Alcom-FT Technical Report Series
ALCOMFT-TR-01-55

Wavelength Routing in All–Optical Tree Networks: A Survey*

Ioannis Caragiannis† Christos Kaklamanis‡ Pino Persiano§

March 5, 2001

Abstract

We study the problem of allocating optical bandwidth to sets of communication requests in all–optical networks that utilize Wavelength Division Multiplexing (WDM). WDM technology establishes communication between pairs of network nodes by establishing transmitter–receiver paths and assigning wavelengths to each path so that no two paths going through the same fiber link use the same wavelength. Optical bandwidth is the number of distinct wavelengths. Since state–of–the–art technology allows for a limited number of wavelengths, the engineering problem to be solved is to establish communication between pairs of nodes so that the total number of wavelengths used is minimized; this is known as the *wavelength routing problem*.

In this paper, we survey recent advances in bandwidth allocation in tree–shaped WDM all–optical networks:

– We present hardness results and lower bounds for the general problem and the special case of symmetric communication.

– We give the main ideas of deterministic greedy algorithms and study their limitations.

– We demonstrate how we can achieve optimal and nearly–optimal bandwidth utilization in networks with wavelength converters using simple algorithms.

– We also present recent results about the use of randomization for wavelength routing.

1 Introduction

Optical fiber is rapidly becoming the standard transmission medium for backbone networks, since it can provide the required data rate, error rate and delay performance necessary for high speed networks of next generation [21, 41]. However, data rates are limited in opto–electronic networks by the need to convert the optical signals on the fiber to electronic signals in order to process them at the network nodes. Although electronic parallel processing techniques are capable, in principle, to meet future high data rate requirements, the opto–electronic conversion is itself expensive. Thus, it appears likely that, as optical technology improves, simple optical processing will remove the need for optoelectronic conversion. Networks using optical transmission and maintaining optical data paths through the nodes are called *all–optical networks*.

Optical technology is not yet as mature as conventional technology. There are limits as to how sophisticated optical processing at each node can be done.

*Work supported in part by the European Union under IST FET Project ALCOM–FT and Improving RTN Project ARACNE, Progetto MURST Cofinanziato REACTION, and research grants from CNR and the Università di Salerno.

†Computer Technology Institute and Department of Computer Engineering and Informatics, University of Patras, 26500 Rio, Greece.E–mail: caragian@cti.gr

‡Computer Technology Institute and Department of Computer Engineering and Informatics, University of Patras, 26500 Rio, Greece.E–mail: kakl@cti.gr

§Dipartimento di Informatica ed Applicazioni, Università di Salerno, 84081 Baronissi, Italy. E–mail: giuper@dia.unisa.it

Multiwavelength communication [21, 41] is the most popular communication technology used on optical networks. Roughly speaking, it allows to send different streams of data on different wavelengths along an optical fiber. Multiwavelength communication is implemented through *Wavelength Division Multiplexing* (WDM). WDM takes all data streams traveling on an incoming link and route each of them to the right outgoing link, provided that each data stream travels on the same wavelength on both links.

In a WDM all–optical network, once the data stream has been transmitted in the form of light, it continues without conversion to electronic form until it reaches its destination. For a packet transmission to occur, a transmitter at the source must be tuned to the same wavelength as the receiver at the destination for the duration of the packet transmission and no data stream collision may occur at any fiber.

We model the underlying fiber network as a directed graph, where vertices are the nodes of the network and links are optical fibers connecting nodes. Communication requests are ordered pairs of nodes, which are to be thought of as transmitter–receiver pairs. WDM technology establishes connectivity by finding transmitter–receiver directed paths and assigning a wavelength (color) to each path, so that no two paths going through the same link use the same wavelength.

Optical bandwidth is the number of available wavelengths. Optical bandwidth is a scarce resource. State–of–the–art technology allows some hundreds wavelengths per fiber in the laboratory, even less in manufacturing, and there is no anticipation for dramatic progress in the near future. At the state of the art there is no WDM all–optical network that uses the optical bandwidth in an efficient way. However, for a realistic use of WDM all–optical networks for long distance communication networks, it seems necessary a significant progress in the protocols for allocation of the available bandwidth. Thus, the important engineering problem to be solved is to establish communication between pairs of nodes so that the total number of wavelengths used is minimized; this is known as the *wavelength routing problem* [1, 34].

Given a pattern of communication requests and a corresponding path for each request, we define the *load* of the pattern as the maximum number of requests that traverse any fiber of the network. For tree networks, the load of a pattern of communication requests is well defined, since transmitter–receiver paths are unique. Clearly, for any pattern of requests, its load is a lower bound on the number of necessary wavelengths.

Theoretical work on optical networks mainly focuses on the performance of wavelength routing algorithms on regular networks using oblivious (predefined) routing schemes. We point out the pioneering work of Pankaj [34] who considered shuffle exchange, De Bruijn, and hypercubic networks. Aggarwal *et al.* [1] consider oblivious wavelength routing schemes for several networks. Raghavan and Upfal in [37] consider mesh–like networks. Aumann and Rabani [8] improve the bounds of Raghavan and Upfal for mesh networks and also give tight results for hypercubic networks. Rabani in [35] gives almost optimal results for the wavelength routing problem on meshes.

These topologies reflect architectures of optical computers rather than wide–area networks. For fundamental practical reasons, the telecommunication industry does not deploy massive regular architectures: backbone networks need to reflect irregularity of geography, non–uniform clustering of users and traffic, hierarchy of services, dynamic growth, etc. In this direction, Raghavan and Upfal [37], Aumann and Rabani [8], and Bermond *et al.* [12], among other results, focus on bounded–degree networks and give upper and lower bounds as functions of the network expansion.

However, wide–area multiwavelength technology is expected to grow arround the evolution of current networking principles and existing fiber networks. These are mainly SONET (Synchronous Optical Networking Technology) rings and trees [33]. In this sense, even asymptotic results for expander graphs do not address the above telecommunications scenario.

In this work, we consider tree topologies, with each edge of the tree consisting of two opposite directed fiber links. Raghavan and Upfal [37] considered trees with single undirected fibers carrying undirected paths. However, it has since becomes apparent that optical amplifiers placed on fiber will be directed devices. Thus, directed graphs are essential to model state–of–the–art technology.

In particular, we survey recent methods and algorithms for efficient use of bandwidth in trees considering arbitrary patterns of communication requests. All the results are given in terms of the load of the pattern of requests that have to be routed. Surveys on bandwidth allocation for more specific communication patterns like *broadcasting*, *gossiping*, and *permutation routing* can be found in [10, 25].

The rest of this paper is structured as follows. In Section 2, we formalize the wavelength routing problem and present hardness results and lower bounds for the general case and the special case of symmetric communication. In Section 3, we describe deterministic greedy algorithms and present the best known results on them. We briefly discuss the special case of symmetric communication in Section 4. In Section 5, we relax the model and introduce devices called *converters* which can improve bandwidth allocation with some sacrifice in network cost. In Section 6, we briefly describe the first randomized algorithm for the problem. We conclude, in Section 7, with a list of open problems.

2 Hardness Results and Lower Bounds

As we mentioned in Section 1, the load of the communication pattern is a lower bound on the number of necessary colors (wavelengths). The following question now arises. Given any communication pattern of load L, can we hope for a wavelength routing with no more than L wavelengths? The answer is negative for two reasons. The former is that this problem is $\mathcal{NP}$–hard. The latter is that there are patterns that require more than L wavelengths.

We formalize the problem as follows.

> WAVELENGTH ROUTING IN TREES
> INSTANCE: A directed tree T and a pattern of communication requests (i.e., a set of directed paths) P of load L.
> QUESTION: Is it possible to assign wavelengths (colors) from $\{1, 2, ..., L\}$ to requests of P in such a way that requests that share the same directed link are assigned different wavelength?

