Alcom-FT Technical Report Series
ALCOMFT-TR-03-187

Approximate Path Coloring with Applications to Wavelength Assignment in WDM Optical Networks *

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

December 2, 2003

Abstract

Motivated by the wavelength assignment problem in WDM optical networks, we study path coloring problems in graphs. Given a set of paths P on a graph G, the path coloring problem is to color the paths of P so that no two paths traversing the same edge of G are assigned the same color and the total number of colors used is minimized. The problem has been proved to be NP-hard even for trees and rings.

Using optimal solutions to fractional path coloring, a natural relaxation of path coloring, on which we apply a randomized rounding technique combined with existing coloring algorithms, we obtain new upper bounds on the minimum number of colors sufficient to color any set of paths on any graph. The upper bounds are either existential or constructive.

The existential upper bounds are significantly better than existing ones provided that the cost of the optimal fractional path coloring is sufficiently large and the dilation of the set of paths is small. Our algorithmic applications include improved approximation algorithms for path coloring in rings and in bidirected trees. Our results extend to variations of the original path coloring problem arizing in multifiber WDM optical networks.

1 Introduction

We study path coloring problems in graphs. Let P be a set of paths on a graph G and $k > 0$ be an integer. The paths of P and the edges of G may be directed or undirected. The path k-coloring problem (or, simply, path coloring when $k = 1$) is to assign colors to the paths of P in such way that at most k paths with the same color share an edge of the graph and the total number of colors is minimized. The problem has been proved to be NP-hard, even for

*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.

$k = 1$ and even for the simplest topologies of rings and trees. Thus, approximation algorithms are essential.

The problem has application to Wavelength Division Multiplexing (WDM) optical networks [18]. Such networks consists of nodes connected with fibers. Connection requests are pairs of nodes to be thought of as transmitter-receiver pairs. For each connection request, WDM technology routes the request through a path from its transmitter to its receiver and assigns this path a wavelength in such way that paths going through the same fiber are assigned different wavelengths. Recently, the multifiber WDM network model was introduced [14, 11]. In these networks each fiber of the standard model is replaced by k identical "parallel" fibers.

For path coloring problems, bounds on the number of colors are usually expressed as a function of the load of the set of paths given as input, i.e., the maximum number of paths going through any edge of the graph. Erlebach et al. [5] present an algorithm that colors any set of paths of load L on a bidirected tree with at most $5L/3$ colors. Auletta et al. [1] present a randomized algorithm that colors any set of paths of load L on a bidirected binary tree of depth $o(L^{1/3})$ with at most $7L/5 + o(L)$ colors, with high probability. In rings, Tucker's algorithm [19] colors any set of paths of load L with $2L$ colors or with $\lceil \frac{l-1}{l-2} L \rceil + 1$ colors where l is the minimum number of paths necessary to cover the ring, as shown recently in [12, 20]. The interested reader may refer to [2] for a survey on path coloring results motivated by WDM optical networks.

Upper bounds of $(1 + 1/k)\frac{L}{k} + c_k$ (where c_k depends only on k) for path k-coloring in rings are presented in [14, 11]. The results in [5, 1] can be trivially modified to give $\lceil \frac{5L}{3k} \rceil$ and $\frac{7L}{5k} + o(L/k)$ upper bounds for path k-coloring in arbitrary and binary bidirected trees, respectively. Note that L/k is a lower bound on the minimum number of colors necessary to k-color any set of paths of load L. Thus, by dividing the upper bound on the number of colors achieved by an algorithm by L/k we obtain an upper bound on its approximation ratio.

Another line of research is to design approximation path coloring algorithms which use an optimal fractional coloring to obtain a provably good approximation of the optimal path coloring. Given a set of paths on a graph, we may think of the path k-coloring problem as the problem of covering the paths by as few as possible k-independent sets of paths, i.e., sets of paths in which at most k paths share an edge of the graph. This can be captured by the following integer linear program

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

where $\mathcal{I}$ denotes the set of the k-independent sets of P. This formulation has a natural relaxation into the following linear program:

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

The corresponding combinatorial problem is called the fractional (path) k-coloring problem [3, 7] and any feasible solution to the linear program is called a fractional k-coloring of P. Given a set of paths P on a graph G, we denote by $w_k(P, G)$ and $f_k(P, G)$ the cost of the optimal solution of the integer linear program and its relaxation, respectively. Alternatively, one may see the (fractional) path coloring problem for a set of paths P on a graph G as a (fractional) graph coloring problem on the conflict graph of P, i.e., the graph which has a node for each path of P and an edge between two nodes if the corresponding paths traverse the same edge on G.

In general, fractional path coloring is hard to approximate while it can be solved in polynomial time provided that an indepedent set of maximum size can be computed efficiently [7]. Jansen [8] and Jansen and Porkolab [9] extend this result by showing how to compute α-approximate solutions for fractional coloring using α-approximation algorithms for computing maximum independent sets. The techniques of [7, 8, 9] can be applied to fractional path k-colorings as well. However, they constitute general ways for approximating the optimal objective value of the corresponding linear program with an exponential number of variables while, in approximation algorithms for path coloring, a provably good solution for fractional path coloring (i.e., the values of the variables of the corresponding linear program) is rounded to an integral one which gives a path coloring. So, previous work (as well as this paper) seeks for formulations of fractional path coloring as a linear program with a polynomial number of variables.

The work of Kumar [10] on the path coloring problem in rings (also known as circular arc coloring problem) uses a reduction to instances of integral multicommodity flow due to Tucker [19]. Kumar solves the relaxation of the multicommodity flow problem optimally (this is equivalent to computing the optimal fractional coloring almost exactly) and then performs randomized rounding [16, 17] to obtain the path coloring. The resulting path coloring is proved to be within $1.37 + o(1)$ of the optimal number of colors.

In [3], it is shown that the fractional path coloring can be solved in polynomial time in bounded-degree bidirected trees. By applying a randomized rounding method similar to that used by Kumar and using the algorithm of Erlebach et al. [5] as a subroutine, a $(1.613+o(1))$-approximation algorithm is obtained.

The contribution of this paper can be summarized as follows:

- We introduce a new randomized rounding method applied to fractional path k-colorings. For the analysis, we study a generalization of a classical occupancy problem which may be of interest in other applications as well.
- Using the randomized rounding we obtain new existential upper bounds on the minimum number of colors sufficient to k-color any set of paths provided that the cost of the optimal fractional coloring is sufficiently large and the dilation (i.e., the length of the longest path) is small. Existential upper bounds for arbitrary k are also obtained for arbitrary trees and rings.
- We also discuss two algorithmic applications of the method. For constant k, we present polynomial time approximation path k-coloring algorithms in bidirected trees of bounded-degree and in rings. The algorithms have better approximation ratios than existing ones provided that the load is not small. The same restriction exists in previous results [3, 10]. For WDM networks, this is a realistic assumption.
  - We give a method which computes an almost optimal fractional k-coloring of a set of paths on a bounded-degree bidirected tree. For $k = 1$, this method is slightly weaker than the method in [3] but it is suitable for our purposes. The fractional k-coloring is then used to perform randomized rounding and, using the algorithm of Erlebach et al. [5] as a subroutine, we obtain a $(1.511 + o(1))$-approximation

algorithm for path k-coloring. In binary trees, we use the randomized algorithm in [1] as a subroutine to obtain a $(1.336 + o(1))$-approximation algorithm.

– In rings, we present a reduction of path k-coloring to instances of an integral constrained multicommodity flow problem, generalizing in this way Tucker's reduction for $k > 1$. This reduction is used for computing almost optimal fractional k-colorings, which, combined with randomized rounding and existing algorithms [11, 12, 14, 19, 20], give better approximation algorithms for path k-coloring ($k \geq 2$) and for special instances of path coloring in rings.

The main strength of our randomized rounding technique is that it uses a parameter which can be adjusted according to the approximation ratio of the k-coloring algorithm used as a subroutine. It can be used to give path k-coloring algorithms with improved approximation ratio in any graph (directed or undirected) where the best known upper bound is expressed in terms of the load, provided that an almost optimal fractional k-coloring can be computed efficiently.

The rest of the paper is structured as follows. In Section 2, we present the occupancy problem and study the behaviour of random variables related to it. We present the randomized rounding technique in Section 3 together with its analysis and applications. We devote Section 4 to describe how to compute almost optimal fractional k-colorings in bidirected trees and in rings and how to perform randomized rounding according to them. Due to lack of space, most of the proofs have been put in appendix.

2 An occupancy problem

In this section, we study the behaviour of random variables in a new occupancy problem which is a generalization of classical "balls-to-bins" processes [15]. Our analysis will be very useful for proving bounds on the performance of our randomized rounding method.

Let $k \geq 1$ be an integer, $n > 0$ be an integer multiple of k and $q > 0$. Consider the following "balls-to-bins" process. We have n/k balls and n bins. Associated with each ball i and each subset of bins s_j of size k is a non-negative number p_{ij} such that $\sum_j p_{ij} = 1$ for any ball i, and $\sum_{i=1}^{n/k} \sum_{j:\ell \in s_j} p_{ij} = 1$ for each bin ℓ. For each ball $i = 1, ..., n/k$, we toss a coin with $\Pr[\text{HEADS}] = q - \lfloor q \rfloor$. On HEADS, we execute $\lfloor q \rfloor + 1$ rounds, otherwise we execute $\lfloor q \rfloor$ rounds. In each round executed for ball i, a subset of bins of size k is selected randomly among all possible subsets according to the probabilities p_{ij}, and one copy of ball i is thrown to each bin of the selected set. We denote by $\mathcal{Q}$ the random variable representing the number of empty bins after the execution of the process and by $\mathcal{R}$ the random variable representing the total number of rounds executed.

