Approximation Results for Wavelength Routing in Directed Trees

Klaus Jansen
Fachbereich IV - Mathematik
Universität Trier
54 286 Trier, Germany
jansen@dm3.uni-trier.de

Abstract

The paper deals with wavelength routing in WDM (wavelength division multiplexing) optical networks. Wavelengths must be assigned to connection requests which are represented as directed paths, and it is required that paths receive different wavelengths if they share a directed link. The goal is to minimize the number of wavelengths used. Mihail et al. [10] conjectured that there is a solution for directed trees with at most \( \frac{3}{2} L_{\text{max}} \) wavelengths where \( L_{\text{max}} \) is the maximum load on a directed link.

In this paper, we show that there exists a binary tree with load \( L_{\text{max}} \neq 3 \) such that every solution needs at least 5 wavelengths. Furthermore, we present an approximation algorithm for binary trees with approximation ratio \( \frac{5}{3} L_{\text{max}} \leq \frac{5}{3} \cdot \text{OPT} \). This improves the bound \( \frac{2}{5} L_{\text{max}} \) for binary trees given by Mihail et al. [10]. Finally, we show that no local greedy algorithm for binary trees can go below \( \frac{5}{3} L_{\text{max}} \). This ratio is the best ratio that can be achieved by any local greedy algorithm for this problem.

1. Introduction

Optics is a technology that promises very high speed networks in the future. The high speeds in these networks arise from maintaining signals in optical form. We refer to the books of Green [5] and McAulay [9] for an overview of the physical theory and application of this technology. Multiple laser beams can be propagated over the same fiber on distinct optical wavelengths; this is achieved by WDM (wavelength division multiplexing) [2]. A single optical link in a network can carry several logical signals if they are transmitted on different wavelengths. The information is transmitted as light and must not be converted during the transmission. Signals for different calls may travel on the same link into a node \( v \) on different wavelengths and can exit node \( v \) along different links keeping their original wavelength.

To do this, switches based on acousto-optic filters [1] are used. Such a switch can differentiate between several wavelengths coming along a link to a node and can direct them to different outputs of the switch. The only constraint is that paths (calls) sharing the same link must have different wavelengths. We note that these optical switches can not modulate the wavelengths of the signals passing through them. Intuitively, we may think of wavelengths as light rays of different colors. Wavelengths are a limited resource, however, and solutions to the problem of efficient wavelength allocation have importance for the future development of this technology.

In this paper we consider tree topologies where each edge of the tree consists of two opposite directed fiber links. A call request \( r \) is a directed path between two vertices \( x \) and \( y \) of the network. Given a set \( R \) of call requests, the maximum number of paths (the maximum load) through any directed link is denoted by \( L_{\text{max}} \). Two call requests \( r \) and \( r' \) are in conflict if the corresponding paths go through the same directed fiber link. Calls using an edge of the tree in different directions do not interfere with each other. The goal is to minimize the number of required wavelengths such that no two calls that are in conflict get the same wavelength. We refer to wavelengths as colors, and we can view the wavelength assignment problem as a path coloring problem. Let \( \text{OPT} \) be the minimum number of needed wavelengths or colors. Then, we have \( L_{\text{max}} \leq \text{OPT} \).

Raghavan and Upfal [11] studied the undirected version of the wavelength routing problem. The directed version of the wavelength routing problem was first considered in [10]. Mihail et al. presented approximation algorithms for trees,
rings and trees of rings. For trees, they gave an approximation algorithm with bound $\frac{15}{8} L_{\text{max}} \leq \frac{15}{8} OPT$. They presented a 2-approximation algorithm for rings and a $\frac{15}{8}$-approximation algorithm for trees of rings. Furthermore, they had a $\frac{7}{4}$-approximation algorithm for binary trees and noticed that no greedy algorithm will go below $\frac{7}{4} L_{\text{max}}$. Finally, they conjectured that there is a coloring for trees with at most $\frac{3}{2} L_{\text{max}}$ colors. Erlebach and Jansen [3] have proved that the wavelength routing problem is NP-complete even for binary directed trees. Independently, Kaklamanis and Persiano [6] and Kumar and Schwabe [8] improved the ratio $\frac{15}{8}$ to $\frac{7}{4}$ for arbitrary directed trees. All these algorithms are local greedy in the sense that they consider the nodes of the tree in depth-first order and extend an existing partial coloring by choosing colors for the uncolored paths touching the current node.

In this paper, we show that there exists even a binary tree with load $L = 3$ such that every coloring requires at least 5 wavelengths. Moreover, we present an approximation algorithm for binary trees with approximation ratio $\frac{5}{8} L_{\text{max}} \leq \frac{5}{8} OPT$. Finally, we show that no local greedy algorithm for binary trees can go below $\frac{5}{8} L_{\text{max}}$. This ratio is the best ratio that can be achieved by any local greedy algorithm for the wavelength routing problem in directed trees.

2. The 5/3 example

Mihail, Kaklamanis and Rao [10] conjectured that there is a feasible coloring for the wavelength routing problem in directed trees that uses at most $\frac{1}{2} L_{\text{max}}$ colors. We prove that this conjecture is not true even for binary trees with load $L_{\text{max}} = 3$.

