, 2 and 3 at node r. Also, under weight assignment x_r, there are $2L$ independent sets with weight $1 - D$,

L^2 independent sets with weight D/L while all other independent sets have zero weight. Thus, Property 4 is satisfied since

$$\sum_{I\in\mathcal{I}} x_r(I) = (2-D)L < \frac{4D^2-4D+4}{3D}L.$$

Assuming that the execution of FRACT-COLOR at a node u with label i has produced a non-negative weight assignment x_u that satisfies Properties 1, 2 and 3 at u and Property 4, we can easily show that the execution of FRACT-COLOR at a node v with label $i+1$ creates a non-negative weight assignment x_v that satisfies Properties 1, 2 and 3 at node v and Property 4. By the description of FRACT-COLOR, we just have to show that the weight assignment x_u which is part of the input of the execution of FRACT-COLOR at node v satisfies Properties 1, 2 and 3 at node $p(v)$ and Property 4. Property 4 is obviously satisfied. Since either $p(v)=u$ or $p(v)$ has a label smaller than u's, x_u satisfies Properties 1, 2 and 3 at $p(v)$.

Let p be a path traversing an edge e between some node v and a child of v in some direction. Note that p is contained only in independent sets of the disjoint sets I_e^S and I_e^D. Furthermore, x satisfies Properties 1, 2 and 3 at node v and there are L paths that traverse the edge e in opposite direction to p. We obtain that

$$\sum_{I\in\mathcal{I}:p\in I} x(I) = \sum_{I\in I_e^S:p\in I} x(I) + \sum_{I\in I_e^D:p\in I} x(I) = 1.$$

Also, x is non-negative. Thus, the algorithm specified above finds a fractional coloring x of the paths of $\mathcal{P}$ of cost at most $\frac{4D^2-4D+4}{3D}L$. In the following section, we describe the procedure FRACT-COLOR and prove that it works correctly (i.e., as specified above) provided that $D\in[2/3, \frac{2+\sqrt{2}}{4}]$. Substituting $D=\frac{2+\sqrt{2}}{4}$, we obtain the following theorem.

Theorem 7 *For any locally-symmetric set of paths $\mathcal{P}$ of load L on a binary tree T, there exists a fractional coloring of cost at most $1.367L$.*

5 The procedure FRACT-COLOR

Let T be a binary tree and $\mathcal{P}$ a normal locally-symmetric set of paths on T. We show how the procedure FRACT-COLOR works when executed at a non–leaf node v. Let P_v be the set of paths of $\mathcal{P}$ touching node v. We have to consider two cases: one for Scenario I and one for Scenario II. Due to lack of space, we present only the case of Scenario I here. The description and sketch of proof for Scenario II are given in Appendix A.

We denote by I_v the set of independent sets of $\mathcal{I}$ that contain only paths touching nodes with label smaller than v. We denote by I_v^0 the set of independent sets in I_v that do not contain paths traversing the edge between v and its parent in some direction. We denote by I_v^i for $i=1,2,3,4,7,8$, the set of independent sets of I_v which contain a path p of M_v^i and no path traversing the edge between v and its parent in opposite direction to p, and by I_v^{ij} the set of independent sets of $\mathcal{I}$ which contain a path of M_v^i and a path of M_v^j, such that paths of M_v^i traverse the edge between v and its parent in opposite direction than paths in M_v^j.

When executed at v, FRACT-COLOR takes as input the set P_v and a non-negative weight assignment $x_{p(v)}(I)$ on the independent sets of $\mathcal{I}$ that satisfies Properties 1, 2, and 3 at $p(v)$ and Property 4. FRACT-COLOR computes a non-negative weight assignment $x_v(I)$ on the independent sets of $\mathcal{I}$ that satisfies Properties 1, 2, and 3 at v and Property 4. This is done in the following way:

- For each independent set I' of I_v^1 and I_v^4 and each path p of M_v^5, we set to $x_v(I) = \frac{\alpha}{L-|M_v^1|}x_{p(v)}(I')$ the weight of the independent set consisting of the paths of I' and path p and $x_v(I') = (1-\alpha)x_{p(v)}(I')$.

- For each independent set I' of I_v^2 and I_v^3 and each path p of M_v^6, we set to $x_v(I) = \frac{\alpha}{L-|M_v^1|}x_{p(v)}(I')$ the weight of the independent set consisting of the paths of I' and path p and $x_v(I') = (1-\alpha)x_{p(v)}(I')$.

