Randomized Path Coloring on Binary Trees*

Vincenzo Auletta[1], Ioannis Caragiannis[2], Christos Kaklamanis[2], and Pino Persiano[1]

[1] Dipartimento di Informatica ed Applicazioni
Università di Salerno, 84081 Baronissi, Italy
{auletta,giuper}@dia.unisa.it

[2] Computer Technology Institute and
Dept. of Computer Engineering and Informatics
University of Patras, 26500 Rio, Greece
{caragian,kakl}@cti.gr

Abstract. Motivated by the problem of WDM routing in all–optical networks, we study the following NP–hard problem. We are given a directed binary tree T and a set R of directed paths on T. We wish to assign colors to paths in R, in such a way that no two paths that share a directed arc of T are assigned the same color and that the total number of colors used is minimized. Our results are expressed in terms of the depth of the tree and the maximum load l of R, i.e., the maximum number of paths that go through a directed arc of T.
So far, only deterministic greedy algorithms have been presented for the problem. The best known algorithm colors any set R of maximum load l using at most $5l/3$ colors. Alternatively, we say that this algorithm has performance ratio 5/3. It is also known that no deterministic greedy algorithm can achieve a performance ratio better than 5/3.
In this paper we define the class of greedy algorithms that use randomization. We study their limitations and prove that, with high probability, randomized greedy algorithms cannot achieve a performance ratio better than 3/2 when applied to binary trees of depth $\Omega(l)$, and $1.293 - o(1)$ when applied to binary trees of constant depth.
Exploiting inherent properties of randomized greedy algorithms, we obtain the first randomized algorithm for the problem that uses at most $7l/5 + o(l)$ colors for coloring any set of paths of maximum load l on binary trees of depth $o(l^{1/3})$, with high probability. We also present an existential upper bound of $7l/5 + o(l)$ that holds on any binary tree.
In the analysis of our bounds we use tail inequalities for random variables following hypergeometrical probability distributions which may be of their own interest.

* This work is partially supported by EU IST Project ALCOM–FT, EU RTN Project ARACNE, Progetto Cofinanziato of MURST–Italy and by a research grant from the Università di Salerno.

K. Jansen and S. Khuller (Eds.): APPROX 2000, LNCS 1913, pp. 60–71, 2000.

1 Introduction

Let $T(V, E)$ be a directed tree, i.e., a tree with each arc consisting of two opposite directed arcs. Let R be a set of directed paths on T. The path coloring problem is to assign colors to paths in R so that no two paths that share a directed arc 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 [20] 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 [5], 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 R, i.e., the maximum number of paths that share a directed arc of T. An algorithm that assigns at most $2l$ colors to any set of paths of load l can be derived by the work of Raghavan and Upfal [19] on the undirected version of the problem. Alternatively, we say that this algorithm has performance ratio 2. Mihail et al. [16] give an 15/8 upper bound. Kaklamanis and Persiano [11] and independently Kumar and Schwabe [15] improve the upper bound to 7/4. The best known upper bound is 5/3 [12].

All the above algorithms are deterministic and greedy in the following sense: they visit the tree in a top to bottom manner and at each node v color all paths that touch node v and are still uncolored; moreover, once a path has been colored, it is never recolored again. In the context of WDM routing, greedy algorithms are important as they are simple and, more importantly, they are amenable of being implemented easily and fast in a distributed environment. Kaklamanis et al. [12] prove that no greedy algorithm can achieve better performance ratio than 5/3.

The path coloring problem on binary trees is also NP–hard [6]. In this case, we express the results in terms of the depth of the tree, as well. All the known upper and lower bounds hold in this case. Simple deterministic greedy algorithms that achieve the 5/3 upper bound in binary trees are presented in [3] and [10]. The best known lower bound on the performance ratio of any algorithm is 5/4 [15], i.e., there exists a binary tree T of depth 3 and a set of paths R of load l on T that cannot be colored with less than $5l/4$ colors.

