WAVELENGTH ASSIGNMENT ALGORITHMS FOR WDM OPTICAL NETWORKS

by

Zhengbing Bian
B.Eng., Huazhong University of Science & Technology, 1996
M.A.Sc., The University of British Columbia, 2001

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
in the School
of
Computing Science

SIMON FRASER UNIVERSITY
August 2003

APPROVAL

Name:	Zhengbing Bian
Degree:	Master of Science
Title of thesis:	Wavelength Assignment Algorithms for WDM Optical Networks
Examining Committee:	Dr. Wo-Shun Luk Chair
	Dr. Qian-ping Gu, Senior Supervisor
	Dr. Joseph G. Peters, Supervisor
	Dr. Tiko Kameda, SFU Examiner
Date Approved:	Aug. 7, 2003

SIMON FRASER UNIVERSITY

PARTIAL COPYRIGHT LICENSE

Title of Thesis/Project/Extended Essay:

Wavelength Assignment Algorithms for WDM Optical Networks

Author: ______________________
(signature)

Zhengbing Bian
____
(name)

August 8, 2003
____
(date)

:vg

Abstract

The explosive growth of the Internet and bandwidth-intensive applications such as video-on-demand and multimedia conferences require high-bandwidth networks. The current high-speed electronic networks cannot provide such capacity. Optical networks offer much higher bandwidth than traditional networks. When employed with the *wavelength division multiplexing* (WDM) technology, they can provide the huge bandwidth needed. The *tree of rings* is a popular topology which can often be found in WDM networks. In this thesis, we first study *wavelength assignment* (WA) algorithms for trees of rings.

A tree of rings is a graph obtained by interconnecting rings in a tree structure such that any two rings intersect in at most one node and any two nodes are connected by exactly two edge-disjoint paths. The WA problem is that given a set of paths on a graph, assign wavelengths to the paths such that any two paths sharing a common edge are assigned different wavelengths and the number of wavelengths is minimized. The WA problem on trees of rings is known to be NP-hard. A trivial lower bound on the number of wavelengths is the maximum number L of paths on any link. In this thesis, we propose a greedy approximation algorithm which uses at most $3L$ wavelengths on a tree of rings with node degree at most 8. This improves the previous $4L$ upper bound. Our algorithm uses at most $4L$ wavelengths for a tree of rings with node degree greater than 8. We also show that $3L$ is the lower bound for some instances of the WA problem on trees of rings. In addition, we show that our algorithm achieves approximation ratios of $2\frac{1}{16}$ and $2\frac{3}{13}$ for trees of rings with node degrees at most 4 and 6, respectively.

Optical switches, which keep the data stream transmitted in optical form from source to destination to eliminate the electro-optic conversion bottleneck at intermediate nodes, are key devices in realizing the huge bandwidth of optical networks. One of the common ways to build large optical switches is to use *directional couplers* (DCs). However, DCs suffer from an intrinsic crosstalk problem. In this thesis, we study the nonblocking properties of *Benes networks* and *Banyan-type networks* with extra stages under crosstalk constraints.

To my parents, my sisters and my wife

Acknowledgments

I would like to express my gratitude to my academic advisor Dr. Qian-ping Gu. Without his support and guidance, this thesis would not have been possible. I would like to thank Dr. Joseph G. Peters and Dr. Tiko Kameda for being my thesis committee members. I would also like to thank my colleagues for many discussions on network topologies.

I am especially grateful to my parents and sisters for their love, support and encouragement over the years. I am equally grateful to my wife Zheqiong Wang for her understanding and support during my study.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Internet traffic has increased dramatically during the past decade. Meanwhile we are seeing the continuously rising demand for network bandwidth. The major cause of this increasing demand is the tremendous growth of the Internet and World Wide Web, both in terms of number of users and the amount of bandwidth taken by each user [35]. The emerging multimedia applications such as video-on-demand require high bandwidth. Businesses today rely on high-speed networks. It is expected that the current electronic network infrastructure will be unable to meet this ever-increasing demand in the near future. A new technology is needed to support the huge bandwidth needs.

Optical fiber offers much higher bandwidth than conventional copper cables. A single fiber has a potential bandwidth on the order of 50THz [30]. Meanwhile, it has low cost, extremely low bit error rate (typically 10^{-12}, compared to 10^{-6} in copper cables), low signal attenuation and low signal distortion. In addition, optical fibers are more secure from tapping, since light does not radiate from the fiber and it is nearly impossible to tap into it secretly without being detected. As a result, it is the preferred medium for data transmission with bit rate more than a few tens of megabits per second over any distance more than one kilometer. It is also the preferred means of realizing short distance (a few meters to hundreds of meters), high-speed (gigabits per second and above) interconnection inside large systems [35]. In the past few decades, optical fibers have been widely deployed in all kinds of telecommunications networks.

Optical fiber has been used in two generations of optical networks [35]. In the first generation, it was essentially used for transmission and simply to provide capacity, since it provides lower bit error rates and higher capacities than copper cables. All the switching and other intelligent network functions were handled by electronics. Thus, the bandwidth is limited by the electronics at the fiber endpoints. Currently, transmission rates are restricted to 10Gb/s (OC-192) in commercially available systems. Examples of the first generation optical networks are SONET and SDH networks.

In the second generation optical networks, some of the routing, switching and intelligence is handled by optical layer. The fiber bandwidth is further exploited by a technique called *wavelength division multiplexing* (WDM), where the optical bandwidth is partitioned into a large number of channels on different wavelengths (or, equivalently, colors), and each channel works at peak electronic rate. These wavelengths do not interfere with each other as long as the channel space is large enough. Other than providing a huge bandwidth, WDM networks can also provide data transparency in which the network may accept data at any bit rate and any protocol format within the limits. Data transparency may be realized through *all-optical* (or single-hop) transmission and switching of signals. In an all-optical network, data is transfered from source to destination in optical form, without undergoing any optical-to-electrical conversion. Keeping the signal in optical form eliminates the "electronic bottleneck" of communications networks with electronic switching.

1.1 WDM network model

A WDM optical network consists of routing nodes interconnected by point-to-point fiber links. WDM networks provide *lightpaths* to its users, such as SONET terminals or IP routers. Lightpaths are optical connections carried from source node to destination node over a wavelength on each intermediate link. At intermediate nodes in the network, the lightpaths are routed and switched from one link to another. Different lightpaths can use the same wavelength as long as they do not have a common link. In this thesis, we consider switched optical networks with generalized switches based on accousto-optic-filters, as is done in [3, 33]. In this kind of network, the switch can differentiate between several wavelengths coming in along a fiber and direct each of them to a different output of the switch. The only constraint is that no two lightpaths sharing any link have the same wavelength. Such networks are often called *wavelength routed networks.*

A WDM optical network can be modelled in both undirected and directed models. In the undirected model, the optical network is modelled as an undirected graph $G = (V(G), E(G))$, where each undirected edge represents a point-to-point undirected fiber-optic link. A request consists of an unordered pair of nodes. In the directed model, the optical network is modelled as a directed graph $G = (V(G), A(G))$, where each arc represents a point-to-point unidirectional fiber-optic link. A request consists of an ordered pair of nodes, source and destination. An instance consists of a set of requests. A solution for an instance consists of settings for the switches in the network, and an assignment of wavelengths to the requests, such that a path (directed or undirected) is set up between the nodes of each request, and no two paths are assigned the same wavelength if they share a link in the undirected model (or share an arc in the directed model).

In this thesis, we use both undirected and directed models. In particular, both models are used in the literature review (Chapter 3). For simplicity, our algorithm for trees of rings (Chapter 4) only considers the undirected model. We use the directed model for the crosstalk reduction problem at optical switches (Chapter 5).

1.2 Routing and wavelength assignment

The cost of a WDM network is related to the number of wavelengths used: the cost of optical switches and other devices depends on the number of wavelengths they handle. There is also a limit on the number of available wavelengths on a single fiber. Although the number of wavelengths per fiber could be as large as 160 in the laboratory [35], commercially available systems have the limitation of several tens of wavelengths per fiber. A constant goal in WDM optical networks is to minimize the number of wavelengths used. One of the most important problems is the *routing and wavelength assignment* (RWA) problem.

The RWA problem is defined as follows. Given a network topology and a set of routing requests, determine a path and wavelength for each routing request, such that the number of wavelengths used is minimized. A special case of the RWA problem is that the paths are already given and we are asked to find a wavelength assignment with the minimum number of wavelengths. This problem is called the wavelength assignment (WA) problem. To solve the WA problem, the following two constraints apply:

1. *Distinct Channel Assignment (DCA):* Two paths must be assigned different wavelengths on any common link.

2. *Wavelength Continuity:* If no wavelength conversion is available, then a path must be assigned the same wavelength on all the links in it.

Throughout the thesis, we assume the above two constraints hold, unless otherwise specified.

The RWA and WA problems can be studied for both offline (or static) and online (or dynamic) cases. In the offline case, all the routing requests are given at one time. For the online case, the routing request comes in one by one, and there is no knowledge about future requests. In this thesis, we consider the offline WA problem on trees of rings (Chapter 4). The crosstalk reduction problem (Chapter 5) is studied under the online traffic.

1.3 Contributions

In this thesis, we first study the WA problem on a tree of rings. This problem is NP-hard. A trivial lower bound is the maximum number L of paths on any link (called the maximum edge load). We propose a greedy algorithm which uses at most $3L$ wavelengths for trees of rings with maximum node degree 8, and at most $4L$ for trees of rings with node degree greater than 8. This implies that the algorithm achieves an approximation ratio of 3 for node degree at most 8. We also show that $3L$ is the lower bound on the number of wavelengths for some instances of the WA problem on trees of rings. We further prove that our algorithm achieves approximation ratios of $2\frac{1}{16}$ and $2\frac{3}{13}$ on trees of rings with node degrees at most 4 and 6, respectively. This improves over the previous $4L$ result, and the previous 2-approximation algorithm which only works for trees of rings with node degree at most 4.

As introduced in Section 1.1, optical switches are widely used in WDM optical networks. One of the most common ways to construct large optical switches is to use *directional couplers* (DCs). Crosstalk can be a severe problem in DC-based optical switches. Previous results on crosstalk reduction are only available for the special cases where either no crosstalk constraint is enforced or no crosstalk is allowed. In this thesis, we study the general crosstalk reduction problem and show several lower bounds for Benes networks and Banyan-type networks with extra stages under arbitrary crosstalk constraints.

1.4 Organization of the thesis

The thesis will focus on the WA problem on trees of rings, and nonblocking properties of Banyan-type networks under crosstalk constraints. In Chapter 2, we introduce the basic concepts and notation. In Chapter 3, we survey the major results of the RWA and WA problems on trees, rings and trees of rings. In Chapter 4, we propose a greedy algorithm for the WA problem on trees of rings and show that it improves over the previous work. Chapter 5 will be devoted to the analysis of crosstalk reduction in Banyan-type networks. In Chapter 6, we conclude the thesis and discuss the future work.

Chapter 2

Problem Definitions

In this chapter, we will introduce the basic definitions for bipartite graph, matching and graph coloring. These concepts can be found in most graph theory books, such as [7] and [43]. We will give the formal definition of the routing and wavelength assignment problem. We will also introduce several popular network topologies in WDM optical networks.

2.1 Bipartite graph, matching and graph coloring

An independent set in a graph G is a set of pairwise nonadjacent vertices. A graph G is *bipartite* if the node set $V(G)$ is the union of two disjoint independent sets called *partite* sets of G.

A *matching* in a graph G is a set of edges with no shared endpoints. A *maximal matching* in a graph is a matching that cannot be enlarged by adding an edge. A *maximum matching* is a matching of maximum size among all matchings in the graph.

The *clique number* of a graph G is the maximum size of a set of pairwise adjacent vertices (clique) in G.

Definition *(West [43])* A ***vertex k-coloring*** of a graph G is a labeling $f : V(G) \to S$, where $|S| = k$. The labels are **colors**. A graph is **k-colorable** if it has a k-coloring such that adjacent vertices have different labels. The **chromatic number** is the least k such that G is k-colorable.

It is easy to see that the following theorem holds:

Theorem 2.1.1 (Berge [7]) *For every graph G, the clique number is a lower bound on the chromatic number.*

Definition Given a set P of paths in a graph G, the load $L(e, P)$ of an (undirected or directed) edge e is the number of paths in P containing that edge. The maximum load $L(P)$ is the maximum of the values $L(e, P)$, taken over all edges e of the graph.

In this thesis, the maximum load is denoted by L. The set of paths P that creates this maximum load will always be clear from the context.

Definition Given a set P of paths in a graph G, the *conflict graph* associated with P is the undirected graph $G_c(P, E)$ with the node set P such that each node of G_c corresponds to a path in P and two nodes of G_c are adjacent if and only if the corresponding paths in P share an edge of G.

Definition Given a graph G and a set P of paths, a valid coloring for P is that any two paths sharing a common edge are assigned different colors. Let $\omega(P)$ represent the maximum number of pairwise intersecting paths, and let $\chi(P)$ represent the minimum possible number of colors required in a valid coloring for P.

It is easy to see that $\omega(P)$ and $\chi(P)$ are equal to the clique number and the chromatic number of the conflict graph $G_c(P, E)$, respectively.

Since all paths that cross a given edge are pairwise intersecting and thus must be assigned different colors, for any set P, the following lemma holds:

Lemma 2.1.2 $L(P) \leq \omega(P) \leq \chi(P)$.

In the following discussion, we use ω to denote the maximum number of pairwise intersecting paths in G which is equal to the clique number of the associated conflict graph G_c.

2.2 Routing and wavelength assignment

In this thesis, an optical network will be represented as an undirected graph $G = (V(G), E(G))$ in the undirected model and as a directed graph $G = (V(G), A(G))$ in the directed model.

We use $p(x,y)$ to denote a path (undirected or directed) in G connecting node x and node y. A *connection request (call)* consists of a pair of nodes (x,y) (unordered in undirected model, or ordered in directed model) in G, and the call is realized along a path between these two end nodes. An instance I is a multi-set of requests (two requests are considered as distinct elements of I even if they have the same end points). We say that a set P_I of paths realizes an instance I if for each $(x,y) \in I$, there is exactly one path $p(x,y)$ in P_I. We will simply denote P_I as P if there is no ambiguity.

Let G be a graph, I be an instance of requests, and P_I be a set of paths that realizes I. The wavelength assignment (WA) problem, denoted by (G, I, P_I), assigns wavelengths to paths in P_I with the minimum number of wavelengths such that the paths receive different wavelengths if they intersect. This minimum number of wavelengths is denoted by $\chi(G, I, P_I)$. If we consider wavelengths as colors, for the problem (G, I, P_I), we try to find a vertex coloring of the corresponding conflict graph $G_c(P_I, E)$, such that any two adjacent vertices are colored differently, and the number of colors used is minimized. It is easy to see that $\chi(G, I, P_I)$ is equal to the chromatic number of $G_c(P_I, E)$. The routing and wavelength assignment (RWA) problem, denoted by (G, I), seeks for a set P_I of paths that realizes I and a solution for the WA problem (G, I, P_I) such that $\chi(G, I, P_I)$ is minimized among all P_I's. We use $\chi(G, I)$ to denote the smallest $\chi(G, I, P_I)$ over all P_I's.

There are several related optimization problems (definitions adopted from [15]):

Path Coloring (PC): Given a set of calls, assign paths and colors to the calls such that calls receive different colors if their paths intersect. The optimization goal is to minimize the number of colors. PC is the same as the RWA problem in optical networks. (Application: Minimize the number of wavelengths in an all-optical WDM network.)

PC with Pre-specified Paths (PCwPP): Same as path coloring, but the paths are specified as part of the input. PCwPP is the same as the WA problem.

Path Packing (PP) Given a set of calls, assign paths to the calls such that the maximum edge load L is minimized. (Application: Minimize the required link capacity if all calls request the same bandwidth.)

Maximum Edge-Disjoint Paths (MEDP): Given a set of calls, select a subset of the calls and assign edge-disjoint paths to the calls in that subset. Maximize the cardinality of the subset. (Application: Maximize the number of established calls when at most one call is allowed on any edge.)

Maximum Weight Edge-Disjoint Paths (MWEDP): Given a set of calls that are assigned positive weights, select a subset of the calls and assign edge-disjoint paths to the calls in that subset. Maximize the total weight of the subset.

MEDP with Pre-specified Paths (MEDPwPP): Same as MEDP, but the paths are specified as part of the input.

MWEDP with Pre-specified Paths (MWEDPwPP): Same as MWEDP, but the paths are specified as part of the input.

In the following discussions, we will use wavelength assignment and path coloring interchangeably.

For offline maximization problems, an algorithm is a ρ-approximation algorithm if it runs in polynomial time and always computes a solution whose objective value is at least $\frac{1}{\rho}$ of the optimum. For offline minimization problems, an algorithm is a ρ-approximation algorithm if it runs in polynomial time and always computes a solution whose objective value is at most ρ times the optimum [12].

By convention, for online problems, we use *competitive ratios* [9] to evaluate the performance of an algorithm. For online maximization problems, an algorithm is a ρ-competitive algorithm if it runs in polynomial time and always computes a solution whose objective value is at least $\frac{1}{\rho}$ of the optimum (normally the optimum offline solution). For online minimization problems, an algorithm is a ρ-competitive algorithm if it runs in polynomial time and always computes a solution whose objective value is at most ρ times the optimum [9].

