Georgia State University
ScholarWorks @ Georgia State University

Computer Science Dissertations | Department of Computer Science

1-12-2006

Upper Bound Analysis and Routing in Optical Benes Networks

Jiling Zhong

Follow this and additional works at: http://scholarworks.gsu.edu/cs_diss

Recommended Citation
Zhong, Jiling, "Upper Bound Analysis and Routing in Optical Benes Networks." Dissertation, Georgia State University, 2006.
http://scholarworks.gsu.edu/cs_diss/4

This Dissertation is brought to you for free and open access by the Department of Computer Science at ScholarWorks @ Georgia State University. It has been accepted for inclusion in Computer Science Dissertations by an authorized administrator of ScholarWorks @ Georgia State University. For more information, please contact scholarworks@gsu.edu.

UPPER BOUND ANALYSIS AND ROUTING IN OPTICAL BENES NETWORKS

by

JILING ZHONG

Under the Direction of Yi Pan

ABSTRACT

Multistage Interconnection Networks (MIN) are popular in switching and communication applications. It has been used in telecommunication and parallel computing systems for many years. The new challenge facing optical MIN is crosstalk, which is caused by coupling two signals within a switching element. Crosstalk is not too big an issue in the Electrical Domain, but due to the stringent Bit Error Rate (BER) constraint, it is a big major concern in the Optical Domain. In this research dissertation, we will study the blocking probability in the optical network and we will study the deterministic conditions for strictly non-blocking Vertical Stacked Optical Benes Networks (VSOBN) with and without worst-case scenarios. We will establish the upper bound on blocking probability of Vertical Stacked Optical Benes Networks with respect to the number of planes used when the non-blocking requirement is not met.

We will then study routing in WDM Benes networks and propose a new routing algorithm so that the number of wavelengths can be reduced. Since routing in WDM optical network is an NP-hard problem, many heuristic algorithms are designed by many researchers to perform this

routing. We will also develop a genetic algorithm, simulated annealing algorithm and ant colony technique and apply these AI algorithms to route the connections in WDM Benes network.

INDEX WORDS: Benes Network, Mins, Genetic Algorithms, Simulated Annealing, Ant Colony Algorithms

UPPER BOUND ANALYSIS AND ROUTING IN OPTICAL BENES NETWORKS

by

JILING ZHONG

A Dissertation Submitted in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

in the College of Arts and Sciences

Georgia State University

2005

Copyright by
Jiling Zhong
2005

UPPER BOUND ANALYSIS AND ROUTING IN OPTICAL BENES NETWORKS

by

JILING ZHONG

Major Professor: Yi Pan
Committee: Alex Zelikovsky
Anu Bourgeois
Upkar Varshney

Electronic Version Approved:

Office of Graduate Studies
College of Arts and Sciences
Georgia State University
December 2005

ACKNOWLEDGEMENTS

I am deeply indebted to my advisor, Professor Yi Pan, for his constant support and guidance. During my years of study, from time to time, I may be bewildered and tired but I have never doubted that Dr. Pan was there to consult, to clear my thoughts, and to seek strength and direction from. I can never forget those hours after hours Dr. Pan has spent with me in discussing the problems, finding the solutions…. In him, I have seen not only a prominent researcher, but also a person with great determination, compassion and vision, a true scholar.

I must also thank my committee members, Dr. Alex Zelikovsky, Dr. Anu Bourgeois and Dr. Upkar Varshney. Their advice and patience have proved invaluable. During the course of this work, Dr. Zelikovsky and Dr. Bourgeois have given me many enlightening comments and made this work more mathematically rigorous. Dr. Varshney has reminded me of some very interesting future research topics; his keen observation has paved way to even brighter future for this work. Special thanks also go to Dr. Bourgeois for her proofreading of the draft and thanks to her this final version is more sound and readable.

I would also like to thank my family and friends for their support and beliefs. There must have been those whom I have sought help from without asking for name. Though I may never know their names, their help are humbly appreciated.

Without the support from all these people, this work will never be possible. Thank you.

TABLE OF CONTENTS

Acknowledges iv

List of Figures v

List of Tables x

Chapter 1. INTRODUCTION 1

1.1 Motivation ……. 1

1.2 Research Contribution …… 3

Chapter 2. BACKGROUND 5

2.1. Optical Multistage Interconnection Networks ……... 5

2.2. Multistage Interconnection Networks (MINs) …….. 7

2.3. A generalized MIN model ……. 8

2.4. MIN Switches ……10

2.4.1 Banyan Network ……. 10

2.4.2 Strictly Non-blocking Network ……. 14

2.4.3 Wide Sense Non-blocking Network …… 15

2.4.4 Rearrangeably Non-blocking Network …… 16

2.5. Problems in OMIN …… 18

2.5.1. Bottlenecks at the switch element …… 18

2.5.2. Crosstalk in OMIN ……. 18

2.6. Approaches to avoid crosstalk ……. 19

Chapter 3. PREVIOUS RELATED WORK 23

3.1 Analysis of Blocking Probability of Stacked Banyan Networks …….. 24

3.2 Semi-permutation in Benes Network …… 25

3.3 Optical Window in Omega Network …… 27

3.4 Conflict Graph and Graph Coloring ……. 29

3.5 Parallel Routing in Benes Network …….. 29

3.6 Parallel Routing in Benes Network without Crosstalk ……. 31

Chapter 4. UPPER BOUND ON BLOCKING PROBABILITY OF VSOBN 33

4.1 Strictly Nonblocking Without Worst Case Scenarios …… 33

4.2 Upper Bound on Blocking Probability …….. 41

Chapter 5. WDM ROUTING IN BENES NETWORK 52

5.1. Route construction …… 52

5.1.1 Motivation …… 53

5.2. Wavelength Awareness Routing …… 57

5.3 Wavelength Assignment …… 61

Chapter 6. GENETIC ALGORITHMS 68

6.1. Introduction to Genetic Algorithm …… 68

6.2 Operators of Genetic Algorithm …… 70

6.3 Parameter of Genetic Algorithm …… 71

6.4 A modified GAs …… 72

6.4.1 Take Relative Fitness Information into Account …… 74

6.4.2 Take Fitness Distribution Information into Account ……. 75

6.5 Genetic Algorithms in This Thesis …… 75

6.5.1 Cross Over Rate ……. 76

6.5.2 Initial Population Size …… 76

6.5.3 Number of Generations …….. 76

6.5.4 Objective Function …….. 76

6.5.5 Mutation Rate …… 77

6.6 Results and Discussion …….. 77

Chapter 7 SIMULATED ANNEALING 86

7.1 Inversion …… 88

7.2 Translation …… 88

7.3 Switching …….. 89

7.4 The Experiment and Analysis of The Test Result …… 89

Chapter 8. ANT COLONY 105

8.1 Introduction to Ant Colony …….. 105

8.2 Application of Ant Colony Optimization to Benes Network Routing …… 107

8.3 Experiment and Analysis of the Results …… 108

8.4 A Comparison among GA, SA and ACO ……. 111

Chapter 9. CONCLUSION AND FUTURE WORK 115

BIBLIOGRAPHY 119

List of Figures

Figure 2.1. A generalized MIN with *N* inputs, *M* outputs, and *g* stages …… 8

Figure 2.2. A closer view of one of the stages …… 9

Figure 2.3. Two SE states …… 11

Figure 2.4. Non-traditional cross bar ……... 11

Figure 2.5. A recursive construction of baseline network …… . 12

Figure 2.6 Self-routing in Banyan network …… 13

Figure 2.7. Blocking in Banyan network …… 14

Figure 2.8. 3-stage Clos network …… 15

Figure 2.9. A 4 x 4 Crossbar …… 16

Figure 2.10. By appending extra stages, alternative routes are available …… 17

Figure 2.11. Cross talk in a switch element …… 19

Figure 2.12. WDM ……. 20

Figure 2.13. TDM ……. 21

Figure 2.14. A hybrid of WDM and TDM ……. 22

Figure 2.15. Legal passing connections in a SE at a time …… 22

Figure 3.1. A Stacked Banyan Networks with Two Planes …… 24

Figure 3.2 Vertically stacked Banyan network …… 25

Figure 3.3 A permutation …… 27

Figure 3.4. Source-Destination permutation …….. 28

Figure 3.5. Optical Window …….. 28

Figure 3.6. An example of conflict graph …….. 29

Figure 4.1 Different input (output) links have different blocking capabilities …… 34

Figure 4.2: Blocking requirement and actual number of planes needed for worst cases …….. 37

Figure 4.3. Savings(percentage) without 1st and 2nd worst case …….. 40

Figure 4.4. Illustration of paths for calculating α and γ ……. 49

Figure 4.5. Blocking Probability for Traffic Rate r = 0.8 and 0.1 …… 51

Figure. 5.1 Benes network exhibits symmetry respect to the center stage ……. 53

Figure 5.2. Conflict graph for a given source-destination permutation ……. 55

Figure 5.3. Two of the possible SE settings for a permutation and their conflict graphs ….. 56

Figure 5.4. Figure 5.4. C_1 is conflicting with C_2 …….57

Figure 5.5. C_3 is chosen to be conflicting with C_1 …… 57

Figure 5.6. C_4 is chosen to be conflicting with C_2 …… 57

Figure 5.7. C_4 is chosen to be conflicting with C_3 instead of C_1 …… 58

Figure 5.8. Greedy algorithms is not optimal …… 58

Figure 5.9. A permutation …….. 61

Figure 5.10. The network setting for the permutation in Figure 5.9 ……. 62

Figure 5.11 Step 1 in the first part …… 63

Figure 5.12 Address change in Baseline network …… 64

Figure 5.13 Step 2 in the first part ……... 64

Figure 5.14 Step 3 in the first part …… 65

Figure 5.15 Step 1 in Second Part …… 65

Figure 5.16 Step 2 in Second Part …… 66

Figure 5.17 Step 3 in Second Part …… 66

Figure 5.18 Conflict graph – result of the modified windows method ……... 67

Figure 6.1 Genetic Algorithms …… 69

Figure 6.2 Adjust the selective pressure according to relative fitness …… 74

Figure 6.3 Adjust the selective pressure according to chromosome distribution ……... 75

Figure 6.4. A permutation …….. 77

Figure 6.5 The graph coloring of the above permutation in [62] ……. 78

Figure 6.6 The graph coloring of the above permutation using GAs ……. 78

Figure 6.7 number of wavelength required in 8 x 8 Benes network, cross over rate = 0.7 …. 79

Figure 6.8 number of wavelength required in 8 x 8 Benes network, cross over rate = 1 …….. 80

Figure 6.9 No. of wavelength required in 16 x 16 Benes network, cross over rate = 0.7 ……. 81

Figure 6.10 No. of wavelength required in 16 x 16 Benes network, cross over rate = 1 ……. 82

Figure 6.11 No. of wavelength required in 32 x 32 Benes network, cross over rate = 0.7 … 83

Figure 6.12 No. of wavelength required in 32 x 32 Benes network, cross over rate = 1 …….. 84

Figure 6.13 average saving of No. of wavelength over the original algorithm ……..… 85

Figure 7.1 A SA flow chart …… 88

Figure 7.2 A permutation …… 89

Figure 7.3 The graph coloring of the above permutation in [62] …… 90

Figure 7.4 The graph coloring of the above permutation using SA …….. 90

Figure 7.5 No. of wavelengths in 8 x 8 Benes network under different move set …….. 91

Figure 7.6. No. of wavelengths in 16 x 16 Benes network under different move set ……. 93

Figure 7.7. No. of wavelengths in 32 x 32 Benes network under different move set ……. 94

Figure 7.8. Number of wavelength in Benes network under SA with changing starting …… 95 temperature in 8 x 8 Benes network

Figure 7.9 Number of wavelength in Benes network under SA with changing starting ……. 96
temperature in 16 x 16 Benes network

Figure 7.10 Number of wavelength in Benes network under SA with changing …… 97
starting temperature in 32 x 32 Benes network

Figure 7.11 Number of wavelength in Benes network under SA with changing ……. 98
ending temperature in 8 x 8 Benes network

Figure 7.12 Number of wavelength in Benes network under SA with changing ……. 99
ending temperature in 16 x 16 Benes network

Figure 7.13 Number of wavelength in Benes network under SA with changing ……. 100
ending temperature in 32 x 32 Benes network

Figure 7.14. Number of wavelength in Benes network under SA with …… 102
changing cooling rate in 8 x 8 Benes network

Figure 7.15. Number of wavelength in Benes network under SA with …… 103
changing cooling rate in 16 x 16 Benes network

Figure 7.16. No. of wavelength in Benes network under SA with ……. 104
changing cooling rate in 32 x 32 Benes network

Figure 8.1 Ants are looking for food ……. 105

Figure 8.2 An obstacle is in the way …… 105

Figure 8.3 Ants randomly choose a route …… 106

Figure 8.4 Ants have found the shortest path …….. 106

Figure 8.5. The number of wavelength with respect to number of ants 8 x 8 …….. 108

Figure 8.6. The number of wavelength with respect to number of ants 16 x 16 …….. 109

Figure 8.7 The number of wavelength with respect to number of ants 32 x 32 ……... 110

Figure 8.8 Comparison among GA, SA and ACO …… 111

Figure 8.9 Comparisons of running time among GA, SA and ACO and Heuristics ……... 113

Figure 8.10 Comparisons of running time among SA and ACO and Heuristics ……... 114

List of Tables

Table 4.1 Blocking Capacity and No. of planes required …….. 37

Table 4.2 Savings (percentage) without 1st and 2nd worst case …….. 40

Table 6.1 Number of wavelength required in 8 x 8 Benes network, cross over rate = 0.7 … 79