Randomization has been used as a tool for the design of path coloring algorithms on rings and meshes. Kumar [14] presents an algorithm that takes advantage of randomization to round the solution of an integer linear programming relaxation of the circular arc coloring problem. As a result, he improves the upper bound on the approximation ratio for the path coloring problem in rings to $1.37 + o(1)$. Rabani in [18] also follows a randomized rounding approach and presents an existential constant upper bound on the approximation ratio for the path coloring problem on meshes.

In this paper we define the class of greedy algorithms that use randomization. We study their limitations proving lower bounds on their performance when they are applied to either large or small trees. In particular, we prove that, with high probability, randomized greedy algorithms cannot achieve a performance ratio better than 3/2 when applied to binary trees of depth $\Omega(l)$, while their performance is at least $1.293 - o(1)$ when applied to trees of constant depth.

We also exploit inherent advantages of randomized greedy algorithms and, using limited recoloring, we obtain the first randomized algorithm that colors sets of paths of load l on binary trees of depth $o(l^{1/3})$ using at most $7l/5 + o(l)$ colors. For the analysis, we use tail inequalities for random variables that follow the hypergeometrical distribution. Our upper bound holds with high probability under the assumption that the load l is large. Our analysis also yields an existential upper bound of $7l/5 + o(l)$ on the number of colors sufficient for coloring any set of paths of load l that holds on any binary tree.

The rest of the paper is structured as follows. In Section 2, we present new technical lemmas for random variables following the hypergeometrical probability distribution that might be of their own interest. In Section 3, we give the notion of randomized greedy algorithms and study their limitations proving our lower bounds. Finally, in Section 4, we present our constructive and existential upper bounds.

Due to lack of space, we omit formal proofs from this extended abstract. Instead, we provide outlines for the proofs, including formal statements of most claims and lemmas used to prove our main theorems. The complete proofs can be found in the full version of the paper [1].

2 Preliminaries

In this section we present tail bounds for hypergeometrical (and hypergeometrical–like) probability distributions. These bounds will be very useful for proving both the upper and the lower bound for the path coloring problem. Our approach is similar to the one used in [13] (see also [17]) to calculate the tail bounds of a well known occupancy problem. We exploit the properties of special sequences of random variables called martingales, using Azuma's inequality [2] for their analysis. Similar results in a more general context are presented in [21].

Consider the following process. We have a collection of n balls, of which αn are red and $(1-\alpha)n$ are black ($0 \leq \alpha \leq 1$). We select without replacement uniformly at random βn balls ($0 \leq \beta \leq 1$). Let Ω_1 be the random variable representing the number of red balls that are selected; it is known that Ω_1 follows the hypergeometrical probability distribution [7]. We give bounds for the tails of the distribution of Ω_1.

Lemma 1. *The expectation of* Ω_1 *is*

$$\mathcal{E}[\Omega_1] = \alpha\beta n$$

and

$$\Pr\left[|\Omega_1 - \mathcal{E}[\Omega_1]| > \sqrt{2\beta\gamma n}\right] \leq 2e^{-\gamma},$$

for any $\gamma > 0$.

Now consider the following process. We have a collection of n balls, of which αn are red and $(1-\alpha)n$ are black $(0 \leq \alpha \leq 1)$. We execute the following two step experiment. First, we select without replacement uniformly at random $\beta_1 n$ out of the n balls, and then, starting again with the same n balls, we select without replacement uniformly at random $\beta_2 n$ out of the n balls $(0 \leq \beta_1, \beta_2 \leq 1)$. We study the distribution of the random variable Ω_2 representing the number of red balls that are selected in both selections.

Lemma 2. *The expectation of* Ω_2 *is*

$$\mathcal{E}[\Omega_2] = \alpha\beta_1\beta_2 n$$

and

$$\Pr\left[|\Omega_2 - \mathcal{E}[\Omega_2]| > 2\sqrt{2\min\{\beta_1, \beta_2\}\gamma n}\right] \leq 4e^{-\gamma},$$

for any $\gamma > 0$.