**Theorem 2.1** There exists a binary tree $T = (V, E)$ and a set of directed calls $R$ such that

1. the maximum number of paths through any directed link is three and
(2) every coloring needs at least five wavelengths or colors.

Proof: Components $A$, $B$ and $C$: First, we construct a component $A$ with three nodes $v_0, v_1, v_2$ and calls $x_1, \ldots, x_4, z_1, z_2, z_3, y$ as given in Figure 1.

Let $G(A)$ be the conflict graph corresponding to component $A$. We denote by $G_0(A)$ the subgraph of $G(A)$ induced by $\{x_1, x_2, x_3, x_4\}$, $G_1(A)$ the subgraph induced by $\{x_1, x_2, y, z_1, z_2, z_3\}$ and $G_2(A)$ the subgraph induced by $\{x_3, x_4, y, z_1, z_2, z_3\}$. The graph $G(A)$ obeys the following properties:

(1) Any 3-coloring of the graph $G_0(A)$ can be extended only to a coloring of $G(A)$ with at least four colors in $G_1(A)$ or $G_2(A)$.

(2) Any 4-coloring of $G_0(A)$ can be extended only to a coloring of $G(A)$ with at least five colors.

Next, we construct a component $B$ with three nodes as given in Figure 2. We define $G(B)$, $G_0(B)$, $G_1(B)$ and $G_2(B)$ in the same way as above. Then, we have the following property.

(3) Each 4-coloring of $G_0(B)$ can be extended only to a 4-coloring of $G(B)$ with at least four colors in $G_1(B)$ or $G_2(B)$ or to a coloring of $G(B)$ with at least five colors.

Assertion (3) can be proved by case analysis. As example, consider the case with $f(x) = 1$, $f(y) = 2$, $f(a) = 1$, $f(z) = 3$, $f(b) = 2$ and $f(c) = 4$. For a 4-coloring of $G(B)$ we must have $f(d_1) = 3$, $f(d_2) = 2$, $f(d_3) = 4$ or $f(d_1) = 3$, $f(d_2) = 4$, $f(d_3) = 2$. In both cases, we obtain four colors in $G_1(B)$.

The last component $C$ (see Figure 3) is used for the case that the calls $x_1, x_2, x_3, x_4$ in component $A$ are colored with only two colors. Again, we use $G(C)$, $G_1(C)$, $G_2(C)$ as above and consider a coloring $f$ of $G(C)$. If $\{f(x_1), f(x_2), f(x_3), f(x_4)\} = \{1, 2\}$ then we have $f(x_5) \notin \{1, 2\}$ and $\{f(x_3), f(x_4)\} = \{1, 2\}$. This implies that $G_2(C)$ now contains three colors. Therefore, a 2-coloring of $G_1(C)$ implies a 3-coloring of $G_2(C)$.

Construction: Now we explain the construction of the example. First, we take a $C$-component, connect the calls in the left child with an $A$ - component extending the paths $x_1, x_2, x_3, x_4$ and connect the calls in the right child with another $A$-component extending the paths $x_1, x_3, x_4, x_5$. $A'$ denotes the second $A$-component.

Then, we build recursively a tree with $B$-components as given in Figure 4. First, we take one $B$ - component with nodes $A$, $B$ and $C$. In the second step, we use two $B$ - components with nodes $B$, $D$, $E$ and $C$, $F$, $G$, respectively. The key idea of this tree is the use of property (3) and an adding of further $B$ - components at the leaves successively such that every 4 - coloring of $\{x_1, \ldots, x_6\}$ requires at least five colors. The call $a_1$ (and similar the other calls $a_2, \ldots, a_3'$) does not go directly from $B$ to $C$ as drawn in Figure 4; the call $a_1$ starts at $H$, runs through $D, B, A, C, G$ and ends at $O$. This means that we have drawn the paths from $v_1$ to $v_2$ (and from $v_2$ to $v_1$) in a $B$-component as a direct horizontal arc $(v_1, v_2)$ (and $(v_2, v_1)$) in Figure 4. Taking these horizontal arcs into account we have at most three paths on each directed link. For example, we have three paths $x_1, x_2, a_1$ from $B$ to $A$ and three paths $x_3, a_2, a_3$ in the opposite direction. Observe the symmetric construction of the directed paths in the tree in Figure 4.
Figure 4. The tree with $B$-components to achieve a fifth color for each 4-coloring of the calls $\{x_1, \ldots, x_6\}$. 
Figure 5. The local extension problem.

By case analysis (and using the symmetry of the tree in Figure 4), it can be proved that every 4-coloring of \(x_1, \ldots, x_6\) creates a fifth color in this tree. As an example, we consider a coloring \(f\) with \(f(x_1) = 1, f(x_2) = 2, f(x_3) = 3, f(x_4) = 1, f(x_5) = 4\) and \(f(x_6) = 3\). This implies \(f(a_1) = 3\) and \(\{f(a_2), f(a_3)\} \subset \{1, 2, 4\}\) or we have five colors directly. If \(\{f(a_2), f(a_3)\} = \{1, 2\}\), then \(c_1\) gets the fifth color. If \(\{f(a_2), f(a_3)\} = \{1, 4\}\), then \(b_1\) obtains the fifth color. In the last case \(\{f(a_2), f(a_3)\} = \{2, 4\}\), \(g_1\) gets the fifth color. We identify the paths \(x_1, \ldots, x_6\) at the root of this tree with the paths \(x_1, x_2, z_1, z_2, z_3, y\) in \(G_1(A)\). We use the same construction (three copies of the tree with \(B\)-components) for the paths in \(G_2(A), G_3(A')\) and \(G_2(A')\).