Lemma 1
a. $E[\mathcal{Q}] \leq ne^{-q}$
b. For any $\lambda > 0$, $\Pr[|\mathcal{Q} - E[\mathcal{Q}]| > \lambda] \leq 2\exp\left(-\frac{\lambda^2}{2\lceil q \rceil nk}\right)$
c. $E[\mathcal{R}] = qn/k$
d. If q is not integer, then for any $\lambda > 0$, $\Pr[\mathcal{R} - E[\mathcal{R}] > \lambda] \leq \exp\left(-\frac{\lambda^2 k}{4(q-\lfloor q \rfloor)n}\right)$

3 The randomized rounding technique

In this section we present the randomized rounding technique. The technique is applied to normal sets of paths.

Definition 2 *A set of (directed) paths P on a network G is called normal if it has the same load on every (directed) edge of G.*

The main idea is to round a fractional k-coloring of the set of paths P' and obtain a k-coloring of some of the paths of P. In particular, we use a family of fractional k-coloring functions as a representation of a fractional k-coloring.

Definition 3 *Let P be a normal set of paths of load kZ (where Z is an integer). A set of non-negative weight functions x_j for $j = 1, ..., Z$ on the k-independent sets of P is called a family of fractional k-coloring functions for P if*

$$\sum_{I \in \mathcal{I}: p \in I} \sum_{j=1}^{Z} x_j(I) = 1, \text{ for any path } p \in P, \text{ and}$$

$$\sum_{I \in \mathcal{I}} x_j(I) = 1, \text{ for any } j = 1, ..., Z,$$

where $\mathcal{I}$ is the set of the k-independent sets of P.

Observe that if a set of paths P of load kZ (where Z is an integer) on a graph G has a family of fractional k-coloring functions, then it has a fractional k-coloring of cost exactly Z since the weight function x defined as $x(I) = \sum_{i=1}^{Z} x_i(I)$ for $I \in \mathcal{I}$, is a fractional k-coloring of P of cost Z. The opposite also holds as the following lemma states.

Lemma 4 *Let $k \geq 1$ be an integer constant and let P be a normal set of paths of load kZ (where Z is integer) on a graph G. Given a fractional k-coloring x of P of cost Z, we can construct a family of fractional k-coloring functions y_j for $j = 1, ..., Z$.*

The following lemma implies that, for any set of paths, there exists a superset which has a family of fractional k-coloring functions.

Lemma 5 *Let $k > 0$ be an integer and let P be a set of paths on a graph G. Consider the normal set of paths P' of load $k(1 + \lceil f_k(P, G) \rceil)$ on G obtained by adding single-hop paths to P. It is $f_k(P', G) = 1 + \lceil f_k(P, G) \rceil$.*

We are now ready to describe the randomized rounding technique. The technique applies to normal sets of paths having a family of fractional k-coloring functions. On input a set of paths P of load $kf_k(P, G)$ (where $f_k(P, G)$ is integer) on a graph G, the randomized rounding technique uses a parameter $q > 0$ and a family of fractional k-coloring functions x_i, $i = 1, ..., f_k(P, G)$ for P to properly k-color some of the paths of P as follows. Initially, all paths of P are uncolored. For each $i = 1, ..., f_k(P, G)$, randomized rounding proceeds by tossing a coin with $\Pr[\text{HEADS}] = q - \lfloor q \rfloor$. On HEADS, it executes $\lfloor q \rfloor + 1$ rounds, otherwise it executes $\lfloor q \rfloor$ rounds. In each round associated with some i, a k-independent set is selected by casting a die with a face for each k-independent set with $x_i(I) > 0$ and probability $x_i(I)$ associated with the face corresponding to the k-independent set I. At the end of the round, all the paths of the selected k-independent set which are still uncolored are colored with a new color.

In the rest of this section we will use the randomized rounding technique either to prove existential upper bounds on the minimum number of colors sufficient to k-color a set of paths or to obtain polynomial time approximation algorithms for k-coloring sets of paths with a provably small number of colors.

3.1 Existential upper bounds

We start by proving a simple upper bound on the number of colors sufficient for k-coloring any set of paths on any (directed or undirected) graph.

Theorem 6 *For any set of paths P on a graph G with m edges and for any integer $k > 0$, it is*

$$w_k(P,G) \leq 1 + (1 + \lceil f_k(P,G) \rceil) \left\lceil \ln\left(\frac{km(1+\lceil f_k(P,G)\rceil)}{k+1}\right)\right\rceil.$$

Proof: Let P' be the normal set of paths of load $k(1+\lceil f_k(P,G)\rceil)$ obtained by adding single-hop paths to P. By Lemma 5, it is $f_k(P',G) = 1+\lceil f_k(P,G)\rceil$ and P' has a family of fractional k-coloring functions x_i for $i = 1, ..., \lceil f_k(P,G)\rceil$. We will apply the randomized rounding to P' with $q = \left\lceil \ln\left(\frac{km(1+\lceil f_k(P,G)\rceil)}{k+1}\right)\right\rceil$ using the family of fractional k-coloring functions x_i. Since q is integer, the number of colors used during the randomized rounding is $q(1 + \lceil f_k(P,G)\rceil)$. We will show that with non-zero probability, only one more color is sufficient to complete the k-coloring of P'. This will imply a coloring of P with the same number of colors and will complete the proof of the theorem.

Let e be an edge of G. We denote by $\mathcal{Q}_e$ the random variable representing the number of paths traversing e which are left uncolored after the application of randomized rounding. We may view the randomized rounding as a balls-to-bins process like the one described in Section 2. The paths traversing edge e are the bins and the paths of the k-independent set traversing e which are selected during a round correspond to copies of a ball thrown into the k corresponding bins. The probabilities on the sets of k bins where copies of balls are thrown in the corresponding balls-to-bins process are defined by the family of fractional k-coloring functions. Thus, the random variable $\mathcal{Q}_e$ corresponds to the number of empty bins in the balls-to-bins process.

By applying the Markov inequality (see [15], Theorem 3.2, page 46) for the random variable $\mathcal{Q}_e$ (using Lemma 1 to upper-bound the expectation of $\mathcal{Q}_e$) we have that the probability that more than k paths traversing edge e are left uncolored after rounding is at most

$$\frac{E[\mathcal{Q}_e]}{k+1} \leq \frac{k(1+\lceil f_k(P,G)\rceil)}{(k+1)e^q} < 1/m.$$

Since there are m edges in G, the probability that the load of paths left uncolored after rounding exceeds k is less than 1, and, hence, these paths can be k-colored with one extra color. ■

An upper bound of $f_k(P,G)(1 + \ln(km))$ for $w_k(P,G)$ can be obtained by using the techniques of Lovász [13]. For most values of the parameters, our result is slightly weaker but note that it was proved using the simplest probabilistic tool. In the following we give better upper bounds for $w_k(P,G)$ provided that $f_k(P,G)$ is sufficiently large.

Lemma 7 *Let P be a set of paths on a graph G with $m > 3$ edges, $k > 0$ be an integer and β be such that*

$$\beta \geq \max_{P' \subseteq P} \left\{ \frac{k w_k(P', G)}{L(P', G)} \right\}$$

where $L(P', G)$ denotes the load of the set of paths P' on G. If $f_k(P,G) = \Omega\left(\frac{\beta^2 \ln m}{\ln \beta}\right)$, then $w_k(P,G) \leq f_k(P,G) O(\ln \beta)$, and, if $f_k(P,G) = \omega\left(\frac{\beta^2 \ln m}{\ln \beta}\right)$, then $w_k(P,G) \leq f_k(P,G)(1 + \ln \beta + o(1))$.

Proof: Let P' be the normal set of paths of load $k(1 + \lceil f_k(P,G) \rceil)$ obtained by adding single-hop paths to P. By Lemma 5, it is $f_k(P', G) = 1 + \lceil f_k(P,G) \rceil$ and P' has a family of fractional k-coloring functions x_i for $i = 1, ..., 1 + \lceil f_k(P,G) \rceil$. We apply randomized rounding to P' with $q = \ln \beta$ using the family of fractional k-coloring functions x_i. We define $Z = 1 + \lceil f_k(P,G) \rceil$.

By Lemma 1, we obtain that $E[\mathcal{R}] = Z \ln \beta$ and that, for any $\lambda > 0$, the probability that $\mathcal{R} \geq E[\mathcal{R}] + \lambda$ is at most $\exp\left(-\frac{\lambda^2}{4Z}\right)$. By setting $\lambda = 2\sqrt{Z \ln m}$, we have that the probability that the number of colors used during rounding exceeds $Z \ln \beta + 2\sqrt{Z \ln m}$ is at most $1/m$.

Let e be an edge of G and $\mathcal{Q}_e$ be the random variable representing the number of paths traversing e which are left uncolored after the application of randomized rounding. Again, as in the proof of Theorem 6, we observe that the random variable $\mathcal{Q}_e$ has identical behaviour to the number of empty bins in a corresponding balls-to-bins process.

Using Lemma 1, we obtain that $E[\mathcal{Q}_e] \leq \frac{kZ}{\beta}$ and that, for any $\lambda > 0$, the probability that $\mathcal{Q}_e \geq E[\mathcal{Q}_e] + \lambda$ is at most $2\exp\left(-\frac{\lambda^2}{2k^2 Z \lceil \ln \beta \rceil}\right)$. By setting $\lambda = 2k\sqrt{Z \lceil \ln \beta \rceil \ln m}$, we have that the probability that $\mathcal{Q}_e$ exceeds $\frac{kZ}{\beta} + 2k\sqrt{Z \lceil \ln \beta \rceil \ln m}$ is less than $2/m^2$. Since there are m edges in G, the load of the paths left uncolored after the application of the randomized rounding technique is at most $\frac{kZ}{\beta} + 2k\sqrt{Z \lceil \ln \beta \rceil \ln m}$, with probability at least $1 - 2/m$.