2.3 Specific network topologies

There are several network topologies which are of particular interest in communications networks. In this section, we will introduce some of these networks. The definitions given below are for undirected graphs. Definitions for directed graphs can be easily obtained by replacing each undirected edge with two directed edges in opposite directions.

2.3.1 Ring networks

A ring $R_n = (\{1, 2, ..., n\}, \{(n, 1)\} \cup \{(i, i + 1)|i = 1, ..., n - 1\})$ is a graph that consists of a single cycle of length at least three. The ring is a popular topology in communications networks. It is simple and node-symmetric. There are two edge-disjoint paths between any two nodes on a ring. A ring network remains connected if any single node or link fails, thus providing good fault tolerance. Many practical networks, such as SONET rings, are arranged in a ring structure.

2.3.2 Trees and binary trees

A tree is a connected graph with no cycles. A *rooted tree* is a tree with one vertex chosen as root. A *rooted plane tree* or *planted tree* is a rooted tree with a left-to-right ordering specified for the children of each vertex. A *binary tree* is a rooted plane tree where each vertex has at most two children [43].

Trees are of particular interest since many practical networks in the telecommunications industry have a tree-like structure. Binary trees are important in data storage and information retrieval.

2.3.3 Trees of rings

The tree of rings is a graph that can be defined inductively as follows [15]:

1. A single ring is a tree of rings.
2. If TR is a tree of rings, then the graph obtained by adding a node-disjoint ring R to TR and then identifying one node of R with one node of TR is also a tree of rings.
3. No other graphs are trees of rings.

Equivalently, a tree of rings (Figure 2.1) is a connected graph TR whose edges can be partitioned into rings such that any two rings have at most one node in common, and for all pairs (u, v) of nodes in TR, all simple paths from u to v touch precisely the same rings. We say that a path touches a ring if it contains at least one edge of that ring. Furthermore, a path touches a node if it starts at that node, ends at that node, or passes through that node. Two paths intersect if they share an edge.

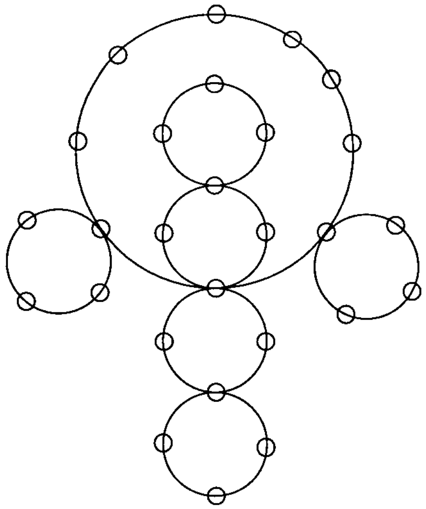


Figure 2.1: A tree of rings

Trees of rings can often be found in local-area networks: there is a main ring, with several sub-rings dangling from it, and sub-subrings from the sub-rings, and so on, as observed in [33]. Trees of rings are not expensive to build and require few additional links as compared to a tree topology. A tree of rings remains connected even if an arbitrary link fails in each ring, thus providing better fault tolerance than a tree network [15]. Many research efforts have been devoted to the study of trees of rings [13, 15, 29, 33].

2.4 Special instances

There are several special instances which are of particular interest [3]:

- An *h-k relation* is an instance in which each node is a source of at most h requests and a destination of at most k requests. In the directed model, a *1-1* relation is also known as a *permutation* instance.
- The *One-to-All* (also called *broadcast*) instance from a source node $x_0 \in V(G)$,

  $I_0 = \{(x_0, y) | y \in V(G), y \neq x_0\}$. Note that the instance I_0 is an *(N-1)-1* relation, where $N = |V(G)|$.
- The *All-to-All* instance $I_A = \{(x, y) | x \in V(G), y \in V(G), x \neq y\}$. Note that the instance I_A is an *(N-1)-(N-1)* relation, where $N = |V(G)|$.

Chapter 3

Previous Work

The RWA and WA problems on trees of rings are closely related to the RWA and WA problems on trees and rings. We will first review the existing results on these topologies.

3.1 Wavelength assignment on trees

In this section, we review the main results for the WA problem on trees. It has been proved that the problem is NP-hard, both for undirected and directed trees. These results have been extended to binary trees.

Theorem 3.1.1 (Erlebach and Jansen [17]) *For undirected and directed trees, the wavelength assignment problem is NP-hard.*

The above theorem holds even if we restrict instances to arbitrary trees and communication patterns with maximum load 3. The following result applies to binary trees and communication patterns of arbitrary load.

Theorem 3.1.2 (Erlebach and Jansen [18]) *For undirected and directed binary trees, the wavelength assignment problem is NP-hard.*

Since the WA problem on trees is NP-hard, there is no polynomial time algorithm to solve this problem unless $P = NP$. Nevertheless, there are approximation algorithms. All known wavelength assignment approximation algorithms for trees are greedy algorithms in essence (so are the algorithms for trees of rings). The algorithm in [26] uses a reduction

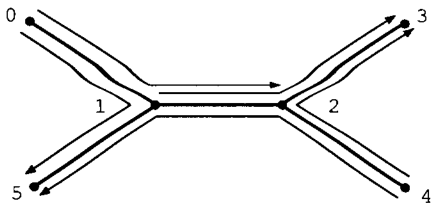


Figure 3.1: The communication pattern for the proof of the $5L/4$ lower bound

of the problem to an edge coloring problem on a bipartite graph. They have the following result:

Theorem 3.1.3 (Kaklamanis *et al.* [26]) *There exists a polynomial time greedy algorithm which realizes any pattern of communication requests of load L on a directed tree using at most* $5L/3$ *wavelengths.*

Using an adversary argument, a lower bound can be found for any greedy wavelength assignment algorithms for trees. An adversary algorithm ADV knows how a greedy algorithm performs the coloring, thus it can construct a communication pattern and it can be proved that there exists a lower bound on the number of wavelengths used by any greedy algorithm.

Theorem 3.1.4 (Caragiannis *et al.* [11]) *Let A be a deterministic greedy wavelength assignment algorithm in trees. There exists an algorithm ADV which, on any input $\delta > 0$ and integer $L > 0$, output a binary tree T and a pattern of communication requests P of load L on T, such that A colors P with at least* $(5/3 - \delta)L$ *colors.*

From the above theorem, it is easy to see that any deterministic greedy algorithm cannot use less than $5L/3$ colors. Therefore, the result of Theorem 3.1.3 is the best possible within the class of deterministic greedy algorithms.

Kumar [28] proved that there exists a communication pattern with load L which requires at least $5L/4$ wavelengths for any routing algorithms:

Theorem 3.1.5 (Kumar *et al.* [28]) *For any integer $l > 0$, there exists a communication pattern of load $L = 4l$ on a directed binary tree T that requires at least* $5L/4$ *wavelengths.*

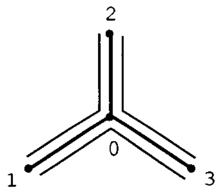


Figure 3.2: The communication pattern for the proof of the $3L/2$ lower bound

The proof of the above theorem can be shown in Figure 3.1. Each arrow represents $L/2$ requests. Note that there exist $5L/2$ requests and no more than 2 can be assigned the same color. Thus at least $5L/4$ colors are needed.

For undirected trees, Tarjan [38] proved that $3L/2$ is the sufficient number of colors for path coloring. From Figure 3.2, it is easy to see that $3L/2$ is also the lower bound [3]. In the figure, each line represents $L/2$ requests. There are $3L/2$ requests and none of them can be assigned the same colors. Thus at least $3L/2$ colors are needed.

For the all-to-all instance on directed trees, it has been proved that the chromatic number is equal to the load:

Theorem 3.1.6 (Gargano *et al.* [22]) *For the all-to-all instance I_A in any symmetric directed tree G, $\chi(G, I_A) = L(G, I_A)$, and there is an efficient algorithm to color the paths using $L(G, I_A)$ colors.*

It is observed that the above theorem does not hold for the undirected model. There are examples where $\chi(G, I_A) > L(G, I_A)$ for undirected trees.

The online WA problem on undirected trees was studied in [24]. They have the following result:

Theorem 3.1.7 (Gerstel *et al.* [24]) *For the online WA problem on a tree network with N nodes and load L, the number of wavelengths required is bounded by $(2L-1)\lceil \log_2 N \rceil$.*

Note that L is a lower bound for the online WA problem on trees. Thus, their algorithm achieves a competitive ratio of $O(\log N)$.

Bartal *et al.* [2] obtained an $O(\log N)$-competitive algorithm for the online WA problem on trees by reducing the problem to the online coloring of a d-inductive graph (a graph is d inductive if its vertices can be numbered in such a way that each vertex has at most d links to vertices with higher number). They also proved that any algorithm for online path coloring on trees with N nodes has a competitive ratio of $\Omega(\frac{\log N}{\log \log N})$.

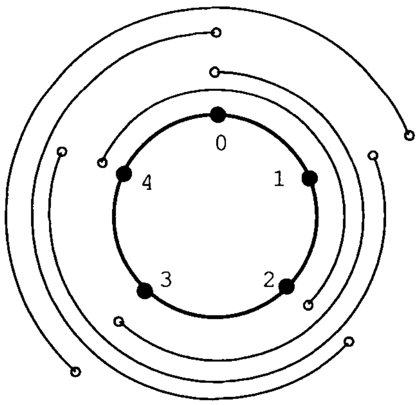


Figure 3.3: A lower bound for the wavelength assignment problem on rings

3.2 Routing and wavelength assignment on rings

Given a set of routing requests on an undirected ring network, there is an efficient algorithm to find a routing such that the maximum load is minimized:

Theorem 3.2.1 (Frank *et al.* [20]) *There is a linear time algorithm to find a routing for any instance I in any undirected ring network such that the maximum load is minimized.*

Wilfong [44] extended this result to the directed case:

Theorem 3.2.2 (Wilfong *et al.* [44]) *For any instance I in any directed ring network, there is an efficient algorithm to find a routing such that the maximum load is minimized.*

However, it has been proved that the RWA and WA problems are NP-hard, both for undirected and directed rings.

Theorem 3.2.3 (Garey *et al.* [21]) *The RWA and WA problems on rings are NP-hard.*

There is a $2L$ algorithm for the WA problem on rings:

Theorem 3.2.4 (Tucker [40]) *Given a set P of paths on a ring network with no wavelength conversion, $\chi(P) \leq 2L - 1$.*

This upper bound is tight for some instances. Consider the routing requests in Figure 3.3. There are 5 requests. The load is 3. It is easy to see that all these 5 paths must be assigned different colors. Thus $W = 2 \times 3 - 1 = 5$. In the general case, for a ring network with

$N = 2L - 1$ nodes, consider the N distinct requests $(x, x+L)$ (the numbers are modulo N). The load is L, and the number of wavelengths needed is $N = 2L - 1$.

Combine Theorems 3.2.1 and 3.2.2 with Theorem 3.2.4, one can get an efficient algorithm of approximation ratio 2 for the RWA problem on undirected and directed ring networks.

Upper bounds on $\chi(P)$ have also been found in terms of the clique number of the conflict graph of $G_c(P, E)$.

Theorem 3.2.5 (Karapetian [27]) *On a ring network, any set P of paths can be colored using at most 1.5ω colors such that any two paths sharing an edge are assigned different colors.*

For the all-to-all instance on rings, it has been proved that the chromatic number is equal to the maximum load:

Theorem 3.2.6 (Bermond *et al.* [8]) *For the all-to-all instance I_A in any directed ring G with N nodes, $\chi(G, I_A) = L(G, I_A) = \lceil \frac{1}{2} \lfloor \frac{N^2}{4} \rfloor \rceil$, and there is an efficient algorithm to color the paths using $L(G, I_A)$ colors.*

It is well known that the number of wavelengths needed can be reduced if we use wavelength converters. Ramaswami has the following result:

Theorem 3.2.7 (Ramaswami [34]) *Given a set P of paths on a ring with fixed wavelength conversion at one node and no conversion at all other nodes, $\chi(P) \leq L + 1$. There is a traffic of load L which requires $L + 1$ wavelengths on any rings with fixed conversion at every node and sufficiently large size.*

The online WA problem on rings was studied in [24]. They have the following result:

Theorem 3.2.8 (Gerstel *et al.* [24]) *For the online WA problem on a ring network with N nodes and load L, the number of wavelengths required is bounded by $L + L\lceil \log_2 N \rceil$.*

In [23], it was proved that $0.5L \log_2 N$ is a lower bound for the online WA problem on a ring network with N nodes and load L:

Theorem 3.2.9 (Gerstel *et al.* [23]) *On a ring network with N nodes, for every wavelength allocation algorithm, there is a communication pattern of load L which requires at least $0.5L \log_2 N$ wavelengths.*

3.3 Routing and wavelength assignment on trees of rings

3.3.1 Path packing

Given a set of routing requests on a tree of rings, the problem of minimizing the maximum load (path packing) can be reduced to path packing in rings [15]. The idea was to consider the path packing problem for each ring in the given tree of rings. Erlebach has the following result:

Theorem 3.3.1 (Erlebach [15]) *Path Packing can be solved optimally in polynomial time for undirected and directed trees of rings.*

3.3.2 Routing and wavelength assignment

As observed in [13, 15], the RWA and WA problems are NP-hard on trees of rings since the RWA and WA problems are NP-hard on rings. It is known that an algorithm for the WA problem on trees that uses at most αL colors can be used to obtain a 2α-approximation algorithm for the RWA problem on trees of rings, both in the undirected case [33] and in the directed case [29]: It is sufficient to remove an arbitrary link from each ring in the tree of rings (the *cut-one-link* heuristic) and to use the tree algorithm in the resulting tree; the maximum load of the obtained paths is at most twice the load of the paths in the optimal solution, which in turn is a lower bound on the optimal number of colors. In this way, a 3-approximation algorithm is obtained in the undirected case and a $\frac{10}{3}$-approximation algorithm in the directed case.

The all-to-all instance for the RWA problem on directed trees of rings was studied in [4]. It was shown that a routing that minimizes the maximum load L can be computed in polynomial time, and that the resulting paths can be colored optimally with L colors.

Theorem 3.3.2 (Beauquier *et al.* [4]) *For the all-to-all instance I_A in any directed tree of rings G, $\chi(G, I_A) = L(G, I_A)$, and there is an efficient algorithm to color the paths using $L(G, I_A)$ colors.*

From Theorems 3.1.6, 3.2.6 and 3.3.2, we know that for the all-to-all instance in directed trees, rings and trees of rings, the chromatic number is equal to the maximum load. It is not known whether the equality $\chi(G, I_A) = L(G, I_A)$ holds for any directed graph G.

For the online RWA problem on trees of rings, Bartal *et al.* [2] gave an $O(\log N)$-competitive algorithm. The idea is to use the cut-one-link heuristic: remove one edge from each ring on the given tree of rings, thus obtain a tree. The maximum load of the obtained tree is at most twice the maximum load of the original tree of rings. Combine this with the $O(\log N)$-competitive algorithm on trees, one can get an $O(\log N)$-competitive algorithm on trees of rings.

3.3.3 Wavelength assignment

The cut-one-link heuristic does not work for the WA problem on trees of rings, since the paths are already given and we are not allowed to re-route the paths. For the WA problem, Erlebach [15] proposed a greedy algorithm which used at most $4L$ and $8L$ colors on undirected and directed trees of rings, respectively. For undirected trees of rings, a 2-approximation algorithm for the WA problem was given in [13] for the special case in which each node is contained in at most two rings (i.e., in trees of rings with maximum node degree of four). Since our algorithm is related to these two algorithms, we will introduce these algorithms in some detail.

Given a set P of paths in a tree of rings $TR = (V, E)$, Erlebach [15] proposed the following greedy approximation algorithm.

Algorithm GreedyColoring (TR, P):

1. Initially, all paths are uncolored.

2. Process each node u of TR in a depth first search (DFS) order starting from an arbitrary node $s \in V$ as follows:

   Let P_u be the set of uncolored paths that touch u. Assign every path p of P_u, in arbitrary order, the color c with the smallest index such that no path intersecting p is already colored by c.

The coloring strategy used for P_u in the algorithm is known as *first-fit strategy.* The following result is known:

Theorem 3.3.3 (Erlebach [15]) *For the WA problem on trees of rings, GreedyColoring is a polynomial time algorithm that uses at most* $4L$ *colors in the undirected case.*

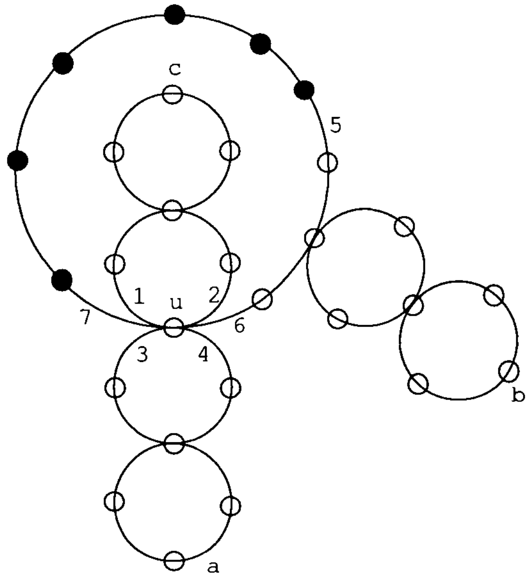


Figure 3.4: Coloring steps for Algorithm GreedyColoring

Proof In order to derive an upper bound on the number of colors used by the greedy algorithm, we consider an arbitrary path p at the time it is assigned its color and show that it can intersect only a bounded number of paths that have already been assigned a color prior to p. In Figure 3.4, assume that the dark nodes have been processed based on the *DFS* order already, and the algorithm is now processing u. Let r_0 denote the ring containing u and a dark node adjacent to u that has been processed already (there is only one such ring). Call ring r_0 the *current ring*. The uncolored paths touching u do not touch any dark node (otherwise they would have been colored in previous stages). They can be classified into two basic types. The first type uses at least one edge in r_0; for the arguments given below, take a path connecting nodes a and b in Figure 3.4 as a representative path of the first type. The second type does not use any edge in r_0; take a path connecting a and c as a representative path.