Intuitively, we may think of the wavelengths as colors and the wavelength routing problem as a coloring problem of directed paths. In the rest of the paper, we use the terms wavelength (wavelength routing) and color (coloring or proper coloring), interchangeably.

Erlebach and Jansen [16] have proved the following hardness result.

Theorem 1 (Erlebach and Jansen [16]) WAVELENGTH ROUTING IN TREES *is $\mathcal{NP}$–complete.*

Note that the above statement holds even if we restrict instances to arbitrary trees and communication patterns of load 3. The following result, which is due to Erlebach and Jansen [17] as well, applies to binary trees and communication patterns of arbitrary load.

Theorem 2 (Erlebach and Jansen [17]) WAVELENGTH ROUTING ON BINARY TREES *is $\mathcal{NP}$–complete.*

Thus, the corresponding optimization problems (minimizing the number of wavelengths) are $\mathcal{NP}$–hard, in general.

Now, we give the second reason why, given a communication pattern of load L, a wavelength routing with not much more than L wavelengths is infeasible.

Theorem 3 (Kumar and Schwabe [28]) *For any integer $l > 0$, there exists a communication pattern of load $L = 4l$ on a binary tree T that requires at least $5L/4$ wavelengths.*

The proof of the theorem is depicted in Figure 1.

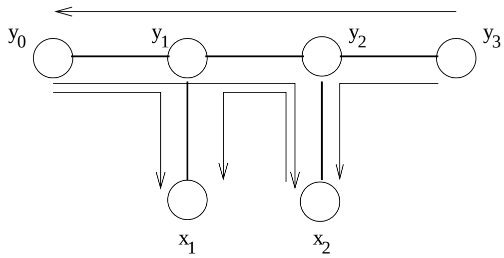

Figure 1: The communication pattern for the proof of Theorem 3. Each arrow represents $L/2 = 2l$ requests. Note that there exist $5L/2$ requests and no more than 2 can be assigned the same wavelength. This yields at least $5L/4$ necessary wavelengths.

As we will see below (Section 4), these statements hold for the special case of symmetric communication.

3 Greedy Algorithms

All known wavelength routing algorithms [15, 22, 23, 24, 28, 32] belong to a special class of algorithms, the class of *greedy* algorithms. We devote this section to their study. Given a tree network T and a pattern of requests P, we call greedy a wavelength routing algorithm that works as follows:

> Starting from a node, the algorithm computes a breadth–first (BFS) numbering of the nodes of the tree. The algorithm proceeds in phases, one per each node u of the tree. The nodes are considered following their BFS numbering. The phase associated with node u assumes that we already have a proper coloring where all requests that touch (i.e. start, end, or go through) nodes with numbers strictly smaller than u's have been colored and no other request has been colored. During this phase, the partial proper coloring is extended to one that assigns proper colors to requests that touch node u but have not been colored yet. During each phase, the algorithm does not recolor requests that have been colored in previous phases.

Thus, various greedy algorithms differ among themselves with respect to the strategies followed to extend the partial proper coloring during a phase. The algorithms in [23, 24, 28, 32] make use of complicated subroutines in order to extend the partial coloring during a phase; in particular, their subroutines include a reduction of the problem to an edge coloring problem on a bipartite graph. On the other hand, the algorithms in [15, 22] use much simpler methods to solve the wavelength routing problem on binary trees. The common characteristic for all these algorithms is that they are deterministic.

3.1 The Reduction to Constrained Bipartite Edge Coloring

The algorithms in [23, 24, 28, 32] reduces the coloring of a phase associated with node u to an edge coloring problem on a bipartite graph. In the following we describe this reduction.

Let v_0 be u's parent and let $v_1, \cdots, v_k$ be the children of u. The algorithm constructs the *bipartite graph associated with u* in the following way. For each node v_i, the bipartite graph has four vertices

W_i, X_i, Y_i, Z_i and the left and right partitions are $\{W_i, Z_i | i = 0, \cdots k\}$ and $\{X_i, Y_i | i = 0, \cdots k\}$. For each request of the tree directed out of some v_i into some v_j, we have an edge in the bipartite graph from W_i to X_j. For each request directed out of some v_i and terminating on u, we have an edge from W_i to Y_i. Finally, for each request directed out of u into some v_i, we have an edge from Z_i to X_i. See Figure 2. The above edges are called *real*. Notice that all edges that are adjacent to either X_0 or W_0 have already been colored, as they correspond to requests touching a node with BFS number smaller than u's (in this specific case, the requests touch u's parent) that have been colored at some previous phase. We call the edges incident to either X_0 or W_0 *color–forced edges.*

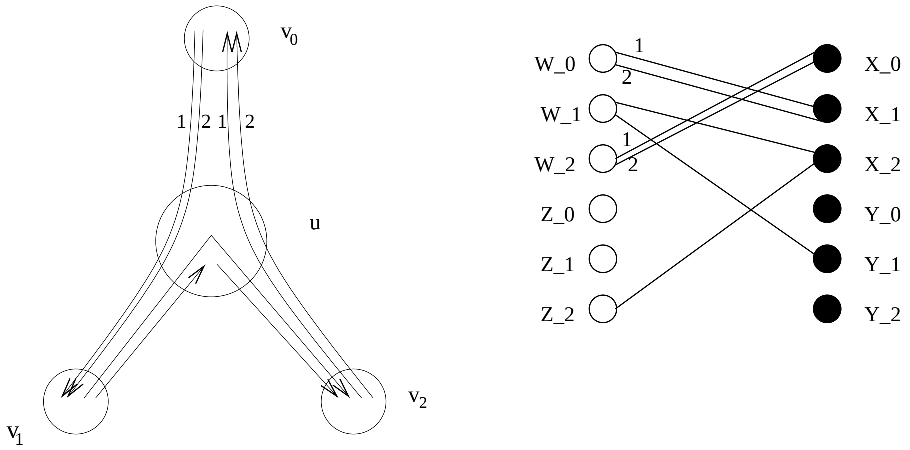

Figure 2: Requests touching node u and the relative bipartite graph (only real edges are shown).

Notice that no real edge extends across opposite vertices Z_i and Y_i or W_i and X_i. Indeed vertex Z_i has edges only to vertices of type X_i; on the other hand, an edge from W_i to X_i would correspond to a request in the tree going from c_i to itself. We call a pair of opposite vertices a *line*. Notice also that all vertices of the bipartite graph have degree at most L and, thus, it is possible to add *fictitious edges* to the bipartite graph so that all vertices have degree exactly L. The following claim holds.

Claim 4 (Mihail *et al.* **[32])** *Let P be a pattern of communication requests on a tree T. Consider a specific BFS numbering of the nodes of T, a node u and a partial coloring χ of the requests of P that touch nodes with BFS number smaller than u's. Then, any coloring of the edges of the bipartite graph associated with u corresponds to a proper coloring of the requests of P that touch node u.*

Thus, the problem of coloring requests is reduced to the problem of coloring the edges of an L–regular bipartite graph, under the constraint that some colors have already been assigned to edges adjacent to W_0 and X_0. We call this problem an *α–constrained bipartite edge coloring problem* on an L–regular bipartite graph. The parameter α denotes that edges incident to nodes W_0 and X_0 have been colored with αL colors.

The objective is to extend the coloring to all the edges of the bipartite graph. We first describe a simple (and weak) solution to the problem. Then, we briefly describe the main ideas that lead to an improved solution of the problem in [24].

We call *single* colors the colors that appear only in one color–forced edge and double colors the colors that appear in two color–forced edges (one incident in W_0 and one incident in X_0). We denote by S and D the number of single and double colors, respectively. Clearly, $2D + S = 2L$.