Table 6.2 Number of wavelength required in 8 x 8 Benes network, cross over rate = 1 ….. 79

Table 6.3 Number of wavelength required in 16 x 16 Benes network, cross over rate = 0.7 … 80

Table 6.4 Number of wavelength required in 16 x 16 Benes network, cross over rate = 1 …. 81

Table 6.5 Number of wavelength required in 32 x 32 Benes network, cross over rate = 0.7 … 82

Table 6.6 Number of wavelength required in 32 x 32 Benes network, cross over rate = 1 … 83

Table 7.1. No. of wavelengths in 8 x 8 Benes network under different move set …… 91

Table 7.2. No. of wavelengths in 16 x 16 Benes network under different move set …… 92

Table 7.3. No. of wavelengths in 32 x 32 Benes network under different move set …… 93

Table 7.4. Number of wavelength in Benes network under SA with changing …… 94
starting temperature in 8 x 8 Benes network

Table 7.5. Number of wavelength in Benes network under SA with changing ……..… 95
starting temperature in 16 x 16 Benes network

Table 7.6. Number of wavelength in Benes network under SA with changing ……. 96
starting temperature in 32 x 32 Benes network

Table 7.7. Number of wavelength in Benes network under SA with changing …… 98
ending temperature in 8 x 8 Benes network

Table 7.8. Number of wavelength in Benes network under SA with changing …… 99
ending temperature in 16 x 16 Benes network

Table 7.9. Number of wavelength in Benes network under SA with changing ……... 100

ending temperature in 32 x 32 Benes network

Table7.10. Number of wavelength in Benes network under SA with changing ……. 101
cooling rate in 8 x 8 Benes network

Table 7.11. Number of wavelength in Benes network under SA with changing ……. 102
cooling rate in 16 x 16 Benes network

Table 7.12. Number of wavelength in Benes network under SA with changing ……. 103
cooling rate in 32 x 32 Benes network

Table 8.1. The number of wavelength needed with respect to different number …...…… 108
of ants in 16 x 16 Benes network, with cycle = 100

Table 8.2. The number of wavelength needed with respect to different number ……. 109
of ants in 16 x 16 Benes network, with cycle = 100

Table 8.3. The number of wavelength needed with respect to different number …….. 110
of ants in 16 x 16 Benes network, with cycle = 100

Table 8.4. Comparison among GA, SA and ACO …… 111

Table 8.5 Comparison of running time among the AI techniques and the heuristic …… 112

Chapter 1

Introduction

Multistage Interconnection Network (MIN) is very popular in switching and communication applications [47]. It has been used in telecommunication and parallel computing systems for many years. This network consists of N inputs, N outputs, and n stages ($n = \log_2 N$). Each stage has N/2 Switching Elements (SEs), each SE has two inputs and two outputs connected in a certain pattern. The most widely used MINs are the electronic MINs. As optical technology advances, there is a considerable interest in using optical technology to implement interconnection networks and switches [31, 32, 41, 44]. In electronic MINs electricity is used, where as in optical MINs light is used to transmit the messages.

The electronic MINs and the optical MINs have many similarities [47], but there are some fundamental differences between them such as the optical-loss during switching [2, 17] and the crosstalk problem in the optical switches [4, 31]. To avoid the crosstalk problem, various approaches have been proposed by many researchers [1, 4, 30, 37].

1.1 Motivation

In this research, we are interested in a network called Benes Network. Benes networks are extensions to Banyan networks, which have a unique connection pattern [14, 47]. Previous results [12],[13] focus on determining the minimum number of planes required for nonblocking Vertical Stacked Optical Benes Networks (VSOBN) networks. These results

indicate that the vertical stacking scheme, although is attractive, requires a prohibitively high hardware cost for building a nonblocking VSOBN network.

Analysis of blocking probability of a network that does not meet the hardware requirement for nonblocking is an effective approach to studying network performance. In [19], blocking probability of stacked banyan networks is analyzed. In this dissertation, We shall analyze the blocking properties of VSOBN networks, and derive an upper bound of the blocking probability with respect to the number of planes employed. As it turns out, we do not need a full number of planes to guarantee non-blocking, and our analysis will show what the probability of blocking is if a smaller number of planes is used.

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 [31]. In this dissertation, we study the nonblocking properties of WDM Benes networks under crosstalk free constraints. WDM routing in Benes network consists of two parts of work, one is route construction, or the setup the SEs, the other is wavelength assignment, which is to assign different wavelength to routes so that they do not interfere with each other. Previous research and work was concentrated on wavelength assignment [6, 20]. But in our research we would like to propose a very interesting idea of wavelength aware route construction. The idea is to set up the SEs such that the wavelengths used in the second phase, the wavelength assignment phase, can be reduced.

In this research, we will use genetic algorithms, simulated annealing and ant colony algorithms to assign the wavelength. Since the wavelength is equivalent to the graph coloring problem and therefore is NP-Complete, some heuristic solution is acceptable and necessary.

1.2 Research Contribution

In summary, this dissertation has the following contributions:

1. Proposes a mathematical model to analyze the blocking probability of stacked Benes network; derive an upper bound of the blocking probability with respect to the number of planes in stacked Benes network.
2. Proposes a new algorithm in WDM Benes network routing that reduces the number of wavelengths needed; proposes modified windows method for baseline network.
3. Develops AI techniques, such as the genetic algorithm (GA), simulated annealing (SA) algorithm and the ant colony optimization (ACO) technique that are applied to the problem to calculate the number of wavelength required for routing a given permutation.

The remainder of the dissertation is organized as follows. Chapter 2 provides the required background information by briefly discussing these networks and introduces one of the important problems called crosstalk in OMIN and some approaches such as space division, time division and wavelength division multiplexing to avoid crosstalk. Chapter 3 introduces the Benes network and some relevant issues in Benes network such as crosstalk, routing and blocking. Chapter 4 provides a survey of the previous related work. Chapter 5 proposes a mathematical model to analyze the stacked Benes network and derives the upper bound of blocking probability without some worst cases. Chapter 6 will introduce wavelength aware route construction in WDM Benes network. Chapter 7 introduces a modified genetic algorithm to solve routing problem. Chapter 8 explores the use of simulated annealing to solve the routing problem. Chapter 9 explores the use of ant colony algorithms

to solve the routing problem. Chapter 10 summarizes the dissertation and provides some possible future research topics.

Chapter 2

Background

Optical switches are used in optical networks for a variety of applications. The different applications require different switching times and number of switch ports. In terms of the switching function achievable, switches are of two types: blocking or non-blocking. A switch is said to be non-blocking if an unused input port can be connected to any unused output port. If some interconnection cannot be realized, the switch is said to be blocking, to be non-blocking otherwise. There are three types of non-blocking networks, namely strictly non-blocking, wide sense non-blocking, and rearrangeably non-blocking. This research will focus on one of the rearrangeably non-blacking networks, Benes network.

2.1. Optical Multistage Interconnection Networks

Optical Multistage Interconnection Networks (OMIN) differ from Electrical Multistage Interconnection Networks; in which optical signal is converted to/from electrical signal at the network input/output, optical Multistage Interconnection Networks work in optical domain. This advantage makes the signal transmission in optical network faster.

As networks face increasing bandwidth demand and diminishing fiber availability [2, 4], network providers are moving towards a crucial milestone in network evolution: the optical network. Optical networks, based on the emergence of the optical layer in transport networks, provide higher capacity and reduced costs for new applications such as the Internet, video and multimedia interaction, and advanced digital services.

Optical networks are high-capacity telecommunications networks based on optical technologies and components that provide routing, grooming, and restoration at the wavelength

level as well as wavelength-based services. Customers are demanding more services and options and are carrying more and different types of data traffic. Optical networks provide the required bandwidth and flexibility to enable end-to-end wavelength services and meet all the high-capacity and varied needs [2, 14, 17].

Optical fiber offers much higher bandwidth than conventional copper cables. A single fiber has a potential bandwidth on the order of 50THz [8]. 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 [11]. 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 network [11]. 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 was 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, optical layer handles switching and some of the routing. 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 transferred 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 [31].

2.2. Multistage Interconnection Networks (MINs)

Electronic multistage interconnection networks (MINs) have been studied extensively as an important interconnecting scheme for communication and parallel computing systems [14, 16]. MINs connect input devices to output devices through a number of switch stages, where each switch element (SE) is a crossbar network [31]. The number of stages and the connection patterns between stages determine the routing capability of the networks.

MINs were initially proposed for telephone networks and later for array processors [14]. In these cases, a central controller establishes the path from input to output. In cases where the number of inputs equals the number of outputs, each input synchronously transmits a message to one output, and each output receives a message from exactly one input. Such

unicast communication patterns can be represented as a permutation of the input addresses [10]. On the other hand, in asynchronous multiprocessors, centralized control and permutation routing are infeasible. In this case, a routing algorithm is required to establish the path across the stages of a MIN [14].

Depending on the interconnection scheme employed between two adjacent stages and the number of stages, various MINs have been proposed [14, 16]. MINs are good for constructing parallel computers with hundreds of processors and have been used in commercial machines.

2.3. A Generalized MIN Model

There are many ways to interconnect adjacent stages [14]. Figure 1 shows a generalized multistage interconnection network with N inputs and M outputs. It has *g*

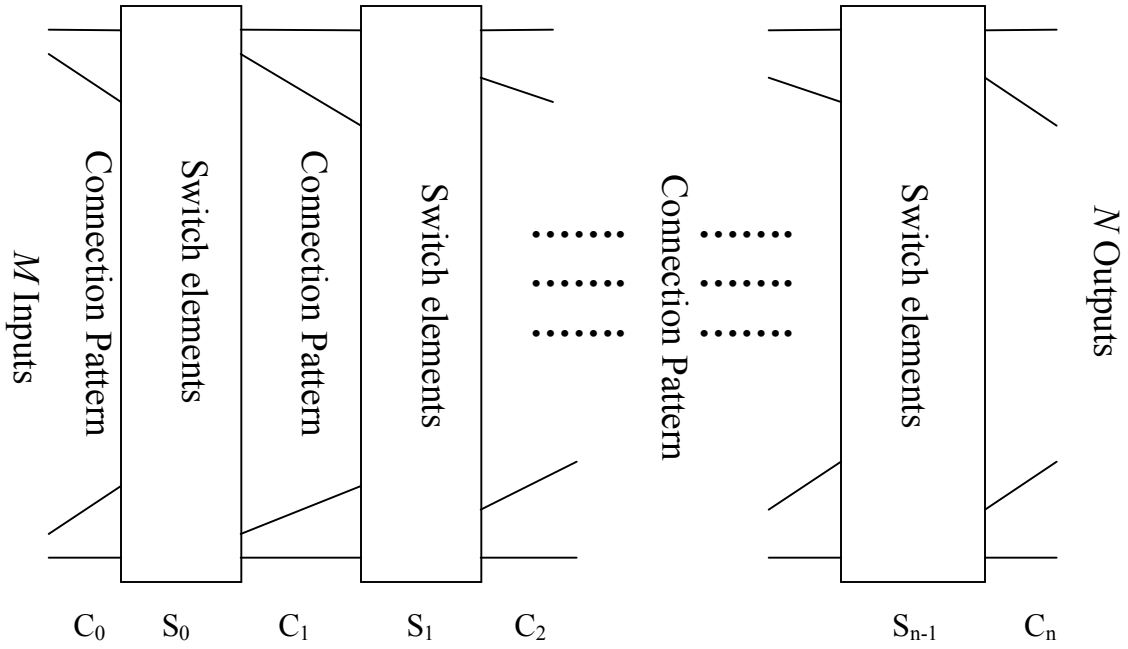

Figure 2.1. A generalized MIN with *M* inputs, *N* outputs, and *n* stages

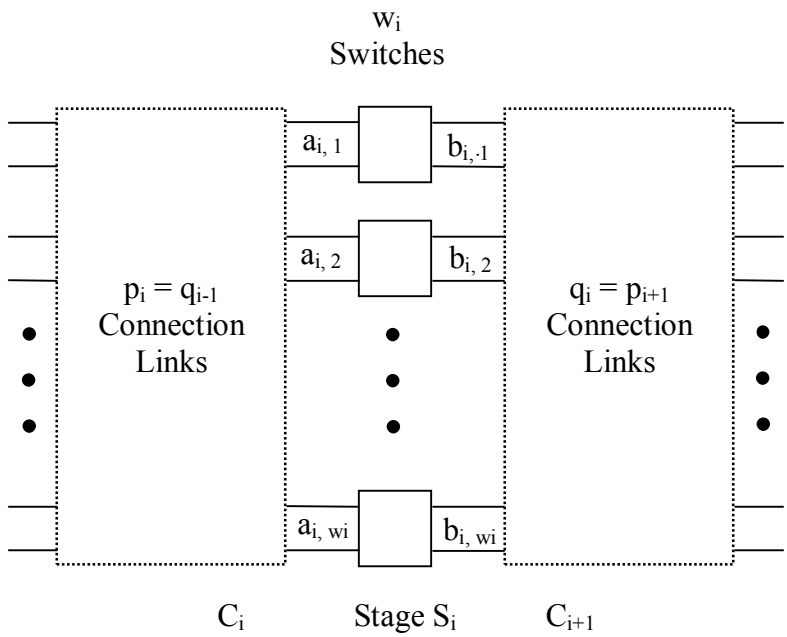

Figure 2.2. A closer view of on of the stages

stages, G_0 to G_{g-1}. As shown in Figure 2, each stage, say G_i has w_i switches of size $a_{i,j} \times b_{i,j}$, where $1 \leq j \leq w_i$. Thus, stage G_i has p_i inputs and q_i outputs, where

$$p_i = \sum_{j=1}^{w_i} a_{i,j} \text{ and } q_i = \sum_{j=1}^{w_i} b_{i,j}$$

The connection between two adjacent stages, G_{i-1} and G_i, denoted C_i, defines the connection pattern for $p_i = q_{i-1}$ links, where $p_0 = N$ and $q_{g-1} = M$. A MIN thus can be represented as

$$C_0(M)S_0(w_0)C_1(p_1)S_1 \ldots S_{n-1}(w_{n-1})C_n(N)$$

A connection pattern $C_i(p_i)$ defines those p_i links should be connected between the $q_{i-1} = p_i$ outputs from stage G_{i-1} and the p_i inputs to stage G_i. Different connection patterns give different characteristics and topological properties of MINs. The links are labeled from 0 to p_{i-1} at C_i.