If p belongs to the first type, all pre-colored paths that intersect p have to use one or two of the links 3, 4, 5 and 6 (see Figure 3.4). There can be at most $4L - 2$ such paths: at most L paths use link 5, at most $L-1$ paths use link 6, and at most $2L-1$ paths use links 3 and 4 (the path p itself has to pass through link 6, thus there are at most $L - 1$ pre-colored paths which use link 6; similarly, there can be at most $2L - 1$ pre-colored paths which use links 3 and 4). Thus, $4L$ colors are enough to ensure a valid coloring.

If p belongs to the second type, all previously colored paths that intersect p must also touch u and touch a ring containing u that is touched by p. There are at most four links incident to u that belong to the two rings touched by p (in the figure, these are links 1, 2, 3 and 4). All pre-colored paths that intersect p must pass through one of these four links. Therefore, the total number of such paths is at most $4L - 2$ (because the path p itself has to pass through two of these four links), and $4L$ colors are enough to ensure a valid coloring. ▌

Since L is a lower bound for the WA problem on trees of rings, GreedyColoring achieves an approximation ratio of 4. This is the best published result for the WA problem on trees of rings with arbitrary node degree.

For undirected trees of rings, there is a 2-approximation algorithm [13]. It was later found to work only for trees of rings with node degree at most 4 (i.e., each node is contained in at most two rings) [16]. The main idea of the algorithm and the counterexample are described as follows. First, we will introduce the terminology and notation that were used in the algorithm.

For a given tree of rings *TR*, a tree structure underlying *TR* (also called *underlying tree*) can be defined as follows [1]. Pick an arbitrary ring r_0 as the root and mark r_0 as *ready*, and mark all other rings as *available*. Then repeat the following process: Let v be a node in a ring r that is marked ready and that is contained in at least one ring that is marked available. Then make all available rings that contain v to be chilrdren of r and mark them ready as well. This procedure induces a unique parent ring to each ring (except the root ring), and the set of all rings forms a tree structure.

In the algorithm [13], they first fix a *DFS* order of the vertices of the underlying tree (note that this is different from the *DFS* order of the vertices of the tree of rings in Erlebach's algorithm). They process the rings of the tree of rings *TR* one by one in the order defined by the *DFS* order in the corresponding underlying tree. Call the processing of a ring r a stage. Let Q denote the set of all the paths containing some edges in r. At a stage for processing r, they color all the uncolored paths in Q. Their colors will not be changed thereafter. Repeat the above process until all rings are processed.

At each stage, the algorithm consists of two parts. In the first part, along r in the clockwise direction, they construct m sets A_i ($1 \leq i \leq m$) of paths, such that each A_i

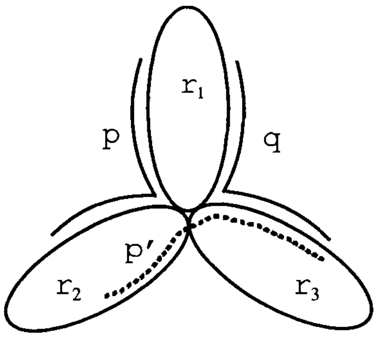


Figure 3.5: A counterexample to Deng's algorithm

contains a set of edge-disjoint paths. In the second phase, along r in the counterclockwise direction, they partition the rest of Q into t sets B_j $(1 \leq j \leq t)$ of paths, such that each B_j contains a set of edge-disjoint paths. The outputs of each stage are $A = \{A_i \mid 1 \leq i \leq m\}$ and $B = \{B_j \mid 1 \leq j \leq t\}$.

After processing each ring (except the first one), they assign colors to the newly generated classes as follows. Suppose r is the ring which has just been processed with output A_i's and B_j's, and suppose $D = \{D_l \mid 1 \leq l \leq n\}$ consists of the n sets D_l of paths generated before processing r. They merge a set in A or B with a set in D as long as they have a common path. Repeat this process until no such two sets exist. For the remaining sets in A, B and D, arbitrarily merge a set in A or B with a set in D. Repeat the process until no such two sets exist. The resulting sets are the color classes obtained after processing r.

They have the following result:

Theorem 3.3.4 (Deng *et al.* [13]) *For the WA problem on trees of rings with node degree at most* 4, *there is a polynomial time approximation algorithm which uses at most* 2ω *colors in the undirected case.*

The algorithm was intended for general trees of rings. Later it was found out that for trees of rings with node degree greater than 4, the algorithm will create invalid colorings [16]. Consider the example shown in Figure 3.5. The tree of rings consists of three rings r_1, r_2 and r_3 (in *DFS* order). At each stage, the algorithm only considers the paths which share some edges with the current ring. After processing ring r_1, paths p and q get the same color (say, color 1), since they are edge-disjoint. When the algorithm processes ring r_2, only paths p and p' are considered. Since p' is edge-disjoint with p, according to the algorithm, p' will get color 1, too. However, this is an invalid coloring since q and p' intersect on ring r_3. The reason for the problem is that an uncolored path that shares some edges with the current ring can intersect a colored path that does not share an edge with the current ring. Such conflicts do not occur for trees of rings with node degree at most 4.

Chapter 4

Wavelength Assignment on Trees of Rings

In this chapter, we consider undirected trees of rings and give our new wavelength assignment algorithm. Let P be a set of paths in a tree of rings $TR = (V, E)$. Recall that $\chi(P)$ is the number of colors required to color P in an optimal solution, i.e., the chromatic number of the associated conflict graph, ω is the maximum number of pairwise intersecting paths, and L is the maximum load among all edges. From Lemma 2.1.2, we know that $L \leq \omega \leq \chi(P)$. We will show that for a tree of rings with node degree at most 8, $3L$ is both the upper and lower bounds for the WA problem. Furthermore, $2\frac{1}{16}\omega$ and $2\frac{3}{13}\omega$ are upper bounds for trees of rings with node degrees at most 4 and 6, respectively.

4.1 3L upper bound

4.1.1 Algorithm

Inspired by the work of Deng [13] and Erlebach [15], we propose the following greedy approximation algorithm. First, we choose an arbitrary node $s \in V$ and fix a depth-first search (*DFS*) order of the vertices of *TR*. Then we process the nodes in *TR* one by one in this *DFS* order. Call the processing of a node a *stage*. At a stage for processing u (call u the *current node*), we color all the uncolored paths that touch u (recall that a path touches a node if it starts at that node, ends at that node, or passes through that node), using the previously used colors as many as possible. At the end of each stage, all the paths which

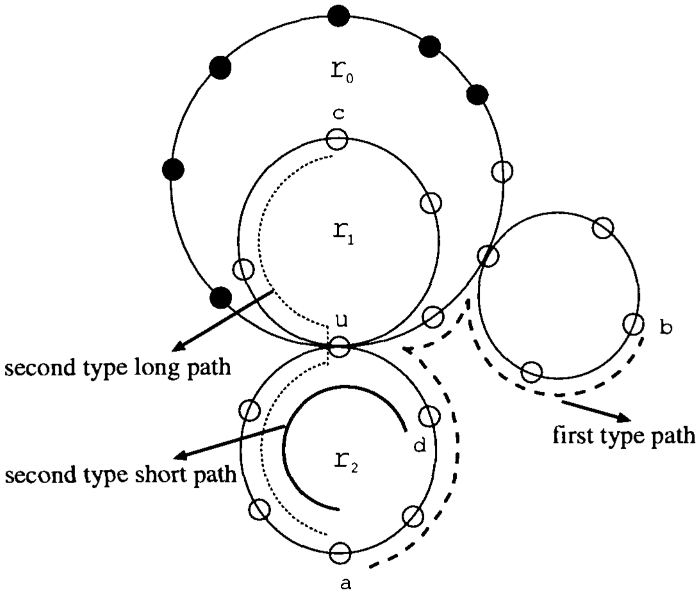


Figure 4.1: Illustration of the coloring steps and different types of paths

touch the current node u are colored. Their colors will not be changed thereafter. After u is processed, we process the next node in the *DFS* order until all nodes are processed.

In order to derive an upper bound on the number of colors used by the greedy algorithm, we show that an arbitrary path p can intersect only a bounded number of paths that have already been assigned a color prior to p. In Figure 4.1, assume that the dark nodes have been processed based on the *DFS* order, and that the algorithm is now processing node u. Let r_0 denote the ring containing u and a dark node adjacent to u that has been processed already (there is only one such ring). Call ring r_0 the *current ring*. The uncolored paths touching u do not touch any dark node (otherwise they would have been colored in previous stages). These paths can be classified into two basic types. The first type paths touch r_0 (i.e., use at least one edge of r_0); for the arguments given below, take a path connecting nodes a and b in Figure 4.1 as a representative path of the first type. The second type paths do not use any edge in r_0, and they can be further divided into two types: *long paths* and *short paths*. The long paths pass through two rings which contain node u, and the short paths only pass through one ring which contains node u. In Figure 4.1, the path connecting a and c is a long path, and the path connecting a and d is a short path. Note that if p belongs to the second type, all previously colored paths that intersect p must also touch u

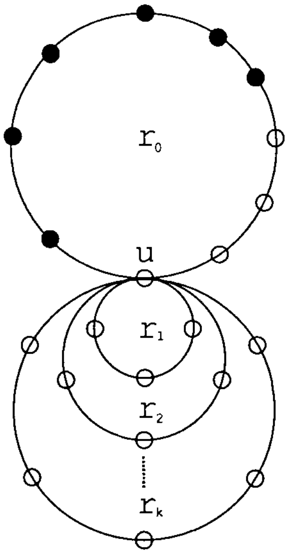


Figure 4.2: Coloring steps for Algorithm G_Coloring

and touch a ring containing u that is touched by p. Our strategy is to color the uncolored long paths first, since these paths are more difficult to color. After coloring all the long paths, we color the short paths.

In the following algorithm, we process the uncolored paths which touch node u and one of the rings r_i, for i in the order from 0 to k (suppose the maximum node degree is $2(k+1)$, see Figure 4.2). Ring r_0 is the unique ring which contains node u and a dark node adjacent to u. Rings r_1 to r_k are other rings which contain node u. It is easy to see that uncolored paths which touch ring r_0 belong to the first type. In our algorithm, the paths which touch ring r_0 are colored before the paths which do not touch ring r_0. When processing ring r_i, suppose the uncolored paths form a set U and all previously colored paths form a set V. We first construct a bipartite graph $G(U, V)$ consisting of two sets of nodes, U and V, such that (p, q) is an edge of G if and only if $p \in U$ and $q \in V$ are edge-disjoint. (Note that U and V are sets of paths. For simplicity, in the discussion, we use them to denote the node sets in the corresponding bipartite graph). Then we find a maximum matching M in G. For each edge $(p, q) \in M$, color p using the color of q. Color the remaining paths in U using colors by first-fit strategy. In this way, the number of colors needed is reduced.

The detailed algorithm is given in Figure 4.3.

```
Algorithm G_Coloring ($TR$, $P$)
Input: A set $P$ of paths on a tree of rings $TR$
Output: A valid wavelength assignment for $P$
begin
  1. Initially, all paths are uncolored.
  2. Process every node $u$ of $TR$ one by one based on the $DFS$ order starting from
       an arbitrary node $s$.
     For node $u$ being processed, let $P_u$ be the set of uncolored paths that touch $u$ and
       let $r_i$ ($0 \leq i \leq k$) be the rings touching $u$ with $r_0$ being the current ring.
     Color the paths of $P_u$ as follows:
     2.1 Color long paths (see Figure 4.2)
       For $i = 0, 1, ..., k - 1$ do the following steps:
         Let $V^i$ be the set of pre-colored paths and $U^i$ be the set of uncolored long
           paths that touch node $u$ and ring $r_i$.
         Construct a bipartite graph $G(U^i, V^i)$ such that there is an edge $(p, q)$ in $G$
           if and only if paths $p \in U^i$ and $q \in V^i$ are edge-disjoint.
         Find a maximum matching $M$ in G.
         For each $(p, q) \in M$, color $p$ with the color of $q$.
         Coloring the remaining paths in $U^i$ using new colors by first-fit strategy.
     2.2 Color short paths
       For $i = 0, 1, ..., k$ do the following:
         Color the uncolored short paths that touch $u$ and ring $r_i$ by first-fit strategy.
end.
```

Figure 4.3: The greedy algorithm for the WA problem on trees of rings

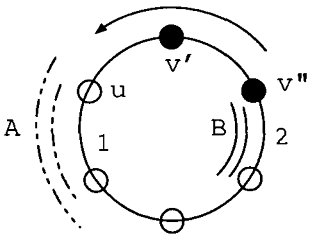


Figure 4.4: The coloring step for a node of degree 2

4.1.2 Analysis

In this section, we first show that G_Coloring uses at most $3L$ colors for trees of rings with node degree at most 8, then we show that for general trees of rings, G_Coloring uses at most $4L$ colors.

We have the following result:

Theorem 4.1.1 *For the WA problem on trees of rings with maximum node degree* 8*, G_Coloring is a polynomial-time algorithm that uses at most* $3L$ *colors in the undirected case.*

Proof Step 2.1 of Algorithm G_Coloring may involve constructing a bipartite graph and finding a maximum matching from the bipartite graph. To construct a bipartite graph efficiently when we process each node u, we do a pre-processing as follows: For the set P of the given paths and the tree of rings TR, we construct the conflict graph $G_c(P, E)$. This can be done in $O(N \times L^2)$ time, where N is the number of nodes in TR and L is the maximum load of TR. By refering the conflict graph $G_c(P, E)$, a bipartite graph G can be constructed in $O(L^2)$ time at Step 2.1 because G has $O(L)$ nodes. The time complexity for finding a maximum matching in G is $O(L^{2.5})$ [25]. Since Step 2.1 is executed N times, the time complexity of G_Coloring is $O(N \times L^{2.5})$. Clearly, the algorithm runs in polynomial time.

If the degree of the current node u is 2, for any uncolored path p which contains u, the number of pre-colored paths which can intersect p is at most $2L-1$. To see this, suppose we have reached u from a dark node v' along the counterclockwise direction (see Figure 4.4). In other words, v' is the *predecessor* of u in the DFS order. We can find another dark node v'' (possibly the same as v') if we go from u along the current ring in the counterclockwise direction. If we denote the edges as shown in Figure 4.4, there can be at most $L - 1$ pre-colored paths using link 1 (path p itself has to pass through link 1, thus there are at most

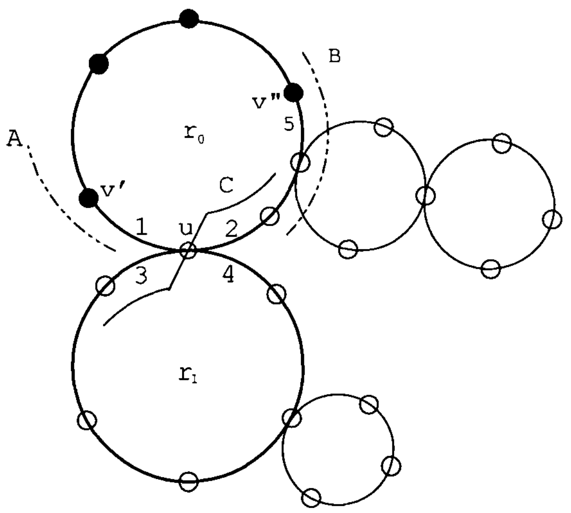


Figure 4.5: First type paths

$L-1$ instead of L pre-colored paths which pass through link 1), and at most L pre-colored paths using link 2. Thus, at most $2L-1$ pre-colored paths can intersect p, and $2L$ colors are enough to ensure a valid coloring. From now on, we only consider the case where the degree of the current node is greater than 2.

Since the node degree is restricted to 8, there can be at most 4 rings which contain node u. These rings are called r_0, r_1, r_2 and r_3, respectively. Step 2.1 of the algorithm only needs to do the coloring for rings r_0, r_1, and r_2 ($i = 0, 1, 2$, respectively), because an uncolored long path which touches node u and ring r_i can only touch a ring r_j with $j > i$. We prove the theorem by showing that in Step 2.1, for each of these three rings, the number of colors needed is bounded by $3L$.

We first show that when we process the first ring r_0 ($i = 0$), for any uncolored path p which touches u and r_0, the number of pre-colored paths that can intersect p is less than $3L$. All the pre-colored paths which can possibly intersect p are shown in Figure 4.5 (in the figure, rings r_2 and r_3 are not shown). These paths use some of the links 1, 2 and 5. They can be classified into the following three types:

- A: pass through link 1;

- B: pass through link 5 but do not pass through link 1;

- C: pass through link 2 but do not pass through link 1 or link 5.