Now we decompose the L–regular bipartite graph into L matchings (this can be done in polynomial time; see [11]). At least $S/2$ of these matchings have at least one single color in one of the two colored–forced edges. Thus, we can use this single color to color the uncolored edges of the matching. The uncolored edges of the matchings that have no color–forced edges colored with a single color are colored with extra colors (one extra color per matching). In total, we have

$$D + S + L - S/2 \leq L + D + S/2 = 2L$$

colors. By using the simple way for solving the α–constrained bipartite edge coloring problem during each phase, we obtain a simple greedy wavelength routing algorithm that uses at most $2L$ colors (wavelengths) for any pattern of requests of load L.

The works [23, 24, 28, 32] give better solutions to this problem. In order to obtain improved results, they either consider matchings in pairs and color them in sophisticated ways using detailed potential and averaging arguments for the analysis [28, 32] or partition matchings into groups which can be colored and accounted for indepedently [23, 24]. In particular, Kaklamanis *et al.* [24] solve the problem proving the following theorem.

Theorem 5 (Kaklamanis *et al.* **[24])** *For any $\alpha \in [1, 4/3]$ and integer $L > 0$, there exists a polynomial time algorithm for the α–constrained bipartite edge coloring problem on an L–regular bipartite graph which uses at most $\left(1 + \frac{\alpha}{2}\right) L$ total colors and at most $4L/3$ colors per line.*

The interested reader may refer to the papers [23, 24, 28, 32] for detailed description of the techniques. Note that one might think of bipartite edge coloring problems with different constraints. Tight bounds on the number of colors for more generalized constrained bipartite edge coloring problems can be found in [14].

Using the coloring algorithm presented in [24] for $\alpha = 4/3$ as a subroutine, the wavelength routing algorithm maintains at each phase the following two invariants:

I. The total number of colors is no greater than $5L/3$.

II. The number of colors seen by two opposite directed links is at most $4L/3$.

In this way, the following result can be proved.

Theorem 6 (Kaklamanis *et al.* **[24], see also [18])** *There exists a polynomial time greedy algorithm which routes any pattern of communication requests of load L on a tree using at most $5L/3$ wavelengths.*

3.2 A Lower Bound Technique for Greedy Algorithms

In this section, we present a lower bound technique for greedy wavelength routing algorithms. We first briefly describe the technique; then we give the best known statement for deterministic greedy algorithms.

The technique is based on an adversary argument. An adversary algorithm $\mathcal{ADV}$ is exhibited that constructs a communication pattern for which it can be proved that there exists a lower bound on the number of colors used by any greedy algorithm.

The adversary $\mathcal{ADV}$ constructs the communication pattern P in an incremental way visiting the vertices of the tree according to a BFS visit. At each vertex v, the $\mathcal{ADV}$ deals with the set of requests traversing one of the two parallel directed links between u and its parent $p(u)$. For each *downward* request p (that is, for each request including the directed link $(p(u), u)$), $\mathcal{ADV}$ has two options:

1. do nothing; i.e., make p stop at u;

2. propagate p to the left child $l(u)$ by appending arc $(u, l(u))$ to p.

Similarly, for each *upward* request p (that is, for each request including the directed link $(u, p(u))$), $\mathcal{ADV}$ has two options:

1. do nothing; i.e. make p start from u;

2. make p originate from the right child $r(u)$ by pre-pending arc $(r(u), u)$ to p;

Moreover, the adversary algorithm $\mathcal{ADV}$ can introduce requests between the two children of u (see Figure 3). Initially, these requests will consist of only two arcs (from a child to u and from u to the other child) and can be augmented when the adversary reaches the children of u.

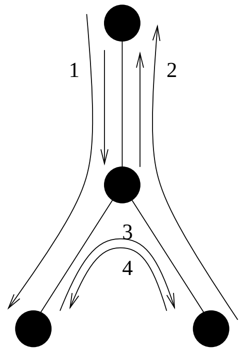

Figure 3: The construction for the lower bound for greedy algorithms.

At each step, since the adversary $\mathcal{ADV}$ may know how a greedy algorithm performs the coloring, it can choose to augment requests in such a way that the number of common colors used in downward requests from $p(u)$ to $l(u)$ and upward requests from $r(u)$ to $p(u)$ is small. In this way, no matter what a greedy algorithm can do, both the total number of colors and the number of colors seen by the opposite directed links between u and its children will increase.

By constructing such an adversary for deterministic greedy algorithms, Kaklamanis el al. [24] (see also [18]) prove the following lower bound. The same lower bound technique can be used for greedy algorithms that use randomization (see Section 6).

Theorem 7 (Kaklamanis *et al.* [24], see also [18]) *Let $\mathcal{A}$ be a deterministic greedy wavelength routing algorithm in trees. There exists an algorithm $\mathcal{ADV}$ which, on input $\delta > 0$ and integer $L > 0$, outputs a binary tree T and a pattern of communication requests P of load L on T, such that $\mathcal{A}$ colors P with at least $(5/3 - \delta)L$ colors.*

Thus, the greedy algorithm presented in [24, 18] is best possible within the class of deterministic greedy algorithms. In Section 6 we demonstrate how randomization can be used to beat the barrier of 5/3 at least on binary trees.

4 Symmetric Communication

In this section we consider the special case of patterns of symmetric communication requests, i.e., for any transmitter–receiver pair of nodes (v_1, v_2) in the communication pattern, its symmetric pair (v_2, v_1) also belongs to the communication pattern.

We can restrict the wavelength routing of patterns of symmetric communication requests to use the same wavelength for each pair of symmetric requests. Then, we can solve the wavelength routing problem using the algorithm of Raghavan and Upfal [37] for the undirected version of the problem.

This algorithm is deterministic and greedy. In this way, we can route any pattern of symmetric communication requests of load L using at most $3L/2$ wavelengths. For undirected patterns of requests, this bound is tight in general, i.e., there exist patterns of undirected requests of load L that cannot be routed with less than $3L/2$ wavelengths. The following two questions now arise.

- Can we improve this bound by assigning wavelengths to each request independently?
- Is the wavelength routing problem easier than in the general case if we restrict the input instances to patterns of symmetric requests?

Although we are not aware of complete answers for these questions, the following discussion shows some inherent similarities and differences between the general wavelength routing problem and the case where the input instance is restricted to patterns of symmetric communication requests.

For these patterns, Caragiannis *et al.* [13] have proved some interesting statements (lower bounds). Both $\mathcal{NP}$–completeness results (Theorems 1 and 2) hold in the case of symmetric patterns of communication requests [13]. Notice that the lower bound of Figure 1 apply to non–symmetric patterns of requests. A similar lower bound (but much more complicated than the one of Figure 1) holds for symmetric communication patterns as well.

Theorem 8 (Caragiannis *et al.* **[13])** *For any $\delta > 0$ and integer $l > 0$, there exists a binary tree T and a pattern of symmetric communication requests P with load $L = 4l$ on T, such that no algorithm can route P using less than $(5/4 - \delta)L$ wavelengths.*

Does Theorem 8 indicate that the symmetric version of the problem is as "hard" as the general one? The following result gives evidence for a negative answer. Notice that for symmetric communication patterns, there exists an algorithm which may route requests in such a way that each pair of opposite directed fiber links sees at most L wavelengths. This can be done in a trivial way if we consider two symmetric requests as an undirected one and use any algorithm for the undirected problem. However, this is not the case for the non–symmetric problem.