From a practical point of view, it is interesting that all the switches are identical, thus amortizing the design cost. Banyan networks are a class of MINs with the property that there is a unique path between any pair of source and destination [16]. An N-node ($N=k^n$) Delta network is a sub class of banyan networks, which is constructed from identical $k \times k$ switches in n stages, where each stage contains $\frac{N}{k}$ switches. Many of the known MINs, such as Omega, flip, cube, butterfly, and baseline [14], belong to the class of Delta networks [33] and have been shown to be topologically and functionally equivalent [48]. A good survey of those MINs can be found in [14, 47].

2.4 MIN Switchers

There are two types of witches: blocking or non-blocking. A switch is said to be non-blocking if an unused input port can be connected to any unused output port. Thus a non-blocking switch is capable of realizing every interconnection pattern between the inputs and the outputs. If some interconnection pattern(s) cannot be realized, the switch is said to be blocking. One of the popular blocking networks is Banyan network.

2.4.1 Banyan Networks

Banyan networks were first introduced by Goke and Lipovski [16]. Banyan network is a multistage interconnection network (MIN), which usually consists of a number of switching elements (SEs) grouped into several stages interconnected by a set of links. Usually, a 2 x 2 crossbar can implement each Sea. The traditional cross bar has two states, namely, cross state and bar state (Figure 2.3).

Figure 2.3 Two SE states

There is another type of cross bar, which is often referred as non-traditional, where one input (output) can be directed to (form) both of the outputs (inputs), as in Figure 2.4. This type is more powerful but also is more expensive and therefore not popular. In this dissertation, we shall consider traditional cross bar only.

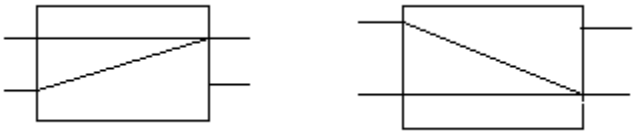

Figure 2.4 Non-traditional cross bar

The formal definition of Banyan networks is as follows,

1. It has N inputs, N outputs, $logN$ stages and $N/2$ SEs in each stage.
2. There is a unique path between each input and each output.
3. Let u and v be two SEs in stage i, and let $S_j(u)$ and $S_j(v)$ be two sets of SEs to which u and v can reach in stage j, $0<i+1=j<=n$. Then $S_j(u) \wedge S_j(v) = \varnothing$ or $S_j(u) = S_j(v)$ for any u and v.

Banyan networks are widely used as switch networks or interconnection networks due to its nice properties such as (uniformed connection pattern, self-routing, and short network diameter). There are several well-known Banyan networks, such as Omega, Shuffle

Exchange, Butterfly, and Baseline networks. It has been proved that all these networks have the same topology; therefore, they are equivalent. In this dissertation, we use Baseline network as the representative of Banyan network.

An *N x N* Baseline network, denoted by BL(*N*) is constructed recursively. A BL(2) is a 2 x 2 SE. A BL(*N*) consists of a switching stage of *N/2* SEs, and a shuffle connection, followed by a stack of two BL(*N/2*). A BL(*N*) has logN stages and each stage has *N/2* SEs, denoted as $s_0, s_1 \ldots s_{n/2-1}$. If we pair them as follows, (s_0, s_1), (s_2, s_3), …, $(s_{n/2-2}, s_{n/2-1})$, for each of such a pair, the upper outputs of both switches are linked to a switch in the upper BL(*N/2*) network, the lower outputs of both switches are linked to a switch in the lower BL(*N/2*) network, and only input switches in the pair are linked to those two switches in the next stage (one is in the upper BL(*N/2*) network, and the other is in the lower BL(*N/2*) network). Figure 2.5 shows the recursive pattern of a Baseline network.

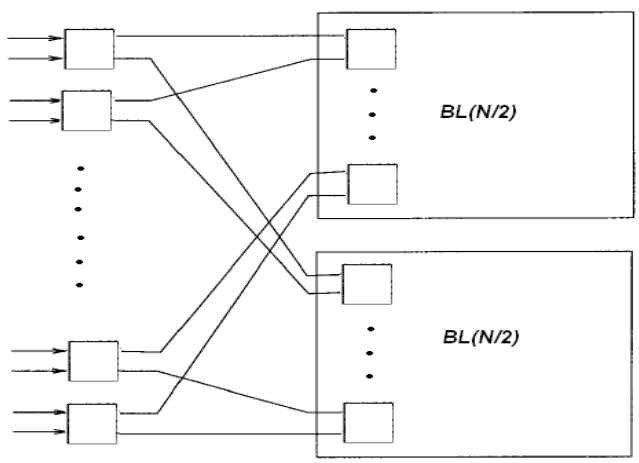

Figure 2.5. A recursive construction of baseline network

One of the properties of Banyan network is that it is self-routing. And the routing is decided by the destination. For example, a binary form of a destination is $d_{n-1}d_{n-2}\ldots d_0$. When

making a routing decision, if the (n-i)th bit of the destination equals to 0, the input of the SE on the connection path in stage i is connected to the SE's upper output link; if it equals1, the input of the SE on the connection path in stage i is connected to the SE's lower output link. As can be seen in Figure 2.6, the connection is from 001 to 111, since the destination is 111, the path will take the lower, lower, and lower output at each or the states it goes through.

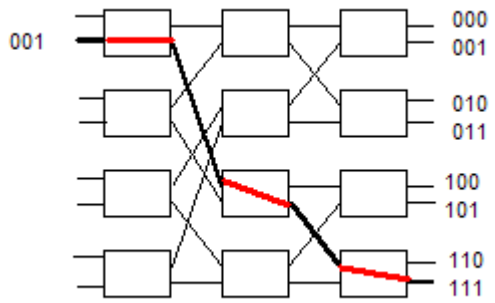

Figure 2.6 Self-routing in Banyan network

Banyan network has some very nice properties, such as short path and uniform path length. The length of the path is logN, and each path has the same length. Moreover, the number of SEs, or cross point, is NlogN. What makes Banyan network even more appealing is its self-routing property, which makes local routing decision making possible. However, Banyan network has a very serious drawback, it being blocking network. For example, in Figure 2.7, two connection, (001-111) and (011-110), both require the lower output link in the second stage, exhibiting a conflict.

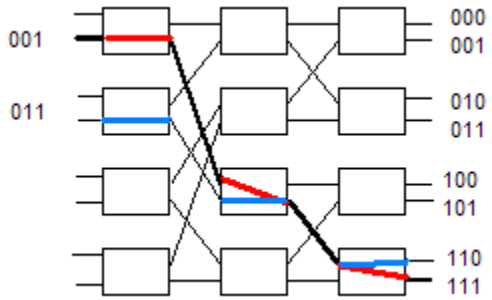

Figure 2.7. Blocking in Banyan network

Because of its being a blocking network, other non-blocking networks are usually preferred. As previous noted, there are three types of non-blocking, namely, strictly non-blocking, wide sense non-blocking, and rearrangeabely non-blocking, each of which is elaborated further.

2.4.2 Strictly Non-blocking Network

A strict non-blocking switch allows any unused input to be connected to any unused output regardless of how previous connections were made through the switch. One example of strict non-blocking network is Clos network [11], Clos networks has three stages of SEs, which can be implemented by crossbars. The first stage contains r_1 SEs, each of which has n_1 inputs and m outputs, and each of the m outputs goes to each of the m r_1 x r_2 SEs in stage 2; each of the r_2 outputs in stage 2 in turn goes to one of the r_2 m x n_2 inputs in the third stage. A Clos network is usually denoted by $C(n_1, r_1, m, n_2, r_2)$, and if $n_1=n_2=n$ and $r_1 = r_2 =r$, Clos network can be denoted as $C(n, m, r)$, as in Figure 2.8

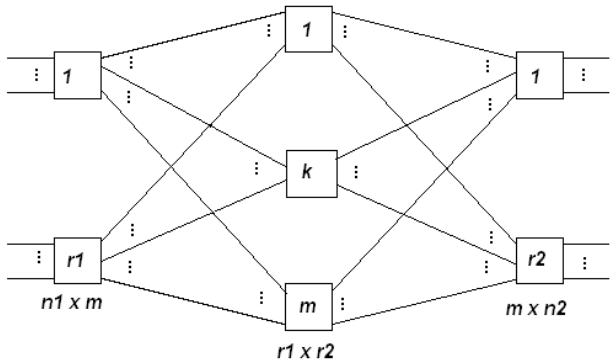

Figure 2.8. 3-stage Clos network

If $m >= 2n - 1$, then Clos network is strict non-blocking. The number of SEs used in Clos network is $O(N^{3/2})$ [11]; each path in Clos network goes through the same number of SEs, therefore the signal energy loss along the path is uniformed.

2.4.3 Wide Sense Non-blocking Network

A switch is said to be wide-sense non-blocking if any unused input can be connected to any unused output, without requiring any existing connection to be rerouted. Wide-sense non-blocking switches usually make use of specific routing algorithms to route connections so that future connections will not be blocked. An example of wide sense non-blocking network is crossbar. A crossbar consists of a matrix of $N \times N$ SEs, as in Figure 2.9, to connect input i to output j, the path taken traverses the SE in row i till it reaches column j and then traverses the switches in column j till it reaches output j. Thus the SE on this path in row i and column j must be set appropriately for this connection to be made. An $N \times N$ crossbar requires n^2 SEs. The shortest path length is 1 and the longest path length is $2N - 1$. Obviously, paths consist of deferent number of SEs and therefore the signal energy loss along different paths may be different and that is not desirable.

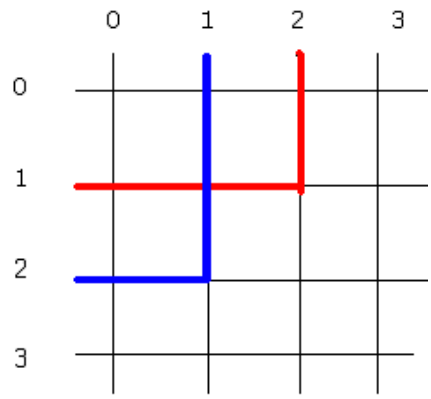

Figure 2.9. A 4 x 4 Crossbar

2.4.4 Rearrangeably Non-blocking Networks

A non-blocking switch that may require rerouting of connections to achieve the non-blocking property is said to be rearrangeably non-blocking. The Benes network is a rearrangeably nonblocking switch architecture, and is one of the most efficient switch architectures in terms of the number of 2x2 switches it uses to build larger switches.

Banyan-type networks have a single path between an input–output pair. A common design technique for creating alternate paths is to append *x* extra stages to the back of a regular Banyan-type network in which case the number of paths between an input–output pair becomes 2^x (see Fig. 2.10). The maximum number of stages that can be added to such network is (log N – 1), which corresponds to the Benes network

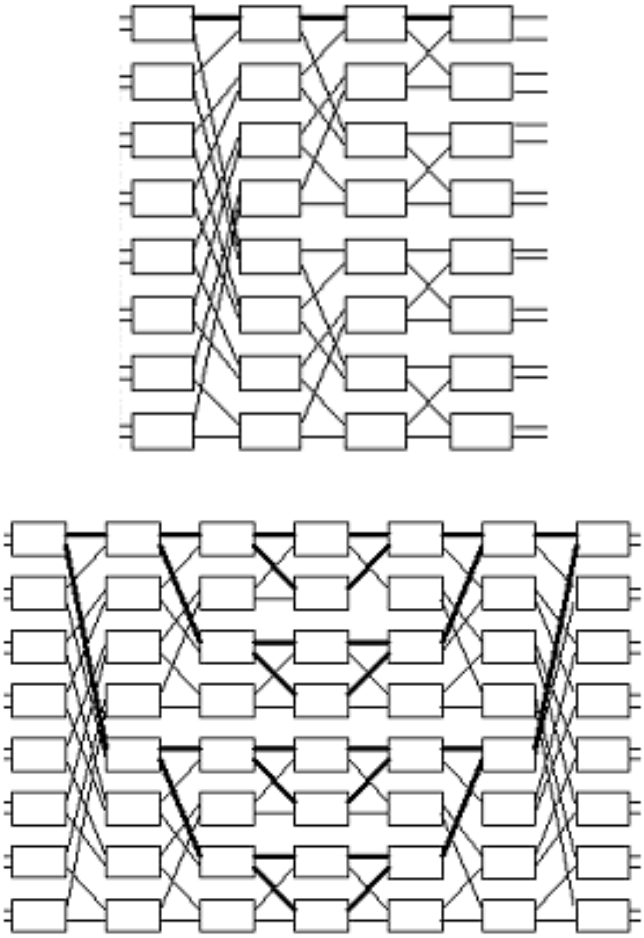

Figure 2.10: By appending extra stages, alternative routes are available

A Benes networks a small number of SEs along a path between an input–output pair. These characteristics have made Benes networks attractive for constructing DC-based optical switching networks because loss and attenuation of an optical signal are proportional to the number of couplers that the optical signal passes through. Benes networks proves to be rearrangeably non-blocking network [26]. A Benes networks of *N* inputs has *2logN-1* stages, and each stage has *N/2* SEs. Therefore, A Benes networks has O(NlogN) SEs, which is smaller than either the Crossbar or the Clos network.