- For each independent set I' of I_v^{14} and each path p of M_v^5, we set to $x_v(I) = \frac{\beta}{L-|M_v^1|}x_{p(v)}(I')$ the weight of the independent set consisting of the paths of I' and path p and $x_v(I') = (1-\beta)x_{p(v)}(I')$.

- For each independent set I' of I_v^{23} and each path p of M_v^6, we set to $x_v(I) = \frac{\beta}{L-|M_v^1|}x_{p(v)}(I')$ the weight of the independent set consisting of the paths of I' and path p and $x_v(I') = (1-\beta)x_{p(v)}(I')$.

- For each independent set I' of I_v^{18} and I_v^{47} and each path p of M_v^5, we set to $x_v(I) = \frac{\gamma}{L-|M_v^1|}x_{p(v)}(I')$ the weight of the independent set consisting of the paths of I' and path p and $x_v(I') = (1-\gamma)x_{p(v)}(I')$.

- For each independent set I' of I_v^{27} and I_v^{38} and each path p of M_v^6, we set to $x_v(I) = \frac{\gamma}{L-|M_v^1|}x_{p(v)}(I')$ the weight of the independent set consisting of the paths of I' and path p and $x_v(I') = (1-\gamma)x_{p(v)}(I')$.

- Let
$$k_1 = \sum_{I \in I_v^0 \cup I_v^7 \cup I_v^8 \cup I_v^{78}} x_{p(v)}(I)$$
and
$$k_2 = 2n(L - |M_v^1|) + \frac{D(L-|M_v^1|)^2}{L}.$$

- For each independent set I' of $I_v^0 \cup I_v^7 \cup I_v^8 \cup I_v^{78}$ and each path p of M_v^5 and M_v^6, we set to $x_v(I) = \frac{n}{\max\{k_1,k_2\}}x_{p(v)}(I')$ the weight of the independent set consisting of the paths of I' and path p.

- For each independent set I' of $I_v^0 \cup I_v^7 \cup I_v^8 \cup I_v^{78}$ and each pair of paths p, q of M_v^5 and M_v^6 respectively, we set to $x_v(I) = \frac{D}{\max\{k_1,k_2\}L}x_{p(v)}(I')$ the weight of the independent set consisting of the paths of I' and paths p and q.

- For each independent set I of $I_v^0 \cup I_v^7 \cup I_v^8 \cup I_v^{78}$ we set $x_v(I) = \max\{0, 1 - \frac{k_2}{k_1}\}x_{p(v)}(I)$.

- For each path p of M_v^5 and M_v^6, we set to $x_v(I) = n\max\{0, 1 - \frac{k_1}{k_2}\}$ the weight of the independent set consisting only of p.

- For each pair of paths p, q of M_v^5 and M_v^6 respectively, we set to $x_v(I) = \frac{D}{L}\max\{0, 1-\frac{k_1}{k_2}\}$ the weight of the independent set consisting only of p and q.

- For all other independent sets I (i.e., independent sets that contain at least one path touching only nodes with label greater than v's) we set $x_v(I) = 0$.

Now, we have to set values to the parameters α, β, γ, n so that FRACT-COLOR is correct (i.e., it matches the input-output specification given in the previous section). The next lemma gives sufficient conditions for this purpose.

Lemma 8 (Correctness conditions) *If*

$$0 \leq \alpha, \beta, \gamma \leq 1 \tag{1}$$
$$n \geq 0 \tag{2}$$
$$\alpha L(1-D) + \beta|M_v^1|D + \gamma(L - 2|M_v^1|)D = D(L - |M_v^1|) \tag{3}$$
$$-\beta|M_v^1|^2 D + nL(L - |M_v^1|) = (L - |M_v^1|)((1-D)L - D|M_v^1|) \tag{4}$$
$$2n(L - |M_v^1|) \leq \frac{D^2 - 4D + 4}{3D} L - (4 - 2D)|M_v^1| + \frac{3D|M_v^1|^2}{L} \tag{5}$$

then FRACT-COLOR *is correct.*

Proof: It can be easily verified that conditions (1) and (2) and the fact that the weight assignment $x_{p(v)}$ is non-negative guarantee that x_v is non-negative. Also, by the definition of FRACT-COLOR, x_v satisfies Property 3.

By making calculations, it can be verified that

$$\sum_{I \in \mathcal{I}} x_v(I) = \sum_{I \in \mathcal{I}} x_{p(v)}(I) + \max\{0, k_2 - k_1\}.$$

If $k_1 \geq k_2$, x_v satisfies Property 4 since $x_{p(v)}$ satisfies Property 4. If $k_1 < k_2$, condition (5) and the fact that $x_{p(v)}$ satisfies Property 4 guarantee that x_v satisfy Property 4 as well.