Theorem 9 (Caragiannis *et al.* **[13])** *For any integer $l > 0$, there exists a tree T and a pattern of communication requests with load $L = 8l$ that cannot be routed in such a way that any pair of opposite directed links of T sees less than $9L/8$ wavelengths.*

5 Wavelength Conversion

In Section 3, we saw that (deterministic) greedy wavelength routing algorithms in tree networks cannot use, in the worst case, less than $5L/3$ wavelengths to route sets of communication requests of load L, resulting to 60% utilization of available bandwidth. Furthermore, even if a better (non-greedy or randomized) algorithm is discovered, we know by Theorem 3 that there exist communication patterns of load L that require $5L/4$ wavelengths, meaning that 20% of the available bandwidth across the fiber links will remain unutilized.

The inefficiency of greedy algorithms in the allocation of the bandwidth is due to the fact that greedy algorithms color requests going through node u without "knowing" which requests go through its children.

Thus, if we are seeking better utilization of optical bandwidth, we have to relax some of the constraints of the problem. In particular, *wavelength converters* allow to relax the restriction that a request has to use the same wavelength along the whole request from the transmitter to the receiver. For example, if we can change wavelengths assigned to requests we can correct at u the "bad" assignments made by the greedy algorithm and improve bandwidth utilization (see Figure 4).

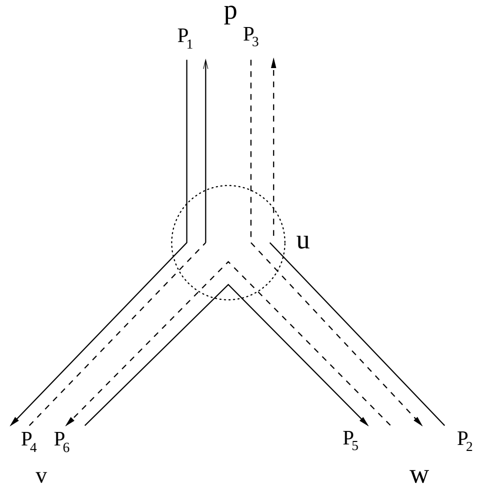

Figure 4: An example of greedy coloring with wavelength conversion. The wavelengths assigned to requests P_1 and P_4 are changed at u in order to make possible to route all requests with two wavelengths.

A possibility would be to convert the optical signal into electronic form and to retransmit it at a different wavelength. If there is no restriction on the wavelengths on which the message can be retransmitted, then it is possible to route all patterns of communication requests of load L with L wavelengths. In fact, in this case the assignment of wavelengths to requests on a link of the network is independent from the wavelengths assigned to the same requests on the other links.

However, converting optical signals to electronic signals has the drawback of wasting the benefits of using optical communication. Recently, a new technology has been proposed that allows to change the wavelength of an optical signal without converting it into electronic form. Wavelength converters have been designed and constructed [48]. A wavelength converter, placed at a node of the network, can be used to change the wavelengths assigned to requests traversing that node. The effects of wavelength conversion have been extensively studied in different models and it has been proved that it can dramatically improve the efficiency in the allocation of the optical bandwidth [9, 27, 38, 39, 40, 43, 47]. However, wavelength conversion is a very expensive technology and it is not realistic to assume to give to all the nodes of the network the capability of changing wavelengths to all requests. This motivates the study of all-optical networks that allow for some form of restricted wavelength conversion.

In the literature, two prevalent approaches are used to address WDM optical networks with wavelength conversion: *sparse conversion* and *limited conversion*. In a network with sparse wavelength conversion, only a fraction of the nodes are equipped with wavelength converters that are able to perform an arbitrary number of simultaneous conversions; in a network with limited conversion, instead, each node hosts wavelength converters, but these devices can perform a limited number of conversions.

Sparse conversion optical networks have been considered in [26, 39, 43, 46]. Wilfong and Winkler [46] consider the problem of minimizing the number of nodes of a network that support wavelength conversion in order to route any communication pattern using a number of wavelengths equal to the optimal load. They prove that the problem is $\mathcal{NP}$–hard. Kleinberg and Kumar [26] present a 2–approximation to this problem for general networks, exploiting its relation to the problem of computing the minimum vertex cover of a graph. Ramaswami and Sasaki [39] show that, in a ring network, a simple converter is sufficient to guarantee that any pattern of requests of load L can be routed with L wavelengths. Subramaniam *et al.* [43] give heuristics to allocate wavelengths, based on probabilistic models of communication traffic.

Variants of the limited conversion model have been considered by Ramaswami and Sasaki [39], Yates *et al.* [47], and Lee and Li [29]. Ramaswami and Sasaki [39] propose ring and star networks with limited wavelength conversion to support communication patterns efficiently. Although they address the undirected case, all their results for rings translate to the directed case as well. Furthermore, they propose algorithms for bandwidth allocation in undirected stars, trees, and networks with arbitrary topologies for the case where the length of requests is at most two.

5.1 The Network Model

In our network model, some nodes of the network host wavelength converters. A wavelength converter can be modeled as a bipartite graph $G = (X, Y, E)$. Each one of the sets of vertices X and Y have one vertex for each wavelength and there is an edge between vertices $x \in X$ and $y \in Y$ if and only if the converter is capable of converting the wavelength corresponding to x to the wavelength corresponding to y. For example, a full converter corresponds to a complete bipartite graph and a fixed conversion converter corresponds to a bipartite graph where the vertices of U have degree one. Some examples of wavelength converters are depicted in Figure 5.

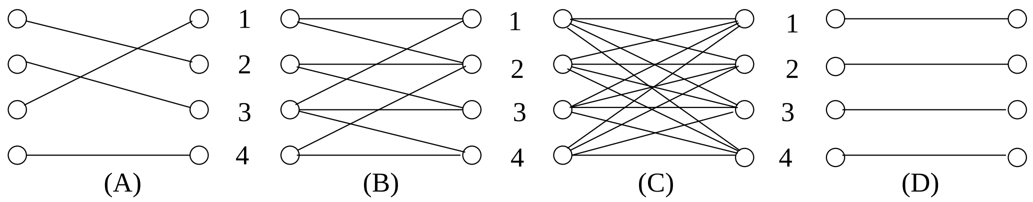

Figure 5: Wavelength conversion bipartite graphs: (A) fixed conversion, (B) partial conversion, (C) full conversion, and (D) no conversion.

In order to increase network performance without tremendous increase in cost, we mainly use converters of limited functionality. The term limited reflects the fact that the converters are simple according to two measures: their *degree* and their *size*. We proceed now to define these two measures.

Definition 10 *The degree of a converter is the maximum degree of the vertices of its corresponding bipartite graph.*

Definition 11 *The size of a converter is the number of edges in the corresponding bipartite graph.*

Let u be a node that hosts converters: each converter located at u is assigned to a pair of incoming and outgoing directed links adjacent to u. This converter will be used only to change colors assigned to requests containing the two directed links, while traversing u. Thus, a request may have one color on the incoming link and a different color on the outgoing link. In other words, the connection request corresponding to the request may travel on a wavelength on the segment ending at u and on a different wavelength on the segment starting from u.

In the discussion that follows, we consider three models of limited conversion networks, differing in the number of converters placed at nodes and in the placement pattern. Denote by d the degree of u and by p the parent of u. The models we consider are the following:

all-pairs There is one converter for each pair of incoming and outgoing links adjacent to u. Thus, the number of converters at u is $d(d-1)/2$ and we can change color to all requests traversing u.

top-down For each child v of u, there is a converter between links (p, u) and (u, v) and another converter between links (v, u) and (u, p). The number of converters at u is $2(d-1)$ and we can change color only to requests coming from or going to p.

down For each child v of u, there is a converter between links (p, u) and (u, v) and a converter between links (w, u) and (u, v), for each child w of u different from v. The number of converters at u is $(d-1)^2$ and we can change color only to requests going from p to a descendant of u or to requests traversing two distinct children of u.