For every 3-coloring of the calls \(x_1, x_2, x_3, x_4\) in \(G_0(A)\) and every 3-coloring of the calls \(x_1, x_3, x_4, x_5\) in \(G_0(A')\) we obtain four colors in \(G_1(A)\) or \(G_2(A)\) (and four colors in \(G_1(A')\) or \(G_2(A')\)). If \(G_0(A)\) (or \(G_0(A')\)) is colored with four colors, then we obtain directly a fifth color in \(G(A)\) (or \(G(A')\)). Moreover, if the paths \(x_1, x_2, x_3, x_4\) are colored with only two colors, then (using the \(C\)-component) the paths \(x_1, x_3, x_4, x_5\) are colored with three colors. Consider an arbitrary 3-coloring of \(G_0(A)\) and an extension of this coloring to a 4-coloring of \(G(A)\). Then, using property (1) \(G_1(A)\) or \(G_2(A)\) contains at least four colors. Suppose that the left child contains four colors. Then, we have already five colors or we get the fifth color in the tree with the \(B\)-components corresponding to \(G_1(A)\).

3. A 5/3-approximation algorithm

In this section we improve the bound \(\frac{2}{3}L_{max}\) for binary trees given by Mihail et al. [10] to the new bound \(\lceil \frac{2}{3}L_{max} \rceil\). Suppose that a set of directed paths (or calls) in a binary tree is given such that the number of paths through any directed link is at most \(L_{max}\). We assign colors to the paths such that no paths going through the same link have the same color. We prove the following result.

**Theorem 3.1** Let \(T = (V, E)\) be a binary tree and \(R\) be a set of directed calls. Then, there is a polynomial time algorithm that finds a feasible coloring using at most \(\lceil \frac{2}{3}L_{max} \rceil \leq \lceil \frac{5}{3}O PT\rceil\) colors.

To prove this, we develop a coloring algorithm that starts at a leaf and runs through the tree \(T\) in depth first search. Let \(v_\ell\) be a leaf and \(v_\ell'\) be the unique neighbor of \(v_\ell\). Then, the paths through \((v_\ell, v_\ell')\) and \((v_\ell', v_\ell)\) can be colored simply with colors of the set \(\{1, \ldots, L_{max}\}\).

Now, we consider an internal node \(v\) with parent \(p(v)\) and two children \(v_1\) and \(v_2\). We assume that the paths through \((p(v), v)\) and \((v, p(v))\) are colored in a previous step. In the following, we must compute an extension of this coloring. To do this we color the new paths through the links \((v, v_1), (v_1, v), (v_2, v)\) and \((v, v_2)\). We may assume that no path starts (or ends) at node \(v\) and goes through only one of these four links. Otherwise, we can delete these paths and can simply color them after the coloring of the remaining paths. If we have used \(k\) colors for the remaining paths through \((v, v_1)\) and \((v_1, v)\), then we can
color the additional paths through \((v, v_1), (v_1, v)\) with at most \(\max (k, L_{\text{max}})\) colors. Furthermore, we can use this idea to extend a coloring for a node \(v\) with parent \(p(v)\) and only one child \(v_1\).

Using this assumption, we have to analyse the following local extension problem. The paths through \((p(v), v_1), (p(v), v_2), (v_1, p(v))\) and \((v_2, p(v))\) are colored previously with at most \(k\) colors. The goal is to find a compatible coloring of the paths through \((v_1, v_2)\) and \((v_2, v_1)\) such that we have at most

1. \(k\) colors for the paths through the links \((v, v_1)\) or \((v_1, v)\),
2. \(k\) colors for the paths through the links \((v, v_2)\) or \((v_2, v)\),
3. \(k'\) colors for all paths.

This problem is illustrated in Figure 5. Depending on \(L_{\text{max}}\) we define the parameters \(k, k'\) as in Table 1. We notice that \(k' = \lceil \frac{3}{2} L_{\text{max}} \rceil\). In the last section, we show that these parameters \(k\) and \(k'\) are required for this and other greedy approximation algorithms.

We may assume that the number of paths through \((v_1, v_2)\) and through \((v_2, v_1)\) is maximal with respect to \(L_{\text{max}}\); otherwise we can include some new paths to simplify the analysis. We denote by \(f_0\) the number of different colors of paths through \((p(v), v)\) and \((v, p(v))\). Given a computed extension of this precoloring, we denote by \(f_i\) the number of different colors of paths through \((v, v_i)\) or \((v_i, v)\), for \(i \in \{1, 2\}\). Moreover, \(f_2\) denotes the total number of colors used.