3 Randomized Greedy Algorithms

Greedy algorithms have a top–down structure as the algorithms presented in [16,11,15,12,3,10]. Starting from a node, the algorithm computes a breadth–first numbering of the nodes of the tree. The algorithm proceeds in phases, one per each node v of the tree. The nodes are considered following their breadth first numbering. In the phase associated with node v, it is assumed that we already have a partial proper coloring where all paths that touch (i.e., start, end, or go through) nodes with numbers strictly smaller than v's have been colored and that no other path has been colored. During this phase, the partial coloring is extended to one that assigns proper colors to all paths that touch v but have not been colored yet. During each phase, the algorithm does not recolor paths that have been colored in previous phases. So far, only deterministic greedy algorithms have been studied. The deterministic greedy algorithm presented in [12] guarantees that any set of paths of load l can be colored with $5l/3$ colors.

A randomized greedy algorithm $\mathcal{A}$ uses a palette of colors and proceeds in phases. At each phase associated with a node v, A picks a random proper coloring of the uncolored paths using colors of the palette according to some probability distribution.

We can prove that no randomized greedy algorithm can achieve a performance ratio better than $3l/2$ if the depth of the tree is large.

Theorem 3. *Let* $\mathcal{A}$ *be a (possibly randomized) greedy path coloring algorithm on binary trees. There exists a randomized algorithm* $\mathcal{ADV}$ *which, on input* $\epsilon > 0$ *and integer* $l > 0$*, outputs a binary tree* T *of depth* $l + \epsilon \ln l + 2$ *and a set* R *of paths of maximum load* l *on* T*, such that the probability that* $\mathcal{A}$ *colors* R *with at least* $3l/2$ *colors is at least* $1 - \exp(-l^{\epsilon})$.

We can also prove the following lower bound that captures the limitations of randomized greedy algorithms even on small trees.

Theorem 4. *Let $\mathcal{A}$ be a (possibly randomized) greedy path coloring algorithm on binary trees. There exists a randomized algorithm $\mathcal{ADV}$ which, on input $\delta > 0$ and integer $l > 0$, outputs a binary tree T of constant depth and a set R of paths of maximum load l on T, such that the probability that $\mathcal{A}$ colors R with at least $(1.293 - \delta - o(1))l$ colors is at least $1 - O(l^{-2})$.*

Furthermore, Theorem 4 can be extended for the case of randomized algorithms with a greedy structure that allow for limited recoloring like the one we present in the next section.

4 Upper Bounds

In this section we present our randomized algorithm for the path coloring problem on binary trees. Note that this is the first randomized algorithm for this problem. Our algorithm has a greedy structure but allows for limited recoloring. We first present three procedures (namely Preprocessing Procedure, Recoloring Procedure, and Coloring Procedure) that are used as subroutines by the algorithm, in Sections 4.1, 4.2, and 4.3, respectively. Then, in Section 4.4, we give the description of the algorithm and the analysis of its performance. In particular, we show how our algorithm can color any set of paths of load l on binary trees of depth $o(l^{1/3})$ using at most $7l/5 + o(l)$ colors. Our analysis also yields an existential upper bound on the number of colors sufficient for coloring any set of paths of load l on any binary tree (of any depth), which is presented in Section 4.5.

4.1 The Preprocessing Procedure

Given a set R^* of directed paths of maximum load l on a binary tree T, we want to transform it to another set R of paths that satisfies the following properties:

- **Property 1:** They have full load l at each directed arc.
- **Property 2:** For every node v, paths that originate or terminate at a node v appear on only one of the three arcs adjacent to v.
- **Property 3:** For every node v, the number of paths that originate from v is equal to the number of paths that are destined for v (note that property 3 is a corollary from properties 1 and 2).

This is done by a Preprocessing Procedure which is described in the following. At first, the set of paths is transformed to a full load superset of paths by adding single–hop paths at the directed arcs that are not fully loaded. Next the non–leaf nodes of the tree are traversed in a BFS manner. We consider a step of the traversal associated with a node v. Let R_v be the set of paths that touch v. R_v is the union of two disjoint sets of paths: S_v which is the set of paths that either originate or terminate at v, and P_v which is the set of paths that go through v. Pairs of paths (v_1, v), (v, v_2) in S_v are combined and create one path (v_1, v_2) that goes through v. Paths (v_1, v), (v, v_2) are deleted by S_v and the path (v_1, v_2) is inserted to P_v.