2.5 Problems in OMIN

Fiber optic communications promise to meet the increasing demand of communication systems, and received much attention in parallel processing community as well [31]. Although optical MIN has great promises and has some advantages over the electronic MIN, it leads to some other problems too. Optical MINs suffer with problems such as path loss, conversion of the signal at the switch and crosstalk [2, 4, 17].

2.5.1. Bottlenecks at The Switch Element

An optical cross-connect switch may be thought of as a black box with multiple input and output fibers carrying network traffic. The optical cross-connect switches used in today's networks rely on electronic cores. An optical signal arriving at a switch input port is converted to an electronic signal by a high-speed photo detector (receiver). Electronic circuits in the switch core then direct the signal to the desired output port. A final electrical-to-optical conversion is performed by a laser diode, transforming the signal back into light for onward transmission on the fiber network [2].

The fundamental problem with these electronic cores is that they do not scale well to large port counts (numbers of input and output channels) and are costly to replace for network upgrades to the higher data rates needed for the growing demand for bandwidth. In order to avoid this problem, the need is to develop all optical switching technologies with low-optical-loss switching and extremely high reliability [17].

2.5.2. Crosstalk in OMIN

One of the most serious problems is optical crosstalk in optical MIN. This crosstalk occurs when two signal channels interact with each other. When a crosstalk happens, a small

fraction of the input signal power may be detected at another output although the main signal is injected at the right output. For this reason, when a signal passes many switching elements, the input signal will be distorted at the output due to the loss and crosstalk introduced on the path [4, 31]. This was not too big an issue in electrical MINs, but because the more stringent bit error rate in optical network, it has become a big problem. There are two ways in which optical signals can interact in a planar switching network. The channels carrying the signals could cross each other in order to embed a particular topology. Alternatively, two paths sharing a SE will experience some undesired coupling from one path to another within a SE [31]. This is shown in Figure 2.11:

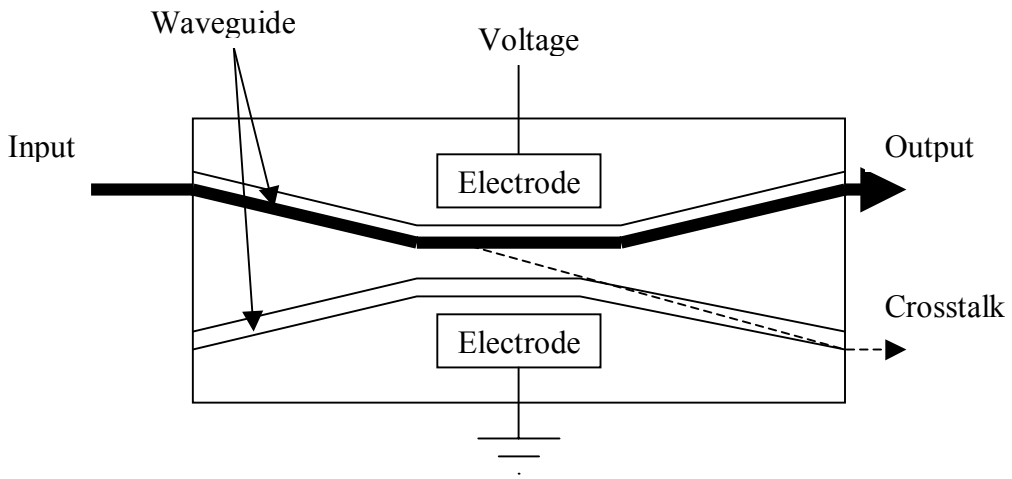

Figure 2.11. Cross talk in a switch element

Each switching element can be in two connecting schemes as shown in Figure 2.11. Since these two ways in Figure 2.11 will cause crosstalk, what we need to do is to prevent these situations from occurring in all the switching elements.

2.6. Approaches to avoid crosstalk

To reduce the negative effect of crosstalk, many approaches have been proposed. One way to solve crosstalk is to use a $2N \times 2N$ regular MIN to provide the $N \times N$ connection [4]. But half the inputs and outputs are wasted in this approach. We can use wave division multiplexing (WDM) [1, 27, 30, 40] and time division multiplexing (TDM) [31, 37] to solve this problem. WDM is based on a well-known concept of called frequency division multiplexing or FDM. With this technology, the bandwidth of a channel (its frequency domain) is divided into multiple channels, and each channel occupies a part of the larger frequency spectrum. In WDM networks, each channel is called a *wavelength*. This name is used because each channel operates at a different frequency and at a different optical wavelength. Because we are dealing with optical networks, wavelengths on the fiber are separated from each other by a guard band, which helps prevent their interfering with each other. This idea is called channel spacing, or simply spacing [4]. It is similar to the idea of guard bands used in electrical systems. In Figure 2.12, small gaps between each channel represent the guard band.

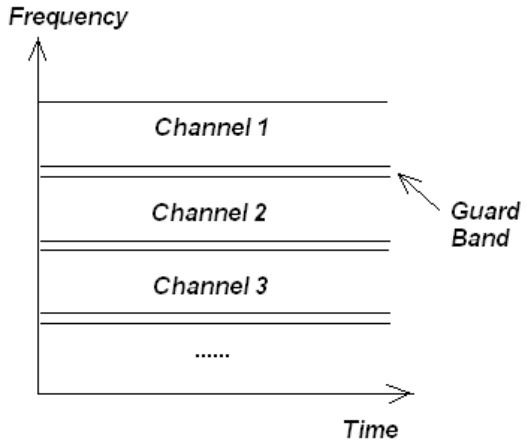

Figure 2.12. WDM

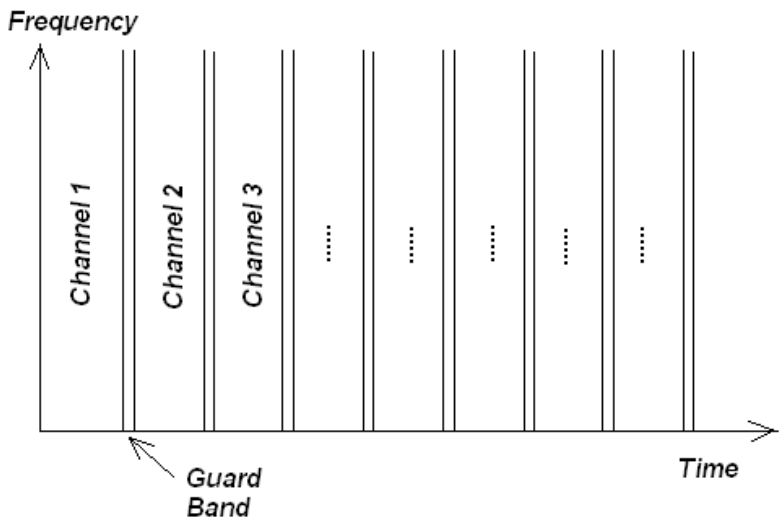

Figure 2.13. TDM

TDM provides full channel capacity but divides the channel usage into time slots. Each user is given a slot and the slots are rotated among the users. A pure TDM system cyclically scans the input signals (incoming traffic) from the multiple incoming data sources (communication links, for example) [4]. Figure 2.13 shows on how TDM can be implemented.

Most optical networks (or, for that matter, most networks in general) use a combination of WDM and TDM by time-division multiplexing fixed slots onto a specific wavelength, as in Figure 2.14, in which, each color represents a message, and as can be seen, messages are spread among several time slots and several frequencies. This concept is quite valuable because it allows multiple users to share one WDM wavelength's capacity.

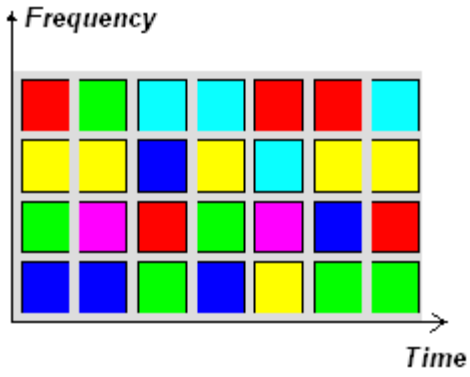

Figure 2.14. A hybrid of WDM and TDM

In this research we are interested in routing traffic through an N × N optical network to avoid coupling two signals within each switching element. We can implement this idea using time division multiplexing (TDM) [37]. This means, in every switching element, there are only four legal passing connections as shown in Figure 2.15.

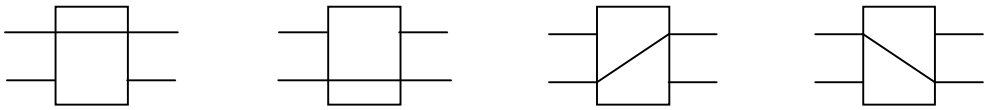

Figure 2.15. Legal passing connections in a SE at a time

Chapter 3

Related Work

The Benes architecture is a rearrangeably nonblocking switch architecture, and is one of the most efficient switch architectures in terms of the number of 2x2 switches it uses to build larger switches.

In this dissertation, we will focus on the optical Benes networks that are free of first-order crosstalk in SEs (we refer to this as crosstalk-free hereafter). It is the crosstalk-free constraint that makes the analysis of optical Benes networks different from that for electronic ones. In Benes networks, when two connections intend to use the same link, one of them will be blocked. This is called link-blocking. There is, however, another type of blocking in optical Benes networks. If adding the connection causes some paths including the new one to violate the crosstalk-free constraint, the connection cannot be added even if the path is available. We refer to this second type of blocking as crosstalk-blocking. Since the crosstalk-free constraint requires that only one signal be allowed to pass through a SE at a time, it thus has a larger contribution to the overall blocking probability than that of link-blocking.

Vertical stacking of multiple copies of an optical Benes network is a novel scheme for constructing nonblocking (crosstalk-free) optical switching networks with neither increasing the number of stages nor sacrificing the self-routing property of the Benes network [12], as shown in Figure 3.1.

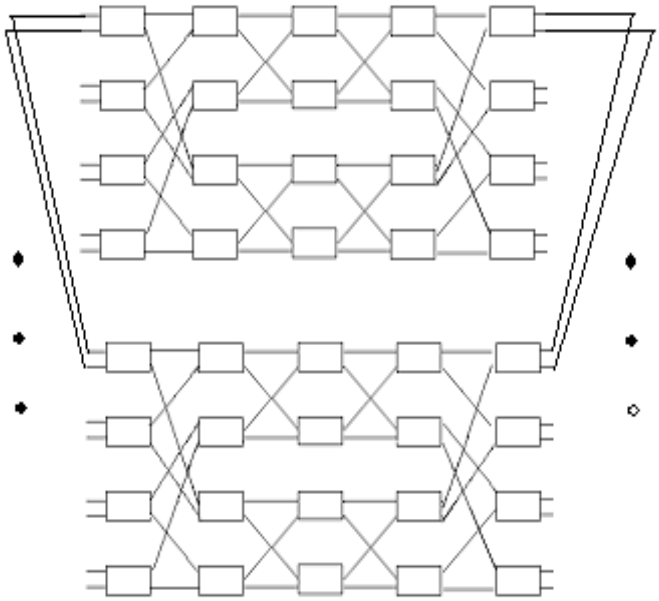

Figure 3.1. A Stacked Banyan Networks with Two Planes.

We use VSOBN to denote vertically stacked optical Benes networks and VSOBN(N,m) to denote an VSOBN network that has m stacked copies (planes) of an NxN Benes network. Previous results [12],[13] focus on determining the minimum number of planes required for nonblocking VSOBN(N,m) networks. These results indicate that the vertical stacking scheme, although is attractive, requires a prohibitively high hardware cost for building a nonblocking VSOBN network.

In this section, we will briefly introduce some of the previous related work.

3.1 Analysis of Blocking Probability of Stacked Banyan Networks

Analysis of blocking probability of a network that does not meet the hardware requirement for non-blocking is an effective approach to studying network performance. In

[19], blocking probability of stacked Banyan networks is analyzed. As in Figure 3.2, vertically stacked Banyan network is one of the ways to achieve non-blockingness [12].

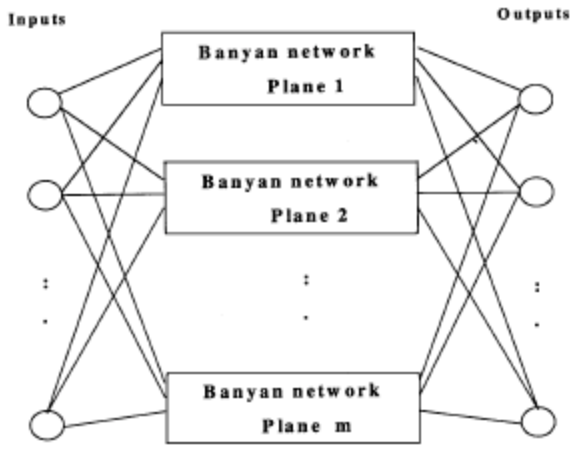

Figure 3.2 Vertically stacked Banyan network

The deterministic condition for the network being strictly non-blocking is explore and as it turns out, the hardware cost for strictly non-blocking stacked Banyan network is too high [19]. Then, the blocking probability of stacked Banyan network under the condition that the number of planes do not meet the non-blocking requirement is analyzed. The probability for the worst case to happen is derived and it is very small, which justifies that in most cases, there is really no need to meet the non-blocking requirement. The author then derives the upper bound and lower bound of the stacked Banyan network. These bounds provide quantitative measurements for tradeoffs between network hardware cost and blocking probability, and show that network hardware cost can be dramatically reduced if a small and predictable blocking probability is allowed.