In Figure 6, it is shown how converters are positioned at a node of a limited conversion binary tree.

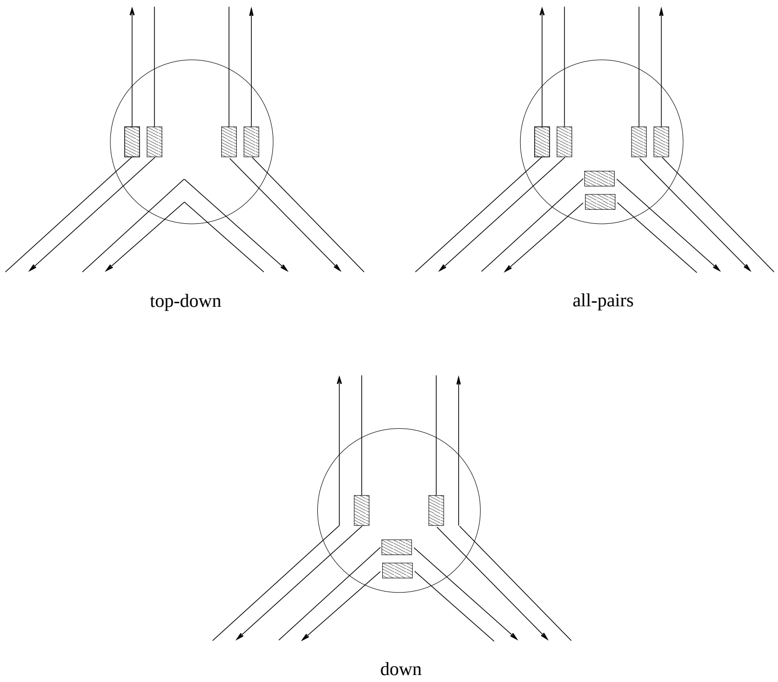

Figure 6: The position of wavelength converters at a node of a limited conversion binary tree network.

Clearly, a wavelength routing algorithm for down and top–down limited conversion networks also work for all–pairs limited conversion network. However, the converse does not necessarily hold.

5.2 Optimal Bandwidth Utilization in Sparse Conversion Trees

In this section, we study the problem of placing wavelength converters with full conversion capabilities at the nodes of a sparse conversion tree in order to guarantee complete utilization of available bandwidth. In other words, given a directed tree network T, which supports L wavelengths, we want to place wavelength converters at some of the nodes of T in order to route any set of communication requests of load L using exactly L wavelengths. We consider the problem of deciding how many full converters are necessary and sufficient to guarantee optimal bandwidth utilization.

Let u be a node of the tree T. If we locate at u a full converter for each pair of incoming and outgoing directed links, then we can change the wavelengths assigned to all the requests going through u. This is equivalent to splitting each request going through u into two requests (the first ending at u and the second starting from u) and color each one independently. In the sequel we say that u is a *full conversion point* if it hosts a full wavelength converter for each pair of incoming and outgoing directed links. If u is a full conversion point, then we can consider the forest obtained from T by removing u and color requests on each tree of the forest, independently.

We proceed by considering the application of the greedy algorithm to *spiders*; a spider is a tree having at most one node of degree greater than 2. It is known [20, 46] that spiders guarantee optimal bandwidth utilization. In [3], it is demonstrated how this can be achieved by a greedy algorithm.

Notice that the tree depicted in Figure 1 is the smallest tree that is not a spider; it is easy to see that each tree that is not a spider contains a subtree that is homomorphic to this one. By Theorem 3, we obtain that there exists a gap for the number of colors necessary to color a pattern of requests on a tree: either the tree is a spider and, thus, any pattern of requests of load L can be colored with exactly L wavelengths, or the tree is not a spider and thus there exists a pattern of requests of load L that requires at least $5L/4$ wavelengths. In other words, if we seek complete bandwidth utilization, then we need to locate full conversion points at a set S of nodes of T in such a way that the forest obtained by splitting T at the vertices of S consists of spiders.

The following two results are proved in [5] (see also [3]) and give tight bounds on the number of full conversion points.

Theorem 12 (Auletta *et al.* **[5], see also [3])** *For each integer $n \geq 6$, there exists a tree T with n nodes such that for each set X of nodes of T, with $|X| < \left\lfloor \frac{1}{2}\left(\frac{n}{2}-1\right)\right\rfloor$, the forest which results by splitting T at the nodes of X contains a tree that is not a spider.*

Theorem 13 (Auletta *et al.* **[5], see also [3])** *For each tree T with n nodes, there exists a set X of nodes of T, with $|X| \leq \left\lfloor \frac{1}{2}\left(\frac{n}{2}-1\right)\right\rfloor$, such that all trees of the forest which results by splitting T at the nodes of X are spiders.*

5.3 Limited Conversion Top–Down Binary Trees

In this section, we present a greedy algorithm for limited conversion top-down binary trees. We first give a sufficient condition so that the algorithm can route any pattern of communication requests of load L with W wavelengths ($W \geq L$). Then, we describe how to construct converters of small degree and size that allow for optimal ($W = L$) or nearly-optimal bandwidth utilization using the algorithm.

We concentrate on the converters located at a node u. We denote by p the parent of u, by v the left child of u and by w the right child of u. Also, we denote by $P(u)$ the set of requests touching u. We partition $P(u)$ into six subsets:

- $P_1(u)$ consisting of the requests going from p to v;
- $P_2(u)$ consisting of the requests going from v to p;
- $P_3(u)$ consisting of the requests going from p to w;
- $P_4(u)$ consisting of the requests going from w to p;
- $P_5(u)$ consisting of the requests going from v to w;
- $P_6(u)$ consisting of the requests going from w to v.

We denote by $C_1(u)$, $C_2(u)$, $C_3(u)$ and $C_4(u)$ the four converters placed at u. Converter $C_i(u)$ is devoted to the requests of $P_i(u)$ (see Figure 7).

We now present the wavelength routing algorithm. The algorithm visits the nodes of the tree in BFS order, starting from the root. The algorithm proceeds in phases, one for each node of the tree. The phase corresponding to node u assumes that all requests touching nodes previously visited have already been colored. In particular, it assumes that all requests traversing the links between u and

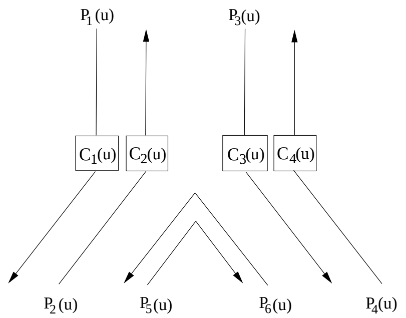

Figure 7: Converters located at node u of a top–down limited conversion binary tree.

p (in any direction) have already been assigned a color on the segment between p and u. Then, the algorithm extends this partial coloring by assigning a color to the segments of all requests touching u that consist of links between u and its children.

More formally, let A_1 be the set of colors assigned to requests in $P_1(u)$, A_2 the set of colors assigned to requests in $P_2(u)$, A_3 the set of colors assigned to requests in $P_3(u)$, and A_4 the set of colors assigned to requests in $P_4(u)$. The algorithm performs two independent steps:

STEP 1: Converters $C_1(u)$ and $C_4(u)$ are set in such way that:

- for each $r \in P_1(u)$, the algorithm assigns a color c_r to the segment (u, v) of r such that the color of r on (p, u) can be converted to c_r by converter $C_1(u)$.
- for each $r \in P_4(u)$, the algorithm assigns a color c_r to the segment (w, u) of r such that c_r can be converted into the color of r on (u, p) by converter $C_4(u)$.