Lemma 5. *Consider a set of directed paths R^* of maximum load l on a binary tree T. After the application of the Preprocessing Procedure, the set R of directed paths that is produced satisfies properties 1, 2, and 3.*

It is easy to see how to obtain a legal coloring for the original pattern if we have a coloring of the pattern which is produced after the application of the Preprocessing Procedure.

In the rest of the paper we consider only sets of paths that satisfy properties 1, 2 and 3. Let R be a set of paths that satisfies properties 1, 2 and 3 on a binary tree T, and let v be a non–leaf node of the tree. Assuming that the nodes of T are visited in a BFS manner, let $p(v)$ be the parent node of v, and $l(v)$, $r(v)$ its left and the right child nodes, respectively.

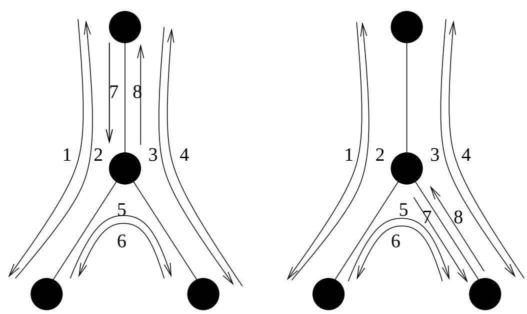

Fig. 1. The two cases that need to be considered for the analysis of path coloring algorithms on binary trees. Numbers represent groups of paths (number 1 implies the set of paths M_v^1, etc.).

We partition the set P_v of paths that go through v to the following disjoint subsets which we call groups: the group M_v^1 of the paths that come from $p(v)$ and go to $l(v)$, the group M_v^2 of the paths that come from $l(v)$ and go to $p(v)$, the group M_v^3 of the paths that come from $p(v)$ and go to $r(v)$, the group M_v^4 of the paths that come from $r(v)$ and go to $p(v)$, the group M_v^5 of the paths that come from $l(v)$ and go to $r(v)$, and the group M_v^6 of the paths that come from $r(v)$ and go to $l(v)$. Since P_v satisfies properties 1, 2, and 3, we only need to consider the two cases for paths in S_v depicted in Figure 1.

- **Scenario I:** The paths in S_v touch node $p(v)$: the set S_v is composed by the group M_v^7 of the paths that come from $p(v)$ and stop at v and the group M_v^8 of the paths that originate from v and go to $p(v)$.

- **Scenario II:** The paths in S_v touch a child node of v (wlog $r(v)$): the set S_v is composed by the group M_v^7 of the paths that originate from v and go to $r(v)$ and the group M_v^8 of the paths that come from $r(v)$ and stop at v.

4.2 The Recoloring Procedure

Let T be a binary tree with 4 nodes and of the form depicted in Figure 1. Let v be the node with degree 3, $p(v)$ its parent, and $l(v)$, $r(v)$ its left and right child node, respectively. Consider a set R_v of paths of full load l on T which is partitioned to groups M_v^i as described in the previous section.

Also consider a random coloring of the paths that traverse the opposite directed arcs $(p(v), v)$ with tl colors $(1 \leq t \leq 2)$. We assume that the coloring has been selected among all possible proper colorings according to some probability distribution $\mathcal{P}$. Let S be the number of single colors, i.e., colors assigned only to one path that traverse arc $(p(v), v)$, and D the number of double colors, i.e., colors assigned to two paths that traverse arc $(p(v), v)$ in opposite directions. Since the load of R_v is l we obtain that $D = (2-t)l$ and $S = 2(t-1)l$.