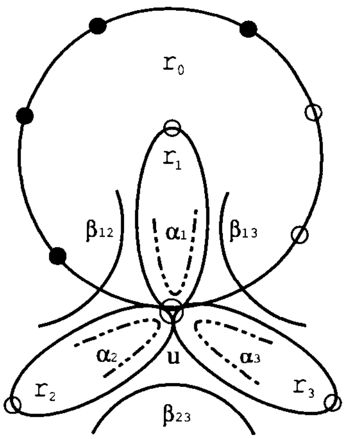


Figure 4.6: Second type paths

Note that types A and B paths were colored in previous stages, whereas type C paths are colored in the current stage. In Algorithm G_Coloring, to minimize the use of new colors, we try to color the uncolored paths using those colors of the paths in A and B, then use new colors if no more old colors can be used. For any uncolored path p, there are at most $2L$ types A and B paths, and at most $L-1$ colored type C paths (because path p itself must pass through link 2). Thus, at most $3L-1$ pre-colored paths can intersect p, and $3L$ colors are enough to ensure a valid coloring in this case.

Now we prove that $3L$ colors are enough for coloring ring r_1 in Step 2.1 ($i = 1$).

Note that an uncolored path which touches node u and ring r_i can only touch exactly one ring r_j with $j > i$. Suppose the sets of pre-colored paths on rings r_1, r_2 and r_3 are α_1, α_2 and α_3, respectively. The sets of uncolored long paths are β_{12}, β_{13} and β_{23}, respectively, as shown in Figure 4.6. It is easy to see that a path in α_1 and a path in β_{23} can never intersect, so they can be assigned the same color. Similarly, α_2 and β_{13}, α_3 and β_{12} can never intersect. Thus the size of the maximum matching M for ring r_1 ($i = 1$) in Step 2.1 is at least min($|\alpha_2|$, $|\beta_{13}|$)+min($|\alpha_3|$, $|\beta_{12}|$). The number of new colors needed is at most $w_{new} = |\beta_{12}| + |\beta_{13}| - min(|\alpha_2|, |\beta_{13}|) - min(|\alpha_3|, |\beta_{12}|)$. The total number of colors used after coloring the long paths touching ring r_1 is at most $w_{new} + |\alpha_1| + |\alpha_2| + |\alpha_3|$.

We have the following constraints due to the maximum load L on any edge:

$$|\alpha_1| + |\alpha_2| + |\alpha_3| \leq 2L, \tag{4.1}$$

$$|\alpha_1| + |\beta_{12}| + |\beta_{13}| \leq 2L, \tag{4.2}$$

$$|\alpha_2| + |\beta_{12}| + |\beta_{23}| \leq 2L, \tag{4.3}$$

and

$$|\alpha_3| + |\beta_{13}| + |\beta_{23}| \leq 2L. \tag{4.4}$$

To get Inequality (4.1), consider ring r_0 in Figure 4.6. The pre-colored paths in α_1, α_2 and α_3 must pass through one of the two links on ring r_0 which are incident to node u. Since the maximum load is L, there can be at most $2L$ such paths. Other inequalities can be derived similarly.

Consider the relations between $|\alpha_2|$ and $|\beta_{13}|$, and between $|\alpha_3|$ and $|\beta_{12}|$, we have the following four cases:

1. $|\alpha_2| \geq |\beta_{13}|$ and $|\alpha_3| \geq |\beta_{12}|$

   In this case, $w_{new} = 0$ and we do not need any new color, so the number of colors is bounded by $2L$.

2. $|\alpha_2| \geq |\beta_{13}|$ and $|\alpha_3| < |\beta_{12}|$

   The number of new colors needed is at most $w_{new} = |\beta_{12}| - |\alpha_3|$. Using Inequalities (4.1), (4.2) and (4.3), we can get that the total number of colors is:

$$\begin{aligned} w_{new} + |\alpha_1| + |\alpha_2| + |\alpha_3| &= |\beta_{12}| - |\alpha_3| + |\alpha_1| + |\alpha_2| + |\alpha_3| \\ &= |\beta_{12}| + |\alpha_1| + |\alpha_2| \\ &= \frac{1}{2}(2|\alpha_1| + 2|\alpha_2| + 2|\beta_{12}|) \\ &= \frac{1}{2}(|\alpha_1| + |\beta_{12}| + |\alpha_2| + |\beta_{12}| + |\alpha_1| + |\alpha_2|) \\ &\leq \frac{1}{2}(2L + 2L + 2L) \\ &= 3L. \end{aligned}$$

3. $|\alpha_2| < |\beta_{13}|$ and $|\alpha_3| \geq |\beta_{12}|$

   The number of new colors needed is at most $w_{new} = |\beta_{13}| - |\alpha_2|$. Using Inequalities (4.1), (4.2) and (4.4), we can get that the total number of colors is:

$$\begin{aligned}
w_{new} + |\alpha_1| + |\alpha_2| + |\alpha_3| &= |\beta_{13}| - |\alpha_2| + |\alpha_1| + |\alpha_2| + |\alpha_3| \\
&= |\beta_{13}| + |\alpha_1| + |\alpha_3| \\
&= \frac{1}{2}(2|\alpha_1| + 2|\beta_{13}| + 2|\alpha_3|) \\
&= \frac{1}{2}(|\alpha_1| + |\beta_{13}| + |\alpha_3| + |\beta_{13}| + |\alpha_1| + |\alpha_3|) \\
&\leq \frac{1}{2}(2L + 2L + 2L) \\
&= 3L.
\end{aligned}$$

4. $|\alpha_2| < |\beta_{13}|$ and $|\alpha_3| < |\beta_{12}|$

   The number of new colors needed is at most $w_{new} = |\beta_{12}| - |\alpha_3| + |\beta_{13}| - |\alpha_2|$. Using Inequality (4.2), we can get that the total number of colors is:

$$\begin{aligned}
w_{new} + |\alpha_1| + |\alpha_2| + |\alpha_3| &= |\beta_{12}| - |\alpha_3| + |\beta_{13}| - |\alpha_2| + |\alpha_1| + |\alpha_2| + |\alpha_3| \\
&= |\alpha_1| + |\beta_{12}| + |\beta_{13}| \\
&\leq 2L.
\end{aligned}$$

We have proved that for $i = 1$, we use at most $3L$ colors for long paths. Suppose that in coloring ring r_1, we used a set α'_1 of colors from α_1, used some colors from $\alpha_2 \cup \alpha_3$ and used a set δ of new colors. Clearly, $|\alpha'_1| \leq |\alpha_1|$. Now we prove that $3L$ is enough for coloring the long paths on ring r_2.

To color ring r_2, the number of colors needed is at most $W = |\beta_{23}| + |\delta| + |\alpha'_1| + |\alpha_2| + |\alpha_3|$. If $|\delta| = 0$, on ring r_2 and r_3, we have the following constraints:

$$2|\beta_{23}| + |\alpha'_1| + |\alpha_2| + |\alpha_3| \leq 4L,$$

and

$$|\alpha'_1| + |\alpha_2| + |\alpha_3| \leq 2L.$$

Adding them, we can get that $W = |\beta_{23}| + |\alpha'_1| + |\alpha_2| + |\alpha_3| \leq 3L$.

Assume $|\delta| > 0$. From previous analysis, we know that

$$|\delta| \leq |\beta_{12}| + |\beta_{13}| - |\alpha_2| - |\alpha_3|.$$

On ring r_1, we have

$$|\beta_{12}| + |\beta_{13}| + |\alpha'_1| \leq 2L.$$

Adding the preceding two inequalities, we have

$$|\delta| + |\alpha'_1| + |\alpha_2| + |\alpha_3| \leq 2L.$$

On ring r_2 and r_3, we have

$$2|\beta_{23}| + |\alpha'_1| + |\alpha_2| + |\alpha_3| + |\delta| \leq 4L.$$

Adding the last two inequalities, we can get that

$$W = |\delta| + |\alpha'_1| + |\beta_{23}| + |\alpha_2| + |\alpha_3| \leq 3L.$$

Thus, on rings r_1 and r_2, the long paths can be colored using at most $3L$ colors.

For the remaining short paths, they only pass through one of these three rings. All pre-colored paths have to use one of the two links incident to u. There are at most $2L - 1$ such paths, and $2L$ colors are enough to ensure a valid coloring.

Thus, G_Coloring uses at most $3L$ colors in the undirected case for trees of rings with node degree at most 8. ▮

Unfortunately, the above technique cannot be applied directly to the general case (arbitrary node degree). The algorithm G_Coloring works for trees of rings with node degree greater than 8 using at most $4L$ colors. This follows directly from [15], but we are unable to prove an upper bound better than $4L$ for the general case. The author of the thesis thinks that in practical optical networks, node degree of 8 is probably enough if we consider load balance and fault tolerance issues.

The 3-approximation cut-one-link heuristic [33] works only for the RWA problem on trees of rings in which the routes are not fixed, not for the WA problem on trees of rings with pre-specified paths. The 2-approximation algorithm for the WA problem given in [13] works only in the case of node degree four, and it uses at most 2ω colors, where ω is the maximum number of pairwise intersecting paths. In the worst case scenario given in Section 4.3, $\omega = 3L$. It is not hard to see that our greedy algorithm runs more efficiently.

4.2 A more efficient greedy algorithm

The time complexity of the general algorithm G_Coloring is $O(N \times L^{2.5})$ (Section 4.1). With some modification, the algorithm can run more efficiently while still achieving the $3L$ upper bound for the special case of node degree at most 8. The modified algorithm is given in Figure 4.7 and has time complexity $O(N \times L^2)$.

We have the following result:

Theorem 4.2.1 *For the WA problem on trees of rings with maximum node degree at most 8, G_Coloring_I is a polynomial time algorithm that uses at most 3L colors in the undirected case.*

Proof The time complexity of Algorithm G_Coloring_I is $O(N \times L^2)$, since it does not involve finding a maximum matching in a bipartite graph, and the time is still needed for checking whether two paths intersect.

From the previous analysis, we know that if the degree of the current node is 2, or if p belongs to the first type, the number of colors needed is bounded by $3L$.

If p belongs to the second type (i.e., p does not use any edge from ring r_0), we have the following constraints due to the maximum load L on any edge (see Figure 4.6):

$$\begin{aligned} |\alpha_1| + |\alpha_2| + |\alpha_3| &\leq 2L, &(4.5)\\ |\alpha_1| + |\beta_{12}| + |\beta_{13}| &\leq 2L, &(4.6)\\ |\alpha_2| + |\beta_{12}| + |\beta_{23}| &\leq 2L, &(4.7)\end{aligned}$$

and

$$|\alpha_3| + |\beta_{13}| + |\beta_{23}| \leq 2L. \tag{4.8}$$

According to the algorithm, the maximum number of colors needed is

$$w = max\{|\alpha_1|, |\beta_{23}|\} + max\{|\alpha_2|, |\beta_{13}|\} + max\{|\alpha_3|, |\beta_{12}|\}.$$

There are only eight possible cases:

1. $|\alpha_1| \geq |\beta_{23}|$ and $|\alpha_2| \geq |\beta_{13}|$

   (a) $|\alpha_3| \geq |\beta_{12}|$

   From Inequality (4.5), we can get that:

$$\begin{aligned} w &= max\{|\alpha_1|, |\beta_{23}|\} + max\{|\alpha_2|, |\beta_{13}|\} + max\{|\alpha_3|, |\beta_{12}|\}\\ &= |\alpha_1| + |\alpha_2| + |\alpha_3|\\ &\leq 2L. \end{aligned}$$

```
**Algorithm** G_Coloring_I ($TR$,$P$)
**Input:** A set $P$ of paths on a tree of rings $TR$
**Output:** A valid wavelength assignment for $P$
**begin**
    1. Initially, all paths are uncolored.
    2. Process every node $u$ of $TR$ one by one based on the $DFS$ order starting from
          an arbitrary node $s$.
       For node $u$ being processed, let $P_u$ be the set of uncolored paths that touch $u$,
          and $r_i$'s ($0 \leq i \leq 3$) be the rings touching $u$ with $r_0$ being the current ring.
       Color the paths of $P_u$ as follows:
       2.1 Color the first type paths using first-fit strategy.
       2.2 Color the second type long paths (see Figure 4.6)
           Let the pre-colored paths on rings $r_1$, $r_2$ and $r_3$ be $\alpha_1$, $\alpha_2$ and $\alpha_3$, respectively,
              and the uncolored *long paths* be $\beta_{12}$, $\beta_{13}$ and $\beta_{23}$, respectively.
           Color $\beta_{12}$ using the colors of $\alpha_3$ if possible.
           Color $\beta_{13}$ using the colors of $\alpha_2$ if possible.
           Color $\beta_{23}$ using the colors of $\alpha_1$ if possible.
           Coloring the remaining *long paths* using colors by first-fit strategy.
       2.3 Color short paths
           For $i = 0, 1, ..., 3$ do the following:
              Color the uncolored short paths that touch $u$ and ring $r_i$ by first-fit strategy.
**end.**
```

Figure 4.7: A more efficient greedy coloring algorithm

(b) $|\alpha_3| < |\beta_{12}|$

From Inequalities (4.5), (4.6) and (4.7), we can get that:

$$\begin{aligned} w &= max\{|\alpha_1|, |\beta_{23}|\} + max\{|\alpha_2|, |\beta_{13}|\} + max\{|\alpha_3|, |\beta_{12}|\} \\ &= |\alpha_1| + |\alpha_2| + |\beta_{12}| \\ &= \frac{1}{2}(2|\alpha_1| + 2|\alpha_2| + 2|\beta_{12}|) \\ &= \frac{1}{2}(|\alpha_1| + |\beta_{12}| + |\alpha_2| + |\beta_{12}| + |\alpha_1| + |\alpha_2|) \\ &\leq \frac{1}{2}(2L + 2L + 2L) \\ &= 3L. \end{aligned}$$

2. $|\alpha_1| \geq |\beta_{23}|$ and $|\alpha_2| < |\beta_{13}|$

(a) $|\alpha_3| \geq |\beta_{12}|$

From Inequalities (4.5), (4.6) and (4.8), we can get that:

$$\begin{aligned} w &= max\{|\alpha_1|, |\beta_{23}|\} + max\{|\alpha_2|, |\beta_{13}|\} + max\{|\alpha_3|, |\beta_{12}|\} \\ &= |\alpha_1| + |\beta_{13}| + |\alpha_3| \\ &= \frac{1}{2}(2|\alpha_1| + 2|\beta_{13}| + 2|\alpha_3|) \\ &= \frac{1}{2}(|\alpha_1| + |\beta_{13}| + |\alpha_3| + |\beta_{13}| + |\alpha_1| + |\alpha_3|) \\ &\leq \frac{1}{2}(2L + 2L + 2L) \\ &= 3L. \end{aligned}$$

(b) $|\alpha_3| < |\beta_{12}|$

From Inequality (4.6), we can get that:

$$\begin{aligned} w &= max\{|\alpha_1|, |\beta_{23}|\} + max\{|\alpha_2|, |\beta_{13}|\} + max\{|\alpha_3|, |\beta_{12}|\} \\ &= |\alpha_1| + |\beta_{12}| + |\beta_{13}| \\ &\leq 2L. \end{aligned}$$

3. $|\alpha_1| < |\beta_{23}|$ and $|\alpha_2| \geq |\beta_{13}|$

(a) $|\alpha_3| \geq |\beta_{12}|$

From Inequalities (4.5), (4.7) and (4.8), we can get that:

$$\begin{aligned} w &= max\{|\alpha_1|, |\beta_{23}|\} + max\{|\alpha_2|, |\beta_{13}|\} + max\{|\alpha_3|, |\beta_{12}|\} \\ &= |\beta_{23}| + |\alpha_2| + |\alpha_3| \\ &= \frac{1}{2}(2|\beta_{23}| + 2|\alpha_2| + 2|\alpha_3|) \\ &= \frac{1}{2}(|\alpha_2| + |\beta_{23}| + |\alpha_3| + |\beta_{23}| + |\alpha_2| + |\alpha_3|) \\ &\leq \frac{1}{2}(2L + 2L + 2L) \\ &= 3L. \end{aligned}$$

(b) $|\alpha_3| < |\beta_{12}|$

From Inequality (4.7), we can get that:

$$\begin{aligned} w &= max\{|\alpha_1|, |\beta_{23}|\} + max\{|\alpha_2|, |\beta_{13}|\} + max\{|\alpha_3|, |\beta_{12}|\} \\ &= |\beta_{23}| + |\alpha_2| + |\beta_{12}| \\ &\leq 2L. \end{aligned}$$

4. $|\alpha_1| < |\beta_{23}|$ and $|\alpha_2| < |\beta_{13}|$

(a) $|\alpha_3| \geq |\beta_{12}|$

From Inequality (4.8), we can get that:

$$\begin{aligned} w &= max\{|\alpha_1|, |\beta_{23}|\} + max\{|\alpha_2|, |\beta_{13}|\} + max\{|\alpha_3|, |\beta_{12}|\} \\ &= |\beta_{23}| + |\beta_{13}| + |\alpha_3| \\ &\leq 2L. \end{aligned}$$

(b) $|\alpha_3| < |\beta_{12}|$

From Inequalities (4.6), (4.7) and (4.8), we can get that:

$$\begin{aligned} w &= max\{|\alpha_1|, |\beta_{23}|\} + max\{|\alpha_2|, |\beta_{13}|\} + max\{|\alpha_3|, |\beta_{12}|\} \\ &= |\beta_{23}| + |\beta_{13}| + |\beta_{12}| \\ &= \frac{1}{2}(2|\beta_{23}| + 2|\beta_{13}| + 2|\beta_{12}|) \\ &= \frac{1}{2}(|\beta_{23}| + |\beta_{13}| + |\beta_{23}| + |\beta_{12}| + |\beta_{12}| + |\beta_{13}|) \\ &\leq \frac{1}{2}(2L + 2L + 2L) \\ &= 3L. \end{aligned}$$

Thus, the second type long paths can be colored using at most $3L$ colors.