Now, in order to prove Properties 1 and 2 at node v, we have to prove that for any edge e between a node u with label at most the label of v and a child of u, for any path p traversing e, it is $\sum_{I \in I_e^S : p \in I} x_v(I) = 1 - D$ and for any set of paths p, q traversing e in opposite directions, it is $\sum_{I \in I_e^D : p,q \in I} x_v(I) = D/L$. We have to distinguish between two cases for e: (i) e is an edge between a node u with label smaller than v's and a child of u and (ii) e is an edge between v and a child of v.

First, we give the proof for case (i). Given an independent set I of I_v, we denote by $J_v(I)$ the set of independent sets that contain all paths of I, (possibly) paths of M_v^5 and M_v^6, and no other paths. Since x_v satisfies Property 3 at node v, it is $x_v(I) = 0$ for any independent set I not belonging to $\bigcup_{I \in I_v} J_v(I)$. By carefully reading the procedure FRACT-COLOR, we can see that the weight of I under $x_{p(v)}$ equals the sum of weights of the independent sets in $J_v(I)$ under x_v. Also, since $x_{p(v)}$ satisfies Property 3 at $p(v)$, it is $x_{p(v)} = 0$ for all independent sets not belonging to I_v. Furthermore, $x_{p(v)}$ satisfies Properties 1 and 2 at $p(v)$. We obtain that

$$\sum_{I \in I_e^S : p \in I} x_v(I) = \sum_{I \in I_e^S \cap I_v : p \in I} \sum_{I' \in J_v(I)} x_v(I') = \sum_{I \in I_e^S \cap I_v : p \in I} x_{p(v)}(I) = \sum_{I \in I_e^S : p \in I} x_{p(v)}(I) = 1 - D$$

and

$$\sum_{I \in I_e^D : p,q \in I} x_v(I) = \sum_{I \in I_e^D \cap I_v : p,q \in I} \sum_{I' \in J_v(I)} x_v(I') = \sum_{I \in I_e^D \cap I_v : p,q \in I} x_{p(v)}(I) = \sum_{I \in I_e^D : p,q \in I} x_{p(v)}(I) = D/L.$$

Proving case (ii) is lengthy. We have to prove it for $e \in \{(v, l(v)), (v, r(v))\}$, and, in each case, we have to dinstinguish between subcases for paths p and q. We will just show that for edge $e = (v, l(v))$ for a path $p \in M_v^5$ and a path $q \in M_v^1$, it is $\sum_{I \in I_e^D : p,q \in I} x_v(I) = D/L$.

$$
\begin{aligned}
\sum_{I \in I_e^D : p,q \in I} x_v(I) &= \sum_{\substack{I \in I_v^1 : q \in I \\ I' \in J_v(I) : p \in I'}} x_v(I') + \sum_{\substack{I \in I_v^{14} : q \in I \\ I' \in J_v(I) : p \in I'}} x_v(I') + \sum_{\substack{I \in I_v^{18} : q \in I \\ I' \in J_v(I) : p \in I'}} x_v(I') \\
&= \frac{\alpha}{L - |M_v^1|} \sum_{I \in I_v^1 : q \in I} x_{p(v)}(I) + \frac{\beta}{L - |M_v^1|} \sum_{I \in I_v^{14} : q \in I} x_{p(v)}(I) + \frac{\gamma}{L - |M_v^1|} \sum_{I \in I_v^{18} : q \in I} x_{p(v)}(I) \\
&= \frac{\alpha L(1-D) + \beta D|M_v^1| + \gamma D(L - 2|M_v^1|)}{L(L - |M_v^1|)} \\
&= D/L
\end{aligned}
$$

The last equality follows by condition (3). Conditions (3) and (4) are enough to prove all possible subcases. ■

In order to prove that FRACT-COLOR is correct, we will compute values for α, β, γ, n such that the correctness conditions (1)–(5) of Lemma 8 are satisfied. We distinguish between two cases according to the size of M_v^1.