Definition 6. *Let $D = (2-t)l$. A probability distribution $\mathcal{P}$ over all proper colorings of paths in R_v that traverse the arc $(p(v), v)$ with tl colors is weakly uniform if for any two paths $r_1, r_2 \in R_v$ that traverse arc $(p(v), v)$ in opposite directions, the probability that r_1 and r_2 are assigned the same color is D/l^2.*

We will give an example of a random coloring of paths in R_v that traverse the arc $(p(v), v)$ with tl colors according to the weakly uniform probability distribution. Let $D = (2-t)l$. We use a set $\mathcal{X}_1$ of l colors and assign them to paths in R_v that traverse the directed arc $(p(v), v)$. Then, we define the set $\mathcal{X}_2$ of colors as follows. We randomly select D colors of $\mathcal{X}_1$ and use $l - D$ additional colors. For the paths in R_v that traverse the directed arc $(v, p(v))$, we select uniformly at random a coloring among all possible colorings with colors of $\mathcal{X}_2$.

Let $\mathcal{C}$ be a coloring of paths in R_v that traverse the arc $(p(v), v)$. We denote by A_i the set of single colors assigned to paths in M_v^i, and by A_{ij} the set of double colors assigned to paths in groups M_v^i and M_v^j. Clearly, the numbers $|A_i|$ and $|A_{ij}|$ are random variables following the hypergeometrical distribution with expectation

$$\mathcal{E}[|A_i|] = \frac{|M_v^i|S}{2l} \text{ and } \mathcal{E}[|A_{ij}|] = \frac{|M_v^i||M_v^j|D}{l^2}$$

In the following we use i and ij for indexing of sets of single and double colors, respectively. In addition, we use the expressions "for any i" and "for any pair i, j"; we use them as shorthand for the phrases "for any i such that paths in group M_v^i traverse the arc $(p(v), v)$ in some direction" and "for any pair i, j such that paths in groups M_v^i and M_v^j traverse the arc $(p(v), v)$ in opposite directions", respectively.

Definition 7. *Let $D = (2-t)l$. A probability distribution $\mathcal{P}$ over all proper coloring of paths in R_v that traverse the arc $(p(v), v)$ with tl colors satisfying*

$$|A_{ij}| = \frac{D|M_v^i||M_v^j|}{l^2}$$

for any pair i, j, is strongly uniform if for any two paths $r_1, r_2 \in R_v$ that traverses arc $(p(v), v)$ in opposite directions, the probability that r_1 and r_2 are assigned the same color is D/l^2.

Since, for any i, it is $|M_v^i| = |A_i| + \sum_j |A_{ij}|$, we obtain that a coloring chosen according to the strongly uniform probability distribution satisfies

$$|A_i| = \frac{|M_v^i|S}{2l}.$$

Assume that we are given a tree T of 4 nodes (and of the form depicted in Figure 1) and a set of paths R_v of full load l on T as described, and a random coloring $\mathcal{C}$ of the paths in R_v that traverse the arc $(p(v), v)$ with tl colors chosen according to the weakly uniform probability distribution. The Recoloring Procedure will recolor a set $R'_v \subseteq R_v$ of paths that traverse the arc $(p(v), v)$ so that the coloring $\mathcal{C}'$ of paths in R_v that traverse the arc $(p(v), v)$ is a random coloring with tl colors chosen according to the strongly uniform probability distribution.

We now give the description of the Recoloring Procedure. First, each color is marked with probability p. Let X be the set of marked colors that consists of the following disjoint sets of colors: the sets $X_i = A_i \cap X$, for any i, and the sets $X_{ij} = A_{ij} \cap X$, for any pair i, j.

We set

$$y_i = |A_i| - \frac{|M_v^i|S(1 - l^{-1/3})}{2l} \tag{1}$$

for any i, and

$$y_{ij} = |A_{ij}| - \frac{|M_v^i||M_v^j|D(1 - l^{-1/3})}{l^2} \tag{2}$$

for any pair i, j. Clearly, the conditions $0 \leq y_i \leq |X_i|$ and $0 \leq y_{ij} \leq |X_{ij}|$ are necessary so that the procedure we describe in the following is feasible. For the moment we assume that $0 \leq y_i \leq |X_i|$ and $0 \leq y_{ij} \leq |X_{ij}|$.