For the remaining short paths, they only pass through one of the rings containing u. All pre-colored paths have to use one of the two links incident to u. There are at most $2L - 1$ such paths, and $2L$ colors are enough to ensure a valid coloring.

Thus, G_Coloring_I uses at most $3L$ colors in the undirected case for trees of rings with node degree at most 8. ▮

4.3 3L lower bound

Erlebach [15] proved that $4L$ is an upper bound for the WA problem on general trees of rings, and we have just proved that $3L$ is an upper bound for the WA problem on trees of rings with node degree at most 8. A natural question is, whether this $3L$ upper bound can be further improved? In this section, we show that $3L$ is also a lower bound for the WA problem, even for the restricted trees of rings with node degree at most 4 (note that if a tree of rings contains more than one ring, the node degree is at least 4):

Theorem 4.3.1 *For any integer $l > 0$, there exists a communication pattern of load $L = 2l$ on a tree of rings TR that requires at least $3L$ colors in the undirected case.*

Proof Let A, B, C, D, E and F be the path sets, each having $L/2$ paths, as shown in Figure 4.8. Note that the maximum load of the tree of rings in the figure is L and there are a total of $3L$ requests. None of them can be assigned the same color since they are pairwise intersecting ($3L$ is the maximum clique number in the corresponding conflict graph). This shows that at least $3L$ colors are needed. ▮

This lower bound is very interesting since one cannot do better than $3L$ even for trees of rings with node degree at most 4, no matter what algorithm (greedy or non-greedy) one uses. Our algorithm achieves the $3L$ upper bound for trees of rings with node degree up to 8, which is the best possible result.

4.4 Upper bounds based on clique number

In Section 4.1, we have proved that $3L$ is the upper bound for the WA problem on trees of rings with node degree at most 8. Since L is a lower bound for any optimal solution, it turns out that our algorithm achieves an approximation ratio of 3. As we have pointed out,

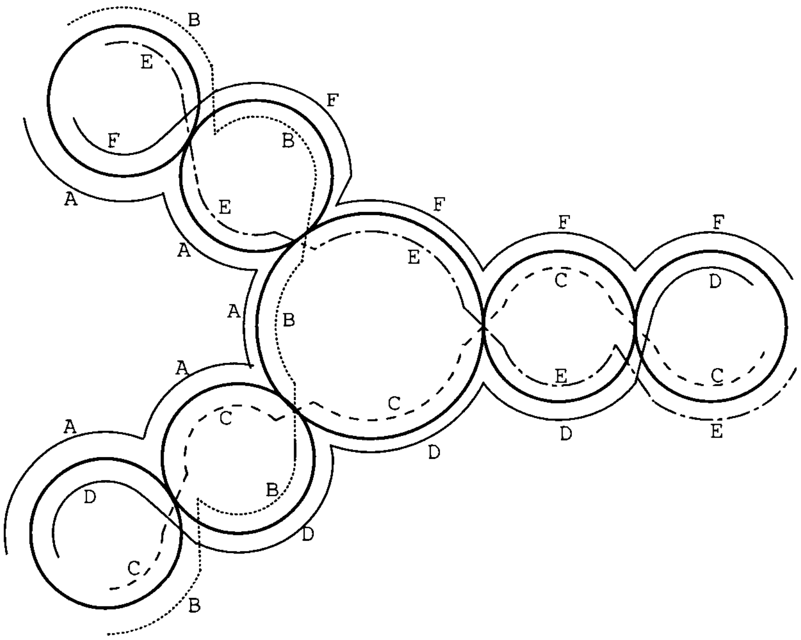


Figure 4.8: The communication pattern for the proof of the 3L lower bound

$L \leq \omega$. With some modifications to the proving technique, we can get better approximation ratios. Recall that $\chi(P)$ is the number of colors required to color P in an optimal solution and ω is the maximum number of pairwise intersecting paths. We can prove that $2\frac{1}{16}\omega$ and $2\frac{3}{13}\omega$ are upper bounds for trees of rings with node degrees at most 4 and 6, respectively. Since ω is a lower bound for $\chi(P)$, this improves the approximation ratios to $2\frac{1}{16}$ and $2\frac{3}{13}$, respectively.

In order to simplify the proof, we first introduce a matching method (called *multiple-step matching*) which is slightly different from the matching methond in Step 2.1 of Algorithm G_Coloring, and this multiple-step matching method works only for node degree at most 6. Then we get the $2\frac{1}{16}\omega$ and $2\frac{3}{13}\omega$ upper bounds for the multiple-step matching. By showing that the general algorithm G_Coloring (which works for arbitrary node degree) is at least as good as the multiple-step matching when used in the special case of node degree at most 6, we conclude that G_Coloring also achieves the $2\frac{1}{16}\omega$ and $2\frac{3}{13}\omega$ upper bounds.

We have the following convention:

1. For a bipartite graph $G(U^a, V^a)$ with node sets U^a and V^a, if M_a is a matching for $G(U^a, V^a)$, then $U^a_{M_a} = \{p|(p,q) \in M_a, p \in U^a\}$, $V^a_{M_a} = \{q|(p,q) \in M_a, q \in V^a\}$. In

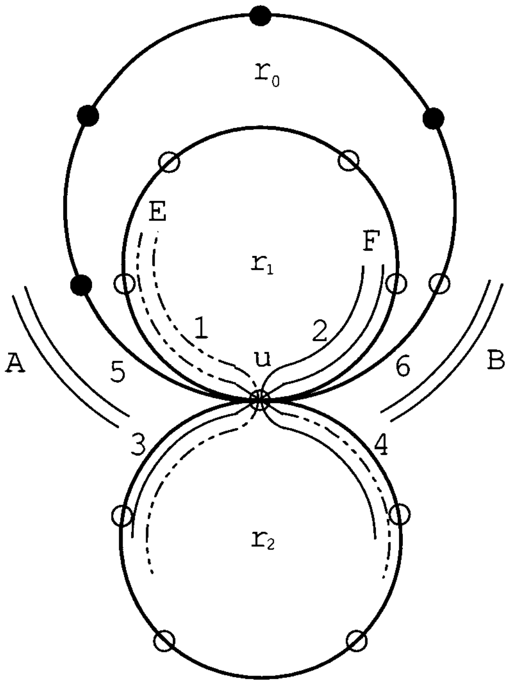


Figure 4.9: The coloring procedure of the multiple-step matching

other words, $U^a_{M_a}$ and $V^a_{M_a}$ are the subsets of U^a and V^a saturated by matching M_a. We will simply denote $U^a_{M_a}(V^a_{M_a})$ as $U_{M_a}(V_{M_a})$ if there is no ambiguity.

2. Suppose p is a path and Q is a set of paths. When we say that p intersects with Q, we mean that p intersects with *every path* in Q. When we say that p is disjoint with Q, we mean that p is disjoint with *some path* in Q.

Since we have restricted the node degree to 6, there can be at most three rings which contain node u. As shown in Figure 4.9, assume that the dark nodes have been processed based on the *DFS* order already, and that the algorithm is now processing node u. We further assume that r_0 is the ring containing node u and a dark node adjacent to u (there is only one such ring). The other two rings are named r_1 and r_2, respectively. According to Step 2.1 of Algorithm G_Coloring, we will first color the uncolored paths which touch ring r_0 and node u, then color the uncolored long paths which touch ring r_1 and node u.

In Algorithm G_Coloring, when processing the uncolored paths which touch ring r_0, a bipartite graph will be contructed and a maximum matching will be found. Call this matching method the *global matching*. As shown in Figure 4.5, the uncolored first type paths form a set C. Recall that the set A contains the pre-colored paths which pass through link 1, and the set B contains the pre-colored paths which pass through link 5 but do not pass through link 1. When coloring ring r_0, replace the global matching method with the following *two-step matching* method:

1. Construct a bipartite graph $G_a(U^a, V^a)$ where $U^a = C$ and $V^a = A$, such that (p, q) is an edge of G_a if and only if $p \in U^a$ and $q \in V^a$ are edge-disjoint. Find a maximum matching M_a in G_a. For each edge $(p, q) \in M_a$, color p using the color of q.

2. Construct a bipartite graph $G_b(U^b, V^b)$ where $U^b = C \backslash U^a_{M_a}$ and $V^b = B$, such that (p, q) is an edge of G_b if and only if $p \in U^b$ and $q \in V^b$ are edge-disjoint. Find a maximum matching M_b in G_b. For each edge $(p, q) \in M_b$, color p using the color of q. Color the rest $(C \backslash \{U^a_{M_a} \cup U^b_{M_b}\})$ using first-fit strategy, and these colors form a set D.

We have the following lemma:

Lemma 4.4.1 *In Step 2.1 of Algorithm G_Coloring, when coloring the long paths touching ring* r_0*, the global matching method uses no more than the colors used by the above two-step matching method.*

Proof When processing ring r_0, let W_{old} be the number of colors used already, W_{new} be the number of new colors used by the global matching method, and W'_{new} be the number of new colors used by the two-step matching. When processing ring r_0, the global matching method in Algorithm G_Coloring uses no more than the colors used by the above two-step matching method. To see this, notice that the size of the maximum matching in Algorithm G_Coloring is greater than or equal to the sum of the sizes of the two maximum matchings in the two-step matching algorithm. Thus, more old colors can be used in the global matching method, hence less new colors are needed. The total number of colors used by the global matching method is $W_{old} + W_{new} \leq W_{old} + W'_{new}$, since $W_{new} \leq W'_{new}$. Thus, the global matching method is at least as good as the two-step matching method. ▮

When processing the uncolored long paths which touch ring r_1, if we denote the edges as shown in Figure 4.9, we can divide the uncolored second type long paths to the following two types:

- E: pass through link 1 and ring r_2;

- F: psss through link 2 and ring r_2.

The pre-colored paths have to use either link 5 or 6. Denote the paths which use link 5 as set A and the paths which use link 6 as set B. Note that paths in A were colored during previous stages, whereas paths in B could be colored during the current stage or previous stages. Clearly, the sizes of both A and B are bounded by ω. In the multiple-step matching method, we will color E using colors from set A and B if possible; for the remaining E, use a set C of new colors. The unused colors in A and B form sets A' and B', respectively. After finishing coloring E, we color F using the colors in set A', B' and C if possible; for the rest of F, use a set D of new colors. The reason that we use sets A' and B' instead of sets A and B in the coloring of F is that the colors in A and B which are already used to color E may conflict with F.

According to the above analysis, when coloring the long paths touching ring r_1, consider the following *three-step matching* method:

1. Construct a bipartite graph $G_{a1}(U^{a1}, V^{a1})$ where $U^{a1} = E$ and $V^{a1} = A$, such that (p, q) is an edge of G_{a1} if and only if $p \in U^{a1}$ and $q \in V^{a1}$ are edge-disjoint. Find a maximum matching M_{a1} in G_{a1}. For each edge $(p, q) \in M_{a1}$, color p using the color of q. Construct a bipartite graph $G_{b1}(U^{b1}, V^{b1})$ where $U^{b1} = E \backslash U^{a1}_{M_{a1}}$ and $V^{b1} = B$, such that (p, q) is an edge of G_{b1} if and only if $p \in U^{b1}$ and $q \in V^{b1}$ are edge-disjoint. Find a maximum matching M_{b1} in G_{b1}. For each edge $(p, q) \in M_{b1}$, color p using the color of q. Color the rest $(E \backslash \{U^{a1}_{M_{a1}} \cup U^{b1}_{M_{b1}}\})$ using first-fit strategy, and these colors form a set C.

2. Construct a bipartite graph $G_{a2}(U^{a2}, V^{a2})$ where $U^{a2} = F$ and $V^{a2} = A'$ (A' is the set of colors $A \backslash V^{a1}_{M_{a1}}$), such that (p, q) is an edge of G_{a2} if and only if $p \in U^{a2}$ and $q \in V^{a2}$ are edge-disjoint. Find a maximum matching M_{a2} in G_{a2}. For each edge $(p, q) \in M_{a2}$, color p using the color of q. Construct a bipartite graph $G_{b2}(U^{b2}, V^{b2})$ where $U^{b2} = F \backslash U^{a2}_{M_{a2}}$ and $V^{b2} = B'$ (B' is the set of colors $B \backslash V^{b1}_{M_{b1}}$), such that (p, q) is an edge of G_{b2} if and only if $p \in U^{b2}$ and $q \in V^{b2}$ are edge-disjoint. Find a maximum matching M_{b2} in G_{b2}. For each edge $(p, q) \in M_{b2}$, color p using the color of q.

3. Construct a bipartite graph $G_c(U^c, V^c)$ where $U^c = F \backslash \{U^{a2}_{M_{a2}} \cup U^{b2}_{M_{b2}}\}$ and $V^c = C$, such that (p, q) is an edge of G_c if and only if $p \in U^c$ and $q \in V^c$ are edge-disjoint. Find a maximum matching M_c in G_c. For each edge $(p, q) \in M_c$, color p using the

color of q. Color the rest $(F \backslash \{U^{a2}_{M_{a2}} \cup U^{b2}_{M_{b2}} \cup U^{c}_{M_c}\})$ using first-fit strategy, and these colors form a set D.

We have the following lemma:

Lemma 4.4.2 *In Step 2.1 of Algorithm G_Coloring, when coloring the long paths touching ring r_1, the global matching method uses no more than the colors used by the above three-step matching method.*

Proof When processing ring r_1, let W_{old} be the number of colors used already, W_{new} be the number of new colors used by the global matching method, and W'_{new} be the number of new colors used by the three-step matching. When processing ring r_1, the global matching method in Algorithm G_Coloring uses no more than the colors used by the above three-step matching method. To see this, notice that the size of the maximum matching in Algorithm G_Coloring is greater than or equal to the sum of the sizes of the first two maximum matchings in the three-step matching algorithm. Thus, more old colors can be used in the global matching method, hence less new colors are needed. The total number of colors used by the global matching method is $W_{old} + W_{new} \leq W_{old} + W'_{new}$, since $W_{new} \leq W'_{new}$. Thus, the global matching method is at least as good as the three-step matching method. ▮

Before giving the main theorem, we have the following lemma:

Lemma 4.4.3 *In the modified coloring method associated with ring r_0, for any edge $(p,q) \in M_a$, p intersects with $A \backslash V_{M_a}$ or q intersects with $U_{M_b} \cup D$. For any edge $(p,q) \in M_b$, p intersects with $B \backslash V_{M_b}$ or q intersects with D.*

Proof We prove the lemma by contradiction. Suppose there exists an edge $(p,q) \in M_a$, such that p is disjoint with some path $f' \in A \backslash V_{M_a}$, and q is disjoint with some path $f'' \in U_{M_b} \cup D$. Then we can assign p the color of f', and assign f'' the color of q. This increases the matching M_a by 1, contradicts to the fact that M_a is a maximum matching. The proof for M_b is similar. ▮

We have the following result:

Theorem 4.4.4 *For the WA problem on trees of rings, when using the multiple-step matching instead of the global matching method, Algorithm G_Coloring is a polynomial time algorithm that uses at most $2\frac{1}{16}\omega$ and $2\frac{3}{13}\omega$ colors for node degrees up to 4 and 6, respectively.*

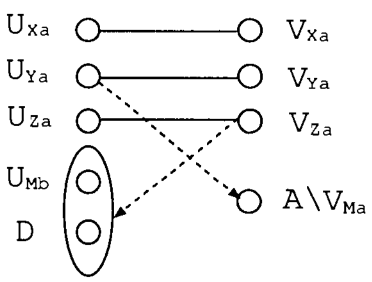


Figure 4.10: The bipartite graph associated with M_a

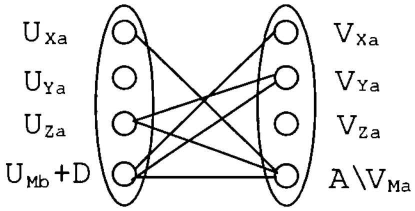


Figure 4.11: The conflict graph associated with M_a

Proof If p belongs to the first type, the configuration can be shown in Figure 4.5. The number of colors needed is $W = |A| + |B| + |D|$. It is easy to see that $|A|$, $|B|$ and $|C|$ are all bounded by ω. If D is empty, the number of colors needed is $|A| + |B|$, which is bounded by 2ω. If $A\backslash V_{M_a}$ or $B\backslash V_{M_b}$ is empty, the number of colors needed is $|B| + |C|$ or $|A| + |C|$, which is also bounded by 2ω. From now on, we assume that $A\backslash V_{M_a}$, $B\backslash V_{M_b}$ and D are not empty.