CASE I. $|M_v^1| \geq \frac{1-D}{D} L$. In this case, we have the following settings

$$
\begin{aligned}
\alpha &= 1 \\
\beta &= \frac{(L - |M_v^1|)(D|M_v^1| - (1-D)L)}{D|M_v^1|^2} \\
\gamma &= \frac{L(1-D)}{|M_v^1| D} \\
n &= 0
\end{aligned}
$$

Clearly, α satisfies condition (1) and n satisfies condition (2). By simple calculations, we obtain that (3), (4), and (5) are also satisfied. Since $|M_v^1| \geq \frac{1-D}{D} L$, it is $\beta \geq 0$. Also,

$$
\beta - 1 = \frac{(L - |M_v^1|)(D|M_v^1| - (1-D)L) - D|M_v^1|^2}{D|M_v^1|^2} = \frac{-2D|M_v^1|^2 + L|M_v^1| - (1-D)L^2}{D|M_v^1|^2}
$$

The expression in the last line is always non-positive provided that $D \in \left[2/3, \frac{2+\sqrt{2}}{4}\right]$. Thus, β satisfies condition (1). Also, since $\frac{1-D}{D} L \leq |M_v^1| \leq L/2$, γ satisfies (1).

CASE II. $|M_v^1| \leq \frac{1-D}{D} L$. In this case, we have the following settings

$$
\begin{aligned}
\alpha &= \frac{D|M_v^1|}{(1-D)L} \\
\beta &= 0 \\
\gamma &= 1 \\
n &= 1 - D - \frac{D|M_v^1|}{L}
\end{aligned}
$$

Clearly, β and γ satisfy condition (1). By making simple calculations, we obtain that (3), (4), and (5) are satisfied. Since $|M_v^1| \leq \frac{1-D}{D} L$, n satisfies condition (2). Also, since $0 \leq |M_v^1| \leq \frac{1-D}{D} L$, α satisfies condition (1).

6 Extensions to the path coloring problem

In this section we outline our path coloring algorithm. Similar ideas are used in [1] to prove a weaker result for general (i.e., non-locally-symmetric) sets of paths on binary trees. Let T be a binary tree and $\mathcal{P}$ a normal locally-symmetric set of paths on T. Our algorithm uses a rational parameter $D \in [2/3, \frac{2+\sqrt{2}}{4})$. We start with a few definitions.

Definition 9 *Let v be a node of T different from the root and $\mathcal{P}$ a normal locally-symmetric set of paths. A probability distribution $\mathcal{Q}$ over all proper colorings of paths of $\mathcal{P}$ traversing the edge e between v and its parent with $(2-D)L$ colors is weakly uniform if for any two paths $p, q \in \mathcal{P}$ that traverse e in opposite directions, the probability that p and q have the same color is D/L.*

Let v be a non-leaf node of T and let $\mathcal{C}$ be a coloring of paths of $\mathcal{P}$ that traverse the edge e between v and its parent. Let χ be a color used by $\mathcal{C}$. χ is called single color if it is used in only one path traversing e and double color if it is used in two paths traversing e in opposite directions. We denote by A_v^i the set of single colors assigned to paths in M_v^i, and by A_v^{ij} the set of double colors assigned to paths in M_v^i and M_v^j.

Definition 10 *Let v be a non-leaf node of T and $\mathcal{P}$ a normal locally-symmetric set of paths. A weakly uniform probability distribution $\mathcal{Q}$ over all proper colorings of paths of $\mathcal{P}$ traversing the edge e between v and its parent with $(2-D)L$ colors is strongly uniform if*

$$|A_v^{ij}| = \frac{D|M_v^i||M_v^j|}{L}$$

for any pair i, j such that paths of M_v^i traverse e in opposite direction to paths of M_v^j.

First, the algorithm performs the following initialization step to produce a random coloring of the paths touching r with $(2-D)L$ colors according to the weakly uniform probability distribution. It colors the paths originating from the root r. It selects uniformly at random without replacement DL colors of the L colors used to color the paths originating from r and assigns them to DL paths which are selected uniformly at random without replacement from the L paths terminating at r. It also assigns $(1-D)L$ new colors to the paths terminating at r which have not been colored.

Then, for $i = 1, \ldots, n-1$, the algorithm executes the procedures Recolor and Color at node v with label i.

The procedure Recolor at a node v takes as input a random coloring $\mathcal{C}_v$ of the paths traversing the edge between v and its parent with $(2-D)L$ colors according to the weakly uniform probability distribution. Recolor stops if v is a leaf. Otherwise, it produces a new random coloring $\mathcal{C}'_v$ of the paths traversing the edge between v and its parent with $(2-D)L$ colors according to the strongly uniform probability distribution.