3.2 Semi-permutation in Benes Network

A permutation [49] for a network is a pairing of its inputs and outputs such that each input appears in exactly one pair and each output appears also in exactly one pair. In other words, a permutation is a full one-to-one mapping between the network inputs and outputs. For an N x N network, suppose input x_i is mapped to output y_i. Then a permutation can be denoted as

$$\begin{matrix} x_0 & x_1 & \dots & x_{n-1} \\ y_0 & y_1 & \dots & y_{n-1} \end{matrix}$$

In addition, a one-to-one mapping between n' network inputs and n' network outputs (n'<n) is denoted as a partial permutation.

A partial permutation of the following form

$$\begin{matrix} x_0 & x_1 & \dots & x_{n/2-1} \\ y_0 & y_1 & \dots & y_{n/2-1} \end{matrix}$$

Where n is a even integer, xi , yi∈ {0, 1, ..., n-1}and x0<x1<…<xn_2-1 is referred to as a semi-permutation [49] of the n-element set, if

$$\left\{\left\lfloor \frac{x_0}{2} \right\rfloor, \left\lfloor \frac{x_1}{2} \right\rfloor, \dots, \left\lfloor \frac{x_{n/2-1}}{2} \right\rfloor\right\} = \{0, 1, \dots, n/2 - 1\}$$

And

$$\left\{\left\lfloor \frac{y_0}{2} \right\rfloor, \left\lfloor \frac{y_1}{2} \right\rfloor, \dots, \left\lfloor \frac{y_{n/2-1}}{2} \right\rfloor\right\} = \{0, 1, \dots, n/2 - 1\}$$

Clearly, a semi-permutation is a partial permutation that ensures that there is only one active link passing through each input switch and output switch, that is, it eliminates crosstalk

in the first and last stages in the network, and thus it has the potential to be realized in an optical network under the constraint of avoiding crosstalk. It is then established that semi-permutation can actually be realized in an optical Benes network without incurring crosstalk. The algorithm used to decompose a permutation into two semi-permutations is proposed and then the algorithm to route the semi-permutation.

3.3. Optical Window

Since we do not allow cross talk in optical muiti-stage networks, we have to use a method to find out which messages should not be assigned to the same wavelength because they will cause crosstalk. The window method in Omega network [32] is used for finding conflicts among all the messages to be sent. This method has already been proved to be correct by other researchers. It can be described roughly as follows. Given a permutation, we combine each source address and its corresponding destination address to produce a matrix. The optical window size is the m-1, where $m = \log_2 N$ and N is the size of the network. We use this window on the produced matrix from left to right except the first column and last column. If two messages have the same bit pattern in any optical window, they will cause conflict in the network. That means they cannot be in the same group, hence, they have to be routed with different wavelength.

To see how the window method works, the source-destination permutation shown in Figure 3.3 is considered, where the network size is 8. The optical window method is applied as shown in Figure 3.5 on the permutation shown in Figure 3.4.

Source:	0 1 2 3 4 5 6 7
Destination:	4 1 2 3 0 5 6 7

Figure 3.3 A permutation

000 → 100
001 → 001
010 → 010
011 → 011
100 → 000
101 → 101
110 → 110
111 → 111

Figure 3.4. Source-Destination permutation

Step 1 (window 0)	Step 2 (window 1)	Step 3 (window 2)
0 0 0 1 0 0	0 0 0 1 0 0	0 0 0 1 0 0
0 0 1 0 0 1	0 0 1 0 0 1	0 0 1 0 0 1
0 1 0 0 1 0	0 1 0 0 1 0	0 1 0 0 1 0
0 1 1 0 1 1	0 1 1 0 1 1	0 1 1 0 1 1
1 0 0 0 0 0	1 0 0 0 0 0	1 0 0 0 0 0
1 0 1 1 0 1	1 0 1 1 0 1	1 0 1 1 0 1
1 1 0 1 1 0	1 1 0 1 1 0	1 1 0 1 1 0
1 1 1 1 1 1	1 1 1 1 1 1	1 1 1 1 1 1
Message Conflicts	Message Conflicts	Message Conflicts
000 and 100	000 and 110	000 and 101
001 and 101	001 and 011	001 and 100
011 and 111	101 and 111	110 and 111
010 and 110	010 and 100	010 and 011

Figure 3.5 Optical Windows Method

In the above example shown in Figure 3.5 message 000 and 100 in Optical window 0 have the same bit pattern of “00” inside the window and hence have a conflict. The bit patterns in the above example can be any of the four combinations of “00”, “01”, “10”, “11”. For

example, in Optical Window 0, message 0 conflicts with message 4, for their bit patterns being both “00”; message 1 conflicts with message 5, for their bit patterns being both “01”; message 2 conflicts with message 6, for their bit patterns both being “10”; and message 3 conflicts with message 7, for their bit patterns both being “11”. Therefore, message 0 and message 4 must be shaded using different colors; message 1 and message 5 must be shaded using different colors; message 2 and message 6 must be shaded using different colors; and message 3 and message 7 must be shaded using different colors. At each Optical Window the messages shaded with the same color inside an optical window have conflicts between them.

4.4. Conflict Graph and Graph Coloring

The conflict graph [36] of a permutation π is a graph G (V, E), where V is the set of all the messages, V is the set of pair of messages that ever being colored the same. For example, the conflict graph for the above example would be

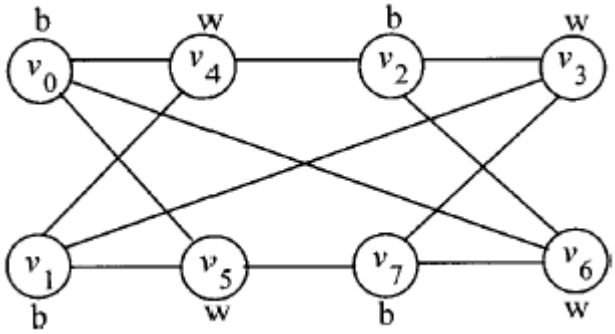

Figure 3.6. An example of conflict graph

Then the graph is colored, which is known to be NP-complete [60].

4.5. Parallel Routing in Benes Networks

The decomposition of a permutation into 2 semi-permutations takes O (N) time [36], and since there are O(logN) SEs along a path, the time complexity of routing in Benes network

is O(NlogN). Tony T. Lee [6] develops a parallel algorithm to decompose a permutation into 2 semi-permutations, and the time complexity of his algorithm is $O(\log^2 N)$, which is sketched as follows.

Packets are forwarded to one of the two output links at each SEs, and each of these output links is labeled as forward routing bit, or FRB (for the baseline network) and reverse forward routing bit or RRB (for the reverse baseline network) A set of routing paths in is said to be non-blocking if and only if the following are satisfied.

1) Symmetric Routing Constraint: The FRB of input and the RRB of output must be the same if and are in a connection request. It is because they must be connected to the same central module in order to establish a path.

2) Internally Conflict-Free Constraint: The FRBs (RRBs) of the two input (output) ports on the same switching module must be distinct in order to avoid internal conflict.

When performing switching function, a 2 x 2 switching element can either be in *bar state (state 0)* in which the two positions remain, or in *cross state (state 1)* in which the two positions exchange from the inputs to the outputs. We use a_i and b_i to denote the state of the th input (output) switching element. When a packet gets across a switching element, its routing bit determines the state of the element, and vice versa. For a permutation as

$$\begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 4 & 5 & 1 & 6 & 0 & 2 & 7 & 3 \end{pmatrix}$$

Let α and β denote the FRBs and the RRBs, respectively, of packets. The symmetric self-routing constraint requires that $\alpha(k) = \beta(\pi(k))$, for k = 0, 1, …, N-1. and the internal conflict-free constraints require that $\alpha(k) = \alpha'(k+1)$. Combine these two formula, we will get $\beta(\pi(k)) =$

$\alpha(k) = \alpha'(k+1) = \beta'(\pi(k+1))$. With the state variables of switching elements, the above routing bits can be as well given by

$$\alpha(k) = \begin{cases} a_{k/2}, & \text{k is even} \\ a_{(k-1)/2}, & \text{k is odd} \end{cases}$$

$$\beta(k) = \begin{cases} b_{k/2}, & \text{k is even} \\ b_{(k-1)/2}, & \text{k is odd} \end{cases}$$

By such algebraic interpretation, we can set up the Boolean equations for the route assignment as demonstrated below. The permutation given in the previous example can first be mapped into

$a_0 \quad a_0' \quad a_1 \quad a_1' \quad a_2 \quad a_2' \quad a_3 \quad a_3'$

$b_2 \quad b_2' \quad b_0 \quad b_3' \quad b_0 \quad b_1' \quad b_3 \quad b_1'$

It then gives

$$\mathbf{a_0 = b_2, \quad a_0' = b_2', \quad a_1 = b_0', \quad a_1' = b_3,}$$
$$\mathbf{a_2 = b_0, \quad a_2' = b_1, \quad a_3 = b_3', \quad a_3' = b_1'}$$

We can eliminate all a's from the above and obtain a set of equations composed only of b's as

$$\mathbf{b_2 = b_2, \quad b_0 = b_3, \quad b_1 = b_0', \quad b_3 = b_1'}$$

This set of equations is called the initializing equations. As it turns out, the permutation is partitioned into several equivalent classes. For example, the above permutation is partitioned into 2 equivalent classes, class 1 being $\{b_0, b_1, b_3\}$ and class 2 being $\{b_2\}$, and then a representative from each class is chosen and assigned a value 0 or 1. Up to now, the states of outer stages are set, and algorithm recursively set the states of each inner stages.

4.6 Parallel Routing in Benes Network without Crosstalk

In [62], a parallel routing and wavelength assignment algorithm is proposed. Conceptually, this algorithm has logN rounds, in each round i, the algorithm sets up the SEs at

stages i and 2logN-i-2, and uses at most 2(i + 1) +1 wavelengths to ensure that there is no crosstalk in stages {0 … i}, and {2logN-i-2, …, 2logN-2} , The algorithm is briefly described as follows,

1. Decompose a permutation into two semi-permutations, each named upper or lower semi-permutation, satisfying that two active inputs (outputs) in an SE in the first (last) stage are in different parts.
2. Route one of them through the upper sub network, and the other through the lower sub network.

The above two steps decide the setup of the Benes network, and the following steps are to assign wavelengths to the network.

3. If there is a connection c' so that c and c' pass through the same SE in stage i , record this information and assign c and c' different wavelength from different end of the available wavelength pool

Chapter 4

Upper Bound on Blocking Probability of VSOBN

In this chapter, we shall analyze the worst-case scenarios in Benes networks and develop an upper bound on blocking probability of vertical stacked optical Benes networks.

4.1 Strictly Nonblocking without Worst Case Scenarios

In this section, we briefly describe the deterministic condition for the strictly nonblocking VSOBN network that is obtained based on worst-case and second worst-case analysis. We also evaluate the probability that the worst-case scenario occurs to motivate the work of this paper.

Due to their topological symmetry, all paths in a Benes network have the same property in terms of blocking. To study the blocking probability, we can arbitrarily select an input and an output in the network and set up a connection between them. Throughout this paper, we will select the path between the first input and the first output and try to set up a connection between them. We call the path between this input-output pair the *tagged path*. All the SEs on the tagged path are called *tagged SEs*. In Benes networks, all paths between the targeted pair are called the tagged paths.

The flow of information through the network is assumed to be from left to right—all the inputs being on the left-hand side and all the outputs on the right-hand side of the network. The stages of SE's are numbered from left (stage 1) to right (2logN -1 stage). The stages of links are also numbered from left to right, but starting from 0 (input links) to (2logN -1destination links). For a tagged path, an *input intersecting set* (IIS) I_i associated with stage i (1<=i<=2logN-1) is defined as the set of all inputs that intersect a tagged SE at stage. Likewise, an *output intersecting set* O_i (OIS) associated with stage i is the set of all outputs that intersect a tagged SE at stage 2LogN-i. Fig. 4.1 shows some examples.

We are interested in an optical network that is nonblocking and crosstalk-free. This can be achieved at the cost of extra hardware. For a VSOBN network, the following theorem gives the deterministic condition for strictly nonblocking [12], we are going to discuss the noblocking requirement without the worst case scenarios.

Theorem 4.1: VSOBN is strictly nonblocking if and only if the following condition is true:

$$m >= 2x + (2N/2^x)^{1/2} - 1 \text{ in which } x = \log N - 1$$

The above result was obtained based on worst-case analysis. That is, to find the maximum possible number of connections that will conflict the tagged path and let each of these connections block a distinct plane.