Now, using the definition of β, it can be easily verified that, since the set of paths left uncolored after rounding consists of a subset of the original set of paths P and (possibly) some additional single-hop paths, it can be k-colored with at most β/k times its load colors.

Hence, with probability at least $1 - 3/m > 0$, the total number of colors is at most

$$Z \ln \beta + 2\sqrt{Z \ln m} + Z + 2\beta\sqrt{Z \lceil \ln \beta \rceil \ln m}.$$

The proof completes by observing that if $f_k(P,G) = \Omega\left(\frac{\beta^2 \ln m}{\ln \beta}\right)$ (resp. $\omega\left(\frac{\beta^2 \ln m}{\ln \beta}\right)$), then the sum of the second and the fourth term in the above expression is of order $O(f_k(P,G) \ln \beta)$ (resp. $o(f_k(P,G) \ln \beta)$). ■

We will apply Lemma 7 to obtain existential upper bounds for $w_k(P,G)$ in general (directed or undirected graphs) and in bidirected trees.

Theorem 8 *Let P be a set of paths on a graph G with dilation D and $k > 0$ be an integer. If $f_k(P,G) = \Omega\left(\frac{D^2 \ln m}{\ln D}\right)$, then $w_k(P,G) \leq f_k(P,G) O(\ln D)$, and, if $f_k(P,G) = \omega\left(\frac{D^2 \ln m}{\ln D}\right)$, then $w_k(P,G) \leq f_k(P,G)(1 + \ln D + o(1))$.*

Proof: Observe that the conflict graph of any set of paths of dilation D and load L has degree at most $D(L-1)$ and, hence, can be k-colored with at most $\left\lceil \frac{DL - D + 1}{k} \right\rceil$ colors. The proof completes by applying Lemma 7 with $\beta = D$. ■

Theorem 9 *Let $k > 0$ be an integer and P be a set of paths of load $\omega(k \ln m)$ on a bidirected tree T with m directed edges. It holds that $w_k(P,T) \leq (1.511 + o(1)) f_k(P,T)$.*

Proof: Erlebach et al. [5] present an algorithm which colors any set of paths of load L on a bidirected tree with at most $5L/3$ colors. Clearly, it can be slightly modified to k-color any set of paths of load L with at most $\left\lceil \frac{5L}{3k} \right\rceil$ colors. Thus, we may apply Lemma 7 with $\beta = \frac{5}{3} + \frac{1}{\ln m}$ (observe that the lower bound on the load implies that $f_k(P,T) = \omega(\ln m)$) and obtain the desired bound. ■

3.2 Algorithmic applications

Observe that the path k-coloring algorithm we used in the proof of Lemma 7 would run in polynomial time on input a set of paths P on a graph G if

(1). A normal superset P' of P of load $k\lceil f_k(P,G) \rceil$ on G can be computed in polynomial time.

(2). Die-casting according to a family of fractional k-coloring functions implied by the fractional k-coloring of P' can be performed in polynomial time.

(3). For any set of paths P, a k-coloring of the paths in P with at most β/k times the load of P colors can be computed in polynomial time.

Although in both Theorems 8 and 9 property (3) is guaranteed by a polynomial time algorithm, (1) and (2) are infeasible in general unless $P = NP$. This is due to the fact that fractional path coloring is as hard to approximate as fractional graph coloring (it is easy to see that for any graph H, we can construct a set of paths on a graph G having H as its conflict graph) which, in turn, is almost as hard to approximate as graph coloring [7, 13]. Moreover, a family of fractional k-coloring functions x_i may have $x_i(I) > 0$ for exponentially many k-independent sets of P.

In Section 4, given a set of paths P on a graph G which is either a bidirected tree of bounded degree or a ring, we show how to construct a normal superset of P of load kZ having a fractional k-coloring of integer cost $Z \leq 1 + \lceil f_k(P,G) \rceil$ and how to perform die-casting according to a family of fractional k-coloring functions implied by this fractional k-coloring, both in polynomial time when $k > 0$ is an integer constant.

For bidirected trees of bounded degree, following this approach, applying randomized rounding with $q = \ln \frac{5}{3} \approx 0.511$, and using the algorithm of Erlebach et al. [5] to k-color the paths left uncolored after the application of randomized rounding, we obtain the following result.

Theorem 10 *Let $k \geq 1$ be an integer constant. There exists a polynomial-time algorithm which, on input a set of paths P of load $\omega(\ln m)$ on a bounded-degree bidirected tree with m directed edges, computes a $(1.511 + o(1))$-approximate k-coloring of P, with high probability.*

For sets of paths of load L on binary trees of depth $o(L^{1/3})$, there exists a randomized algorithm that colors them using at most $7L/5 + o(L)$ colors, with high probability [1]. Thus, we may follow the same approach used for bounded-degree trees, apply randomized rounding with $q = \ln \frac{7}{5} \approx 0.336$, and use this randomized algorithm to k-color the paths left uncolored to obtain the following result.

Theorem 11 *Let $k \geq 1$ be an integer constant. There exists a polynomial-time algorithm which, on input a set of paths P of load $\omega(\ln m)$ on a binary bidirected tree with m directed edges and of depth $o(L^{1/3})$, computes a $(1.336 + o(1))$-approximate k-coloring of P, with high probability.*

We now present an improved approximation for some instances of the path coloring problem in rings. On input a set of paths P on a ring, we use randomized rounding with $q = \ln \frac{l-1}{l-2}$ where l is the minimum number of paths of P necessary to cover the ring, and Tucker's algorithm [19] to color the paths left uncolored after randomized rounding. Li and Simha [12] and, independently, Valencia-Pabon [20] show that Tucker's algorithm colors P with at most $\left\lceil \frac{l-1}{l-2} L \right\rceil + 1$ colors. We obtain the following result.

Theorem 12 *There exists a polynomial-time algorithm which, on input a set of paths P of load $\omega(\ln m)$ on a ring with m edges, computes a $\left(1 + \ln \frac{l-1}{l-2} + o(1)\right)$-approximate coloring of P, with high probability, where l is the minimum number of paths in P necessary to cover the ring.*

It can be easily seen that for sets of paths with $l \geq 5$, the approximation ratio of our algorithm is better than the approximation ratio of the algorithms in [10], [20], and [12].

We can also improve the best known approximation ratio for k-coloring of sets of paths in rings by using randomized rounding with $q = \ln(1 + 1/k)$ and an algorithm presented in [14, 11] to complete the k-coloring. This algorithm (due to Margara and Simon [14] and to Li and Simha [11]) k-colors any set of paths of load L on a ring using at most $\left(1 + \frac{1}{k}\right) \frac{L}{k} + c_k$ colors (where c_k may depend on k). We obtain the following result.

Theorem 13 *Let $k \geq 2$ be an integer constant. There exists a polynomial-time algorithm which, on input a set of paths P of load $\omega(\ln m)$ on a ring with m edges, computes a $(1 + \ln(1 + 1/k) + o(1))$-approximate k-coloring of P, with high probability.*

4 Computing families of fractional k-coloring functions

In this section, given a set of paths P on a bounded-degree tree or on a ring, we show how to compute normal supersets of P of load kZ which have a fractional k-coloring of integer cost $Z \leq 1 + \lceil f_k(P, G) \rceil$.

In both cases, we follow the same augmentation procedure. Starting with a set of paths P of load L on a network G, we construct a normal superset P_0 of P having load the first multiple of k greater or equal to L (i.e., $k\lceil L/k \rceil$). This is done by adding single-hop paths traversing the edges of the tree which are not fully loaded. We run a procedure called checker on the set of paths P_0. The checker returns YES if the set of paths taken as input has a fractional k-coloring of cost equal to its load over k; it returns NO otherwise. If the checker returns NO, we continue this procedure for $i = 1, 2, \ldots$, by constructing a normal superset P_i of P of load $k(i + \lceil L/k \rceil)$ and running the checker on P_i, until it returns YES.

By Lemma 5, we know that the augmentation procedure terminates after at most $1 + \lceil f_k(P, G) \rceil$ executions of the checker. Clearly, $\lceil f_k(P, G) \rceil$ is polynomial in L and the size of the graph. Furthermore, the load of the set of paths given as input to the checker in each execution is also polynomial in L and the size of the graph. In what follows, we will describe how the checker works in bounded-degree bidirected trees and in rings and we will prove that it runs in polynomial time in terms of the load of the set of paths taken as input and the size of the graph. As a result, we will obtain that the whole augmentation procedure runs in polynomial time.

In both cases, we will also show how to use the fractional k-coloring computed during the last execution of the checker to perform die-casting in polynomial time according to a family of fractional k-coloring functions implied by this fractional k-coloring.

4.1 Bidirected trees

In this section, we will describe the checker TREE-k-CHECKER for checking whether a normal set of paths P of load L which is a multiple of k on a bidirected tree T has a fractional k-coloring of cost L/k.

Given a non-leaf node v of the tree, consider the subset P_v of P containing the paths that touch node v. We denote by $\mathcal{I}(P_v)$ the set of all k-independent sets of paths of P_v which have full load k on each directed edge adjacent to v.