The procedure Color at a node v takes as input a random coloring $\mathcal{C}'_v$ of the paths traversing the edge between v and its parent with $(2-D)L$ colors according to the strongly uniform probability distribution. Color stops if v is a leaf. Otherwise, it produces a random coloring $\mathcal{C}''_v$ of the paths touching node v in such a way that:

- The restriction $\mathcal{C}_{l(v)}$ of $\mathcal{C}''_v$ on the paths traversing the edge between v and $l(v)$ is a random coloring with $(2-D)L$ colors according to the weakly uniform probability distribution.

- The restriction $\mathcal{C}_{r(v)}$ of $\mathcal{C}''_v$ on the paths traversing the edge between v and $r(v)$ is a random coloring with $(2-D)L$ colors according to the weakly uniform probability distribution.
- The coloring $\mathcal{C}''_v$ uses at most $\frac{4D^2-4D+4}{3D}$ colors.

Finally, the algorithm uses a deterministic algorithm to color the set of paths whose color has been changed by RECOLOR.

In the following we briefly discuss the main ideas used for the precise definition of procedures RECOLOR and COLOR. We remark that the procedures RECOLOR and COLOR can be implemented to run in time polynomial in L, thus, our algorithm runs in time polynomial in L and the number of nodes of the tree.

Let v be a non-leaf node of T. Observe that if $\mathcal{C}_v$ is a random coloring of the paths traversing the edge between v and its parent, then the numbers $|A_v^{ij}|$ are random variables with expectation

$$\mathcal{E}[|A_v^{ij}|] = \frac{|M_v^i||M_v^j|D}{L}.$$

In [1], it is shown that $|A_v^{ij}|$'s follow a hypergeometrical-like distribution and are sharply concentrated around their expectations. The procedure RECOLOR uses this property and, with high probability, by recoloring a small number of paths traversing the edge between v and its parent, it produces a random coloring $\mathcal{C}'_v$ which follows the strongly uniform probability distribution. This is done in such a way that after all executions of RECOLOR the load of the paths of $\mathcal{P}$ whose color has been changed at least once is $o(L)$ with high probability.

The procedure COLOR mimics FRACT-COLOR in some sense. During its execution at a non-leaf node v such that P_v belongs to Scenario I (the case of Scenario II is similar to Scenario II of FRACT-COLOR) it colors the paths of M_v^5 and M_v^6 using colors of A_v^i and A_v^{ij} in such a way that the probability that two paths traversing one of the edges between v and a child of v in opposite directions are assigned the same color equals D/L (recall that this equals the sum of the weights x_v produced by FRACT-COLOR of the independent sets containing both paths). The analysis is much similar to the analysis of FRACT-COLOR. By working on the details of the analysis, we conclude to similar correctness conditions with those of FRACT-COLOR.

We now state our result for the path coloring of locally-symmetric sets of paths.

Theorem 11 *There exists a randomized polynomial-time algorithm which, for any constant $\delta < 1/3$, colors any locally-symmetric set of paths of load L on a binary tree of depth at most $L^\delta/8$, using at most $1.367L + o(L)$ colors, with probability at least $1 - \exp\left(-\Omega(L^\delta)\right)$.*

The restriction in the depth of the tree is necessary so that the upper bound on the number of colors is guaranteed with high probability. The number of nodes of the tree may be exponential in L (i.e., up to $2^{L^\delta/8}$).

References

[1] V. Auletta, I. Caragiannis, C. Kaklamanis, and P. Persiano. Randomized Path Coloring on Binary Trees. *Theoretical Computer Science,* Vol. 289(1), 2002, pp. 355-399.

[2] I. Caragiannis, A. Ferreira, C. Kaklamanis, S. Perennes, and H. Rivano. Fractional Path Coloring with Applications to WDM Networks. In *Proc. of the 28th International Colloquium on Automata, Languages, and Programming (ICALP '01)*, LNCS 2076, Springer, 2001, pp. 732–743.

[3] I. Caragiannis and C. Kaklamanis. Approximate Path Coloring with Applications to Wavelength Assignment in WDM Optical Networks. *Submitted manuscript*, 2003.

[4] I. Caragiannis, C. Kaklamanis, and P. Persiano. Symmetric Communication in All–Optical Tree Networks. *Parallel Processing Letters*, Vol. 10(4), 2000, pp. 305–313.

[5] S. Corteel, D. Gardy, D. Barth, A. Denise, and M. Valencia-Pabon. On the Complexity of Routing Permutations on Trees by Arc-Disjoint Paths. In *Proc. of the 4th LATIN: Theoretical Informatics*, Springer, LNCS 1776, 2000, pp. 308-317.