Let A'_1 and A'_4 the set of colors assigned to the segments (u, v) and (w, u) of the requests in $P_1(u)$ and $P_4(u)$, respectively. Requests in $P_6(u)$ are assigned colors not in $A'_1 \cup A'_4$.

STEP 2: This step is symmetric to step 1. Converter $C_3(u)$ converts colors assigned to requests of $P_3(u)$ on segment (p, u) to colors assigned on segment (u, w); the algorithm assigns colors to requests of $P_2(u)$ for the segment (v, u) that can be changed by $C_3(u)$ into the colors assigned to the same requests on segment (u, p).

Let A'_2 and A'_3 the set of colors assigned to the segments (v, u) and (u, w) of the requests in $P_2(u)$ and $P_3(u)$, respectively. Requests in $P_5(u)$ are assigned colors not in $A'_2 \cup A'_3$.

The following lemma gives sufficient conditions for the correctness of the algorithm; that is, conditions so that the algorithm can route any communication pattern of load L using W wavelengths ($W \geq L$).

Lemma 14 *The algorithm is correct if for all sets $P(u)$ of requests touching a vertex u, for all sets of colors A_i used to color the segments of the requests in $P_i(u)$ between u and p, there exist sets of colors A'_i such that:*

1. *For $i = 1, 2, 3, 4$, colors of A_i can be converted by $C_i(u)$ into colors of A'_i, and*
2. *$|A'_1 \cup A'_4| \leq W - |P_5|$ and $|A'_2 \cup A'_3| \leq W - |P_6|$.*

Especially in the case where we demand optimal bandwidth utilization using the algorithm described above, Auletta *et al.* [5] (see also [3]) prove that, in order to prove correctness of the algorithm, it is sufficient to prove that by putting two converters back-to-back we obtain a depth–two *L–superconcentrator.*

Definition 15 *An L–superconcentrator is a directed graph with L distinguished vertices called inputs, and L other distinguished vertices called outputs, such that for any $1 \leq k \leq L$, any set X of k inputs and any set Y of k outputs, there exist k vertex–disjoint paths from X to Y.*

The converter which is proposed in [3, 5] for optimal bandwidth utilization using the algorithm above supports L wavelengths and has degree $2\sqrt{L}-1$. Such a converter for $L = 9$ is depicted in Figure 8. The reader may examine Figure 9 where two such converters have been put back-to-back and see that the resulting graph is indeed depth–two superconcentrator. For a formal proof, see [3, 5].

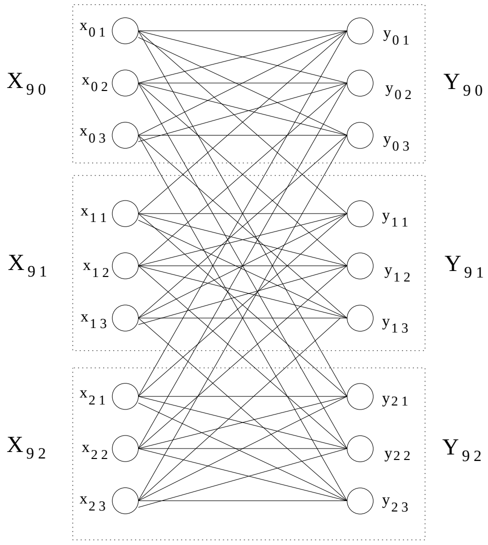

Figure 8: A wavelength converter with 9 wavelengths which allows for optimal bandwidth utilization using the greedy algorithm presented in Section 5.3.

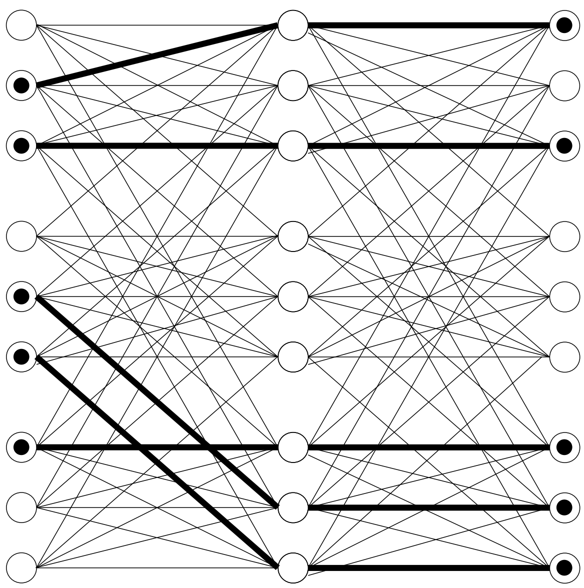

Figure 9: The graph obtained by concatenating two copies of the converter of Figure 8.

Thus, we obtain the following result.

Theorem 16 (Auletta *et al.* **[5], see also [3])** *There exist converters of degree* $2\sqrt{L}-1$ *that allow greedy wavelength routing of any communication pattern of load* L *with* L *wavelengths.*

Although the depth–two superconcentrator produced by putting back–to–back two copies of the converter depicted in Figure 8 has asymptotically optimal degree, the total number of possible conversions is too large.

Auletta *et al.* [7] (see also [3]), extending techniques presented in [36], construct wavelength converters of small size that allow for optimal and nearly–optimal conversion on binary trees. The construction is based on properties of dispersers [42] and Ramanujan graphs, which have been explicitly constructed in [30, 31, 45].

Theorem 17 (Auletta *et al.* **[7], see also [3])** *There exist converters of size* $O\left(L\frac{\log^2 L}{\log\log L}\right)$ *that allow greedy wavelength routing of any communication pattern of load* L *with* L *wavelengths.*

Theorem 18 (Auletta *et al.* **[7], see also [3])** *Let* $f(L) = o(L)$ *be an increasing function. There exist converters of size* $O\left(L\frac{\log^2 f(L)}{\log\log f(L)}\right)$ *that allow greedy wavelength routing of any communication pattern of load* L *with* $L + \frac{L}{f(L)}$ *wavelengths.*

The techniques used for proving Theorems 17 and 18 are similar in spirit with those used for proving Theorem 16.

5.4 All–Pairs and Down Conversion Trees

In this section we present results about optimal and nearly–optimal bandwidth utilization in all–pairs and down conversion trees. All results make use of the expansion properties of Ramanujan graphs. The interested reader may see [3, 6, 19] for a complete discussion and proofs.

Gargano [19] and independently Auletta *et al.* [6] (see also [3]) show that using Ramanujan graphs of constant degree, we can achieve optimal bandwidth utilization in binary trees with all–pairs conversion.

Theorem 19 (Auletta *et al.* **[6], Gargano [19], see also [3])** *Let* T *be an all–pairs limited conversion binary tree. If all the converters located at the nodes of* T *are 15–regular Ramanujan graphs, then there exists a greedy algorithm that colors any communication pattern of load* L *on* T *using* L *wavelengths.*

For down conversion arbitrary trees, Gargano in [19] (see also [3]) shows how to obtain near–optimal utilization of bandwidth using converters of constant degree. In particular, the following result is proved.

Theorem 20 (Gargano [19], see also [3]) *Let* T *be a down limited conversion (arbitrary) tree with* W *wavelengths per link. For any* $\epsilon > 0$*, there exists an integer* $k_\epsilon = O(1/\epsilon)$ *such that, if all converters located at nodes of* T *are* k_ϵ*–regular Ramanujan graphs, then it is possible to greedily route any communication pattern of load at most* $(1-\epsilon)W$*.*

6 Randomized Algorithms

In an attempt to beat the 5/3 lower bound for deterministic greedy algorithms, Auletta *et al.* [4] define the class of randomized greedy wavelength routing algorithms. Randomized greedy algorithms have the same structure as deterministic ones; that is, starting from a node, they consider the nodes of the tree in a BFS manner. Their main difference is that a randomized greedy algorithm $\mathcal{A}$ uses a palette of colors and at each phase associated with a node, $\mathcal{A}$ picks a random proper coloring of the uncolored requests using colors of the palette according to some probability distribution.