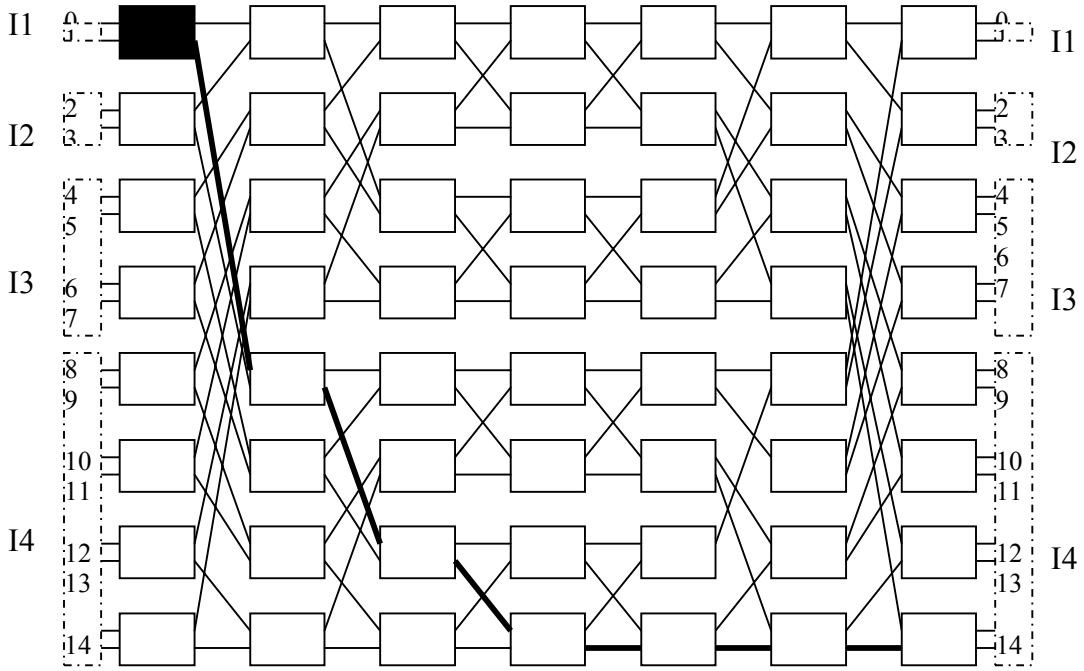

Figure 4.1 Different input (output) links have different blocking capabilities

From figure 4.1, we know that different input(output) links have different blocking capabilities. Inputs(outputs) in I1 have the capabilities to block the whole plane; inputs(outputs)

in I2 have the capabilities to block only ½ plane…; inputs(outputs) in Ik have the capabilities to block only $1/2^{k-1}$plane. When a connection is set up between an input from Ii and an output from Oj, the connection will block $1/2^{\min\{i,j\}-1}$ plane. Therefore, the problem of finding the worst-case traffic pattern can be formulated as follows:

Given a set Γ: {1, ½, ½, ¼, ¼, ¼, ¼, …, 1/ N/2 ,…, 1/ N/2}, find a relation Γ x Γ, such that Σ max(Γ x Γ) is maximized.

It is clear that in order to maximize the sum, the relation must be as unbalanced as possible. For example, for set{1, ½}, (1, ½) and (½,1) would be a better choice than(1,1) and (½, ½) since the former will block 2 plane, but the latter will only block 1½ planes.

Therefore, in order to maximize the sum, we must pick the relation pairs from the two ends of the set. We shall prove this:

Given a set Γ: {1, ½, ½, ¼, ¼, ¼, ¼, …, 1/ N/2 ,…, 1/ N/2}, to find a relation Γ x Γ, such that Σ max(Γ x Γ) is maximized, the relations picked from different end is the optimal solution.

Proof: we sort the list 1, ½, ½, ¼, ¼, ¼, ¼, …, 1/ N/2 ,…, 1/ N/2 in descending order then sort it in ascending order 1/ N/2,…, 1/ N/2,…¼, ¼, ¼, ¼, ½, ½, 1. In the first list, we pick 1 and then pair it with 1/ N/2 from the second list, since 1/ N/2 is the smallest number in the second list and due to the fact that the larger valued between the two in each pair will be produced, pair 1/ N/2 with 1 will in no way decrease the ultimate value. Once this pair has been settled, the lists become ½, ½, ¼, ¼, ¼, ¼, …, 1/ N/2 ,…, 1/ N/2 and 1/ N/2,…¼, ¼, ¼, ¼, ½, ½, 1; and the algorithm will proceed recursively. From this, it turns out that the problem possesses greedy property and it is clear that the greedy solution is optimal in this case. ◘

And the relation is (1,1/N/2), (1/2, 1/N/2),(1/2, 1/N/2),(1/2, 1/N/2),(1/2, 1/N/2),(1/4, 1/N/2),…, (1/N/4, 1/N/2) ,…, (1/N/4, 1/N/2), their respective inversions and (1/N/2, 1/N/2). Add 1,1,

½, ½, ½, ½, ½, ½, ½, ½, ¼, …, 1/N/4 ,…,1/N/4 ,1/N/2 together, we will get the result of theorem 4.1.

From Theorem 4.1, it is clear that the hardware cost for a strictly non-blocking VSOBN network is high.

Let us find out the probability that the worst-case scenario could occur. Let the probability that an input (output) port is busy be r and denote by P_{worst} the probability that the worst-case scenario occurs. P_{worst} is then given in the following lemma under the assumption that statuses of individual input (output) ports are independent.

Lemma 4.1: In an NxN optical benes network, we have

$$P_{worst} = \left(r^{n/2-1} \frac{N/2}{\binom{N-1}{N/2-1}} r \right)^2$$

Proof: Under the constraint of crosstalk-free, the worst-case scenario of conflicts on the tagged path is when all inputs in set I_i(1<=i<=logN-1) are destined for the outputs in O_{logN} and all outputs in set O_i are originated from the inputs in set I_{logN}. Thus, the maximum number of conflicts with the tagged path is determined by both the connections from set I_i and set O_i. Any of the N/2-1 input can reach any of the N-1 output. In the worst case scenario, a connection from I_i must be terminated at an output in O_{logN} and there is N/2 of them. And we have to consider the case in which all the inputs in I_{logN} going to O_i.

Therefore, $P_{worst} = \left(r^{n/2-1} \frac{N/2}{\binom{N-1}{N/2-1}} r \right)^2$ ◘

When r=0.9 and N =64 and r=0.9 and N=128 ,

$P_{64} = 2.63 \times 10^{-34}$

$P_{128} = 3.97 \times 10^{-77}$

This indicates that the probability of worst case from happening is very small or even can be ignored.

Table 4.1: Blocking Capacity and No. of planes required

No. of inputs	16	32	64	128	256
Blocking Capacity	6 1/8	8 1/16	10 1/32	12 1/64	14 1/128
No. of planes required	7	9	11	13	15

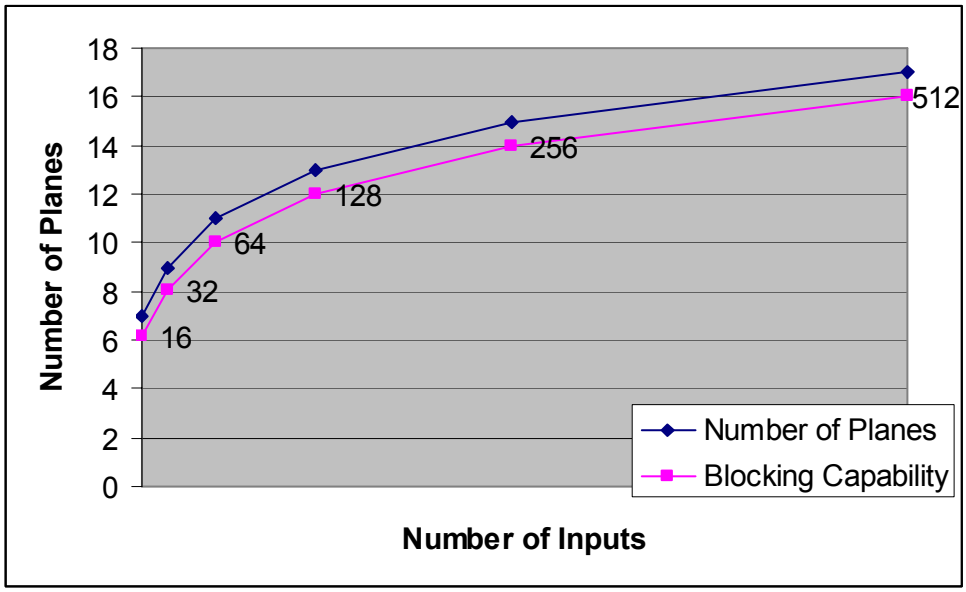

Figure 4.2: Blocking requirement and actual number of planes needed for worst cases.

We now define the term second worst-case scenario as follows:

Second worst case is the case (cases) in which the second largest blocking capability occurs.

It is also important to find out the probability that the second worst-case scenario could occur. The reason is explained as follows:

Take when N = 16 as an example,

I1: {1}	O1: {1}
I2: {½, ½}	O2: {½, ½}
I3: {¼, ¼, ¼, ¼}	O3: {¼, ¼, ¼, ¼}
I4: {1/8,1/8,1/8,1/8,1/8,1/8,1/8,1/8}	O4: {1/8,1/8,1/8,1/8,1/8,1/8,1/8,1/8}

The worst-case is when inputs (outputs) in I_1, I_2, and I_3 are connected to outputs (inputs) in I_4. There will be one pair in I_4 and O_4 left and this pair accounts for the 1/8. In order to be strictly non-blocking, we need 7 planes.

Now let us consider the second worst-case. In the second worst-case, the blocked planes will be 6. The second worst case happens either when I_1, I_2, I_3 going to O_4 and O_1, O_2, O_3 going to I_4 while the remaining one pair in I_4 and O_4 is not connected or one pair in I_3 and O_3 is connected while rest of the inputs (outputs) are connected to O_4 (I_4). Therefore, in the former case, the blocking capability is reduced by 1/8 because the remaining pair in I_4 and O_4 is not connected; in the latter case, because one pair in I_3 and O_3 is connected, while in the worst case these two are both connected with one in I_4 or O_4, the capability of blocking is reduced by ¼. At the same time, the pair in the I_4 and O_4 must be connected together, which increased the blocking capability by 1/8. So, blocking difference between the worst-case and the second worst-case will be ¼-1/8 = 1/8.

This proved Theorem 4.2.

Theorem 4.2: A VSOBN network is strictly nonblocking with m = 2*(logN –1) when worst and second worst case do not occur.

Now it is the time to find out the probability that second worst-case could happen.

Lemma 4.2: In an NxN optical benes network, we have

$$P_{\text{second-worst}}=\left(r^{n/2-1}\frac{N/2}{\binom{N-1}{N/2-1}}(1-r) \right)^2+\left(r^{n/2-2}\frac{\binom{N/2}{N2/-2}}{\binom{N-1}{N/2-2}}r^2 \right)^2$$

Proof: Under the constraint of crosstalk-free, the second worst-case scenario of conflicts on the tagged path happens in two cases.

Case 1: when all inputs in set I_i(1<=i<=logN-2) are destined for the outputs in O_{logN} and all outputs in set O_i (1<=i<=logN-2) are originated from the inputs in set I_{logN} For inputs in I_{logN-1} and outputs in O_{logN-1}, only one pair is connected, rest of them are connected to O_{logN} or I_{logN}. Thus, the maximum number of conflicts with the tagged path is determined by both the connections from set I_i and set O_i. Any of the N/4-1 input can reach any of the N-1 output. In the worst case senario, a connection from I_i must be terminated at an output in O_{logN} and there is N/4 of them. One input from I_{logN-1} and one output in O_{logN-1} is connected. And we have to consider the case in which all the inputs in I_{logN} going to O_i.

Case 2: Exactly the same as in the worst case except that the pair in I_{logN} and O_{logN} is idle. In this case, the total blocking capacity is reduced by 1/8.

Therefore, $P_{\text{second-worst}}=\left(r^{n/2-1}\frac{N/2}{\binom{N-1}{N/2-1}}(1-r) \right)^2+\left(r^{n/2-2}\frac{\binom{N/2}{N/2-2}}{\binom{N-1}{N/2-2}}r^2 \right)^2$

When r=0.9 and N =64 and r=0.9 and N=128 respectively.

$P_{64} = 3.91 \times 10^{-34}$

$P_{128} = 8.38 \times 10^{-74}$

This shows that the probability that second worst case happens is still very small, which justifies theorem 4.2.

The result is significant, because we can save one whole plane if the worst case and second worst-case do not happen very often.

Table 4.2 Savings (percentage) without 1st and 2nd worst case

No. of inputs	16	32	64	128	256	512	1024
No. of planes w/o 1st and 2nd worst scenarios	6	8	10	12	14	16	18
Blocking requirement w/o 1st worst scenarios	6 1/8	8 1/16	10 1/32	12 1/64	14 1/128	16 1/256	18 1/512
Saving(percentage) w/o 2nd worst case	14.3%	11.1%	9.1%	7.7%	6.7%	5.9%	5.3%

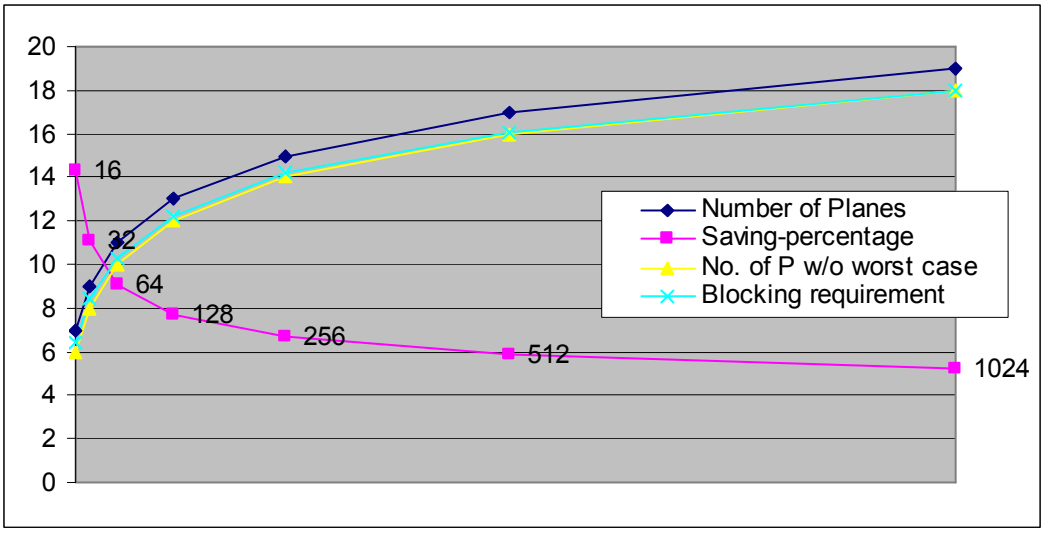

Figure 4.3. Savings (percentage) without 1st and 2nd worst case.