We select a random set Y_i of y_i colors of X_i, for any i, and a random set Y_{ij} of y_{ij} colors of X_{ij}, for any pair i, j. Let Y be the union of all sets Y_i and Y_{ij} and R'_v the set of paths of R_v that traverse the arc $(p(v), v)$ colored with colors in Y. Using (1) and (2), adding all y_i's and y_{ij}'s we obtain

$$|Y| = \sum_i y_i + \sum_{i,j} y_{ij} = tl^{2/3},$$

while the load of R'_v in the opposite directed arcs between $p(v)$ and v is

$$\sum_i y_i + \sum_{i,j} y_{ij} = \sum_j y_j + \sum_{i,j} y_{ij} = l^{2/3},$$

where i and j takes such values that paths in groups M_v^i traverse the directed arc $(p(v), v)$ and paths in groups M_v^j traverse the directed arc $(v, p(v))$. The Recoloring Procedure ends by producing a random coloring $\mathcal{C}''$ of paths in R'_v with the $tl^{2/3}$ colors of Y according to the strongly uniform probability distribution.

Now, our argument is divided in two parts. First, in Claim 8, we show that, if y_i and y_{ij} are of the correct size and $\mathcal{C}''$ is a random coloring of paths in R'_v that traverse the arc $(p(v), v)$ with $tl^{2/3}$ colors according to the strongly uniform probability distribution, then the new coloring $\mathcal{C}'$ of the paths in R_v that traverse the arc $(p(v), v)$ is a random coloring with tl colors according to the strongly uniform probability distribution. Then, in Lemma 9, we show that, for $t = 6/5$ and sufficiently large values of l and the cardinality of each group M_v^i, it is possible to fix the marking probability p so that y_i and y_{ij} have the correct size, with high probability.

Claim 8. *If* $0 \le y_i \le |X_i|$ *for any* i*, and* $0 \le y_{ij} \le |X_{ij}|$ *for any pair* i, j*, the Recoloring Procedure produces a random coloring* $\mathcal{C}'$ *of paths in* R_v *that traverse the arc* $(p(v), v)$ *with* tl *colors according to the strongly uniform probability distribution.*

In the following we concentrate our attention to the case $t = 6/5$ which is sufficient for the proof of the upper bound. The following lemma gives sufficient conditions for the correctness of the Recoloring Procedure. It can be adjusted (with a modification of the constants) so that it works for any $t \in [1, 2]$.

Lemma 9. *Let* $0 < \delta < 1/3$ *be a constant,* $t = 6/5$*,* $p = 5l^{-1/3}$*, and* $l \ge 125$*. If each non–empty group* M_v^i *has cardinality at least*

$$\max\{6.25l^{2/3+\delta}, 1.16l^{8/9+\delta/3}\}$$

then the Recoloring Procedure is correct with probability at least $1 - 63\exp(-l^{\delta}/8)$*.*

For proving the lemma, we use the Chernoff–Hoeffding bound [4,9] together with the tail bounds for hypergeometrical probability distributions (Lemmas 1 and 2) to prove that the hypotheses of Claim 8 are true, with high probability.

Remark. The numbers y_i and y_{ij} given by the equations (1) and (2) must be integral. In the full version of the paper [1], we prove additional technical claims that help us to handle these technical details by adding paths to the original set of paths, increasing the load by an $o(l)$ term.

4.3 The Coloring Procedure

Assume again that we are given a tree T of 4 nodes (and of the form depicted in Figure 1) and a set of paths R_v of full load l on T as described in the previous section, and a random coloring of the paths in R_v that traverse the arc $(p(v), v)$ with $6l/5$ colors according to the strongly uniform probability distribution. The Coloring Procedure extends this coloring to the paths that have not been colored yet, i.e., the paths in R_v that do not traverse the arc $(p(v), v)$. The performance of the Coloring Procedure is stated in the following lemma.

Lemma 10. *The Coloring Procedure colors the paths in R_v that have not been colored yet, in such a way that at most $7l/5$ total colors are used for all paths in R_v, that the number of colors seen by the arcs $(v, l(v))$ and $(v, r(v))$ is exactly $6l/5$, and that paths in R_v traversing these arcs are randomly colored according to the weakly uniform probability distribution.*

According to Section 4.1, for the Coloring Procedure we need to distinguish between Scenarios I and II. The complete description of the Coloring Procedure and the proof of Lemma 10 can be found in [1].