The results presented in the following were originally obtained in [4]. The interested reader may see [4] for further details.

6.1 Lower Bounds

In this section, we present two lower bounds on the number of wavelengths used by randomized greedy algorithms to route patterns of requests of load L. We first present a lower bound for large trees (i.e., trees with height $\Omega(L)$) in Theorem 21; then, in Theorem 22, we present a lower bound for trees of constant height.

The first lower bound states that no randomized greedy algorithm can achieve a performance ratio better than $3/2$ if the depth of the tree is large.

Theorem 21 (Auletta *et al.* **[4])** *Let $\mathcal{A}$ be a (possibly randomized) greedy wavelength routing 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 pattern of communication requests P of load L on T, such that the probability that $\mathcal{A}$ colors P with at least $3L/2$ colors is at least $1 - \exp(-L^{\epsilon})$.*

The proof of the theorem can be modified to prove that if the depth of the tree is $\Omega(L^{0.639})$, then any greedy algorithm requires at least $7L/5$ colors with high probability.

The following lower bound holds even for trees of constant depth.

Theorem 22 (Auletta *et al.* **[4])** *Let $\mathcal{A}$ be a (possibly randomized) greedy wavelength routing 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 pattern of communication requests P with load L on T, such that the probability that $\mathcal{A}$ colors P with at least $(1.293 - \delta - o(1))L$ colors is at least $1 - O(L^{-2})$.*

For the proofs, constructions based on the lower bound technique presented in Section 3.2 are used.

Note that the adversary assumed in Theorems 21 and 22 has no knowledge of the probability distribution according to which the randomized greedy algorithm makes its random choices. Possibly, better lower bounds may be achieved by considering more powerful adversaries.

6.2 Upper Bounds

In this section, we give the main ideas of a randomized wavelength routing algorithm presented in [4]. The algorithm has a greedy structure but allows for limited recoloring at the phases associated with each node.

At each phase, the wavelength routing algorithm maintains the following two invariants:

I. The total number of colors is no greater than $7L/5$.

II. The number of colors seen by two opposite directed links is exactly $6L/5$.

At a phase associated with a node u, a *coloring procedure* is executed which extends the coloring of requests that touch u and its parent node to the requests that touch u and are still uncolored. The coloring procedure is randomized (selects the coloring of requests being uncolored according to a specific probability distribution). In this way, the algorithm can complete the coloring at the phase associated with node u using at most $7L/5$ colors in total, keeping the number of colors seen by the opposite directed links between u and its children to $6L/5$.

At each phase associated with a node u, the algorithm is enhanced by a *recoloring procedure* which recolors a small subset of requests in order to maintain some specific properties on the (probability distribution of the) coloring of requests touching u and its parent. This procedure is randomized as well.

The recoloring procedure at each phase of the algorithm works will very high probability. The coloring procedure at each phase always works correctly maintaining the two invariants. As a result, if the depth of the tree is not very large (no more than $O(L^{1/3})$), the algorithm executes the phases associated with all nodes, with high probability.

After the execution of all phases, the set of requests being recolored by the executions of the recoloring procedure are colored using the simple deterministic greedy algorithm with at most $o(L)$ extra colors due to the fact that as far as the depth of the tree is not very large (no more than $O(L^{1/3})$), the load of the set of requests being recolored is at most $o(L)$, with high probability.

In this way, the following result is proved. The interested reader may look at [4] for a detailed description of the algorithm and formal proofs.

Theorem 23 (Auletta *et al.* **[4])** *Let* $0 < \delta < 1/3$ *be a constant. There exists a randomized wavelength routing algorithm that routes any pattern of communication requests of 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* $1-\exp\left(-\Omega(L^\delta)\right)$.

Furthermore, 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, Auletta *et al.* [4] obtain the following existential upper bound on the number of wavelengths sufficient for routing any pattern of requests of load L. Note that Theorem 24 holds for any binary tree (of any depth).

Theorem 24 (Auletta *et al.* **[4])** *Any pattern of communication requests of load* L *on a binary tree can be routed with at most* $\frac{7}{5}\left(5\left\lceil\frac{\sqrt{L}}{5}\right\rceil+10\right)^2$ *wavelengths.*

Especially for binary trees of depth $O(L^{1/3-\epsilon})$, Theorem 24 improves the large hidden constants implicit in the $o(L)$ term of the constructive upper bound (Theorem 23). In larger binary trees, Theorem 24 significantly improves the 5/3 constructive upper bound for all sets of requests of load greater than $26,000$.

7 Open Problems

Recent work on wavelength routing in trees has revealed many open problems; some of them are listed below.

- The main open problem is to close the gap between $5L/4$ and $5L/3$ for the number of wavelengths sufficient for routing communication patterns of load L on arbitrary trees (see Theorems 3 and 6). Closing the gap between $5L/4$ and $7L/5 + o(L)$ for binary trees also deserves some attention.

- Furthermore, although for deterministic greedy algorithms we know tight bounds on the number of wavelengths, this is not true for randomized greedy algorithms. Exploring the power of randomized greedy algorithms in more depth is interesting as well.

- Notice that the algorithms proposed are too far from optimality. We are not aware of any inapproximability results for the problem. This indicates another direction for future research.

- Many improvements can be made in the work on wavelength conversion. The main open problem here is whether we can achieve optimal bandwidth utilization in arbitrary trees using converters of constant degree (or even linear size). We conjecture that this may be feasible in all–pairs conversion trees with deterministic greedy algorithms.

References

[1] A. Aggarwal, A. Bar-Noy, D. Coppersmith, R. Ramaswami, B. Schieber, and M. Sudan. Efficient Routing and Scheduling Algorithms for Optical Networks. *Journal of the ACM*, Vol. 43, No. 6, 1996, pp. 973–1001.

[2] N. Alon. Eigenvalues and Expanders. *Combinatorica*, 6, 1986, pp. 83–96.

[3] V. Auletta, I. Caragiannis, L. Gargano, C. Kaklamanis, and P. Persiano. Sparse and Limited Wavelength Conversion in All–Optical Tree Networks. *Theoretical Computer Science*, to appear.

[4] V. Auletta, I. Caragiannis, C. Kaklamanis, and P. Persiano. Randomized Path Coloring on Binary Trees. In *Proc. of the 3rd International Workshop on Approximation Algorithms for Combinatorial Optimization (APPROX 2000)*, LNCS 1913, Springer, pp. 60–71, September 2000.

[5] V. Auletta, I. Caragiannis, C. Kaklamanis, and P. Persiano. Bandwidth Allocation Algorithms on Tree–Shaped All-Optical Networks with Wavelength Converters. In *Proc. of the 4th International Colloquium on Structural Information and Communication Complexity (SIROCCO '97)*, July 1997, pp. 13–27.

[6] V. Auletta, I. Caragiannis, C. Kaklamanis, and P. Persiano. Efficient Wavelength Routing in Trees with Low–Degree Converters. *Multichannel Optical Networks: Theory and Practice*, DIMACS Series on Discrete Mathematics and Theoretical Computer Science, Vol. 46, AMS, pp. 1–13, 1998.

[7] V. Auletta, I. Caragiannis, C. Kaklamanis, and P. Persiano. On the Complexity of Wavelength Converters. In *Proc. of the 23rd International Symposium on Mathematical Foundations of Computer Science (MFCS '98)*, LNCS 1450, Springer, August 1998, pp. 771–779.