### 3.1. Transformation into a precoloring configuration

Next, we eliminate two special cases where some specified paths have the same precolor. In these cases, we reduce the capacity \(L_{\text{max}}\) by one or two. After this reduction, we find a simpler precoloring structure of the paths through \((v_i, p(v))\) or \((p(v), v_i)\). After we have computed a coloring for the simpler structure, we include the eliminated colored paths. For the analysis of the worst case bound, we use the color numbers \(f'_i\) \((i \in \{0, 1, 2\})\) and \(f'_2\) of the simpler reduced structure, and calculate later the color numbers \(f_i\) \((i \in \{0, 1, 2\})\) and \(f_2\) of the original local extension problem.

**Reduction (1):** There exist two paths \(P, P'\) through \((v_1, p(v))\) and \((p(v), v_2)\) with the same color. Since we have at most \(L_{\text{max}}\) paths through \((p(v), v)\) and at most \(L_{\text{max}}\) paths through \((v, p(v))\), there are at most

1. \(L_{\text{max}} - 1\) paths through \((v_2, p(v))\) and
2. \(L_{\text{max}} - 1\) paths through \((p(v), v_1)\).

Using the properties (1), (2) and the fact that the number of paths through \((v_2, v_1)\) is maximal with respect to \(L_{\text{max}}\), there exists at least one path through \((v_2, v_1)\). This path can be colored with the same color as the paths \(P\) and \(P'\). Deleting these three paths generates an instance with capacity \(L_{\text{max}} = L_{\text{max}} - 1\). Moreover, it holds \(f_i = f'_i + 1\), \(i \in \{0, 1, 2\}\) and \(f_2 = f'_2 + 1\). The case with paths through \((v_2, p(v))\) and \((p(v), v_1)\) works as well.

**Reduction (2):** There exist four paths \(P_1, P_2, P_3, P_4\) through \((p(v), v_1), (v_1, p(v)), (p(v), v_2), (v_2, p(v))\) where \(P_1\) and \(P_2\) have the same color \(j\), and \(P_3\) and \(P_4\) have the same color \(j'\) with \(j \neq j'\). Again, there are at most \(L_{\text{max}} - 1\) paths through each of the links \((v_i, p(v)), (p(v), v_i), i \in \{1, 2\}\). Since the number of paths through \((v_2, v_1)\) and \((v_1, v_2)\) is maximal, there exists at least one path through \((v_2, v_1)\) and one through \((v_1, v_2)\). Clearly, these two paths can be colored with a third color \(j''\). Eliminating these six paths generate an instance with capacity \(L_{\text{max}} = 2\). Furthermore, we have here \(f_i = f'_i + 2\), \(i \in \{0, 1, 2\}\) and \(f_2 = f'_2 + 3\).

If we apply these reductions iteratively, we obtain w.l.o.g. the following precoloring configuration (see also Figure 6):

1. there are \(a \in \mathbb{N}_0\) pairs of paths through the links \((p(v), v_1)\) and \((v_1, p(v))\) where each pair is colored with the same color \(i \in \{1, \ldots, a\}\),

<table>
<thead>
<tr>
<th>(L_{\text{max}})</th>
<th>(k)</th>
<th>(k')</th>
</tr>
</thead>
<tbody>
<tr>
<td>(3\ell)</td>
<td>(4\ell)</td>
<td>(5\ell)</td>
</tr>
<tr>
<td>(3\ell + 1)</td>
<td>(4\ell + 1)</td>
<td>(5\ell + 1)</td>
</tr>
<tr>
<td>(3\ell + 2)</td>
<td>(4\ell + 3)</td>
<td>(5\ell + 3)</td>
</tr>
</tbody>
</table>
(2) there are $b, c, d, e \in \mathbb{N}_0$ paths through $(p(v), v_1), (v_1, p(v)), (p(v), v_2), (v_2, p(v))$, respectively. These paths are colored pairwise differently and with colors different from $1, \ldots, a$.

We code this precoloring configuration as a vector $(a, b, c, d, e)$. The problem is to find an extension of this coloring for the paths through $(v_1, v_2)$ and for the paths through $(v_2, v_1)$. The maximum number of paths through these links is given by:

$$
m_{(1,2)} = \min(L_{\text{max}} - d, L_{\text{max}} - a - c),
$$

$$
m_{(2,1)} = \min(L_{\text{max}} - e, L_{\text{max}} - a - b).$$

Using the capacity bound $L_{\text{max}}$, the following properties are satisfied:

$$a + c + e \leq L_{\text{max}},$$

$$a + b + d \leq L_{\text{max}}.$$

The number of precolors $f_0$ is equal to $a + b + c + d + e$. In the following, we analyse several cases to show how the paths through $(v_1, v_2)$ and $(v_2, v_1)$ can be colored efficiently. On the assumption that $f_0 \leq k$ (see Table 1), we prove that the generated color numbers $f_1, f_2 \leq k$ and $f_2 \leq k'$.

For the analysis of the precoloring configurations it is helpful to use a third type of a reduction.