[6] T. Erlebach and K. Jansen. The Complexity of Path Coloring and Call Scheduling. *Theoretical Computer Science*, Vol. 255(1–2), 2000, pp. 33–50.

[7] T. Erlebach, K. Jansen, C. Kaklamanis, M. Mihail, and P. Persiano. Optimal Wavelength Routing in Directed Fiber Trees. *Theoretical Computer Science*, Vol. 221(1–2), 1999, pp. 119–137.

[8] V. Kumar. Approximating Circular Arc Colouring and Bandwidth Allocation in All–Optical Ring Networks. In *Proc. of the 1st International Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX '98)*, LNCS 1444, Springer, 1998, pp. 147-158.

[9] V. Kumar and E. Schwabe. Improved Access to Optical Bandwidth in Trees. In *Proc. of the 8th Annual ACM–SIAM Symposium on Discrete Algorithms (SODA '97)*, 1997, pp. 437–444.

[10] P. Raghavan, E. Upfal. Efficient Routing in All-Optical Networks. In *Proc. of the 26th Annual Symposium on Theory of Computing (STOC '94)*, 1994, pp. 133–143.

[11] R. Ramaswami, K. Sivarajan. Optical Networks: A Practical Perspective. *Morgan Kauffman Publishers*, 1998.

A The procedure Fract-Color: Scenario II

Let T be a binary tree and $\mathcal{P}$ a normal locally-symmetric set of paths on T. We show how the procedure Fract-Color works when executed at a non–leaf node v, and the set of paths touching node v belongs to Scenario II.

Let P_v be the set of paths of $\mathcal{P}$ touching node v. We denote by I_v the set of independent sets of $\mathcal{I}$ that contain only paths touching nodes with label smaller than v. We denote by I_v^0 the set of independent sets in I_v that do not contain paths traversing the edge between v and its parent in some direction. We denote by I_v^i for $i = 1, 2, 3, 4$, the set of independent sets of I_v which contain a path p of M_v^i and no path traversing the edge between v and its parent in opposite direction to p, and by I_v^{ij} the set of independent sets of $\mathcal{I}$ which contain a path of M_v^i and a path of M_v^j, such that paths of M_v^i traverse the edge between v and its parent in opposite direction than paths in M_v^j.

We remind that, when executed at v, Fract-Color takes as input the set P_v and a non-negative weight assignment $x_{p(v)}(I)$ on the independent sets of $\mathcal{I}$ that satisfies Properties 1, 2, and 3 at $p(v)$ and Property 4 and computes a non-negative weight assignment $x_v(I)$ on the independent sets of $\mathcal{I}$ that satisfies Properties 1, 2, and 3 at v and Property 4. In the case of Scenario II, this is done in the following way:

- Let

$$\begin{aligned}
k_1 &= \sum_{I \in I_v^0} x_{p(v)}(I) \\
k_2 &= 2n(L - |M_v^1|) + \frac{D(L - |M_v^1|)^2}{L} \\
\lambda_1 &= \sum_{I \in I_v^{12}} x_{p(v)}(I) = |M_v^1|^2 \frac{D}{L} \\
\lambda_2 &= \sum_{I \in I_v^1 \cup I_v^2} x_{p(v)}(I) = 2|M_v^1|(1 - D) \\
\lambda_3 &= 2(1 - D)(2|M_v^1| - L) + \frac{D}{L}(2|M_v^1| - L)^2
\end{aligned}$$

- For each independent set I' of I_v^3 and each path p of M_v^6, we set to $x_v(I) = \frac{\beta}{L - |M_v^1|} x_{p(v)}(I')$ the weight of the independent set consisting of the paths of I' and path p and $x_v(I') = (1 - \beta) x_{p(v)}(I')$.

- For each independent set I' of I_v^4 and each path p of M_v^5, we set to $x_v(I) = \frac{\beta}{L - |M_v^1|} x_{p(v)}(I')$ the weight of the independent set consisting of the paths of I' and path p and $x_v(I') = (1 - \beta) x_{p(v)}(I')$.

- For each independent set I' of I_v^{14} and each path p of M_v^5, we set to $x_v(I) = \frac{\gamma}{L - |M_v^1|} x_{p(v)}(I')$ the weight of the independent set consisting of the paths of I' and path p.

- For each independent set I' of I_v^{14} and each path p of M_v^7, we set to $x_v(I) = \frac{1}{|M_v^1|} x_{p(v)}(I')$ the weight of the independent set consisting of the paths of I' and path p.