TREE-k-CHECKER constructs the linear program described in the following:

> The linear program has a non-negative weight $x(I)$ for each k-independent set of $\mathcal{I}(P_v)$, for any non-leaf node v of the tree. The objective is to maximize the sum of the weights of the k-independent sets of $\mathcal{I}(P_r)$, where r is a specific non-leaf node of T. There are constraints of two types. The first type of constraints is that, for each path $p \in P$ and for each non-leaf node v it touches, the sum of the weights of the k-independent sets of $\mathcal{I}(P_v)$ it belongs to is constrained to be at most 1. The second type of contraints is that, for any pair of adjacent non-leaf nodes v and u of the tree, and any set of k paths $p_1, ..., p_k$ traversing the directed edge (v, u) and any set of k paths $q_1, ..., q_k$ traversing the opposite directed edge (u, v), the sum of the weights of the k-independent sets of $\mathcal{I}(P_u)$ that contain $p_1, ..., p_k, q_1, ...q_k$ is constrained to be equal to the sum of weights of the k-independent sets of $\mathcal{I}(P_v)$ that contain $p_1, ..., p_k, q_1, ...q_k$.

TREE-k-CHECKER solves the above linear program and returns YES if it has a solution of cost L/k. Otherwise, it returns NO.

Lemma 14 *Let $k > 0$ be an integer constant. On input a normal set of paths P of load L which is a multiple of k on a bidirected tree T of bounded degree, TREE-k-CHECKER runs in polynomial time and returns YES iff P has a fractional k-coloring of cost L/k.*

Now consider the application of the augmentation procedure on the original set of paths P of load L on the tree T using TREE-k-CHECKER as checker. We denote by $P_{Z-\lceil L/k \rceil}$ the normal set of paths of load kZ (where Z is an integer) produced when the augmentation procedure terminates. The following statement immediately follows by the definition of the augmentation procedure and by Lemma 14.

Lemma 15 $Z = \lceil f_k(P, T) \rceil$.

When the augmentation procedure terminates we use the solution of the linear program to implicitly build a family of fractional k-coloring functions and perform die-casting according them. Although in the proof of Lemma 14, the fractional k-coloring has non-zero weight for an exponential number of k-independent sets, we can perform die-casting in polynomial time.

Lemma 16 *Die-casting according to a family of fractional k-coloring functions can be done in polynomial time.*

4.2 Rings

In this section, we describe the checker RING-k-CHECKER. It receives as input a normal set of paths P of load L which is a multiple of k on a ring C with m edges and checks whether P has a fractional k-coloring of cost L/k.

We denote by $e_0, e_1, ..., e_{m-1}$ the edges of the ring C (edges e_i and $e_{i+1 \bmod m}$ are consecutive), by P_{e_i} the subset of P consisted of the paths of P traversing edge e_i, and by $\mathcal{I}(P_{e_i})$ the set of all subsets of P_{e_i} of size k. Note that each set of paths in $\mathcal{I}(P_{e_i})$ is a k-independent set.

RING-k-CHECKER considers the following multicommodity flow network $H(P, C)$.

> The network has $m+1$ levels of nodes. Levels $0, ..., m-1$ correspond to the edges $e_0, e_1, ..., e_{m-1}$ of the ring C while level m corresponds to edge e_0 of C as well. In each of these levels corresponding to the edge e_i, the network has $N = \binom{L}{k}$ nodes; one node per each k-independent set of $\mathcal{I}(P_{e_i})$. For each node u of level $i < m$ corresponding to a k-independent set I, we define the forward set of u to be the set of paths of I which traverse edge $e_{i+1 \bmod m}$. For each node u of level $i > 0$ corresponding to a k-independent set I, we define the backward set of u to be the set of paths of I which traverse edge e_{i-1}. The network $H(P, C)$ has a directed edge from a node u of level i to a node v of level $i+1$ iff the forward set of u is the same with the backward set of v. The network $H(P, C)$ has N commodities. Each node of level 0 is the source of a commodity. The sink for each commodity is located at the node of level m which corresponds to the same k-independent set of $\mathcal{I}(P_{e_0})$ with its source.

RING-k-CHECKER solves the maximum multicommodity flow problem on the network $H(P, C)$ under the constraint that for each path p of P, and for each edge e_i traversed by p, the total flow entering (leaving) all the nodes of $H(P, C)$ of level i corresponding to k-independent sets of $\mathcal{I}(P_{e_i})$ that contain the path p is at most 1. RING-k-CHECKER returns YES if there is a total flow of size L/k. Otherwise, it returns NO.

Lemma 17 *Let $k > 0$ be an integer constant. On input a normal set of paths P of load L which is a multiple of k on a ring C, RING-k-CHECKER runs in polynomial time and returns YES iff P has a fractional k-coloring of cost L/k.*

Proof: First, consider a multicommodity flow of size L/k on the network $H(P, C)$ which satisfies the capacity constraint. We will show that there exists a fractional k-coloring of P of cost L/k.

For each commodity with a non-zero flow f, we decompose the flow f into paths $q_1, q_2, ..., q_s$ each carrying an amount f_j of flow from the source to the sink of the commodity, such that $\sum_{j=1}^{s} f_j = f$. This can be done by the following path stripping technique [16]. Consider the edges E of the network carrying non-zero amount of flow for the specific commodity and consider the edge $e \in E$ carrying the smallest amount of flow f' for the specific commodity. Find a path from the source to the sink of the specific commodity which contains only edges of E and contains edge e. Associate amount of flow f' with this path and subtract amount

f' from the flow for the specific commodity carried by each edge along this path. Repeat this procedure until no flow remains.

Note that a flow path from the source to the sink of some commodity in $H(P, C)$ defines a k-independent set of P in the following way. Consider the union I of the k-independent sets of paths in P to which the nodes of the flow path correspond to. Clearly, I is a k-independent set in P.

Let $\mathcal{I}$ be the set of the k-independent sets in P defined by the flow paths produced after the application of path stripping on all non-zero flows. For each k-independent set I of $\mathcal{I}$, we set the weight $x(I)$ to be equal to the flow carried by the flow path corresponding to I. The multicommodity flow maintains the capacity constraint to equality; otherwise, we can easily verify that the size of the flow would be strictly smaller than L/k. Thus, for any path $p \in P$, the sum of the weights of the k-independent sets in $\mathcal{I}$ that contain p is at most 1 and, hence, x defines a feasible fractional k-coloring of P. Its cost equals the total flow size in the network $H(P, C)$, i.e., L/k.

Now consider a fractional k-coloring x of P of cost L/k. We will show that there exists a multicommodity flow on the network $H(P, C)$ of size L/k.

Observe that $x(I)$ can be non-zero only if I has full load k on every edge of the ring; otherwise, the fractional k-coloring would have cost strictly larger than L/k. Any such k-independent set I of P can be mapped to a unique flow path in $H(P, C)$ defined by those nodes $u_0, u_1, ..., u_n$ at levels $0, 1, ..., n$, respectively, such that node u_i corresponds to the set of k paths of P_{e_i} contained in I. Note that, by the definition of the network $H(P, C)$, the directed edges from u_i to u_{i+1}, for $0 \leq i \leq n-1$ exist, and, thus, the flow path exists in $H(P, C)$.

For each element I_i of $\mathcal{I}(P_{e_0})$, denote by $\mathcal{I}_i$ the set of k-independent sets of P with non-zero weight (under x) such that $I_i \subseteq I$ for every $I \in \mathcal{I}_i$. For each commodity i, we map each k-independent set of $\mathcal{I}_i$ to a flow path from the source to the sink of commodity i in the network $H(P, C)$ and, for each edge e of $H(P, C)$, we set the flow for commodity i carried by e equal to the sum of weights of the k-independent sets of $\mathcal{I}_i$ whose corresponding flow path uses e.

Now observe that it is $\sum_{I \in \mathcal{I}: p \in P} = 1$, for any path $p \in P$; otherwise, the fractional k-coloring x would have cost strictly greater than L/k. Thus, the constructed flow does not violate the capacity constraint. Moreover, its size is equal to the sum of weights of all k-independent sets, i.e., it is equal to the cost of the fractional k-coloring x.

It can be easily seen that the multicommodity flow problem can be formulated as a linear program with polynomial number of variables and constraints, thus, it can be solved in polynomial time. ■

Now consider the application of the augmentation procedure on the original set of paths P of load L on the ring C using RING-k-CHECKER as checker. We denote by $P_{Z-\lceil L/k \rceil}$ the normal set of paths of load kZ (where Z is an integer) produced when the augmentation procedure terminates. The following statement immediately follows by the definition of the augmentation procedure and by Lemma 17.

Lemma 18 $Z = \lceil f_k(P, C) \rceil$.

When the augmentation procedure terminates, we use the solution to the multicommodity flow problem on the network $H(P_{Z-\lceil L/k \rceil}, C)$ and apply the same procedure we used in the first part of the proof of Lemma 17 to obtain a fractional k-coloring. I.e., we apply the path stripping technique to decompose the flow for each commodity on $H(P_{Z-\lceil L/k \rceil}, C)$ into flow paths, we map the flow paths into k-independent sets of $P_{Z-\lceil L/k \rceil}$, and we assign to each of these k-independent sets I weight $x(I)$ equal to the flow carried by the corresponding flow path. Since x is a feasible fractional k-coloring of $P_{Z-\lceil L/k \rceil}$, by applying the technique we used in the proof of Lemma 4, we obtain a family of fractional k-coloring functions y_j for $P_{Z-\lceil L/k \rceil}$.

Note that, for each j, the number of the k-independent sets I for which $y_j(I) > 0$ is polynomial. This is due to the fact that the number of flow paths in which the flow of each commodity is decomposed into is upper-bounded by the number of edges in the multicommodity flow network. We obtain the following lemma.

Lemma 19 *For each $j = 1, ..., Z$, die-casting according to function y_j can be done in polynomial time.*

References

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

[2] B. Beauquier, J.-C. Bermond, L. Gargano, P. Hell, S. Perennes, U. Vaccaro. Graph Problems arising from Wavelength-Routing in All-Optical Networks. In *Proc. of the 2nd Workshop on Optics and Computer Science (WOCS '97)*, 1997.