**Reduction (3):** There exists a number $h \in \mathbb{N}$ and a set $C$ of paths between $p(v)$ and $v_1$, between $p(v)$ and $v_2$ and between $v_1$ and $v_2$ such that

(1) the paths between $p(v)$ and $v_1$ and the paths between $p(v)$ and $v_2$ are precolored with $h$ colors,

(2) there exists an $h$-coloring of the entire set $C$ that extends the precoloring,

(3) the remaining paths form also a precoloring configuration with capacity $L'_{\text{max}} = L_{\text{max}} - h$.

Using these properties, we have $f_i' = f_i - h (0 \leq i \leq 2)$ and $f_2' = f_2 - h$.

### 3.2. Idea of the proof

For the idea of our proof we refer to Figure 7. We start with a local extension problem and transform it using reductions of type (1) and (2) into a precoloring configuration. Lemmas 3.10 - 3.14 show us how we can transform a general precoloring configuration (using reductions of type (3)) into one of the following distinguished configurations:

(1) $(0, b, c, d, e)$,

(2) $(a, b, c, d = a + c, e = a + b)$,

(3) $(a, 0, c, d = a + c, e < a)$ and $(a, b, 0, d < a, e = a + b)$.

Moreover, we give efficient colorings of the paths between $v_1$ and $v_2$ (compatible with the precolored paths) for these three distinguished configurations (see also Lemmas 3.2 - 3.8). Then, we consider instances of the local extension problem that can be reduced to a distinguished precoloring configuration. Let $f_0$ be the number of precolors and $L$ be the capacity of the general problem. We show that it is sufficient to transform first a general instance using the reductions (1) – (3) and then to solve the special precoloring configuration problem. We call a coloring of the general problem a $(4, 5)$ - coloring if
Figure 7. The transformation of a general instance into a distinguished configuration.
(1) \( L = 3\ell, f_0 \leq 4\ell \) implies \( f_1, f_2 \leq 4\ell \) and \( f_\Sigma \leq 5\ell \),

(2) \( L = 3\ell + 1, f_0 \leq 4\ell + 1 \) implies \( f_1, f_2 \leq 4\ell + 1 \) and \( f_\Sigma \leq 5\ell + 1 \),

(3) \( L = 3\ell + 2, f_0 \leq 4\ell + 3 \) implies \( f_1, f_2 \leq 4\ell + 3 \) and \( f_\Sigma \leq 5\ell + 3 \).

For the three distinguished precoloring configurations, we can apply the Theorems 3.5, 3.7 and 3.9 and we obtain such (4, 5) - colorings.

3.3. Analysis of distinguished configurations

In the following, we analyse the distinguished precoloring configurations and show how the new paths from \( v_1 \) to \( v_2 \) and from \( v_2 \) to \( v_1 \) can be colored. We use \( L \) instead of \( L_{\text{max}} \). For a \((v_i, v_j)\) path with \( i, j \in \{1, 2\}, i \neq j \) we can use a precolor of a \((p(v), v_i)\) path or a precolor of a \((v_j, p(v))\) path. The goal is to use these precolors in an efficient way.

**Lemma 3.2** For a precoloring configuration \((a, b, c, d, e)\) with \( a = 0, d = c, e = b \) we can compute a compatible coloring of the \((v_1, v_2)\) and \((v_2, v_1)\) paths such that \( f_1, f_2 \leq L + \left[ \frac{L + e}{2} \right] \) and \( f_\Sigma \leq \max(L + \max(b, c), f_0) \).

**Proof:** We may assume \( b \geq c \). This implies that \( L - b \leq L - c \). Using \( a = 0, d = c \) and \( e = b \) we have \( L - c \) paths from \( v_1 \) to \( v_2 \), and \( L - b \) paths from \( v_2 \) to \( v_1 \).

If \( b - c \) is even, we choose \( \frac{L - b}{2} \) precolors of \((p(v), v_1)\) paths and \( \frac{L - c}{2} \) precolors of \((v_2, p(v))\) paths, and use these colors for \( b - c \) paths from \( v_1 \) to \( v_2 \). After that, using \( b - \frac{L - b}{2} = c + \frac{L - c}{2} \), we obtain the precoloring configuration \((0, c + \frac{L - c}{2}, c, c + \frac{L - c}{2})\) with \( L - b \) paths from \( v_1 \) to \( v_2 \), and vice versa.

Next, we choose \( \min(2c, L - b) \) precolors for the \((v_1, v_2)\) paths, and \( \min(2c, L - b) \) precolors for the \((v_2, v_1)\) paths. To do this, we take \( \min(c, \left[ \frac{L - b}{2} \right] ) \) precolors of \((v_i, p(v))\) paths and \( \min(c, \left[ \frac{L - b}{2} \right] ) \) precolors of \((v_i, p(v))\) paths for \( i = 1, 2 \), respectively.

**Case 1:** \( 2c > L - b \). In this case, we need no new colors and obtain \( f_1 = f_2 = b + c + \frac{L - c}{2} + L - b = L + \frac{L - c}{2} \) and \( f_\Sigma = f_0 \).

**Case 2:** \( 2c \leq L - b \). Here, we take \( L - b - 2c \) new colors and conclude \( f_1 = f_2 = b + c + \frac{L - c}{2} + 2c + (L - b - 2c) = L + \frac{L - c}{2} \) and \( f_\Sigma = 2(b + c) + (L - b - 2c) = L + b = L + \max(b, c) \).