- For each independent set I' of I_v^{14}, we set to $x_v(I') = \left(1 - \gamma - \frac{2|M_v^1| - L}{|M_v^1|}\right) x_{p(v)}(I')$.

- For each independent set I' of I_v^{23} and each path p of M_v^6, we set to $x_v(I) = \frac{\gamma}{L-|M_v^1|} x_{p(v)}(I')$ the weight of the independent set consisting of the paths of I' and path p.

- For each independent set I' of I_v^{23} and each path p of M_v^7, we set to $x_v(I) = \frac{1}{|M_v^1|} x_{p(v)}(I')$ the weight of the independent set consisting of the paths of I' and path p.

- For each independent set I' of I_v^{23}, we set $x_v(I') = \left(1 - \gamma - \frac{2|M_v^1|-L}{|M_v^1|}\right) x_{p(v)}(I')$.

- For each independent set I' of I_v^1 and each path p of M_v^5, we set to $x_v(I) = \left(\frac{\alpha}{L-|M_v^1|} - \frac{(2|M_v^1|-L)D}{L|M_v^1|(1-D)}\right) x_{p(v)}(I')$ the weight of the independent set consisting of the paths of I' and path p.

- For each independent set I' of I_v^1 and each pair of paths p, q of M_v^5 and M_v^8 respectively, we set to $x_v(I) = \frac{D}{L|M_v^1|(1-D)} x_{p(v)}(I')$ the weight of the independent set consisting of the paths of I' and paths p and q.

- For each independent set I' of I_v^2 and each path p of M_v^6, we set to $x_v(I) = \left(\frac{\alpha}{L-|M_v^1|} - \frac{(2|M_v^1|-L)D}{L|M_v^1|(1-D)}\right) x_{p(v)}(I')$ the weight of the independent set consisting of the paths of I' and path p.

- For each independent set I' of I_v^2 and each pair of paths p, q of M_v^6 and M_v^7 respectively, we set to $x_v(I) = \frac{D}{L|M_v^1|(1-D)} x_{p(v)}(I')$ the weight of the independent set consisting of the paths of I' and paths p and q.

- For each independent set I' of I_v^{12} and each path p of $M_v^7 \cup M_v^8$, we set to $x_v(I) = \frac{1-D}{\max\{\lambda_1,\lambda_3\}} x_{p(v)}(I')$ the weight of the independent set consisting of the paths of I' and path p.

- For each independent set I' of I_v^{12} and each pair of paths p, q of M_v^7 and M_v^8 respectively, we set to $x_v(I) = \frac{D}{\max\{\lambda_1,\lambda_3\}L} x_{p(v)}(I')$ the weight of the independent set consisting of the paths of I' and paths p and q.

- For each independent set I' of I_v^{12}, we set $x_v(I') = \left(1 - \frac{2(1-D)(2|M_v^1|-L)}{\max\{\lambda_1,\lambda_3\}} - \frac{D(2|M_v^1|-L)^2}{\max\{\lambda_1,\lambda_3\}L}\right) x_{p(v)}(I')$.

- For each independent set I' of $I_v^1 \cup I_v^2$ and each path p of $M_v^7 \cup M_v^8$, we set to $x_v(I) = (1-D)\max\{0, \frac{\lambda_3-\lambda_1}{\lambda_2\lambda_3}\} x_{p(v)}(I')$ the weight of the independent set consisting of the paths of I' and path p.

- For each independent set I' of $I_v^1 \cup I_v^2$ and each pair of paths p, q of M_v^7 and M_v^8 respectively, we set to $x_v(I) = \frac{D}{L}\max\{0, \frac{\lambda_3-\lambda_1}{\lambda_2\lambda_3}\} x_{p(v)}(I')$ the weight of the independent set consisting of the paths of I' and paths p and q.

- For each independent set I' of $I_v^1 \cup I_v^2$, we set $x_v(I') = \left(1 - \alpha - \max\{0, \frac{\lambda_3-\lambda_1}{\lambda_2\lambda_3}\}(2|M_v^1| - L)(2 - 3D + 2|M_v^1|\frac{D}{L})\right) x_{p(v)}(I')$.

- For each independent set I' of I_v^0 and each path p of $M_v^5 \cup M_v^6$, we set to $x_v(I) = \frac{n}{\max\{k_1,k_2\}2(L-|M_v^1|)} x_{p(v)}(I')$ the weight of the independent set consisting of the paths of I' and path p.