[3] I. Caragiannis, A. Ferreira, C. Kaklamanis, S. Perennes, 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, pp. 732–743, 2001.

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

[5] 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), pp. 119-137, 1999.

[6] M.R. Garey, D.S. Johnson, G.L. Miller, and C.H. Papadimitriou. The Complexity of Coloring Circular Arcs and Chords. *SIAM Journal on Alg. Disc. Math.*, Vol. 1(2), pp. 216-227, 1980.

[7] M. Grötschel, L. Lovász, and A. Schrijver. The Ellipsoid Method and its Consequences in Combinatorial Optimization. *Combinatorica*, Vol. 1, pp. 169-197, 1981.

[8] K. Jansen. Approximate Strong Separation with Application in Fractional Graph Coloring and Preemptive Scheduling. *Theoretical Computer Science*, Vol. 302(1-3), pp. 239-256, 2003.

[9] K. Jansen and L. Porkolab. On Preemptive Resource Constrained Scheduling: Polynomial-Time Approximation Schemes. In *Proc. of the 9th International Conference on Integer Programming and Combinatorial Optimization (IPCO '02)*, LNCS 2337, Springer, pp. 329-349, 2002.

[10] V. Kumar. An Approximation Algorithm for Circular Arc Coloring. *Algorithmica*, Vol. 30(3), pp. 406-417, 2001.

[11] G. Li and R. Simha. On the Wavelength Assignment Problem on Multifiber WDM Star and Ring Networks. *IEEE/ACM Transactions on Networking*, Vol. 9(1), pp. 60-68, 2001.

[12] G. Li and R. Simha. On Bounds for the Wavelength Assignment Problem on Optical Ring Networks. *Journal of High-Speed Networks*, Vol. 8(4), pp. 303-309, 1999.

[13] L. Lovász. On the Ratio of Optimal Integral and Fractional Covers. *Discrete Mathematics*, Vol. 13, pp. 383-390, 1975.

[14] L. Margara and J. Simon. Wavelength Assignment Problem on All–Optical Networks with k Fibers per Link. In *Proc. of the 27th International Colloquium on Automata, Languages, and Programming (ICALP '00)*, LNCS 1853, Springer, pp. 768–779, 2000.

[15] R. Motwani, P. Raghavan. Randomized Algorithms. *Cambridge University Press*, 1995.

[16] P. Raghavan. Randomized Rounding and Discrete Ham–Sandwitches: Provably Good Algorithms for Routing and Packing Problems. *PhD Thesis*, CS Division, UC Berkeley, 1986.

[17] P. Raghavan and C.D. Thompson. Randomized Rounding: A Technique for Provably Good Algorithms and Algorithmic Proofs. *Combinatorica*, Vol. 7, pp. 365–374, 1987.

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

[19] A. Tucker. Coloring a Family of Circular Arcs. *SIAM Journal of Applied Mathematics*, Vol. 29(3), pp. 493–502, 1975.

[20] M.E. Valencia–Pabon. Revisiting Tucker's Algorithm to Color Circular–Arc Graphs. *Electronic Notes on Discrete Mathematics*, 2000.

A Proof of Lemma 1

a. The probability that bin ℓ does not receive a copy of ball i in a round executed for ball i is $1-\sum_{j:\ell\in s_j} p_{ij}$. After the execution of the $\lfloor q \rfloor$-th round for ball i, the probability that bin ℓ does not receive any other copy of ball i is 1 if the $(\lfloor q \rfloor+1)$-th round is not executed for ball i (this happens with probability $1-q+\lfloor q \rfloor$) and $1-\sum_{j:\ell\in s_j} p_{ij}$ otherwise. Thus, the probability that bin ℓ does not receive a copy of ball i after the execution of the $\lfloor q \rfloor$-th round is $1-(q-\lfloor q \rfloor)\sum_{j:\ell\in s_j} p_{ij}$. Thus, the probability that bin ℓ is empty is

$$\prod_{i=1}^{n/k}\left(1-\sum_{j:\ell\in s_j} p_{ij}\right)^{\lfloor q \rfloor}\left(1-(q-\lfloor q \rfloor)\sum_{j:\ell\in s_j} p_{ij}\right).$$

It can be easily seen that, under the constraint $\sum_{i=1}^{n/k}\sum_{j:\ell\in s_j} p_{ij}=1$, the above product is maximized for $\sum_{j:\ell\in s_j} p_{ij}=k/n, i=1,...,n/k$ to

$$(1-k/n)^{n\lfloor q \rfloor/k}\,(1-(q-\lfloor q \rfloor)k/n)^{n/k}\leq e^{-\lfloor q \rfloor}e^{-q+\lfloor q \rfloor}=e^{-q}.$$

Thus, the expectation of the number of empty bins $\mathcal{Q}$ is

$$E[\mathcal{Q}]=\sum_{\ell=1}^{n}\Pr[\text{bin } \ell \text{ is empty}]\leq ne^{-q}.$$

b. For proving the tail bound for random variable $\mathcal{Q}$, we will use basic concepts from the theory of martingales (see [15], Chapter 4.4, pages 83-96). We may view the balls-to-bins process as executed in $\lceil q \rceil n/k$ steps, $\lceil q \rceil$ steps associated with each of the n/k balls. In $\lfloor q \rfloor$ of the steps associated with a ball, a round is executed while, if q is not integer, either a round is executed or nothing happens at the $\lceil q \rceil$-th step. Let's denote by Ψ_i the random variable denoting the set of bins chosen for throwing copies of a ball at step i. This can be either a set of bins of size k or the empty set. We can view the number of empty bins $\mathcal{Q}$ as a function $F(\Psi_1,...,\Psi_{\lceil q \rceil n/k})$. Now, consider the Doob martingale $Y_0,Y_1,...,Y_{\lceil q \rceil n/k}$ defined by setting $Y_0=E[F(\Psi_1,...,\Psi_{\lceil q \rceil n/k})]$, and $Y_i=E[F(\Psi_1,...,\Psi_{\lceil q \rceil n/k})|\Psi_1,...,\Psi_i]$. By the definition of the Doob martingale, it is $\mathcal{Q}=Y_{\lceil q \rceil n/k}$ and $E[\mathcal{Q}]=Y_0$. Note that $|Y_i-Y_{i-1}|\leq k$ since changing any one variable Ψ_i can only affect the expected value of $F(\Psi_1,...,\Psi_{\lceil q \rceil n/k})$ by at most k. By applying Azuma's inequality (see [15], Theorem 4.16, page 92) we obtain that

$$\Pr[|Y_{\lceil q \rceil n/k}-Y_0|\geq\lambda]\leq 2\exp\left(-\frac{\lambda^2}{2\lceil q \rceil nk}\right).$$

Since $Y_{\lceil q \rceil n/k}=\mathcal{Q}$ and $Y_0=E[\mathcal{Q}]$, we obtain the desired result.

c. For $i=1,...,n/k$, denote by X_i the Bernoulli random variable indicating whether the $(\lfloor q \rfloor+1)$-th round is executed for ball i. Clearly, the random variable $X=\sum_{i=1}^{n/k}X_i$ denotes the number of balls for which the $(\lfloor q \rfloor+1)$-th round is executed and, hence, $\mathcal{R}=\lfloor q \rfloor n/k+X$. Since $\Pr[X_i=1]=q-\lfloor q \rfloor$ for $i=1,...,n/k$, the expectation of X is $E[X]=(q-\lfloor q \rfloor)n/k$ and $E[\mathcal{R}]=\lfloor q \rfloor n/k+E[X]=qn/k$.

d. Observe that $\Pr[\mathcal{R}-E[\mathcal{R}]>\lambda]=\Pr[X-E[X]>\lambda]$. Since X is the sum of Bernoulli random variables, using a Chernoff bound (see [16] and [15], Theorem 4.3, page 72) we obtain that

$$\Pr[X>(1+\delta)E[X]]\leq\exp\left(-\frac{E[X]\delta^2}{4}\right).$$

Substituting $E[X] = (q - \lfloor q \rfloor)n/k$ and setting $\delta = \frac{\lambda}{E[X]}$ we obtain the desired result. ■

B Proof of Lemma 4

We will construct a set of non-negative weight functions y_j, for $j = 1, ..., Z$, on the k-independent sets of P. Let $\mathcal{I}$ be the set of all k-independent sets of P. We arbitrarily index the k-independent sets of $\mathcal{I}$ and implicitly set $y_j(I_i) = 0$, for $j = 1, ..., Z$ and any $I_i \in \mathcal{I}$. We first find integer t_1 such that

$$\sum_{i=1}^{t_1} x(I_i) \geq 1 > \sum_{i=1}^{t_1-1} x(I_i).$$

We set $y_1(I_i) = x(I_i)$, for $i = 1, ..., t_1 - 1$ and $y_1(I_{t_1}) = 1 - \sum_{i=1}^{t_1-1} x(I_i)$. Then, for $j = 2, ..., Z-1$ we do the following:

- If $x(I_{t_{j-1}}) > y_{j-1}(I_{t_{j-1}})$ we set $y_j(I_{t_{j-1}}) = x(I_{t_{j-1}}) - y_{j-1}(I_{t_{j-1}})$, otherwise we set $y_j(I_{t_{j-1}}) = 0$.
- We find integer t_j such that

$$y_j(I_{t_{j-1}}) + \sum_{i=t_{j-1}+1}^{t_j} x(I_i) \geq 1 > \sum_{i=t_{j-1}+1}^{t_j-1} x(I_i).$$

  We set $y_j(I_i) = x(I_i)$, for $i = t_{j-1} + 1, ..., t_j - 1$ and $y_j(I_{t_j}) = 1 - \sum_{i=t_{j-1}+1}^{t_j-1} x(I_i) - y_j(I_{t_{j-1}})$.