If \( b - c \) is odd, then we use in the first step \( \frac{L - b - 1}{2} \) precolors of \((p(v), v_1)\) paths and \( \frac{L - b - 1}{2} \) precolors of \((v_2, p(v))\) paths. Then, we obtain the precoloring configuration \((0, c + \frac{L - b - 1}{2}, c, c + \frac{L - b - 1}{2})\). Using the same arguments as above, we get \( f_1 = L + \left[ \frac{L - c}{2} \right] , f_2 = L + \left[ \frac{L - c}{2} \right] \) and \( f_\Sigma = f_0 \) or \( f_\Sigma = L + \max(b, c) \).

We notice that \( f_2 = f_0 \) or \( f_\Sigma = L + \max(b, c) \leq L + \frac{L}{2} \). If \( b - c \) is even, then \( f_1 = f_2 = L + \frac{L - c}{2} = L + \frac{L}{2} \).

With similar techniques we can prove results for other configurations \((a, b, c, d, e)\) with \( a = 0 \) (see also our full paper).

**Lemma 3.3** For a precoloring configuration \((a, b, c, d, e)\) with \( a = 0, d \leq c, e \leq b \) we can compute a compatible coloring of the \((v_1, v_2)\) and \((v_2, v_1)\) paths such that \( f_1, f_2 \leq \max(L + \left[ \frac{L - b}{2} \right] , f_0) \) and \( f_\Sigma \leq \max(L + \max(b, c), f_0) \).

**Lemma 3.4** For a precoloring configuration \((a, b, c, d, e)\) with \( a = 0, d \leq c, e > b \) we can compute a compatible coloring of the \((v_1, v_2)\) and \((v_2, v_1)\) paths such that \( f_1, f_2 \leq \max(L + \left[ \frac{L - b}{2} \right] , f_0) \) and \( f_\Sigma \leq \max(L + \max(b, c), f_0) \).

**Theorem 3.5** If an instance of the local extension problem can be transformed using reductions (1), (2) and (3) into a precoloring configuration \((0, b, c, d, e)\) then we obtain a \((4, 5)\) - coloring.

**Proof:** Let us assume that we have a reduction with \( h \) colors into a precoloring configuration with capacity \( L' = L - h \) and \( f_0' = f_0 - h \) precolors. Furthermore, we know that \( f_1 = f_1' + h, f_2 = f_2' + h \) and we can bound \( f_\Sigma \leq f_\Sigma' + \left[ \frac{h}{2} \right] \); the last inequality holds by reason of the second reduction. Lemmas 3.2, 3.3 and 3.4 imply that

\[
\begin{align*}
f_1' &\leq \max(L' + \left[ \frac{\min(b, c) + \min(c, d)}{2} \right] , f_0'), \\
f_2' &\leq \max(L' + \max(\min(b, e), \min(c, d)) , f_0').
\end{align*}
\]

W.l.o.g we may assume \( b \leq e \) and \( c \leq d \). First, we analyse the value of \( f_\Sigma' \). Using the fact that \( f_\Sigma' \leq \max(L' + \left[ \frac{L}{2} \right] , f_0' \), we get \( f_\Sigma \leq \max(L' + h + \left[ \frac{\min(b, e)}{2} \right] , f_0' + \left[ \frac{b}{2} \right] ) = \max(L + \left[ \frac{L}{2} \right] , f_0 + \left[ \frac{L}{2} \right] ) \). Finally, suppose \( f_0 + \left[ \frac{L}{2} \right] \geq L + \left[ \frac{L}{2} \right] \).
Case 1: $h \leq 2\ell$. We have $f_2 \leq f_0 + \frac{h}{4} \leq f_0 + \ell$. The rest is easy to verify.

Case 2: $L = 3\ell$, $h = 2\ell + i$ with $i > 0$. Then, the reduced capacity $L'$ is $\ell - i$ and $f_0' \leq 2(\ell - i)$. We conclude $f_2 \leq f_0' + \frac{h}{4} \leq 5\ell - 2i + \frac{h}{4} < 5\ell$.

Cases 3, 4: $L = 3\ell + 1$ and $L = 3\ell + 2$. The proof goes in the same way as in case 2.

The other case $L + \frac{h}{4} \geq f_0 + \frac{h}{2}$ can be proved by a simple case analysis ($L = 3\ell, \ldots, 3\ell + 2$).

Next, we study the behaviour of $f_1$ and $f_2$. If $b + c$ is even, it holds $f_1, f_2 \leq \max(L' + \frac{h}{4}, f_0)$. Using $f_1 = f_1 + h$ for $i \in \{0, 1, 2\}$ and $L = L' + h$, we obtain $f_1, f_2 \leq \max(L + \frac{h}{4}, f_0)$. Again, a case analysis shows the assertion for $f_1, f_2$.

Suppose now that $b + c$ is odd. Since $b \leq c$ and $e = d$, it holds $2(b + c) \leq f_0 \leq f_0$. Suppose now that $b + c$ is odd. Since $b \leq c$ and $e = d$, it holds $2(b + c) \leq f_0 \leq f_0$.