4.4 The Path Coloring Algorithm

In this section we give the description of our algorithm and discuss its performance. In particular, we prove the following theorem.

Theorem 11. *There exists a randomized algorithm that, for any constant $\delta < 1/3$, colors any set of directed paths of maximum load l on a binary tree of depth at most $l^\delta/8$, using at most $7l/5 + o(l)$ colors, with probability at least $1 - \exp\left(-\Omega(l^\delta)\right)$.*

Let T be a binary tree and R^* a set of paths of load l on T. Our algorithm uses as subroutines the Preprocessing Procedure, the Recoloring Procedure and the Coloring Procedure described in the previous sections.

First, we execute the Preprocessing Procedure on R^* and we obtain a new set R of paths that satisfies properties 1, 2, and 3, as described in Section 4.1.

Then, the algorithm roots the tree at a leaf node $r(T)$, and produces a random coloring of the paths that share the opposite directed arcs adjacent to $r(T)$ with exactly $6l/5$ colors. This can be done by assigning l colors to the paths that originate from $r(T)$, randomly selecting $4l/5$ from these colors, and randomly assigning these colors and $l/5$ additional colors to the paths destined for $r(T)$. Note that, in this way, the coloring of the paths that share the opposite directed arcs adjacent to $r(T)$ is obviously random according to the weakly uniform probability distribution. Then, the algorithm performs the following procedure Color on the child node of $r(T)$.

The procedure Color at a node v takes as input the set of paths R_v that touch v together with a random coloring of paths that traverse the arc $(p(v), v)$ with $6l/5$ colors according to the weakly uniform probability distribution. The procedure immediately stops if v is a leaf. Otherwise, the Recoloring Procedure is executed producing a random coloring of paths traversing arc $(p(v), v)$ with $6l/5$ colors according to the strongly uniform probability distribution. We denote by R'_v the set of paths that are recolored by the Recoloring Procedure. Then, the Coloring Procedure is executed producing a coloring of the paths in R_v that have not been colored yet, using at most $7l/5$ colors in total, in such a way that the number of colors seen by the opposite directed arcs between v and $r(v)$ and the opposite directed arcs between v and $l(v)$) is exactly $6l/5$ and that the colorings of paths traversing these arcs are random according to the weakly uniform probability distribution. The procedure recursively executes Color for the child nodes of v, $r(v)$ and $l(v)$.

After executing procedure Color recursively on every node of T, all paths in R have been properly colored except for the paths in the set $\cup_v R'_v$; these were the paths recolored during the execution of the Recoloring Procedure at all nodes. Our algorithm ends by properly coloring the paths in $\cup_v R'_v$; this can be done easily with the greedy deterministic algorithm using at most $o(l)$ extra colors because of the following Lemma 12.

Lemma 12. *Let $0 < \delta < 1/3$ be a constant. Consider the execution of the algorithm on a binary tree of depth at most $l^\delta/8$. The load of all paths that are recolored by the Recoloring Procedure is at most $2l^{2/3+\delta}$ with probability at least $1 - exp\left(-\Omega(l^{2/3+\delta})\right)$.*

For proving Lemma 12, we actually prove the stronger statement that, with high probability, the load of the paths that are marked (which is a superset of the paths that are recolored) is at most $2l^{2/3+\delta}$.

4.5 An Existential Upper Bound

The analysis of the previous sections implies that, when the algorithm begins the execution of the phase associated with a node v, with some (small) positive probability, the numbers of single and double colors is equal to their expectation. We may assume that the coloring of paths traversing the arc $(p(v), v)$ is random according to the strongly uniform probability distribution. Thus, with non–zero probability, the execution of the Recoloring Procedure is unnecessary at all steps of the algorithm. Using this probabilistic argument together with additional technical claims, we obtain an existential upper bound on the number of colors sufficient for coloring any set of paths of load l.

Theorem 13. *Any set of paths of load l on a binary tree can be colored with at most $\frac{7}{5}\left(5\left\lceil\frac{\sqrt{l}}{5}\right\rceil+10\right)^2$ colors.*

Note that Theorem 13 holds for any binary tree (of any depth). Especially for binary trees of depth $o(l^{1/3})$, it improves the large hidden constants implicit in the $o(l)$ term of our constructive upper bound (Theorem 11). In larger binary trees, it significantly improves the 5/3 constructive upper bound for all sets of paths of load greater than $26,000$.