If $x(I_{t_{Z-1}}) > y_{Z-1}(I_{t_{Z-1}})$ we set $y_Z(I_{t_{Z-1}}) = x(I_{t_{Z-1}}) - y_{Z-1}(I_{t_{Z-1}})$; otherwise, we set $y_Z(I_{t_{Z-1}}) = 0$. Finally, we set $y_Z(I_i) = x(I_i)$ for $i \geq t_{Z-1} + 1$.

It is straightforward to verify that our construction is such that $\sum_{I \in \mathcal{I}} y_j(I) = 1$ for $i = 1, ..., Z$ and $\sum_{i=1}^{Z} y_j(I) = x(I)$ for $I \in \mathcal{I}$. We will complete the proof by showing that $\sum_{I \in \mathcal{I}: p \in I} x(I) = 1$ for any path $p \in P$.

Assume that there is a path $p' \in P$ such that $\sum_{I \in \mathcal{I}: p' \in I} x(I) > 1$. Let e be an edge traversed by p'. Since P has load kZ on e, it is

$$\sum_{p \in P: e \in p} \sum_{I \in \mathcal{I}: p \in I} x(I) > kZ.$$

Also, it is

$$\sum_{p \in P: e \in p} \sum_{I \in \mathcal{I}: p \in I} x(I) \leq k \sum_{I \in \mathcal{I}} x(I)$$

since the term $x(I)$ for any k-independent set I appears at most k times in the left sum. We obtain that $\sum_{I \in \mathcal{I}} x(I) > Z$, which contradicts the assumption of the Lemma.

We conclude that $\sum_{I \in \mathcal{I}: p \in I} x(I) = 1$ for any path $p \in P$ and, furthermore, that

$$\sum_{p \in P: e \in p} \sum_{I \in \mathcal{I}: p \in I} x(I) = k \sum_{I \in \mathcal{I}} x(I)$$

for each edge e, which means that $x(I)$ is non-zero only for those k-independent sets $I \in \mathcal{I}$ having load k on each edge. These are two important properties of optimal fractional k-colorings on normal sets of paths and we will use them several times in the rest of the paper. ■

C Proof of Lemma 5

Let $\mathcal{I}$ be the set of k-independent sets of P. Consider an optimal fractional k-coloring x of P of cost $f_k(P,G)$. We first increase the cost of x to $1+\lceil f_k(P,G)\rceil$ by setting $x(\emptyset) = 1+\lceil f_k(P,G)\rceil - f_k(P,G)$. Without loss of generality, we may assume that, for any path $p \in P$, it is $\sum_{I\in\mathcal{I}:p\in I} x(I) = 1$. If this is not the case for some path p, then we can always decrease $\sum_{I\in\mathcal{I}:p\in I} x(I)$ by decreasing $x(I)$ and increasing $x(I\backslash p)$ by the same amount, for some independent set that contains p.

We arbitrarily index the edges $e_1,...,e_m$ of G and perform the following procedure. Starting with the original set of paths P, we visit the edges $e_1,...,e_m$ one by one and, when visiting edge e_i, we add single-hop paths on e_i so that the resulting set of paths P_i has load $k(1+\lceil f_k(P,G)\rceil)$ on edge e_i, and we construct a fractional k-coloring x_i for P_i of cost $1+\lceil f_k(P,G)\rceil$.

We first show how to do this when visiting edge e_1. Let L be the load of P on the edge e_1. For $i=0,1,...,k$, we denote by $\mathcal{I}^i_{e_1}$ the subset of $\mathcal{I}$ that consists of the k-independent sets of P having i paths that traverse edge e_1. We add $k(1+\lceil f_k(P,G)\rceil)-L$ single-hop paths on edge e_1. In this way, we obtain the set of paths P_1 which has load $k(1+\lceil f_k(P,G)\rceil)$ on e_1. For $i=1,...,k$, we denote by $\mathcal{J}^i_{e_1}$ the set of all sets of i new single-hop paths. Observe that since we have added $k(1+\lceil f_k(P,G)\rceil)-L \geq k(1+\lceil f_k(P,G)\rceil)-kf_k(P,G) \geq k$ single-hop paths on edge e_1, the sets $\mathcal{J}^i_{e_1}$ are not empty, for $i=1,...,k$. We also denote by $\mathcal{J}$ the set of all k-independent sets of P_1.

We construct a fractional k-coloring x_1 of P_1 as follows. For each k-independent set I of $\mathcal{I}^k_{e_1}$, we set $x_1(I)=x(I)$. For $i=0,...,k-1$, for each $I_1\in\mathcal{I}^i_{e_1}$ and each $I_2\in\mathcal{J}^{k-i}_{e_1}$, we set $x_1(I_1\cup I_2)=\frac{x(I_1)}{|\mathcal{J}^{k-i}_{e_1}|}$. We set $x_1(I)=0$ for all other k-independent sets of $\mathcal{J}$.

We will show that x_1 is a feasible fractional k-coloring of P_1, i.e., we will show that, for any path $p\in P_1$, it is $\sum_{I\in\mathcal{J}:p\in I} x_1(I)=1$. Let p be a path of P_1 which also belongs to P. It is

$$\begin{aligned}
\sum_{I\in\mathcal{J}:p\in I} x_1(I) &= \sum_{I\in\mathcal{I}^k_{e_1}:p\in I} x_1(I) + \sum_{i=0}^{k-1}\sum_{I\in\mathcal{I}^i_{e_1}:p\in I}\sum_{I'\in\mathcal{J}^{k-i}_{e_1}} x_1(I\cup I') \\
&= \sum_{I\in\mathcal{I}^k_{e_1}:p\in I} x(I) + \sum_{i=0}^{k-1}\sum_{I\in\mathcal{I}^i_{e_1}:p\in I}\sum_{I'\in\mathcal{J}^{k-i}_{e_1}} \frac{x(I)}{|\mathcal{J}^{k-i}_{e_1}|} \\
&= \sum_{I\in\mathcal{I}^k_{e_1}:p\in I} x(I) + \sum_{i=0}^{k-1}\sum_{I\in\mathcal{I}^i_{e_1}:p\in I} x(I) \\
&= \sum_{I\in\mathcal{I}:p\in I} x(I) \\
&= 1.
\end{aligned}$$

Let $p\in P_1\backslash P$. Observe that for $i=1,...,k$, it is $|\mathcal{J}^i_{e_1}| = \binom{k(1+\lceil f_k(P,G)\rceil)-L}{i}$, while the number of sets of $\mathcal{J}^i_{e_1}$ that contain p is $\binom{k(1+\lceil f_k(P,G)\rceil)-L-1}{i-1}$. We have that

$$\sum_{I\in\mathcal{J}:p\in I} x_1(I) = \sum_{i=0}^{k-1}\sum_{I\in\mathcal{I}^i_{e_1}}\sum_{I'\in\mathcal{J}^{k-i}_{e_1}:p\in I'} x(I\cup I')$$

$$
\begin{aligned}
&= \sum_{i=0}^{k-1} \sum_{I \in \mathcal{I}_{e_1}^i} \sum_{I' \in \mathcal{J}_{e_1}^{k-i} : p \in I'} \frac{x(I)}{|\mathcal{J}_{e_1}^{k-i}|} \\
&= \sum_{i=0}^{k-1} \sum_{I \in \mathcal{I}_{e_1}^i} \binom{k(1+\lceil f_k(P,G)\rceil)-L-1}{k-i-1} \frac{x(I)}{\binom{k(1+\lceil f_k(P,G)\rceil)-L}{k-i}} \\
&= \sum_{i=0}^{k-1} \sum_{I \in \mathcal{I}_{e_1}^i} \frac{(k-i)x(I)}{k(1+\lceil f_k(P,G)\rceil)-L} \\
&= \frac{k}{k(1+\lceil f_k(P,G)\rceil)-L} \sum_{i=0}^{k} \sum_{I \in \mathcal{I}_{e_1}^i} x(I) - \frac{1}{k(1+\lceil f_k(P,G)\rceil)-L} \sum_{i=0}^{k} \sum_{I \in \mathcal{I}_{e_1}^i} i x(I) \\
&= \frac{k}{k(1+\lceil f_k(P,G)\rceil)-L} \sum_{I \in \mathcal{I}} x(I) - \frac{1}{k(1+\lceil f_k(P,G)\rceil)-L} \sum_{p' \in P : e_1 \in p'} \sum_{i=0}^{k} \sum_{I \in \mathcal{I}_{e_1}^i : p' \in I} x(I) \\
&= \frac{k(1+\lceil f_k(P,G)\rceil)}{k(1+\lceil f_k(P,G)\rceil)-L} - \frac{1}{k(1+\lceil f_k(P,G)\rceil)-L} \sum_{p' \in P : e_1 \in p'} \sum_{I \in \mathcal{I} : p' \in I} x(I) \\
&= \frac{k(1+\lceil f_k(P,G)\rceil)}{k(1+\lceil f_k(P,G)\rceil)-L} - \frac{L}{k(1+\lceil f_k(P,G)\rceil)-L} \\
&= 1.
\end{aligned}
$$