The table shows that even when N = 1024, which is a quite large network, we can save 5.3%, if we neglect first and second order worst-cases.

Lemma 5.3: In an NxN optical benes network, further neglecting third order worst-case does not save a plane.

For example when N = 16, in the third worst-case, the blocked planes will be 4.875. The reason is that 1/8 is the smallest residue in this system and in order to get the blocked planes in the third worst case, all needs to be done is to let two pair in I_3 and O_3 to be connected; while in the second worst case only one pair is connected the other pair is connected with one in I_4 or O_4. Therefore, the capability of blocking is reduced by ¼. And the same time, the pair in the I_4 and O_4 must be connected together, which increased the blocking capability by 1/8. So, blocking difference between the second worst-case and the third worst-case will be ¼-1/8 = 1/8. Therefore, further neglecting third order worst-case is meaningless.

From the above analysis, it can be seen that spending a large amount of extra hardware in order to guarantee the strictly non-blocking property is not cost-effective in most cases. This motivates us to find out the blocking probability of a VSOBN network with respect to the number of planes (hardware cost), and to seek an approach to making tradeoff between hardware cost and blocking probability.

4.2 Upper Bound on Blocking Probability

Since exact blocking probability is hard to obtain, in this section we will derive various formulas to get an upper bound on the blocking probability of a VSOBN in terms of the number of planes. This bound can be considered as the estimate for the worst case blocking probability. In the following discussion, we give a few definitions and notation that will be used in the analysis.

For an N x N network, a matrix of logN x logN is proposed. An element in the matrix C_{ij} stands for the connection from I_i to O_j.

For an N x N network, a matrix of logN x logN is proposed. An element in the matrix C_{ij} stands for the connection from I_i to O_j.

$$
\begin{matrix}
C_{11} & C_{12} & \dots & C_{1n} \\
C_{21} & C_{22} & \dots & C_{2n} \\
. & & & \\
. & & & \\
. & & & \\
C_{(n-1)1} & C_{(n-1)2} & \dots & C_{(n-1)n} \\
C_{n1} & C_{n2} & \dots & C_{nn}
\end{matrix}
$$

We use C_{Ii} to denote the sum of coefficients of C_{ij} in row i where j>=i. For example:

$C_{I2} = C_{22} + C_{23} + \dots + C_{2n}$

We use C_{Oj} to denote the sum of coefficients of C_{ij} in column j where i>=j. For example:

$C_{O2} = C_{22} + C_{32} + \dots + C_{n2}$

By this definition, C_{Ii} stands for the connections coming from input group i going to ouput groups j (j>=i). C_{Oj} stands for the connections going to ouput group j coming from input groups i (i>=j).

The total blocking capability BC is therefore $C_{I1} + C_{O1} - C_{11} + ½*(C_{I2} + C_{O2} - C_{22}) + ¼*(C_{I3} + C_{O3} - C_{33}) + \dots + 1/2^{logn-1}*(C_{In} + C_{On} - C_{nn})$

C_{I1} stands for the connections coming from input group to every output group. C_{O1} stands for the connections coming from every input group to output group 1. And any connection in C_{I1} and C_{O1} will block a whole plane. But there are some overlaps in these tow groups and the overlap is C_{11}(which are the connections coming from input group 1 to output group 1), since it has been counted twice in C_{I1} and C_{O1}.

C_{I2} stands for the connections coming from input group 2 to every output group other than group 1. C_{O2} stands for the connections coming from every input group other than input group 1 to output group 2. And any connection in C_{I2} and C_{O2} will block 1/2 whole plane. But there are some overlaps in these two groups and the overlap is C_{22}(which are the connections coming from input group 2 to output group 2), since it has been counted twice in C_{I2} and C_{O2}.

Similar consideration will apply to C_{I3}, C_{O3},… C_{In}, C_{On}.

Clearly there will be no blocking if $C_{I1} + C_{O1} - C_{11} + ½*(C_{I2} + C_{O2} - C_{22}) + ¼*(C_{I3} + C_{O3} - C_{33}) + \ldots + 1/2^{logn-1}*(C_{In} + C_{On} - C_{nn}) < m$. (1)

Therefore we have

$$P(\text{nonblocking}) = \sum_{C_{I1}=0}^{\min(2^0,m-1)} \sum_{C_{I2}=0}^{\min(2^1,(m-1)*2)} \cdots \sum_{C_{In}=0}^{\min(n/2,(m-1)*2^{\log n-1})} \sum_{C_{O1}=0}^{\min(2^0,m-1)} \sum_{C_{O2}=0}^{\min(2^1,(m-1)*2)} \cdots \sum_{C_{On}=0}^{\min(n/2,(m-1)*2^{\log n-1})}$$

$P(C_{I1}, C_{O1}, C_{I2}, C_{O2}, \ldots, C_{In}, C_{On}) * P(C_{I1} + C_{O1} - C_{11} + ½*(C_{I2} + C_{O2} - C_{22}) + ¼*(C_{I3} + C_{O3} - C_{33}) + \ldots + 1/2^{logn-1}*(C_{In} + C_{On} - C_{nn}) <= m-1$

$|C_{I1}, C_{O1}, C_{I2}, C_{O2}, \ldots, C_{In}, C_{On})$ (2)

The lower bound of C_{I1} is 0 since it can not be negative. On the other hand, C_{I1} must not be greater than 2^0, since there is only 2^0 input in I_1; and C_{I1} must not be greater than m-1, since there are only m planes in the networks and if C_{I1} is greater than m-1, then the network will be blocked. Therefore, C_{I1} must not be greater than the minimum of 2^0 and m-1.

The lower bound of C_{I2} is 0 since it can not be negative. On the other hand, C_{I2} must not be greater than 2^1, since there is only 2^1 inputs in I_2; and C_{I2} must not be greater than 2*(m-1), since there are only m planes in the networks and if C_{I2} is greater than 2*(m-1), then the network will be blocked. Therefore, C_{I1} must not be greater than the minimum of 2^1 and 2*(m-1).

And the ranges of other connections can be derived accordingly.

From (1)

$C_{I1} + C_{O1} - C_{11} + ½*(C_{I2} + C_{O2} - C_{22}) + ¼* (C_{I3} + C_{O3} - C_{33}) + \ldots + 1/2^{logn-1}*(C_{In} + C_{On} - C_{nn}) <= m-1$,

by simple algebraic manipulation,

We have $C_{11} + ½ C_{22} + ,\ldots, + 1/2^{logn-1}C_{nn} >= C_{I1} + C_{O1} + ½*(C_{I2} + C_{O2}) + \ldots + 1/2^{logn-1} (C_{In} + C_{On}) - m + 1$ (3)

Therefore,

$$P(\text{nonblocking}) = \sum_{C_{I1}=0}^{\min(2^0,m-1)} \sum_{C_{I2}=0}^{\min(2^1,(m-1)*2)} \cdots \sum_{C_{In}=0}^{\min(n/2,(m-1)*2^{\log n-1})} \sum_{C_{O1}=0}^{\min(2^0,m-1)} \sum_{C_{O2}=0}^{\min(2^1,(m-1)*2)} \cdots \sum_{C_{On}=0}^{\min(n/2,(m-1)*2^{\log n-1})}$$

$$P(C_{I1}, C_{O1}, C_{I2}, C_{O2}, \ldots, C_{In}, C_{On}) * \sum_{C_{11}+1/2C_{22}+1/4C_{33}+\ldots+1/2^{\log n-1}C_{nn}=Lower}^{Upper} P(C_{11}, C_{22}, ,\ldots, C_{nn} | C_{I1}, C_{O1}, C_{I2}, C_{O2}, \ldots, C_{In}, C_{On})$$

In which

$Lower = \max[0, C_{I1} + C_{O1} + ½*(C_{I2} + C_{O2}) + \ldots + 1/2^{logn-1} (C_{In} + C_{On}) - m + 1]$

$Upper = \min(C_{I1} , C_{O1}) + ½*\min(C_{I2} ,C_{O2}) + \ldots + 1/2^{logn-1}*\min(C_{In} , C_{On})$

From (3) $C_{11} + ½ C_{22} + ,\ldots, + 1/2^{logn-1}C_{nn} >= C_{I1} + C_{O1} + ½*(C_{I2} + C_{O2}) + \ldots + 1/2^{logn-1} (C_{In} + C_{On}) - m + 1$, we can see that the lower bound of $(C_{11} + ½ C_{22} + ¼ C_{33} + \ldots + 1/2^{logn-1}C_{nn})$ is $C_{I1} + C_{O1} + ½*(C_{I2} + C_{O2}) + \ldots + 1/2^{logn-1} (C_{In} + C_{On}) - m + 1$.

On the other hand, C_{11} must not be greater than C_{I1} or C_{01}; $1/2\ C_{22}$ must not be greater than $1/2C_{I2}$ or $1/2C_{02}$ … and so forth. Here so derives the upper bound of $(C_{11} + ½ C_{22} + ¼ C_{33} + \ldots + 1/2^{logn-1}C_{nn})$.

And

$P(\text{nonblocking}) = P(C_{I1}, C_{O1}, C_{I2}, C_{O2}, \ldots, C_{In}, C_{On})\, P(C_{11}, C_{22},\ldots, C_{nn} \mid C_{I1}, C_{O1}, C_{I2}, C_{O2}, \ldots, C_{In}, C_{On})$

$= P(C_{11}, C_{22},\ldots, C_{nn}, C_{I1}, C_{O1}, C_{I2}, C_{O2}, \ldots, C_{In}, C_{On})$

$= P(C_{I1}\, C_{11})P(C_{22},\ldots, C_{nn}, C_{O1}, C_{I2}, C_{O2}, \ldots, C_{In}, C_{On} \mid C_{I1}\, C_{11})$

$= P(C_{I1}\, C_{11})P(C_{22},\ldots, C_{nn}, C_{O1}, C_{I2}, C_{O2}, \ldots, C_{In}, C_{On} \mid C_{11})$

We drop C_{I1} under the assumption that the connections in C_{I1} are independent with all others, which can be justified if the amount of traffic under consideration is huge.

$= P(C_{I1}\, C_{11})P(C_{11}, C_{22},\ldots, C_{nn}, C_{O1}, C_{I2}, C_{O2}, \ldots, C_{In}, C_{On})/P(C_{11})$

$= P(C_{I1}\, C_{11})\, P(C_{O1}\, C_{11})P(C_{22},\ldots, C_{nn}, C_{I2}, C_{O2}, \ldots, C_{In}, C_{On} \mid C_{O1}\, C_{11})/P(C_{11})$

$= P(C_{I1}\, C_{11})\, P(C_{O1}\, C_{11})P(C_{22},\ldots, C_{nn}, C_{I2}, C_{O2}, \ldots, C_{In}, C_{On} \mid C_{11})/P(C_{11})$

We drop C_{O1} under the assumption that the connections in C_{O1} are independent with all others, which can be justified if the amount of traffic under consideration is huge.

$= P(C_{I1}\, C_{11})\, P(C_{O1}\, C_{11})P(C_{11}, C_{22},\ldots, C_{nn}, C_{I2}, C_{O2}, \ldots, C_{In}, C_{On})/P^2(C_{11})$

= .

.

.

$= P(C_{I1}\, C_{11})\, P(C_{O1}\, C_{11})\, P(C_{I2}\, C_{22})\, P(C_{O2}\, C_{22})\ldots P(C_{In}\, C_{nn})\, P(C_{On}\, C_{nn})P(C_{11}, C_{22}, ,\ldots, C_{nn})/ P^2(C_{11})\, P^2(C_{22})\ldots P^2(C_{nn})$

Since $C_{11}, C_{22}, ,\ldots, C_{nn}$ are independent of each other,

$P(C_{11}, C_{22}, ,\ldots, C_{nn}) = P(C_{11})P(C_{22}) ,\ldots, P(C_{nn})$

So $P(\text{nonblocking}) = P(C_{I1}\, C_{11})\, P(C_{O1}\, C_{11})\, P(C_{I2}\, C_{22})\, P(C_{O2}\, C_{22})\ldots P(C_{In}\, C_{nn})\, P(C_{On}\, C_{nn})P(C_{11}, C_{22}, ,\ldots, C_{nn})/ P^2(C_{11})\, P^2(C_{22})\ldots P^2(C_{nn})$

$= P(C_{I1}\ C_{11})\ P(C_{O1}\ C_{11})\ P(C_{I2}\ C_{22})\ P(C_{O2}\ C_{22})\ldots\ P(C_{In}\ C_{nn})\ P(C_{On}\ C_{nn})\ P(C_{11})P(C_{22})\ ,\ldots,\ P(C_{nn})/$

$P^2(C_{11})\ P^2(C_{22})\ldots\ P^2(C_{nn})$

$= P(C_{I1}\ C_{11})\ P(C_{O1}\ C_{11})\ P(C_{I2}\ C_{22})\ P(C_{O2}\ C_{22})\ldots\ P(C_{In}\ C_{nn})\ P(C_{On}\ C_{nn})\ /\ P(C_{11})\ P(C_{22})\ldots\ P(C_{nn})$

$P(C_{11})$ stands for the probability of C_{11} connections coming from input group 1 going to output group 1.