According to Lemma 4.4.3, we can divide M_a into three types:

1. $X_a = \{(p,q)|(p,q) \in M_a$, p intersects with $A\backslash V_{M_a}$ and q intersects with $U_{M_b} \cup D\}$;

2. $Y_a = \{(p,q)|(p,q) \in M_a$, p is disjoint with $A\backslash V_{M_a}$ and q intersects with $U_{M_b} \cup D\}$;

3. $Z_a = \{(p,q)|(p,q) \in M_a$, p intersects with $A\backslash V_{M_a}$ and q is disjoint with $U_{M_b} \cup D\}$;

The definitions for X_b, Y_b and Z_b are similar:

1. $X_b = \{(p,q)|(p,q) \in M_b$, p intersects with $B\backslash V_{M_b}$ and q intersects with $D\}$;

2. $Y_b = \{(p,q)|(p,q) \in M_b$, p is disjoint with $B\backslash V_{M_b}$ and q intersects with $D\}$;

3. $Z_b = \{(p,q)|(p,q) \in M_b$, p intersects with $B\backslash V_{M_b}$ and q is disjoint with $D\}$;

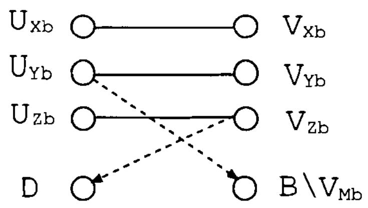


Figure 4.12: The bipartite graph associated with M_b

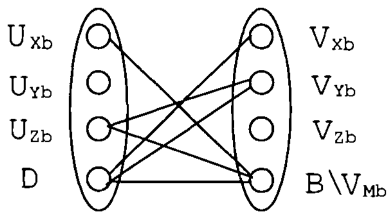


Figure 4.13: The conflict graph associated with M_b

We can show the matching M_a in a bipartite graph (see Figure 4.10). In the figure, a solid line represents a matching. A dashed arrow means that each path in the tail is disjoint from some path in the head. For example, in Figure 4.10, there is a solid line between U_{X_a} and V_{X_a} since this is a matching. There is a dashed arrow from U_{Y_a} to $A\backslash V_{M_a}$, which means that each path in U_{Y_a} is disjoint from some path in $A\backslash V_{M_a}$. We can thus construct a conflict graph (Figure 4.11). In this figure, a solid line means that the two sets of paths are conflicting, i.e., they cannot be assigned the same colors. According to the definitions of X_a, Y_a and Z_a, V_{X_a}, V_{Y_a} and $A\backslash V_{M_a}$ are conflicting with $U_{M_b} \cup D$; U_{X_a} and U_{Z_a} are conflicting with $A\backslash V_{M_a}$. We can show that V_{Y_a} conflicts with U_{Z_a} by contradiction. Suppose there exist two edges $(p, q) \in Y_a$, $(p', q') \in Z_a$ such that p' and q are edge-disjoint, we can improve the matching M_a by 1: assign p the same color as a path in $A\backslash V_{M_a}$, assign p' the same color as q and assign a path in $U_{M_b} \cup D$ the same color as q'. This contradicts to the fact that M_a is a maximum matching. Thus V_{Y_a} and U_{Z_a} must be conflicting.

According to the conflict graph for the matching M_a (Figure 4.11), we have the following constraints:

$$\left.\begin{array}{rcl} |X_a| + |Y_a| + |A| - |M_a| + |M_b| + |D| & \leq & \omega \\ |X_a| + |Z_a| + |A| - |M_a| + |M_b| + |D| & \leq & \omega \\ |Y_a| + |Z_a| + |A| - |M_a| + |M_b| + |D| & \leq & \omega \end{array}\right\}. \tag{4.9}$$

Similarly, according to the bipartite graph for the matching M_b (see Figure 4.12), we can construct the conflict graph for the matching M_b (Figure 4.13). We have the following constraints:

$$\left.\begin{array}{rcl} |X_b| + |Y_b| + |B| - |M_b| + |D| & \leq & \omega \\ |X_b| + |Z_b| + |B| - |M_b| + |D| & \leq & \omega \\ |Y_b| + |Z_b| + |B| - |M_b| + |D| & \leq & \omega \end{array}\right\}. \tag{4.10}$$

The maximum number of colors needed is $|A|+|B|+|D|$. This is our object function and we want to minimize it. Combine Inequalities (4.9) and (4.10) with the fact that $|A|$, $|B|$ and $|C|$ are all bounded by ω, and solve this optimization problem using the Optimization Toolbox in MATLAB, we can get that $|A| + |B| + |D| \leq 2\frac{1}{16}\omega$.

If the maximum node degree is 4, from the proof of Theorem 4.1.1, we know that $2L$ colors are enough to color the second type of short paths, thus 2ω colors are enough, since $L \leq \omega$. Combine this with the $2\frac{1}{16}\omega$ result for the first type paths, we have proved that for trees of rings with node degree at most 4, $2\frac{1}{16}\omega$ is an upper bound on the number of colors needed.

If the maximum node degree is 6, using similar technique, for the second type long paths which touch ring r_1 (Figure 4.9), we can get the following constraints:

$$\left.\begin{array}{rcl} |X_{a1}| + |Y_{a1}| + |A| - |M_{a1}| + |M_{b1}| + |C| & \leq & \omega \\ |X_{a1}| + |Z_{a1}| + |A| - |M_{a1}| + |M_{b1}| + |C| & \leq & \omega \\ |Y_{a1}| + |Z_{a1}| + |A| - |M_{a1}| + |M_{b1}| + |C| & \leq & \omega \end{array}\right\}, \tag{4.11}$$

$$\left.\begin{array}{rcl} |X_{b1}| + |Y_{b1}| + |B| - |M_{b1}| + |C| & \leq & \omega \\ |X_{b1}| + |Z_{b1}| + |B| - |M_{b1}| + |C| & \leq & \omega \\ |Y_{b1}| + |Z_{b1}| + |B| - |M_{b1}| + |C| & \leq & \omega \end{array}\right\}, \tag{4.12}$$

$$\left.\begin{array}{rcl} |X_{a2}| + |Y_{a2}| + |A| - |M_{a1}| - |M_{a2}| + |M_{b2}| + |M_c| + |D| & \leq & \omega \\ |X_{a2}| + |Z_{a2}| + |A| - |M_{a1}| - |M_{a2}| + |M_{b2}| + |M_c| + |D| & \leq & \omega \\ |Y_{a2}| + |Z_{a2}| + |A| - |M_{a1}| - |M_{a2}| + |M_{b2}| + |M_c| + |D| & \leq & \omega \end{array}\right\}, \tag{4.13}$$

$$\left.\begin{array}{rcl} |X_{b2}| + |Y_{b2}| + |B| - |M_{b1}| - |M_{b2}| + |M_c| + |D| & \leq & \omega \\ |X_{b2}| + |Z_{b2}| + |B| - |M_{b1}| - |M_{b2}| + |M_c| + |D| & \leq & \omega \\ |Y_{b2}| + |Z_{b2}| + |B| - |M_{b1}| - |M_{b2}| + |M_c| + |D| & \leq & \omega \end{array}\right\}, \tag{4.14}$$

and

$$\left.\begin{aligned}|X_c|+|Y_c|+|C|-|M_c|+|D| &\leq \omega\\ |X_c|+|Z_c|+|C|-|M_c|+|D| &\leq \omega\\ |Y_c|+|Z_c|+|C|-|M_c|+|D| &\leq \omega\end{aligned}\right\}. \tag{4.15}$$

Note that each path in A and B must use one of the links 1, 2, 3 and 4; each of the long paths must use two of the links 1, 2, 3 and 4 (see Figure 4.9). Combine this with the fact that $L \leq \omega$, we have the following contraint:

$$|A|+|B|+2|M_{a1}|+2|M_{b1}|+2|M_{a2}|+2|M_{b2}|+2|M_c| \leq 4\omega. \tag{4.16}$$

The maximum number of colors needed is $|A|+|B|+|C|+|D|$. This is our object function and we want to minimize it. Combine Inequalities (4.11) to (4.16) with the fact that $|A|$ and $|B|$ are all bounded by ω, and solve this optimization problem using the Optimization Toolbox in MATLAB, we can get that $|A|+|B|+|C|+|D| \leq 2\frac{3}{13}\omega$.

For the second type short paths, from the proof of Theorem 4.1.1, we know that $2L$ colors are enough to color this type of paths, thus 2ω colors are also enough, since $L \leq \omega$. Combine this with the $2\frac{1}{16}\omega$ result for the first type paths and the $2\frac{3}{13}\omega$ result for the second type long paths, we have proved that for the WA problem on a tree of rings with node degree at most 6, $2\frac{3}{13}\omega$ is an upper bound on the number of colors. ∎

According to Lemma 4.4.1 and Lemma 4.4.2, we know that the global-matching method in Algorithm G_Coloring is at least as good as the two-step and three-step matching methods. Thus, G_Coloring achieves upper bounds of $2\frac{1}{16}\omega$ and $2\frac{3}{13}\omega$ as well.

From the above proof, we can see that G_Coloring uses at most $2\frac{1}{16}\omega$ and $2\frac{3}{13}\omega$ on trees of rings with node degrees at most 4 and 6, respectively. Combine this with the $3L$ upper bound, we have proved that G_Coloring achieves approximation ratios of $2\frac{1}{16}$, $2\frac{3}{13}$ and 3 on trees of rings with node degrees at most 4, 6 and 8, respectively.

Chapter 5

Crosstalk Reduction

Optical switches are now widely used in WDM all-optical networks. Directional couplers (DCs) can switch signals with multiple wavelengths. They are commonly used to build large optical switches. However, DCs suffer from an intrinsic crosstalk problem. In this chapter, we first study the nonblocking properties of Benes networks under crosstalk constraints and give lower and upper bounds on the number of wavelengths. We further study the nonblocking properties of the Banyan-type networks with extra stages and give the necessary conditions for the network to be strict-sense nonblocking (a network is called strict-sense nonblocking if any permutation instance can be routed in a link-disjoint manner no matter how the previous paths were established).

5.1 Introduction

A DC is used to combine and split signals in an optical network. It can be designed to be either wavelength selective or wavelength independent over a useful wide range. A 2×2 coupler consists of two input ports and two output ports, as shown in Figure 5.1. The most commonly used DCs are made by fusing two fibers together in the middle. A 2×2 coupler takes a fraction α of the power from input 1 and places it on output 1 and the remaining fraction $1-\alpha$ on output 2. Similarly, a fraction $1-\alpha$ of the power from input 2 is distributed to output 1 and the remaining power to output 2. α is called the coupling ratio [35]. $\alpha = 0$ and $\alpha = 1$ correspond to the *cross* state and the *bar* state, respectively (see Figure 5.2). Couplers are the building blocks for large optical switches and other optical devices.

The major shortcoming of DCs is crosstalk. Crosstalk is the effect of other signals on the

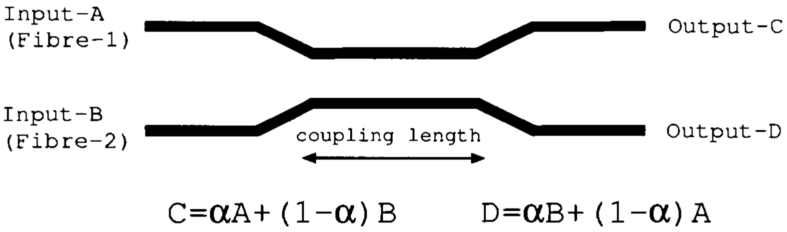


Figure 5.1: A directional coupler

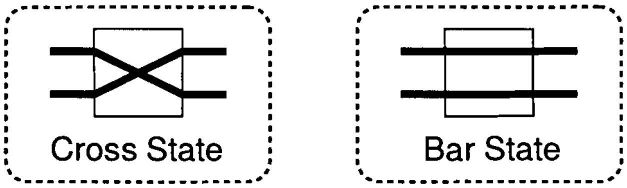


Figure 5.2: Two states of a constraint switching element

desired signal. Optical crosstalk occurs when two signal channels interact with each other. When two signals pass through a DC, a small portion of the signal power will be directed to the unintended output channel. If a signal passes many switches, the input signal will be distorted at the output due to the crosstalk introduced on the path. Studies indicate that switch crosstalk is the most significant factor which reduces the signal-to-noise ratio and limits the size of a network. Two forms of crosstalk exist in WDM systems: *interchannel crosstalk* and *intrachannel crosstalk* [35]. The first case is when the crosstalk signal is at a wavelength sufficiently different from the wavelength of the desired signal. The second case is when the crosstalk signal is at the same wavelength as that of the desired signal. The effect of intrachannel crosstalk can be much more severe than interchannel crosstalk, since the interchannel crosstalk can be filtered at the receiver, whereas intrachannel crosstalk cannot.

One method to reduce crosstalk is to ensure that a switch is not used by two input signals simultaneously. This is called the *space dilation.* Another method is to use *wavelength dilation* approach. The idea is, whenever two signals pass through a coupler, they must use two different wavelengths. To further reduce the interchannel crosstalk, one can add an additional filter for each wavelength at the output. The wavelength dilation approach can reduce the hardware cost. In this thesis, we use wavelength dilation method to achieve strict-sense nonblocking under crosstalk constraints.

5.2 Nonblocking properties

In this section, we give the necessary definitions. A network will be modelled as a directed graph with a subset S of nodes called *inputs* and a subset T of nodes called *outputs*. A routing request is an ordered pair (s, t) of the network, where $s \in S$ is a *source node* and $t \in T$ is a *destination node*. In this chapter, we only consider permutation instances, i.e., each node s can be a source of at most one request and each node t can be a destination of at most one request. Especially, when we say an arbitrary set of routing requests, we mean an arbitrary permutation instance. As is done in previous chapters, circuit-switching model is used instead of packet-switching. We only consider the networks without wavelength converters. For such networks, a path from s to t must be assigned the same wavelength on all of its links. If there is no crosstalk constraint on the switches, the only problem left is to realize the connection requests using the minimum number of wavelengths, while ensuring that the paths with the same wavelength are edge-disjoint. The *wavelength blocking* is the case when two paths try to use the same wavelength on the same link. Under the crosstalk constraint on the switches, however, a DC-based switching system must satisfy a certain crosstalk level. Intrachannel crosstalk will be generated when two light signals of the same wavelength pass through the same switching element (SE). We call such SE a crosstalk SE (CSE). The level of crosstalk can be represented using the number of CSEs along a lightpath.

A network is called *link rearrangeable nonblocking* if any set of connection requests given at once can be routed in a link-disjoint manner. It is called *wide-sense link nonblocking* if there is a routing algorithm that is able to route every sequence of requests in a link-disjoint manner. It is called *strict-sense link nonblocking* if any sequence of requests can be routed in a link-disjoint manner no matter how the previous paths were established. In WDM networks, we are concerned with the *wavelength nonblocking* [36]. A WDM network is called *k-wavelength rearrangeable nonblocking* if any set of connection requests given at once can be routed in such a way that any two paths sharing the same link are assigned different wavelengths and at most k wavelengths are used. We define the *k-wavelength wide-sense nonblocking* (*strict-sense nonblocking*) similarly. Clearly, when $k = 1$, wavelength nonblocking becomes link nonblocking. Under the crosstalk constraint, the nonblocking property is further specified by the number of CSE's allowed on any path. We say a network is *c-CSE* and *k-wavelength rearrangeable nonblocking* if any set of connection requests given at once can be routed in such a way that any two paths sharing the same link are assigned different wavelengths, at most k wavelengths are used, and there are at most c CSE's in each path. Similarly, we can define c-CSE and k-wavelength wide-sense (strict-sense) nonblocking.

In the following discussion, we use the *baseline* topology and all the SEs are assumed to be 2×2. The n-dimensional baseline network, denoted by B_n, consists of $n \times 2^{n-1}$ nodes arranged in n stages connecting $N = 2^n$ inputs and N outputs. In Figure 5.3(a), the stages are numbered $1, 2, ..., n$ from left to right. Each node can be represented by a binary label $v = x_1 x_2 ... x_{n-1} x_n \in \{0,1\}^n$. The ith binary baseline permutation δ_i, for $1 \leq i \leq n$, is defined by

$$\delta_i(x_1 ... x_{i-1} x_i x_{i+1} ... x_{n-1} x_n) = x_1 ... x_{i-1} x_n x_i x_{i+1} ... x_{n-1}$$

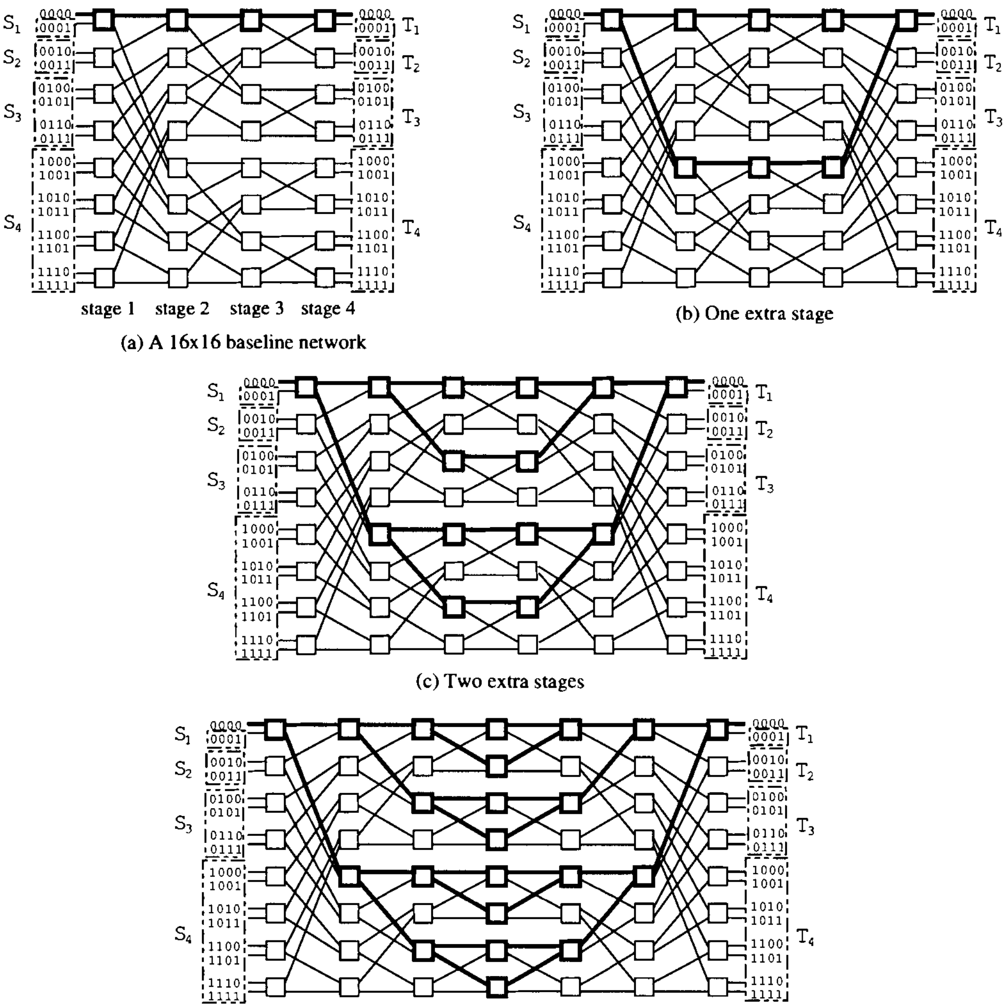
The ith baseline connection performs a cyclic shifting of the $n - i + 1$ least significant digits in the index to the right for one position [14]. Figure 5.3(a) shows a 16×16 baseline network. Baseline networks are topologically equivalent to butterfly and omega networks. They are often called Banyan-type networks.