The cost of the fractional k-coloring x_1 is

$$
\begin{aligned}
\sum_{I \in \mathcal{J}} x_1(I) &= \sum_{I \in \mathcal{I}_{e_1}^k} x_1(I) + \sum_{i=0}^{k-1} \sum_{I \in \mathcal{I}_{e_1}^i} \sum_{I' \in \mathcal{J}_{e_1}^{k-i}} x_1(I \cup I') \\
&= \sum_{I \in \mathcal{I}_{e_1}^k} x(I) + \sum_{i=0}^{k-1} \sum_{I \in \mathcal{I}_{e_1}^i} \sum_{I' \in \mathcal{J}_{e_1}^{k-i}} \frac{x(I)}{|\mathcal{J}_{e_1}^{k-i}|} \\
&= \sum_{I \in \mathcal{I}_{e_1}^k} x(I) + \sum_{i=0}^{k-1} \sum_{I \in \mathcal{I}_{e_1}^i} x(I) \\
&= \sum_{I \in \mathcal{I}} x(I) \\
&= 1 + \lceil f_k(P,G)\rceil.
\end{aligned}
$$

When visiting edge e_i (for $i \geq 2$), we add single-hop paths on edge e_i so that the load on e_i becomes $k(1+\lceil f_k(P,G)\rceil)$. In this way, we obtain the set of paths P_i. A fractional k-coloring x_i for P_i of cost $1+\lceil f_k(P,G)\rceil$ can be obtained by the fractional k-coloring x_{i-1} of P_{i-1} in the same way we obtained x_1 from x (now we start by setting $x_{i-1}(\emptyset) = 0$). After visiting the last edge e_m, we obtain the normal set of paths $P' = P_m$ of load $k(1+\lceil f_k(P,G)\rceil)$ which has the fractional k-coloring x_m of cost $1+\lceil f_k(P,G)\rceil$. Since the load of P' is $k(1+\lceil f_k(P,G)\rceil)$, $1+\lceil f_k(P,G)\rceil$ is a lower bound for $f_k(P',G)$. Thus, we conclude that $f_k(P',G) = 1+\lceil f_k(P,G)\rceil$. ■

D Proof of Lemma 14

We denote by $\mathcal{I}$ the set of all k-independent sets of P. Let y be a fractional k-coloring of P of cost L/k. It can be easily seen that $y(I)$ can be non-zero only for k-independent sets having load k on each directed edge of the tree; otherwise, the fractional k-coloring would have cost strictly larger than L/k.

We construct a solution x to the linear program as follows. For each non-leaf node v and each k-independent set I_v of $\mathcal{I}(P_v)$, we set $x(I_v)$ to be equal to the sum of the weights of the k-independent sets of $\mathcal{I}$ that contain all paths of I_v, i.e., $x(I_v) = \sum_{I \in \mathcal{I}: I \cap I_v = I_v} y(I)$. Clearly, the cost of the solution x is equal to the sum of the weights of the k-independent sets of $\mathcal{I}$, i.e., it is L/k. We will also show that this solution is feasible.

For each path $p \in P$, the sum of the weights of the k-independent sets containing p is exactly 1; otherwise, the fractional k-coloring would have cost strictly larger than L/k. Thus, for any path $p \in P$ and any non-leaf node v, it is $\sum_{I \in \mathcal{I}(P_v): p \in I} x(I) = 1$, and constraints of the first type are maintained.

Constraints of the second type are maintained since, for any pair of adjacent non-leaf nodes v and u of the tree, any set of k paths $p_1, ..., p_k$ traversing the directed edge (v, u) and any set of k paths $q_1, ..., q_k$ traversing the directed edge (u, v), both the sum of the weights of the k-independent sets of $\mathcal{I}(P_u)$ that contain $p_1, ..., p_k, q_1, ..., q_k$ and the sum of the weights of the k-independent sets of $\mathcal{I}(P_v)$ that contain $p_1, ..., p_k, q_1, ..., q_k$ are equal to the sum of the weights under y of the k-independent sets of $\mathcal{I}$ containing $p_1, ..., p_k, q_1, ..., q_k$.

Now consider a solution x to the linear program of cost L/k. We will construct a fractional k-coloring of P of the same cost.

Let I be a k-independent set in $\mathcal{I}$. For any non-leaf node v of T, we denote by $J(I, v)$ the subset of I which contains the paths of I touching node v. Now, starting from node r, consider a BFS (or DFS) ordering $v_0 = r, v_1, v_2, ..., v_{n-1}$ of the n non-leaf nodes of T. For a non-leaf node of T different than r, let $Q(I, v)$ be the set of paths in $J(I, v)$ which traverse the directed edges between v and its parent. We set

$$y(I) = x(J(I, v_0)) \prod_{i=1}^{n-1} \frac{x(J(I, v_i))}{\sum_{I' \in \mathcal{I}(P_{v_i}): Q(I', v_i) \cap I' = Q(I, v_i)} x(I')}$$

if I has full load k on each directed edge of T and $y(I) = 0$ otherwise.

For $i = 0, ..., n-1$, denote by $\mathcal{I}_i$ the subset of $\mathcal{I}$ consisting of the k-independent sets which contain only paths touching nodes $v_0, ..., v_i$ and have full load k on each directed edge adjacent to the nodes $v_0, ..., v_i$. It is $\mathcal{I}_0 = \mathcal{I}(P_r)$ while $\mathcal{I}_{n-1}$ is the set of those k-independent sets I for which $y(I)$ was not explicitly set to zero. We first prove the following technical claim.

Claim 20

$$\begin{aligned} &\sum_{I \in \mathcal{I}_t} x(J(I, v_0)) \prod_{i=1}^{t} \frac{x(J(I, v_i))}{\sum_{I' \in \mathcal{I}(P_{v_i}): Q(I, v_i) \cap I' = Q(I, v_i)} x(I')} \\ = &\sum_{I \in \mathcal{I}_{t-1}} x(J(I, v_0)) \prod_{i=1}^{t-1} \frac{x(J(I, v_i))}{\sum_{I' \in \mathcal{I}(P_{v_i}): Q(I, v_i) \cap I' = Q(I, v_i)} x(I')} \end{aligned}$$

for $t = 2, ..., n-1$ *and*

$$\sum_{I \in \mathcal{I}_1} x(J(I, v_0)) \frac{x(J(I, v_1))}{\sum_{I' \in \mathcal{I}(P_{v_1}): Q(I, v_1) \cap I' = Q(I, v_1)} x(I')} = \sum_{I \in \mathcal{I}_0} x(J(I, v_0)).$$

Proof: Given a k-independent set I of $\mathcal{I}_{t-1}$ for some t such that $1 \le t \le n-1$, we denote by $\mathcal{I}_t^I$ the subset of $\mathcal{I}_t$ that consists of those k-independent sets that contain all paths in I, i.e., $\mathcal{I}_t$ is the disjoint union of the sets $\mathcal{I}_t^I$, for any $I \in \mathcal{I}_{t-1}$. For $t = 2, ..., n-1$, it is

$$\begin{aligned}
&\sum_{I \in \mathcal{I}_t} x(J(I, v_0)) \prod_{i=1}^{t} \frac{x(J(I, v_i))}{\sum_{I' \in \mathcal{I}(P_{v_i}): Q(I, v_i) \cap I' = Q(I, v_i)} x(I')} \\
= &\sum_{I \in \mathcal{I}_{t-1}} \sum_{I'' \in \mathcal{I}_t^I} x(J(I'', v_0)) \prod_{i=1}^{t} \frac{x(J(I'', v_i))}{\sum_{I' \in \mathcal{I}(P_{v_i}): Q(I'', v_i) \cap I' = Q(I'', v_i)} x(I')} \\
= &\sum_{I \in \mathcal{I}_{t-1}} \sum_{I'' \in \mathcal{I}_t^I} x(J(I, v_0)) \prod_{i=1}^{t-1} \frac{x(J(I, v_i))}{\sum_{I' \in \mathcal{I}(P_{v_i}): Q(I, v_i) \cap I' = Q(I, v_i)} x(I')} \frac{x(J(I'', v_t))}{\sum_{I' \in \mathcal{I}(P_{v_t}): Q(I, v_t) \cap I' = Q(I, v_t)} x(I')} \\
= &\sum_{I \in \mathcal{I}_{t-1}} x(J(I, v_0)) \prod_{i=1}^{t-1} \frac{x(J(I, v_i))}{\sum_{I' \in \mathcal{I}(P_{v_i}): Q(I, v_i) \cap I' = Q(I, v_i)} x(I')} \sum_{I'' \in \mathcal{I}_t^I} \frac{x(J(I'', v_t))}{\sum_{I' \in \mathcal{I}(P_{v_t}): Q(I, v_t) \cap I' = Q(I, v_t)} x(I')} \\
= &\sum_{I \in \mathcal{I}_{t-1}} x(J(I, v_0)) \prod_{i=1}^{t-1} \frac{x(J(I, v_i))}{\sum_{I' \in \mathcal{I}(P_{v_i}): Q(I, v_i) \cap I' = Q(I, v_i)} x(I')}
\end{aligned}$$

where the last equality holds since, by the definitions of Q and J, the sums $\sum_{I'' \in \mathcal{I}_t^I} x(J(I'', v_t))$ and $\sum_{I' \in \mathcal{I}(P_{v_i}): Q(I, v_i) \cap I' = Q(I, v_i)} x(I')$ are the same. The second equality can be proved in the same way. ■

Now, the cost of the fractional k-coloring solution y can be expressed as

$$\begin{aligned}
\sum_{I \in \mathcal{I}} y(I) &= \sum_{I \in \mathcal{I}_{n-1}} x(J(I, v_0)) \prod_{i=1}^{n-1} \frac{x(J(I, v_i))}{\sum_{I' \in \mathcal{I}(P_{v_i}): Q(I, v_i) \cap I' = Q(I, v_i)} x(I')} \\
&= \sum_{I \in \mathcal{I}_{n-2}} x(J(I, v_0)) \prod_{i=1}^{n-2} \frac{x(J(I, v_i))}{\sum_{I' \in \mathcal{I}(P_{v_i}): Q(I, v_i) \cap I' = Q(I, v_i)} x(I')} \\
&= \ldots \\
&= \sum_{I \in \mathcal{I}_1} x(J(I, v_0)) \frac{x(J(I, v_1))}{\sum_{I' \in \mathcal{I}(P_{v_1}): Q(I, v_1) \cap I' = Q(I, v_1)} x(I')} \\
&= \sum_{I \in \mathcal{I}_0} x(J(I, v_0)) \\
&= \sum_{I \in \mathcal{I}(P_r)} x(I) \\
&= L/k.
\end{aligned}$$