Case 1: $L = 3\ell$. Since $2(b+c) \leq f_0 \leq 4\ell$ and $b+c$ is odd, we have $b+c \leq 2\ell - 1$. Therefore, $f_1, f_2 \leq L' + \frac{b+c+1}{2} + h \leq 4\ell$.

Case 2: $L = 3\ell + 1$. In this case, we get $b + c \leq 2\ell - 1$ and, therefore, $f_1, f_2 \leq 4\ell + 1$.

Case 3: $L = 3\ell + 2$. In this case, we obtain $b + c \leq 2\ell + 1$ and, therefore, $f_1, f_2 \leq 4\ell + 3$. \(\square\)

By space consideration, the used colorings of the next two configurations and the proofs are omitted and given in the full paper.

**Lemma 3.6** For a precoloring configuration $a, b, c, d, e$ with $d = a + c$, $e = a + b$ we can compute a compatible coloring of the $(v_1, v_2)$ and $(v_2, v_1)$ paths such that $f_1, f_2 \leq L + \left[ \frac{a+b+c}{2} \right]$ and $f_2 \leq \max(L + a, \max(b, c), L + \left[ \frac{a+b+c}{2} \right], f_0)$.

**Theorem 3.7** If an instance of the local extension problem is transformed using reductions (1), (2) and (3) into a precoloring configuration $a, b, c, d, e$ with $d = a + c$ and $e = b + a$ then we obtain a $(4, 5)$-coloring.

**Lemma 3.8** For a precoloring configuration $a, 0, c, d, e$ with $d = a + c$, $e < a$ we can compute a compatible coloring of the paths between $v_1$ and $v_2$ such that $f_1, f_2 \leq L + \left[ \frac{a+c}{2} \right] \leq \frac{L}{2} + \left[ \frac{a+c}{4} \right]$ and $f_2 \leq \max(L + a + c, L + a + \left[ \frac{a+c}{2} \right])$. Furthermore, $f_2 \leq L + \left[ \frac{a+c}{2} \right]$.

**Theorem 3.9** If an instance of the local extension problem is transformed using reductions (1), (2) and (3) into a precoloring configuration $a, 0, c, d, e$ with $d = a + c$ and $e < a$, then we get a $(4, 5)$-coloring.

### 3.4. Reductions into distinguished configurations

Next, we reduce the other precoloring configurations into simpler ones.

**Lemma 3.10** A precoloring configuration $a, b, c, d, e$ with $d \leq c$, $e \leq b$ can be transformed using reductions (3) into the precoloring configuration $(0, b, c, d, e)$.

**Proof:** We define $C$ by a pair of paths between $p(r)$ and $v_1$ where each pair is precolored with the same color. Since $d \leq c$ and $e \leq b$, this defines a reduction of type (3) and we get a $(0, b, c, d, e)$ configuration. \(\square\)

Using similar techniques, we can reduce the remaining configurations (see also our full paper).

**Lemma 3.11** A precoloring configuration $a, b, c, d, e$ with $d \geq c$, $e \geq b$, $a + c \geq d$, $a + b \geq e$ can be transformed using a reduction (3) into a $(a', b', c', a' + c', a' + b')$, $a (d', 0, c, d' + c, e < a')$ or $a (d', b, 0, d < d', a' + b)$ configuration.

**Lemma 3.12** A precoloring configuration $a, b, c, d, e$ with $d \geq c$, $e \geq b$, $e + a \geq d$, $b + a < e$ can be transformed using a reduction (3) into a $(a, b, c, d, e')$ precoloring configuration with $c + a \geq d$ and $b + a = e'$.

We notice that the case with $a + c < d$ and $a + b \geq e$ is symmetric to the case in Lemma 3.12.

**Lemma 3.13** A precoloring configuration $a, b, c, d, e$ with $d \geq c$, $e \geq b$, $a + c < d$, $a + b < e$ can be transformed using a reduction (3) into a $(a, b, c, d', e')$ precoloring configuration with $d' = a + c$, $e' \geq a + b$ or $d' \geq a + c$, $e' = a + b$, or we can compute a compatible coloring with $f_1, f_2, f_2 \leq f_0$.

**Lemma 3.14** A precoloring configuration $a, b, c, d, e$ with $d \leq c$, $e > b$ can be transformed using a reduction (3) into a $(a', b, c', d', e')$ configuration with $a' + e' = d'$, $d' = b + e'$ or into a $(a', b, 0, d, e')$ configuration with $e' = a' + b$ and $d < a'$.

If we put everything together, the proof of our main Theorem 2 is finished. We notice that the proof contains also an approximation algorithm to get such a coloring. To do this,
(1) apply the reductions of type (1), (2) and get a precoloring configuration,

(2) follow the arcs in Figure 7 and apply the reductions of type (3) corresponding to the Lemmas 3.10 - 3.14,

(3) if we reach a leaf that corresponds to a distinguished configuration then we compute a balanced coloring as described in the Lemmas 3.2 - 3.8 for this precoloring configuration.