In a baseline network, there is a unique path between each input and output. Adding extra stages to a baseline network can provide alternative paths. Figures 5.3(b), 5.3(c) and 5.3(d) show baseline networks with 1, 2 and 3 extra stages, respectively. Note that Figure 5.3(d) is actually a Benes network. We adapt the notation of [42] for convenience. For an $N \times N$ Banyan-type switch, we use the notation $B(x, c)$ to denote the Banyan-type network with x extra stages and at most c CSEs are allowed on the path of each connection. We try to find the minimum number of wavelengths w to make the $B(x, c)$ network strict-sense nonblocking for various x and c.

5.3 Previous work

In this section, we will review some of the results on the nonblocking properties of Banyan-type networks under crosstalk constraints.

When a Banyan-type network has x extra stages, there are $\log_2 N + x$ stages. When $c = \log_2 N + x$, no crosstalk constraint is enforced. In this case, we are actually asked to find the minimum w for wavelength nonblocking. Lemma 5.3.1 gives a sufficient condition for a $B(x, \log_2 N + x)$ network to be w-wavelength strict-sense nonblocking. When $c = 0$, no crosstalk is allowed on any node. In this case, the paths with the same wavelength are required to be node-disjoint. Lemma 5.3.2 gives a sufficient condition for a $B(x, 0)$ network



(a) A 16x16 baseline network

(b) One extra stage

(c) Two extra stages

(d) Benes network (x=3)

Figure 5.3: 16 × 16 Banyan-type networks with extra stages

to be 0-CSE and w-wavelength strict-sense nonblocking. Lemma 5.3.3 gives a sufficient condition for an $N \times N$ baseline network $B(0,c)$ to be c-CSE and w-wavelength strict-sense nonblocking, for $0 < c < \log_2 N + x$.

Lemma 5.3.1 [42] *An $N \times N$ $B(x, \log_2 N + x)$ network is w-wavelength strict-sense nonblocking if*

$$w \geq \begin{cases} x + \frac{3}{2}\sqrt{\frac{N}{2^x}} - 1, & \textit{if } \log_2 N + x \textit{ even} \\ x + \sqrt{\frac{2N}{2^x}} - 1, & \textit{if } \log_2 N + x \textit{ odd.} \end{cases}$$

Lemma 5.3.2 [42] *An $N \times N$ $B(x, 0)$ network is 0-CSE and w-wavelength strict-sense nonblocking if*

$$w \geq \begin{cases} 2x + 2\sqrt{\frac{N}{2^x}} - 1, & \textit{if } \log_2 N + x \textit{ even} \\ 2x + \frac{3}{2}\sqrt{\frac{2N}{2^x}} - 1, & \textit{if } \log_2 N + x \textit{ odd.} \end{cases}$$

Lemma 5.3.3 [42] *An $N \times N$ $B(0, c)$ network with even $\log_2 N$ is c-CSE and w-wavelength strict-sense nonblocking if*

$$w \geq \begin{cases} \frac{3}{2}\sqrt{N} + \lfloor \frac{1}{c+1}\sqrt{N} \rfloor - 1, & \textit{if } 0 < c \leq \frac{1}{2}\log_2 N \\ \frac{3}{2}\sqrt{N} - 1, & \textit{if } \frac{1}{2}\log_2 N < c < \log_2 N. \end{cases}$$

When $\log_2 N$ is odd, the condition becomes

$$w \geq \begin{cases} \sqrt{2N} + \lfloor \frac{1}{c+2}\sqrt{2N} \rfloor - 1, & \textit{if } 0 < c \leq \frac{1}{2}(\log_2 N - 1) \\ \sqrt{2N} + \lfloor \frac{1}{2(c+2)}\sqrt{2N} \rfloor - 1, & \textit{if } \frac{1}{2}(\log_2 N - 1) < c < \log_2 N. \end{cases}$$

The detailed proof can be found in [41, 42].

5.4 Benes network

When $x = \log_2 N - 1$, the $B(x,c)$ network becomes the Benes network, as shown in Figure 5.3(d). For an $N \times N$ Benes network, there are $2 \times \log_2 N - 1$ stages. We have the following result:

Theorem 5.4.1 *The necessary and sufficient number of wavelengths for an $N \times N$ $B(\log_2 N - 1, 2 \times \log_2 N - 1)$ Benes network to be w-wavelength strict-sense nonblocking is $w = \log_2 N$.*

Proof Since $c = 2 \times \log_2 N - 1$ and an $N \times N$ Benes network has exactly $2 \times \log_2 N - 1$ stages, no crosstalk constraint is enforced, thus only link blocking can occur. We first show that $\log_2 N$ wavelengths are enough for an $N \times N$ Benes network (BN_n) to be strict-sense nonblocking. Given a set of routing requests I, for any request $(s,t) \in I$, there are $N/2$ paths in BN_n to connect s and t. Call these paths *tagged paths.* Let P be a connection for $I \backslash \{(s,t)\}$. We say a path $q \in P$ gives m blocks to the $N/2$ paths if q shares a common edge with m of the $N/2$ paths. The number m can be easily calculated if the source and destination of q are known. We count the total number of blocks from the paths in P and derive the average number of blocks in one path from s to t.

For $1 \leq k \leq \log_2 N$, let S_k be the set of sources in I that intersect with the tagged paths at stage k. Then S_1 has 1 node and S_k has 2^{k-1} nodes. Similarly, let T_k be the set of destinations in I that intersect with the tagged paths at stage $2 \times \log_2 N - k$. Then T_1 has 1 node and T_k has 2^{k-1} nodes. Call S_k and T_k the *intersecting sets* with respect to (s,t). This can be shown in Figure 5.3(d).

For any path $q : s' \rightarrow t'$ in P, let $j \geq 1$ and $k \geq 1$ be the integers such that $s' \in S_j$ and $t' \in T_k$. Then q contributes 2^r link blocks, where $r = max\{\log_2 N - j - 1, \log_2 N - k - 1\}$. For $r = \log_2 N - 2$, we have $j = k = 1$. Since $|S_1| \leq 1$ and $|T_1| \leq 1$, there can be at most 2 paths, each of which contributes $2^{\log_2 N - 2}$ blocks. In general, for $i \geq 2$ and $r = \log_2 N - i - 1$, we have $k = i$ and $j \geq i$ or $j = i$ and $k \geq i$. Since $|S_i| = 2^{i-1}$ and $|T_i| = 2^{i-1}$, there can be at most $2^{i-1} + 2^{i-1} = 2^i$ paths, each of which contributes $2^{\log_2 N - i - 1}$ link blocks. Summarizing the above and from the fact that I has at most N pairs, the paths in P can contribute at most

$$\sum_{i=1}^{\log_2 N - 1} 2^{\log_2 N - i - 1} \times 2^i = (\log_2 N - 1) \times N/2$$

link blocks. Since there are $N/2$ paths between s and t, the average number of blocks on each path is at most

$$\frac{(\log_2 N - 1) \times N/2}{N/2} = \log_2 N - 1.$$

Therefore, $\log_2 N$ wavelengths are enough to ensure wavelength nonblocking.

Now we show that $\log_2 N$ is also the lower bound. It can be proved by contradiction. Suppose $w < \log_2 N$ wavelengths are enough for a BN_n network to be strict-sense nonblocking. We can disprove this by finding a connection sequence which cannot be satisfied by less than $\log_2 N$ wavelengths. For any request (s,t), we can define intersecting sets S_i's and T_i's with respect to (s,t), where $1 \leq i \leq \log_2 N$. Let connection requests from S_i to $T_{\log_2 N}$, and connection requests from $S_{\log_2 N}$ to T_i use wavelength w_i, for $1 \leq i \leq \log_2 N - 1$. From the proof of the upper bound, we know that there are a total of 2^i elements in S_i and T_i, and in the worst case, each of which can block $2^{\log_2 N-i-1}$ paths of the $N/2$ paths from s to t. Thus, each S_i and T_i can block exactly $2^i \times 2^{\log_2 N-i-1} = N/2$ paths. The total number of the elements in S_i $(1 \leq i < \log_2 N)$ is $\sum_{i=1}^{\log_2 N-1} 2^{i-1} = \frac{N}{2} - 1$, which is strictly less than the number of elements in $T_{\log_2 N}$ $(N/2)$. Thus, we have enough elements in $T_{\log_2 N}$ to create these connections. Similarly, we have enough elements in $S_{\log_2 N}$ for connections destined to T_i $(1 \leq i < \log_2 N)$. Thus (s, t) cannot use wavelength w_i $(1 \leq i \leq \log_2 N - 1)$. It is easy to see that we cannot satisfy the connection request from s to t since all wavelengths have been used, a contradiction. So the lower bound is $\log_2 N$. ▮

Theorem 5.4.2 *The necessary and sufficient number of wavelengths for an $N \times N$ $B(\log_2 N - 1, 0)$ Benes network to be 0-CSE and w-wavelength strict-sense nonblocking is $w = 2 \times \log_2 N - 1$.*

Proof Since $c = 0$, no crosstalk is allowed on any node. Thus, on each wavelength, only one light signal is allowed to pass through an SE. This corresponds to the node-disjoint case. The proof is similar to the proof of Theorem 5.4.1, except that in node-disjoint case, each path in S_i (T_i) can block up to $2^{\log_2 N-i}$ of the $N/2$ paths from s to t. Thus, the total number of node blocking is

$$\sum_{i=1}^{\log_2 N-1} 2^{\log_2 N-i} \times 2^i = (\log_2 N - 1) \times N$$

Since there are $N/2$ paths between s and t, the average number of node blocks on each path is at most

$$\frac{(\log_2 N - 1) \times N}{N/2} = 2 \times (\log_2 N - 1).$$

Therefore, $2 \times \log_2 N - 1$ wavelengths are enough to ensure node-disjoint.

Now we prove that $2 \times \log_2 N - 1$ is also the lower bound. Suppose $w = 2 \times \log_2 N - 2$ is a lower bound. Let connection requests from S_i to $T_{\log_2 N}$ use wavelength w_{2i-1} and connection requests from $S_{\log_2 N}$ to T_i use wavelength w_{2i}, for all $1 \leq i < \log_2 N$. Again we are not able to route the connection request from s to t, since all $2 \times \log_2 N - 2$ wavelengths have been used. Thus, the lower bound is $2 \times \log_2 N - 1$. ▮

Lemma 5.4.3 *In an $N \times N$ $B(\log_2 N - 1, c)$ Benes network, given a set of routing requests I with at most N pairs, for any $(s,t) \in I$, let S_i and T_i be the intersecting sets with respect to (s,t). Connections originating from S_i and connections destined to T_i can intersect with all $N/2$ of the paths from s to t. In addition, each of these paths can have up to $2 \times (i-1)$ CSEs.*

Proof From Theorems 5.4.1 and 5.4.2, it is easy to see that connections originating from S_i and connections destined to T_i can intersect all $N/2$ paths from s to t. Let connections originating from S_i use the same wavelength. It can be seen from Figure 5.3(d) that the 2^{i-1} paths will intersect $i-1$ times until they reach stage i. If we choose the 2^{i-1} destinations such that they are continuous and each output SE is used by two connection requests, these paths will intersect another $i-1$ times after stage $2 \times \log_2 N - (i-1)$ until they reach the last stage. Thus, they can intersect up to $2 \times (i-1)$ times. ▮

Theorem 5.4.4 *A lower bound on the number of wavelengths for an $N \times N$ $B(\log_2 N - 1, c)$ $(0 < c < 2 \times \log_2 N - 1)$ Benes network to be c-CSE and w-wavelength strict-sense nonblocking is $2 \times \log_2 N - 1 - \lceil \frac{c}{2} \rceil$.*

Proof From Theorem 5.4.1 and 5.4.2, we can see that S_i and T_i combined can block N tagged paths (2 wavelengths) in the node-disjoint case, and can block $N/2$ tagged paths (1 wavelength) in the edge-disjoint case. From Lemma 5.4.3, we know that S_i and T_i can still block 2 wavelengths under a crosstalk level c, for $\lceil \frac{c}{2} \rceil < i \leq \log_2 N - 1$. S_i and T_i combined can still block one wavelength no matter what value c is. Thus, the total number of link and crosstalk blocks is

$$2 \times (\log_2 N - 1 - \lceil \frac{c}{2} \rceil) + \lceil \frac{c}{2} \rceil = 2 \times \log_2 N - \lceil \frac{c}{2} \rceil - 2$$

Thus, the number of wavelength needed is at least $2 \times \log_2 N - 1 - \lceil \frac{c}{2} \rceil$. ▮

Unfortunately, we have not found a tight upper bound for the $B(\log_2 N - 1, c)$ networks. There are too many cases. A trivial upper bound is given in Theorem 5.4.2, which corresponds to the node-disjoint case $(c = 0)$.

5.5 Networks with extra stages

Lemma 5.5.1 *In an $N \times N$ $B(x,c)$ Banyan-type network, given a set I of routing requests with at most N pairs, for any $(s,t) \in I$, let S_i and T_i be the intersecting sets with respect* *to (s,t). Connections originating from S_i and connections destined to T_i $(1 \leq i \leq x+1)$ can intersect with all 2^x of the paths from s to t. In addition, paths originating from S_i or destined to T_i can have up to $2 \times (i-1)$ CSEs along the route. Connections originating from S_i and connections destined to T_i can intersect with 2^i of the paths from s to t, where $x+2 \leq i \leq (1/2)(\log_2 N + x)$ if $\log_2 N + x$ even, and $x+2 \leq i \leq (1/2)(\log_2 N + x + 1)$ otherwise. In addition, each 2^x paths originating from S_i or destined to T_i can have up to $2 \times x$ CSEs.*

Proof Let us consider the case where $\log_2 N + x$ is even. From Lemma 5.4.3, it is easy to see that for $1 \leq i \leq x+1$, connections originating from S_i and connections destined to T_i can intersect all 2^x paths from s to t. Suppose connections originating from S_i use the same wavelength. It can be seen from Figure 5.3(c) that the 2^{i-1} paths will intersect $i-1$ times until they reach stage i. If we choose the 2^{i-1} destinations such that they are continuous and each output SE is used by two connection requests, these paths will intersect another $i-1$ times after stage $\log_2 N + x - (i-1)$ until they reach the last stage. Thus, they can intersect up to $2 \times (i-1)$ times. For $x+2 \leq i \leq (1/2)(\log_2 N + x)$, we have 2^{i-1} elements in S_i or T_i and 2^x connections originating from S_i or connections destined to T_i are enough to intersect the 2^x paths from s to t. Thus we can have up to $2 \times x$ CSEs along each of these 2^xpaths. The proof for other cases is similar. ▮

The following proposition can be easily verified.

Proposition 5.5.2 *A lower bound on the number of wavelengths for a $B(x+1,c)$ network to be c-CSE and w-wavelength strict-sense nonblocking is also a lower bound for $B(x,c)$ network.*

In the following discussion, we use $w_{B(x,c)}$ to denote the lower bound on the number of wavelengths for a $B(x,c)$ network to be c-CSE and w-wavelength strict-sense nonblocking.