It remains to show that y is a feasible fractional k-coloring. Let p^* be a path of P and u a non-leaf node it touches. It is $\sum_{I \in \mathcal{I}(P_u): p^* \in I} x(I) = 1$; otherwise, due to the constraints of the second type, we would obtain that the cost of the solution of the linear program is strictly smaller than L/k. Starting from node u, consider a BFS (or DFS) ordering $v_{\pi(0)} = u, v_{\pi(1)}, ..., v_{\pi(n-1)}$ of the non-leaf nodes of T. Due to the constraints of the second type in the linear program, $y(I)$ can be written as

$$y(I) = x(J(I, v_{\pi(0)})) \prod_{i=1}^{n-1} \frac{x(J(I, v_{\pi(i)}))}{\sum_{I' \in \mathcal{I}(P_{v_{\pi(i)}}): Q(I, v_{\pi(i)}) \cap I' = Q(I, v_{\pi(i)})} x(I')}$$

if I has full load k on each directed edge of T.

Following similar reasoning with the one we used to compute the cost of y, we will show that $\sum_{I \in \mathcal{I}: p^* \in I} y(I) = 1$. For $i = 0, ..., n-1$, denote by $\mathcal{J}_i$ the subset of $\mathcal{I}$ consisted of the k-independent sets which contain only paths touching nodes $v_{\pi(0)}, ..., v_{\pi(i)}$ and have full load k on each directed edge adjacent to the nodes $v_{\pi(0)}, ..., v_{\pi(i)}$. It is $\mathcal{J}_0 = \mathcal{I}(P_u)$ while $\mathcal{J}_{n-1}$ is the set of those k-independent sets I for which $y(I)$ was not explicitly set to zero. The following claim can be proved in the same way we proved Claim 20.

Claim 21

$$\begin{aligned}
&\sum_{I \in \mathcal{J}_t: p^* \in I} x(J(I, v_{\pi(0)})) \prod_{i=1}^{t} \frac{x(J(I, v_{\pi(i)}))}{\sum_{I' \in \mathcal{I}(P_{v_{\pi(i)}}): Q(I, v_{\pi(i)}) \cap I' = Q(I, v_{\pi(i)})} x(I')} \\
= &\sum_{I \in \mathcal{J}_{t-1}: p^* \in I} x(J(I, v_{\pi(0)})) \prod_{i=1}^{t-1} \frac{x(J(I, v_{\pi(i)}))}{\sum_{I' \in \mathcal{I}(P_{v_{\pi(i)}}): Q(I, v_{\pi(i)}) \cap I' = Q(I, v_{\pi(i)})} x(I')}
\end{aligned}$$

for $t = 2, ..., n-1$ *and*

$$\sum_{I \in \mathcal{I}_1: p^* \in I} x(J(I, v_{\pi(0)})) \frac{x(J(I, v_{\pi(1)}))}{\sum_{I' \in \mathcal{I}(P_{v_{\pi(1)}}): Q(I, v_{\pi(1)}) \cap I' = Q(I, v_{\pi(1)})} x(I')} = \sum_{I \in \mathcal{J}_0: p^* \in I} x(J(I, v_{\pi(0)})).$$

■

Now, $\sum_{I \in \mathcal{I}: p^* \in I} y(I)$ can be expressed as

$$\begin{aligned}
\sum_{I \in \mathcal{I}} y(I) &= \sum_{I \in \mathcal{J}_{n-1}: p^* \in I} x(J(I, v_0)) \prod_{i=1}^{n-1} \frac{x(J(I, v_i))}{\sum_{I' \in \mathcal{I}(P_{v_i}): Q(I, v_i) \cap I' = Q(I, v_i)} x(I')} \\
&= \sum_{I \in \mathcal{J}_{n-2}: p^* \in I} x(J(I, v_0)) \prod_{i=1}^{n-2} \frac{x(J(I, v_i))}{\sum_{I' \in \mathcal{I}(P_{v_i}): Q(I, v_i) \cap I' = Q(I, v_i)} x(I')} \\
&= \dots \\
&= \sum_{I \in \mathcal{J}_1: p^* \in I} x(J(I, v_0)) \frac{x(J(I, v_1))}{\sum_{I' \in \mathcal{I}(P_{v_1}): Q(I, v_1) \cap I' = Q(I, v_1)} x(I')} \\
&= \sum_{I \in \mathcal{J}_0: p^* \in I} x(J(I, v_0)) \\
&= \sum_{I \in \mathcal{I}(P_u): p^* \in I} x(I) \\
&= 1.
\end{aligned}$$

To prove that TREE-k-CHECKER runs in polynomial time, we will show that the linear program has polynomial number of variables and constraints. First observe that a k-independent set of $\mathcal{I}(P_u)$ can have from $k\Delta$ to $2k\Delta$ paths, where Δ is the degree of the non-leaf node u. Moreover, the number of paths of P touching node u is at most $2\Delta L$. Thus, $|\mathcal{I}(P_u)| \leq \sum_{i=k\Delta}^{2k\Delta} \binom{2\Delta L}{i} = poly(L)$, since both Δ and k are constants. Hence, the number of variables is $poly(L, m)$. The number of constraints of the first type is at most the number of paths times the number of edges which is smaller than mL. Also, for each pair of opposite directed edges between two non-leaf nodes u and v, there are $\binom{L}{k}^2 = poly(L)$ distinct sets of paths containing k paths traversing edge (u, v) and k paths traversing edge (v, u). Hence, the number of constraints of the second type is $poly(L, m)$. ■

E Proof of Lemma 16

Let x be a solution to the linear program of cost Z. First, we use $x(I)$ for $I \in \mathcal{I}(P_r)$ to compute the family x_j of fractional k-coloring functions for P_r by applying Lemma 4. Then, for each $I \in \mathcal{I}$ such that I has load k in each directed edge of T, we set

$$y_j(I) = x_j(J(I, v_0)) \prod_{i=1}^{n-1} \frac{x(J(I, v_i))}{\sum_{I' \in \mathcal{I}(P_{v_i}): Q(I,v_i) \cap I' = Q(I,v_i)} x(I')}.$$

Using technical claims similar in spirit with Claims 20 and 21 used in the proof of Lemma 14, we can show that $\sum_{I \in \mathcal{I}: p \in I} \sum_{j=1}^{Z} y_j(I) = 1$ for any path $p \in P_{Z-\lceil L/k \rceil}$ and $\sum_{I \in \mathcal{I}} y_j(I) = 1$ for $j = 1, ..., Z$, which means that y_j, for $j = 1, ..., Z$, is a family of fractional k-coloring functions for $P_{Z-\lceil L/k \rceil}$. Now, we will show that die-casting according to y_j for some $j = 1, ..., Z$ can be simulated by casting a polynomial number of dice each having a polynomial number of faces.

In particular, we will describe an algorithm which visits the non-leaf nodes following their BFS numbering, and constructs a k-independent set F incrementally. Initially, $F = \emptyset$. At the step associated with node r, the algorithm casts a die which has one face for each k-independent set of $\mathcal{I}(P_r)$ and probability $x_j(I)$ associated with the face corresponding to the k-independent set $I \in \mathcal{I}(P_r)$. The k-independent set F is augmented to the k-independent set selected when casting the die. At the step associated with the non-leaf node v_i, we denote by $Q(F, v_i)$ the subset of F which consists of the $2k$ paths traversing the opposite directed edges between v_i and its parent (k paths per direction). We cast a die which has one face for each k-independent set of $\mathcal{I}(P_v)$ which contains all paths in $Q(F, v_i)$ and probability $\frac{x(I)}{\sum_{I' \in \mathcal{I}(P_{v_i}): Q(F,v_i) \cap I' = Q(F,v_i)} x(I')}$ associated with the face corresponding to the k-independent set I of $\mathcal{I}(P_{v_i})$ that contains all paths in $Q(F, v_i)$. Let F' be the k-independent set selected after casting the die. The k-independent set F is augmented to $F \cup F'$.

The probability that a specific k-independent set $I \in \mathcal{I}$ is constructed by the algorithm is indeed

$$x_j(J(I, v_0)) \prod_{i=1}^{n-1} \frac{x(J(I, v_i))}{\sum_{I' \in \mathcal{I}(P_{v_i}): Q(I,v_i) \cap I' = Q(I,v_i)} x(I')} = y_j(I)$$

since the first factor is the probability that $J(I, v_0)$ is selected after casting the die associated with node r, while factor $\frac{x(I)}{\sum_{I' \in \mathcal{I}(P_{v_i}): Q(F,v_i) \cap I' = Q(F,v_i)} x(I')}$ is the probability that the k-independent set is augmented to include the paths in $J(I, v_i)$ given that paths of $Q(I, v_i)$ have been selected in previous steps.

To show that the algorithm above runs in time polynomial in the load of the set of paths P and in the size of the graph, we observe that we cast n dice and the number of faces of the die associated with a non-leaf node v is upper-bounded by $|\mathcal{I}(P_v)|$ which, since both k and the degree of v are constants, is polynomial in the load of $P_{Z-\lceil L/k \rceil}$. ■