4. Lower bound for local greedy algorithms

Finally, we show that every local greedy algorithm requires \( k \) wavelengths per link and \( k' \) wavelengths altogether, where \( k \) and \( k' \) are the parameters from Table 1. In particular, no local greedy algorithm can do better than our algorithm from the previous section.

**Theorem 4.1** *Every local greedy algorithm even for binary trees requires at least \( \left\lceil \frac{2}{3} L_{max} \right\rceil \) wavelengths or colors.*

We construct recursively a tree with components as drawn in Figure 8. We start with \( f_0 = L \) colors between \( p(v) \) and \( v \) in the first component. The idea is to increase the number of colors recursively in the tree until we reach \( f_0 = k \). The approximation algorithm works only local at a node \( v \) with parent \( p(v) \) and children \( v_1 \) and \( v_2 \). In a local optimization step, the paths between \( v_1 \) and \( v_2 \) get fixed colors. We know the colors of the paths between \( p(v) \) and \( v \) in each component and can use this information to separate paths with different colors. This can be avoided only by a non-local coloring algorithm.

Suppose that we have \( f_0 = L + a \) (with \( 0 \leq a \leq L \)) different colors between \( p(v) \) and \( v \). Then we can separate (as shown in Figure 8) these colors such that we obtain

(1) \( \left\lfloor \frac{a}{2} \right\rfloor \) paths from \( p(v) \) to \( v_1 \) and

(2) \( \left\lceil \frac{a}{2} \right\rceil \) paths from \( v_2 \) to \( p(v) \)

where these paths are precolored differently. To prove the existence of such a separation, we use the fact that there are \( L \) paths from \( p(v) \) to \( v \) and \( a \) paths from \( v_2 \) to \( p(v) \) that are colored differently. The remaining \( L - a \) paths from \( v \) to \( p(v) \) are colored with colors that occur in the first \( L \) paths. Therefore, there exist

(1) \( a \) paths from \( p(v) \) to \( v \) and \( a \) paths from \( v \) to \( p(v) \) that are colored differently and

(2) \( L - a \) pairs of paths between \( p(v) \) and \( v \) where each pair is colored with the same color.
For the separation above we use all paths of type (1). \( \lfloor \frac{L - a}{2} \rfloor \) paths from \( p(v) \) to \( v \) and \( \lceil \frac{L - a}{2} \rceil \) paths from \( v \) to \( p(v) \) where these paths are precolored differently (see also Figure 8 at node \( p(v) \)). Since \( \lfloor \frac{L - a}{2} \rfloor + a = \lfloor \frac{L}{2} \rfloor \), the existence of the separation is proved.

Furthermore (see Figure 8), we insert \( L - \lfloor \frac{L}{2} \rfloor \) new paths from \( v_2 \) to \( v_1 \) and \( L \) new paths from \( v_1 \) to \( v_2 \). This implies that every precoloring extension generates:

\[
E_c \geq L + \lfloor \frac{L}{2} \rfloor, \quad \max(f_1, f_2) \geq L + \lfloor \frac{L}{2} \rfloor.
\]

Moreover, there exists a precoloring extension with \( E_c = L + \lfloor \frac{L}{2} \rfloor \) and \( \max(f_1, f_2) = L + \lfloor \frac{L}{2} \rfloor \).

Next, we study in which cases we generate more colors at the child \( v_1 \) or the child \( v_2 \). To do this we analyze the inequality

\[
f_0' = L + \lfloor \frac{L}{2} \rfloor > f_0.
\]

Let \( L = 3\ell + i \) with \( 0 \leq i < 3 \) and \( f_0 = 4j + j' \) with \( 0 \leq j' < 4 \). Then, the inequality above is equivalent to

\[
3\ell + i + \lfloor \frac{4j + j'}{2} \rfloor > 4j + j'.
\]

By case analysis \( f_0 = 4j \ldots, 4j + 3 \), we can show that the value \( f_0' > f_0 \) if and only if \( f_0 < 4\ell \) \( (f_0 < 4\ell + 1, f_0 < 4\ell + 3) \) for the capacity \( L = 3\ell \) \((L = 3\ell + 1, L = 3\ell + 2)\). Furthermore, the value \( f_0' \) is bounded by \( 4\ell, 4\ell + 1 \) or \( 4\ell + 3 \), respectively. If we use one component with \( f_0 = k \), then we get a coloring with the given parameters \( k \) and \( k' \).

5. Conclusion

In this paper, we have given a polynomial approximation algorithm with worst case ratio \( \lfloor \frac{L}{3} \rfloor \) for the wavelength routing problem restricted to binary trees. Furthermore, we have given an example with load 3 such that every algorithm needs at least five wavelengths. And finally, we have proved that no local greedy algorithm for the wavelength routing problem can go below \( \lfloor \frac{L}{3} \rfloor \). It is an interesting question whether there exists a (non-local) greedy algorithm with a better worst case performance. Recently, Erlebach and Jansen [4] (and independently Kaklamanis and Persiano [7]) have obtained a local greedy algorithm for general trees that routes a set of requests of maximum load \( I_{max} \) using at most \( \lfloor \frac{L}{3} \rfloor \) wavelengths.

Acknowledgement. I thank Thomas Erlebach (TU München) for his helpful comments and many fruitful discussions.

References