Theorem 5.5.3 *A lower bound on the number of wavelengths for an $N \times N$ $B(x,c)$ $(0 < c < \log_2 N + x)$ network with even $\log_2 N + x$ to be c-CSE and w-wavelength strict-sense nonblocking is*

$$w \geq \begin{cases} 2x + 2\sqrt{\frac{N}{2^x}} - 1 - \lceil \frac{c}{2} \rceil, & \textit{if } 0 < c \leq 2x \\ max\{w_{B(x+1,c)}, x + \frac{3}{2}\sqrt{\frac{N}{2^x}} - 1\}, & \textit{if } 2x < c < \log_2 N + x. \end{cases}$$

When $\log_2 N + x$ *is odd, the result becomes*

$$w \geq \begin{cases} 2x + \frac{3}{2}\sqrt{\frac{2N}{2^x}} - 1 - \lceil \frac{c}{2} \rceil, & \textit{if } 0 < c \leq 2x \\ max\{w_{B(x+1,c)}, x + \sqrt{\frac{2N}{2^x}} - 1 + \lfloor \frac{2^{(\log_2 N - x - 1)/2}}{c - 2x + 2} \rfloor\}, & \textit{if } 2x < c < \log_2 N + x. \end{cases}$$

Proof From Theorem 5.4.4 and Lemma 5.5.1, we can see that for $1 \leq i \leq x+1$, S_i and T_i combined can block 2^{x+1} paths (2 wavelengths) in the node-disjoint case, and can block 2^x paths (1 wavelength) in the edge-disjoint case. From Lemma 5.5.1, we know that S_i and T_i can still block 2 wavelengths under a crosstalk level c, for $\lceil \frac{c}{2} \rceil < i \leq x+1$. S_i and T_i combined can still block one wavelength no matter what value c is. Thus, the total number of blocks will reduce by $\lceil \frac{c}{2} \rceil$ when $0 < c \leq 2x$. An obvious lower bound for $2x < c < \log_2 N + x$ is equal to the lower bound of the edge-disjoint case. For odd $\log_2 N + x$, if we include the elements in $S_{(\log_2 N+x+1)/2}$ ($T_{(\log_2 N+x+1)/2}$), we can get a tighter low bound. We noticed that in the edge-disjoint case, the connections originating from $S_{(\log_2 N+x-1)/2}$ should be destined to $T_{(\log_2 N+x+1)/2}$ to create maximum blocking. Similarly, connections destined to $T_{(\log_2 N+x-1)/2}$ should originate from $S_{(\log_2 N+x+1)/2}$. Thus we have $2^{(\log_2 N+x-1)/2}$ extra elements in $S_{(\log_2 N+x+1)/2}$ and $T_{(\log_2 N+x+1)/2}$, which can create additional crosstalk blocks. To block one wavelength, we need 2×2^x elements in $S_{(\log_2 N+x+1)/2}$ and $T_{(\log_2 N+x+1)/2}$ plus $c - 2x$ elements for each of the 2^x connection. So the additional blocking number is

$$\frac{2^{(\log_2 N+x-1)/2}}{2 \times 2^x + (c-2x) \times 2^x} = \frac{2^{(\log_2 N-x-1)/2}}{c - 2x + 2}$$

From Proposition 5.5.2, we know that a lower bound on the number of wavelengths for a $B(x+1, c)$ network to be strict-sense nonblocking is also a lower bound for a $B(x, c)$ network. So we choose the maximum of the calculated value and $w_{B(x+1,c)}$. The proof for even $\log_2 N + x$ is similar. ▮

5.6 Result comparison

Table 5.1 gives lower bounds on the number of wavelengths for Benes networks with various sizes and crosstalk levels to be c-CSE and w-wavelength strict-sense nonblocking. In the table, $c = 0$ is the node-disjoint case and $c = 2 \times \log_2 N - 1$ is the edge-disjoint case. The number of wavelengths reduces by 1 when c decreases by 2. This can be explained by Theorem 5.4.4: a lower bound is $2 \times \log_2 N - 1 - \lceil \frac{c}{2} \rceil$. Table 5.2 gives lower bounds for a 64×64 $B(x, c)$ network to be c-CSE and w-wavelength strict-sense nonblocking. There are still gaps between the necessary and sufficient conditions. We conjecture that the number for the sufficient condition can be reduced.

5.7 Summary

In this chapter, we have analyzed the nonblocking properties of multistage interconnecting networks. We use wavelength dilation to reduce the crosstalk in an optical network. Low bounds and upper bounds for strict-sense nonblocking are given for several well-known network topologies. There are still gaps between the lower and upper bounds for some networks. It is worth investigating whether these bounds can be made tighter. It is also a challenging problem to find the number of wavelengths needed for wide-sense nonblocking on these networks.

Table 5.1: Lower bounds on the number of wavelengths for $B(\log_2 N - 1, c)$ networks

	N								
c	4	8	16	32	64	128	256	512	1024
0	3	5	7	9	11	13	15	17	19
1,2	2	4	6	8	10	12	14	16	18
3,4		3	5	7	9	11	13	15	17
5,6			4	6	8	10	12	14	16
7,8				5	7	9	11	13	15
9,10					6	8	10	12	14
11,12						7	9	11	13
13,14							8	10	12
15,16								9	11
17,18									10

Table 5.2: Lower bounds on the number of wavelengths for 64 × 64 $B(x, c)$ networks

	x					
c	0	1	2	3	4	5
0	15	13	11	11	11	11
1	15	12	10	10	10	10
2	13	12	10	10	10	10
3	13	9	9	9	9	9
4	11	9	9	9	9	9
5	11	8	8	8	8	8
6	11	8	8	8	8	8
7		8	7	7	7	7
8			7	7	7	7
9				6	6	6
10					6	6
11						6

Chapter 6

Conclusion and Future work

WDM all-optical networks can meet the ever-increasing demand for network bandwidth. In this thesis, we studied the WA problem on the tree of rings, a popular topology often found in WDM networks, and the crosstalk reduction problem on optical switches. In this chapter, we will summarize the contributions of this work, and give a few directions for future work.

6.1 Summary of contributions

We have studied the WA problem on a tree of rings. We have shown that the greedy algorithm we proposed uses at most $3L$ wavelengths for trees of rings with maximum node degree 8, where L is the maximum edge load. This improves the previous result of $4L$ [15]. We have also shown that there are instances which require at least $3L$ wavelengths for the WA problem on a tree of rings network. Our algorithm uses at most $4L$ wavelengths for node degree greater than 8.

We have shown that our algorithm achieves approximation ratios of $2\frac{1}{16}$, $2\frac{3}{13}$ and 3 on trees of rings with node degrees at most 4, 6 and 8, respectively. For node degree at most 4, a 2-approximation algorithm has been known [13]. However, this algorithm is much more complex than ours and only works for node degree 4, which allows only two rings intersect at the same node and is probably not enough in practice. Our algorithm works for trees of rings of arbitrary node degree and achieves a good approximation ratio for node degree at most 8, which the author thinks can probably meet most engineering requirements.

Crosstalk can be a severe problem in directional-coupler-based optical switches. In this thesis, we studied the crosstalk reduction problem and showed several lower and upper bounds for Banyan type networks.

6.2 Future work

Many research efforts have been devoted to the RWA and WA problems. However, there are still many open problems. In this section, we give a few directions for the future work.

From Chapter 3, we know that for the WA problem on trees, there is a difference between the $5L/3$ upper bound and the $5L/4$ lower bound. Although greedy algorithms cannot achieve an upper bound better than $5L/3$, it is not known whether other non-greedy algorithms can achieve a better upper bound. It is worth investigating whether the gap can be narrowed.

For the WA problem on trees of rings, we intend to focus on the following several directions in the future:

- We have shown that $3L$ is enough for the WA problem on trees of rings with node degree at most 8, and $3L$ is also necessary. Obviously at least $3L$ is needed for arbitrary node degree. Erlebach [15] has shown that $4L$ is an upper bound. One natural question is, whether $3L$ is enough for trees of rings of arbitrary node degree? We conjecture that $3L$ is also the upper bound for the WA problem on trees of rings with arbitrary node degree, but we do not have a proof yet. Note that although we can only prove that our algorithm achieves the $3L$ upper bound for trees of rings with node degree at most 8, our algorithm works for general trees of rings. In the future, we will evaluate the performance of our algorithm on general trees of rings, and try to find a new non-trivial upper bound which is strictly less than $4L$ (and hopefully equal to $3L$). If we cannot find a better upper bound, then whether it is possible to find a better lower bound which is strictly greater than $3L$? If we cannot find a better lower bound or upper bound (for general algorithms), is it possible to use an adversary argument to prove a better lower bound for the greedy algorithms?

- To the author's best knowledge, there is no published result on the upper bound in terms of ω for trees of rings with arbitrary node degree. Combine the $4L$ result with the fact that $L \leq \omega$, we can get a trivial 4ω upper bound. Although we have several results in terms of ω for trees of rings with bounded node degrees, no better result is known for general trees of rings. In the future, we will try to find a better upper bound in terms of ω for general case.

- In this thesis, we have focused on the undirected trees of rings. However, the optical networks are directed since optical amplifiers placed on the fiber are directed devices. In the future, we intend to extend this algorithm and analysis to the directed model. It is not hard to see that the algorithm can be extended to the directed model without any modifications, but we need to evaluate the performance of the algorithm on directed trees of rings.

- It is also worth investigating the online RWA and WA problems on trees of rings. As introduced in Chapter 3, Bartal *et. al.* [2] showed that there is an $O(\log N)$-competitive algorithm for the online RWA problem (path not fixed) on a tree of rings with N nodes. Since a low bound for the online RWA problem on rings is $O(L \times \log N)$, the lower bound for online RWA problem on trees of rings is also of the order $O(L \times \log N)$. It is worthy finding out the exact coefficients behind these numbers. No result is available for the online WA problem (i.e., path fixed) on trees of rings. This is another future research direction.

For the crosstalk reduction problem on directional-coupler-based Banyan-type networks, we have lower bounds on the number of wavelengths needed for the network to be strict-sense nonblocking. However, we still do not have non-trivial upper bounds for Banyan-type networks with arbitrary extra stages. Further efforts will be devoted to this problem.

Bibliography

[1] R. Sai Anand and Thomas Erlebach. On-line algorithms for edge-disjoint paths in trees of rings. In *Proceedings of the 5th Latin American Symposium on Theoretical Informatics (LATIN'02)*, pages 584–597, Cancun, Mexico, 2002.

[2] Yair Bartal and Stefano Leonardi. Online routing in all-optical networks. In *Proceedings of 24th International Colloquium on Automata, Languages and Programming (ICALP'97)*, pages 516–526, 1997.

[3] B. Beauquier, J-C. Bermond, L. Gargano, P. Hell, S. Perennes, and U. Vaccaro. Graph problems arising from wavelength-routing in all-optical networks. *The 2nd Workshop on Optics and Computer Science (WOCS'97).*

[4] B. Beauquier, S. Perennes, and D. Toth. All-to-all routing and coloring in weighted trees of rings. In *Proceedings of the 11th Annual ACM Symposium on Parallel Algorithms and Architectures SPAA'99*, pages 185–190, 1999.

[5] Bruno Beauquier. All-to-all communication for some wavelength-routed all-optical networks. *Networks*, 33(3):179–187, 1999.

[6] V. E. Benes. *Mathematical Theory of Connecting Networks and Telephone Traffic.* New York: Academic, 1965.

[7] Claude Berge. *Graphs.* North Holland, 1985.

[8] Jean-Claude Bermond, Luisa Gargano, Stephan Perennes, Adele A. Rescigno, and Ugo Vaccaro. Efficient collective communication in optical networks. *Theoretical Computer Science*, 233:165–189, 2000.

[9] Allan Borodin and Ran El-Yaniv. *Online Computation and Competitive Analysis.* Cambridge University Press, 1998.

[10] D.G. Cantor. On non-blocking switching networks. *Networks*, 1:367–377, 1972.

[11] I. Caragiannis, C. Kaklamanis, and P. Persiano. Wavelength routing in optical tree networks: A survey. Bulletin of the European Association for Theoretical Computer Science, 2001.

[12] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms.* MIT Press, Cambridge, Mass., 1990.

[13] X. Deng, G. Li, W. Zang, and Y. Zhou. A 2-approximation algorithm for path coloring on trees of rings. In *Proceedings of the 11th Annual International Symposium on Algorithms and Computation ISAAC 2000, LNCS 1969*, pages 144–155, 2000.

[14] J. Duato, S. Yalamanchili, and L. Ni. *Interconnection Networks: An Engineering Approach.* IEEE Press, 1997.

[15] T. Erlebach. Approximation algorithms and complexity results for path problems in trees of rings. In *Proceedings of the 26th International Symposium on Mathematical Foundations of Computer Science (MFCS 2001), LNCS 2136*, pages 351–362. Springer-Verlag, 2001.

[16] Thomas Erlebach. Counterexample to an algorithm on trees of rings. Private communications, 2003.

[17] Thomas Erlebach and Klaus Jansen. Scheduling of virtual connections in fast networks. In *Proceedings of 4th Workshop on Parallel Systems and Algorithms (PASA '96)*, pages 13–32, 1996.

[18] Thomas Erlebach and Klaus Jansen. Call scheduling in trees, rings and meshes. In *Proceedings of 30th Hawaii International Conference on System Sciences*, pages 221–222, 1997.

[19] Thomas Erlebach and Klaus Jansen. The maximum edge-disjoint paths problem in bidirected trees. *SIAM J. DISCRETE MATH.*, 14(3):326–355, 2001.

[20] Andras Frank, Takao Nishizeki, Nobuji Saito, Hitoshi Suzuki, and Eva Tardos. Algorithms for routing around a rectangle. *Discrete Applied Mathematics*, 40:363–378, 1992.

[21] M. R. Garey, D. S. Johnson, G. L. Miller, and C. H. Papadimitriou. The complexity of coloring circular arcs and chords. *SIAM Journal on Algebraic and Discrete Methods*, 1:216–227, 1980.

[22] Luisa Gargano, Pavol Hell, and Stephane Perennes. Coloring all directed paths in a symmetric tree, with an application to optical networks. *Journal of Graph Theory*, 4:183–196, 2001.

[23] Ori Gerstel and Shay Kutten. Dynamic wavelength allocation in all-optical ring networks. In *Proceedings of IEEE Int. Conf. Communications*, pages 432–436, 1997.

[24] Ori Gerstel, Galen Sasaki, and Rajiv Ramaswami. Dynamic channel assignment for wdm optical networks with little or no wavelength conversion. In *Proceedings of Allerton 1996, Monticello*, pages 32–43, 1996.

[25] J.E. Hopcroft and R.M. Karp. An $n^{2.5}$ algorithm for maximum matchings in bipartite graphs. *SIAM Journal on Computing*, 2(4):225–231, Dec. 1973.

[26] C. Kaklamanis, P. Persiano, T. Erlebach, and K. Jansen. Constrained bipartite edge coloring with applications to wavelength routing. In *Proceedings of 24th International Colloquium on Automata, Languages and Programming (ICALP'97)*, pages 493–504, 1997.

[27] I.A. Karapetian. On coloring of arc graphs. *Dokladi of the Academy of Science of the Armenian SSR*, 70:306–311, 1980.

[28] Vijay Kumar and Eric J. Schwabe. Improved access to optical bandwidth in trees. In *Proceedings of 8th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '97)*, pages 437–444, 1997.

[29] M. Mihail, C. Kaklamanis, and S. Rao. Efficient access to optical bandwidth. In *Proceedings of the 36th Annual Symposium on Foundations of Computer Science FOCS'95*, pages 548–557, 1995.

[30] C. Siva Ram Murthy and Mohan Gurusamy. *WDM optical networks, concepts, design, and algorithms.* Prentice Hall, Upper Saddle River, N.J., 2002.

[31] M. Paterson, H. Schroeder, O. Sykora, and I. Vrto. On permutation communications in all-optical rings. In *Proceedings of 5th International Colloquium on Structural Information and Communication Complexity (SIROCCO'98)*, pages 1–9, 1998.

[32] Qian ping Gu. Online permutation routing on wdm all-optical networks. In *Proceedings of the 31st International Conference on Parallel Processing (ICPP)*, 2002.

[33] P. Raghavan and E. Upfal. Efficient routing in all-optical networks. In *Proceedings of the 26th Annual ACM Symposium on Theory of Computing STOC'94*, pages 134–143, 1994.

[34] Rajiv Ramaswami and Galen Sasaki. Multiwavelength optical networks with limited wavelength conversion. *IEEE/ACM Trans. Networking*, 6:744–754, 1998.

[35] Rajiv Ramaswami and Kumar N. Sivarajan. *Optical networks, a practical perspective.* Morgan Kaufmann Publishers, 2002.

[36] April Rasala and Gordon Wilfong. Strictly non-blocking wdm cross-connects. In *Proceedings of 2000 ACM/SIAM Symposium on Discrete Algorithms (SODA'00)*, pages 606–615, 2000.

[37] Thomas E. Stern and Krishna Bala. *Multiwavelength Optical Networks, A Layered Approach.* Addison Wesley, 1999.

[38] R.E. Tarjan. Decomposition by clique separators. *Discrete Mathematics*, 55(2):221–232, 1985.

[39] V. R. Chinni *et al.* Crosstalk in a lossy directional coupler switch. *J. Lightwave Technol.*, 13(7):1530–1535, July 1995.

[40] A. Tucker. Coloring a family of circular arcs. *SIAM Journal on Math*, 29(3):493–502, 1975.

[41] M. M. Vaez and C.-T. Lea. Wide-sense nonblocking banyan-type switching systems based on directional couplers. *IEEE J. Select. Areas Commun.*, 16:1327–1332, 1998.

[42] M. M. Vaez and C.-T. Lea. Strictly nonblocking directional-coupler-based switching networks under crosstalk constraint. *IEEE Trans. on Commun.*, 48(2), Feb. 2000.

[43] Douglas Brent. West. *Introduction to graph theory.* Prentice Hall, Upper Saddle River, N.J., 2nd edition, 2001.

[44] Gordon Wilfong and Peter Winkler. Ring routing and wavelength translation. In *Proceedings of 9th Annual ACM-SIAM Symp. on Discrete Algorithms*, pages 333–341, 1998.

[45] Bohdan Zelinka. Medians and peripherians of trees. *Arch. Math. (Brno)*, 4:87–95, 1968.