References

1. V. Auletta, I. Caragiannis, C. Kaklamanis, P. Persiano. Randomized Path Coloring on Binary Trees. CTI Technical Report 2000–01–05, January 2000. `http://students.ceid.upatras.gr/~caragian/ackp00.ps`
2. K. Azuma. Weighted Sums of Certain Dependent Random Variables. *Tohoku Mathematical Journal*, 19:357–367, 1967.

3. I. Caragiannis, C. Kaklamanis, P. Persiano. Bounds on Optical Bandwidth Allocation on Directed Fiber Tree Topologies. *2nd Workshop on Optics and Computer Science (WOCS '97)*, 1997.
4. H. Chernoff. A Measure of Asymptotic Efficiency for Tests of a Hypothesis based on the Sum of Observations. *Annals of Mathematical Statistics*, 23:493–509, 1952.
5. T. Erlebach, K. Jansen. Scheduling of Virtual Connections in Fast Networks. In *Proc. of the 4th Workshop on Parallel Systems and Algorithms*, pp. 13–32, 1996.
6. T. Erlebach, K. Jansen. Call Scheduling in Trees, Rings and Meshes. In *Proc. of the 30th Hawaii International Conference on System Sciences*, 1997.
7. W. Feller. An Introduction to Probability Theory and Its Applications: Volume I, John Wiley and Sons, 1968.
8. L. Gargano, P. Hell, S. Perennes, Colouring All Directed Paths in a Symmetric Tree with Applications to WDM Routing. In *Proc. of the 24th International Colloquium on Automata, Languages, and Programming (ICALP '97)*, LNCS 1256, Springer, pp. 505–515.
9. W. Hoeffding. Probability Inequalities for sums of Bounded Random Variables. *Journal of the American Statistics Association*, 58:13–30, 1963.
10. K. Jansen. Approximation Results for Wavelength Routing in Directed Trees. *2nd Workshop on Optics and Computer Science (WOCS '97)*, 1997.
11. C. Kaklamanis, P. Persiano. Efficient Wavelength Routing on Directed Fiber Trees. In *Proc. of the 4th European Symposium on Algorithms (ESA '96)*, LNCS 1136, Springer Verlag, 1996, pp. 460–470.
12. C. Kaklamanis, P. Persiano, T. Erlebach, K. Jansen. Constrained Bipartite Edge Coloring with Applications to Wavelength Routing. In *Proc. of the 24th International Colloquium on Automata, Languages, and Programming (ICALP '97)*, LNCS 1256, Springer Verlag, 1997, pp. 493–504.
13. A. Kamath, R. Motwani, K. Palem, and P. Spirakis. Tail bounds for occupancy and the satisfiability threshold conjecture. In *Proc. of the 35th Annual IEEE Symposium on Foundations of Computer Science (FOCS '94)*, pp. 592–603, 1994.
14. 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)*, 1998.
15. V. Kumar, 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.
16. M. Mihail, C. Kaklamanis, S. Rao. Efficient Access to Optical Bandwidth. In *Proc. of the 36th Annual Symposium on Foundations of Computer Science (FOCS '95)*, 1995, pp. 548–557.
17. R. Motwani, P. Raghavan. Randomized Algorithms. *Cambridge University Press*, 1995.
18. Y. Rabani. Path Coloring on the Mesh. In *Proc. of the 37th Annual Symposium on Foundations of Computer Science (FOCS '96)*, 1996.
19. 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.
20. R. Ramaswami, K. Sivarajan. Optical Networks. *Morgan Kauffman Publishers*, 1998.
21. J. P. Schmidt, A. Siegel, A. Srinivasan. Chernoff–Hoeffding Bounds for Applications with Limited Independence. In *Proc. of the 3rd Annual ACM–SIAM Symposium on Discrete Algorithms (SODA '93)*, 1993, pp. 331–340.