[8] Y. Aumann and Y. Rabani. Improved Bounds for All Optical Routing. In *Proc. of the 6th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '95)*, January 1995, pp. 567–576.

[9] R. A. Barry and P. A. Humblet. Models of Blocking Probability in All–Optical Networks with and without Wavelength Changers. IEEE Journal on Selected Areas of Communication, Vol. 14, June 1996, pp. 858–867.

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

[11] C. Berge. Graphs and Hypergraphs. *North–Holland*, 1973.

[12] J.–C. Bermond, L. Gargano, S. Perennes, A.A. Rescigno, and U. Vaccaro. Efficient Collective Communication in Optical Networks. *Theoretical Compute Science*, 233 (1-2), 2000, pp. 165-189.

[13] I. Caragiannis, C. Kaklamanis, and P. Persiano. Wavelength Routing of Symmetric Communication Requests in Directed Fiber Trees. In *Proc. of the 5th International Colloquium on Structural Information and Communication Complexity (SIROCCO 98)*, pp. 10–19, June 1998. Also: Symmetric Communication in All–Optical Tree Networks. *Parallel Processing Letters*, to appear.

[14] I. Caragiannis, C. Kaklamanis, and P. Persiano. Edge Coloring of Bipartite Graphs with Constraints. In *Proc. of the 23rd International Symposium on Mathematical Foundations of Computer Science (MFCS 99)*, LNCS 1450, Springer, pp. 771–779, August 1999. Also, *Theoretical Computer Science*, to appear.

[15] I. Caragiannis, C. Kaklamanis, and P. Persiano. Bounds on Optical Bandwidth Allocation in Directed Fiber Tree Topologies. *2nd Workshop on Optics and Computer Science*, April 1997.

[16] T. Erlebach and K. Jansen. Scheduling of Virtual Connections in Fast Networks. In *Proc. of 4th Workshop on Parallel Systems and Algorithms (PASA '96)*, World Scientific, 1997, pp. 13–32.

[17] T. Erlebach, K. Jansen. Call Scheduling in Trees, Rings and Meshes. In *Proc. of the 30th Hawaii International Conference on System Sciences*, pp. 221–222, 1997.

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

[19] L. Gargano. Limited Wavelength Conversion in All–Optical Tree Networks. In *Proc. of the 25th International Colloquium on Automata, Languages, and Programming (ICALP '98)*, LNCS 1443, Springer, pp. 544–555, July 1998.

[20] L. Gargano, P. Hell, and 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, July 1997.

[21] P. E. Green. Fiber-Optic Communication Networks. *Prentice–Hall*, 1992.

[22] K. Jansen. Approximation Results for Wavelength Routing in Directed Binary Trees. *2nd Workshop on Optics and Computer Science*, April 1997.

[23] C. Kaklamanis and P. Persiano. Efficient Wavelength Rouiting in Directed Fiber Trees. In *Proc. of the 4th European Symposium on Algorithms (ESA '96)*, LNCS 1136, Springer, pp. 460–470, September 1996.

[24] C. Kaklamanis, P. Persiano, T. Erlebach, and 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, pp. 493–504, July 1997.

[25] R. Klasing. Methods and Problems of Wavelength–Routing in All–Optical Networks. In *Proc. of MFCS 98 Workshop on Communication*, pp. 1–9, August 1998.

[26] J. Kleinberg and A. Kumar. Wavelength Conversion in Optical Networks. In *Proc. of the 10th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '99)*, pp. 566–575, January 1999.

[27] M. Kovacevic and S. Acampora. Benefits of Wavelength Translation in All–Optical Clear–Channel Networks. IEEE Journal on Selected Areas of Communication, Vol. 14, June 1996, pp. 868–880.

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

[29] K. -C. Lee and V.O.K. Li. Routing and Switching in a Wavelength Convertible Lightwave Network. In *Proc. of IEEE INFOCOM 1993*, April 1993, pp. 578–585.

[30] A. Lubotzky, R. Phillips, and P. Sarnak. Ramanujan Graphs. *Combinatorica*, Vol. 8, 1988, pp. 261–278.

[31] G.A. Margulis. Explicit Group–Theoretic Constructions of Combinatorial Schemes and their Applications for the Construction of Expanders and Concentrators. *Problems of Information Transmission*, Vol. 24(1), pp. 39–46, July 1988.

[32] M. Mihail, K. Kaklamanis, and S. Rao. Efficient Access to Optical Bandwidth. In *Proc. of 36th Annual IEEE Symposium on Foundations of Computer Science (FOCS '95)*, October 1995, pp. 548–557.

[33] D. Minoli. Telecommunications Technology Handbook, *Artech House*, 1991.

[34] R. Pankaj. Architectures for Linear Lightwave Networks. PhD. Thesis, Department of Electrical Engineering and Computer Science, MIT, Cambridge MA, 1992.

[35] Y. Rabani. Path Coloring on the Mesh. In *Proc. of the 37th IEEE Symposium on Foundations of Computer Science (FOCS 96)*, pp. 400–409, October 1996.

[36] J. Radhakrishnan and A. Ta-Shma. Tight Bounds for Depth-Two Superconcentrators. In *Proc. of the 38th Annual IEEE Symposium on Foundations of Computer Science*, October 1997, pp. 585–594.

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

[38] B. Ramamurthy and B. Mukherjee. Wavelength Conversion in WDM Networking. IEEE J. on Selected Areas of Communication, Vol. 16, September 1998, pp. 1061–1073.

[39] R. Ramaswami and G.H. Sasaki. Multiwavelength Optical Networks with Limited Wavelength Conversion. In *ACM/IEEE Trans. Networking*, Vol. 6, n. 6, December 1998, pp. 744–754.

[40] R. Ramaswami and K. Sivarajan. Routing and Wavelength Assignment in All–Optical Networks. In *ACM/IEEE Trans. Networking*, Vol. 3, n. 5, October 1995, pp. 489–500.

[41] R. Ramaswami and K. Sivarajan. Optical Networks. *Morgan Kauffman Publishers*, 1998.

[42] M. Sipser. Expanders, Randomness, or Time versus Space. *Journal of Computer and System Sciences*, Vol. 36, n. 3, June 1988, pp. 379–383.

[43] S. Subramaniam, M. Azizoglu, and A.K. Somani. All–Optical Networks with Sparse Wavelength Conversion. *ACM/IEEE Trans. on Networking*, Vol. 4, n. 4, August 1996, pp. 544–557.

[44] R.M. Tanner. Explicit Construction of Concentrators from Generalized n-gons. *SIAM J. Alg. Discr. Math.*, 5, 1984, pp. 287–293.

[45] A. Ta–Shma. Almost Optimal Dispersers. In *Proc. of the 30th Annual ACM Symposium on the Theory of Computing (STOC '98)*, May 1998, pp. 196–202.

[46] G. Wilfong and P. Winkler. Ring Routing and Wavelength Translation. In *Proc. of the 9th Annual ACM–SIAM Symposium on Discrete Algorithms (SODA '98)*, January 1998, pp. 333–341.

[47] J. Yates, J. Lacey, D. Everitt, and M. Summerfield. Limited Range Wavelength Translation in All–Optical Networks. In *Proc. of IEEE INFOCOM '96*, April 1996, pp. 954-961.

[48] J. Zhou, N. Park, K. Vahala, M. Newkirk, and B. Miller. Four–Wave Mixing Wavelength Conversion Efficiency in Semicinductor Travelling–Wave Amplifiers Measured to 65 nm of Wavelength Shift. *IEEE Phot. Tech. Letters*, Vol. 6, n. 8, August 1994, pp. 984–987.