- For each independent set I' of I_v^0 and each pair of paths p, q of M_v^5 and M_v^6 respectively, we set to $x_v(I) = \frac{D}{\max\{k_1,k_2\}(L-|M_v^1|)^2} x_{p(v)}(I')$ the weight of the independent set consisting of the paths of I' and paths p and q.
- For each independent set I' of I_v^0, we set $x_v(I') = \max\{0, 1 - \frac{k_1}{k_2}\} x_{p(v)}(I')$.
- For each path $p \in M_v^5 \cup M_v^6$, we set to $x_v(I) = n \max\{0, 1 - \frac{k_1}{k_2}\}$ the weight of the independent set consisting only of p.
- For each pair of paths p, q of M_v^5 and M_v^6 respectively, we set to $x_v(I) = \frac{D}{L} \max\{0, 1 - \frac{k_1}{k_2}\}$ the weight of the independent set consisting only of p and q.
- For all other independent sets I (i.e., independent sets that contain at least one path touching only nodes with label greater than v's) we set $x_v(I) = 0$.

Now, we have to set values to the parameters α, β, γ, n so that Fract-Color is correct (i.e., it matches the input-output specification given in Section 4). The next lemma gives sufficient conditions for this purpose. Although lengthy, the proof is similar in spirit with the proof of Lemma 8.

Lemma 12 (Correctness conditions) *If*

$$\frac{(L - |M_v^1|)(2|M_v^1| - L)D}{L|M_v^1|(1 - D)} \leq \alpha \tag{6}$$

$$\alpha \leq 1 - \max\{0, \frac{\lambda_3 - \lambda_1}{\lambda_2 \lambda_3}\}(2|M_v^1| - L)(2 - 3D + 2|M_v^1|\frac{D}{L}) \tag{7}$$

$$0 \leq \beta \leq 1 \tag{8}$$

$$0 \leq \gamma \leq 1 - \frac{2|M_v^1| - L}{|M_v^1|} \tag{9}$$

$$n \geq 0 \tag{10}$$

$$\alpha(1 - D) + \gamma \frac{D}{L}(L - |M_v^1|) = \frac{D}{L}(L - |M_v^1|) \tag{11}$$

$$\beta(1 - D) + \gamma \frac{D}{L}|M_v^1| = \frac{D}{L}(L - |M_v^1|) \tag{12}$$

$$n = 1 - D - \beta(1 - D) \tag{13}$$

$$2n(L - |M_v^1|) \leq \frac{7D^2 - 10D + 4}{3D}L - \frac{D|M_v^1|^2}{L} \tag{14}$$

then Fract-Color *is correct.*

In order to prove that Fract-Color is correct, we will compute values for α, β, γ, n such that the correctness conditions (6)–(14) of Lemma 12 are satisfied. We distinguish between two cases according to the size of M_v^1.

CASE I. $|M_v^1| \leq \frac{2D-1}{D}L$. In this case, we have the following settings

$$\alpha = 1$$

$$\beta = \frac{|M_v^1|}{L - |M_v^1|} - \frac{D(2|M_v^1| - L)}{(1 - D)L}$$

$$\gamma = 1 - \frac{(1-D)L}{D(L-|M_v^1|)}$$
$$n = (2|M_v^1| - L)\left(\frac{D}{L} - \frac{1-D}{L-|M_v^1|}\right)$$

Since $\frac{1}{2} \leq \frac{|M_v^1|}{L} \leq \frac{2D-1}{D}$, it follows that α satisfies condition (6), β satisfies condition (8), γ satisfies condition (9), and n satisfies condition (10). Also, since $\lambda_1 \geq \lambda_2$, we obtain that a satisfies condition (7). By simple calculations, we obtain that conditions (11)–(14) are also satisfied.

CASE II. $|M_v^1| \geq \frac{2D-1}{D}L$. In this case, we have the following settings

$$\alpha = \frac{D(L-|M_v^1|)}{(1-D)L}$$
$$\beta = \frac{D(L-|M_v^1|)}{(1-D)L}$$
$$\gamma = 0$$
$$n = 1 - D - \frac{D}{L}(L-|M_v^1|)$$

Since $1 \geq \frac{|M_v^1|}{L} \geq \frac{2D-1}{D}$, it follows that α satisfies conditions (6) and (7), β satisfies condition (8), γ satisfies condition (9), and n satisfies condition (10). By simple calculations, we obtain that conditions (11)–(14) are also satisfied.