Therefore, $P(C_{11})$ is

$P(C_{11}) = \binom{1}{C_{11}} \alpha_1^{C_{11}} (1-\alpha_1)^{1-C_{11}}$, α_1 is the probability that a connection from input group 1 going to the out put groups 1. So α_1 is r * 1/N-1 in this case.(r is the incoming rate or outgoing rate of the traffic)

$P(C_{22})$ stands for the probability of C_{22} connection coming from input group 2 going to output group 2.

Therefore, $P(C_{22})$ is

$P(C_{22}) = \binom{2^1}{C_{22}} \alpha_2^{C_{22}} (1-\alpha_2)^{2^1-C_{22}}$, α_2 is the probability that a connection from input group 2 going to the out put groups 2. So α_2 is r *2^1/(N-1) in this case.

.

.

.

$P(C_{kk})$ stands for the probability of C_{kk} connection coming from input group k going to output group k.

Therefore, $P(C_{kk})$ is

$P(C_{kk}) = \binom{2^{k-1}}{C_{kk}} \alpha_k^{C_{kk}} (1-\alpha_k)^{2^{k-1}-C_{kk}}$, α_k is the probability that a connection from input group k going to the out put groups k. So α_k is $r * (2^{k-1})/(N-1)$ in this case.

$P(C_{I1}\ C_{11})$ is the probability of C_{I1} connections from input group 1 to output groups 1 and above while there are C_{11} connections from input group 1 to output group 1.

$$P(C_{I1}\ C_{11}) = \binom{2^0}{C_{11}} \alpha_1^{C_{11}} * \binom{2^0 - C_{11}}{C_{I1} - C_{11}} * \gamma_1^{C_{I1}-C_{11}} * (1-\alpha_1-\gamma_1)^{2^0-C_{I1}}$$

In which α_1 is the probability that a connection from input group 1 going to the out put groups 1. So it is $r * 2^0/N-1$ in this case. γ_1 is the probability that a connection from input group 1 going to the out put groups above 1 and γ_1 is $r * (N-2^1 / N-1)$

$P(C_{I2}\ C_{22})$ is the probability of C_{I2} connections from input group 2 to output groups 2 and above while there are C_{22} connections from input group 2 to output group 2.

$$P(C_{I2}\ C_{22}) = \binom{2^1}{C_{22}} \alpha_2^{C_{22}} * \binom{2^1 - C_{22}}{C_{I2} - C_{22}} * \gamma_2^{C_{I2}-C_{22}} * (1-\alpha_2-\gamma_2)^{2^1-C_{I2}}$$

In which α_2 is the probability that a connection from input group 2 going to the out put groups 2. So it is $r * 2^1/N-1$ in this case. γ_2 is the probability that a connection from input group 2 going to the out put groups above 2 and γ_2 is $r * (N-2^2 / N-1)$

$P(C_{Ik}\ C_{kk})$ is the probability of C_{Ik} connections from input group K to output groups K and above while there are C_{kk} connections from input group K to output group K.

$P(C_{Ik}\ C_{kk}) = \binom{2^{k-1}}{C_{kk}} \alpha_k^{C_{kk}} * \binom{2^{k-1} - C_{kk}}{C_{Ik} - C_{kk}} * \gamma_k^{C_{Ik} - C_{kk}} * (1 - \alpha_k - \gamma_k)^{2^{k-1} - C_{Ik}}$

In which α_k is the probability that a connection from input group k going to the out put group k. So α_k is $r * 2^{k-1}/N-1$ in this case. γ_k is the probability that a connection from input group k going to the out put groups above k and γ_k is $r * (N-2^k / N-1)$

$P(C_{O1}\ C_{11})$ is the probability of C_{O1} connections going to output group 1 from input groups 1 and above while there are C_{11} connections from input group 1 to output group 1.

$P(C_{O1}\ C_{11}) = \binom{2^0}{C_{11}} \alpha_1^{C_{11}} * \binom{2^0 - C_{11}}{C_{O1} - C_{11}} * \gamma_1^{C_{O1} - C_{11}} * (1 - \alpha_1 - \gamma_1)^{2^0 - C_{O1}}$

In which α_1 is the probability that a connection from input group 1 going to the out put groups 1. So it is $r * 2^0/N-1$ in this case. γ_1 is the probability that a connection going into output group 1 from the input groups above 1 and γ_1 is $r * (N-2^1 / N-1)$

$P(C_{O2}\ C_{22})$ is the probability of C_{O2} connections going into output group 2 coming from input groups 2 and above while there are C_{22} connections from input group 2 to output group 2.

$P(C_{O2}\ C_{22}) = \binom{2^1}{C_{22}} \alpha_2^{C_{22}} * \binom{2^1 - C_{22}}{C_{O2} - C_{22}} * \gamma_2^{C_{O2} - C_{22}} * (1 - \alpha_2 - \gamma_2)^{2^1 - C_{O2}}$ In which α_2 is the probability that a connection from input group 2 going to the out put groups 2. So it is $r * 2^1/N-1$ in this case. γ_2 is the probability that a connection going into output group 2 coming from the input groups above 2 and γ_2 is $r * (N-2^2 / N-1)$

$P(C_{Ok}\ C_{kk})$ is the probability of C_{Ok} connections from input group K to output groups K and above while there are C_{kk} connections from input group K to output group K.

$$P(C_{Ok}\ C_{kk}) = \binom{2^{k-1}}{C_{kk}} \alpha_k^{C_{kk}} * \binom{2^{k-1} - C_{kk}}{C_{Ok} - C_{kk}} * \gamma_k^{C_{Ok} - C_{kk}} * (1 - \alpha_k - \gamma_k)^{2^{k-1} - C_{Ok}}$$

In which α_k is the probability that a connection from input group k going to the out put group k. So α_k is $r * 2^{k-1}/N-1$ in this case. γ_k is the probability that a connection going into output group k coming from the input groups above k and γ_k is $r * (N-2^k / N-1)$

Based on the above formulas, we can calculate the blocking probability with various numbers of planes.

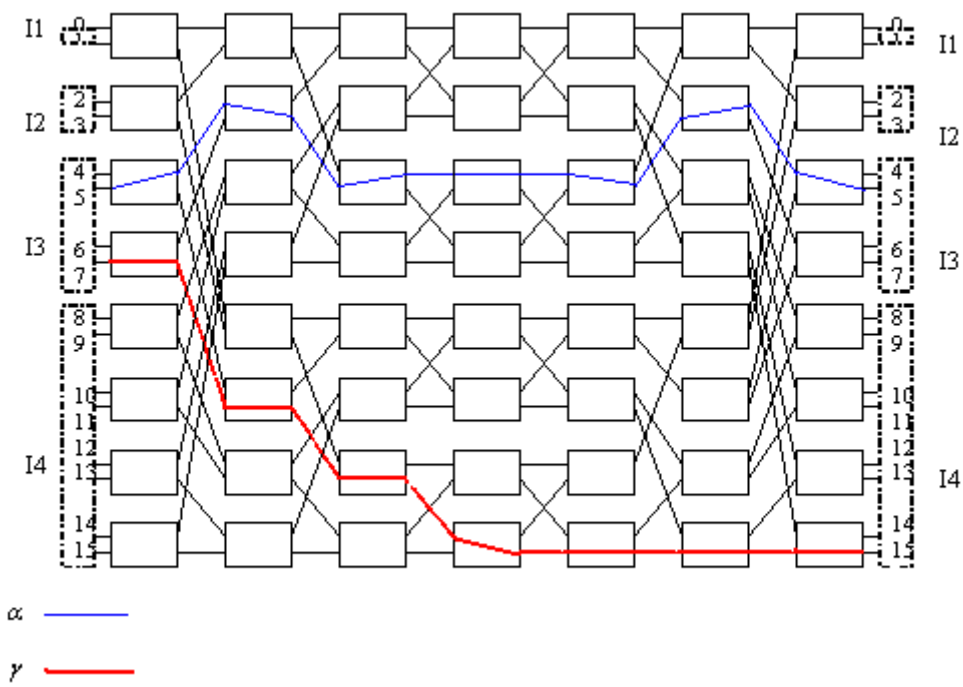

Figure 4.4. Illustration of paths for calculating α and γ

This therefore proved the following theorem 3:

Theorem 3: In a VSOBN(*m,n*) network, the nonblocking probability is $P(C_{I1}\ C_{11})\ P(C_{O1}\ C_{11})\ P(C_{I2}\ C_{22})\ P(C_{O2}\ C_{22})\ldots\ P(C_{In}\ C_{nn})\ P(C_{On}\ C_{nn})\ /\ P(C_{11})\ P(C_{22})\ldots\ P(C_{nn})$; in which $P(C_{kk})$ =

$\binom{2^{k-1}}{C_{kk}} \alpha_k^{C_{kk}} (1-\alpha_k)^{2^{k-1}-C_{kk}}$, α_k is the probability that a connection from input group k going to the out put groups k. So α_k is $r * (2^{k-1})/(N-1)$ in this case, $P(C_{Ik}\ C_{kk})$ is the probability of C_{Ik} connections from input group K to output groups K and above while there are C_{kk} connections from input group K to output group K.

$$P(C_{Ik}\ C_{kk}) = \binom{2^{k-1}}{C_{kk}} \alpha_k^{C_{kk}} * \binom{2^{k-1}-C_{kk}}{C_{Ik}-C_{kk}} * \gamma_k^{C_{Ik}-C_{kk}} * (1-\alpha_k-\gamma_k)^{2^{k-1}-C_{Ik}}$$

In which α_k is the probability that a connection from input group k going to the out put group k. So α_k is $r * 2^{k-1}/N-1$ in this case. γ_k is the probability that a connection from input group k going to the out put groups above k and γ_k is $r * (N-2^k / N-1)$, $P(C_{Ok}\ C_{kk})$ is the probability of C_{Ok} connections from input group K to output groups K and above while there are C_{kk} connections from input group K to output group K.

$$P(C_{Ok}\ C_{kk}) = \binom{2^{k-1}}{C_{kk}} \alpha_k^{C_{kk}} * \binom{2^{k-1}-C_{kk}}{C_{Ok}-C_{kk}} * \gamma_k^{C_{Ok}-C_{kk}} * (1-\alpha_k-\gamma_k)^{2^{k-1}-C_{Ok}}$$

In which α_k is the probability that a connection from input group k going to the out put group k. So α_k is $r * 2^{k-1}/N-1$ in this case. γ_k is the probability that a connection going into output group k coming from the input groups above k and γ_k is $r * (N-2^k / N-1)$ ◘

Now we show some analytical results based on the formulas obtained in the section. Figure 14 show the blocking probability with different number of planes. From the figure, it can be seen that when n =16, the blocking probability is very close to 0 even the number of planes is 5. Hence, in most practical cases, we do not need a full number of planes to guarantee non-blocking, and our analysis will show what the probability of blocking if a smaller number of

planes is used. Figure 4.5 can be used as guidance when a designer makes a trade-off between performance and cost.

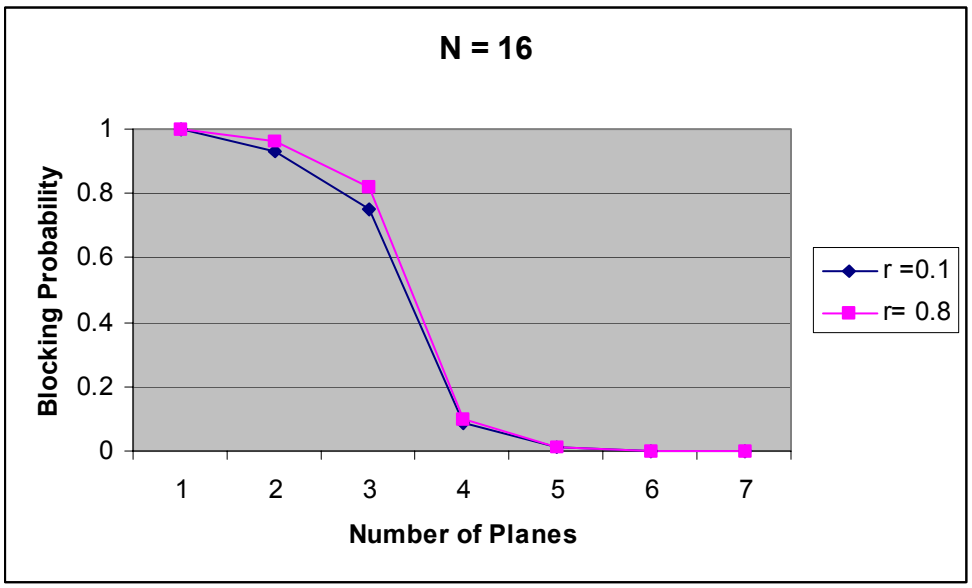

Figure 4.5. Blocking Probability for Traffic Rate r = 0.8 and 0.1

Chapter 5

WDM Routing in Benes network

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 [31]. In this dissertation, we study the non-blocking properties of WDM Benes networks under crosstalk free constraints.

In [49], it has been proved that a permutation can be decomposed into two semi-permutations and each of the two semi-permutations can be routed separately through the Benes network without crosstalk. However, it is not the same as routing the two semi-permutations in the Benes network with two different wavelengths at the same time. In our study, we assume each SE has only two states, either bar or state, therefore, the two semi-permutations might have conflicts in setting the states of the SEs.

Our WDM routing algorithm is composed of two phases, namely route constructing phase and wavelength assigning phase. In route constructing phase, we are given a permutation and the output of this phase is a setup of the switch elements; in wavelength assigning phase, the network setup from the previous phase will be taken and a wavelength assignment will be made on this network so that cross talk will not occur.

5.1 Route Construction

A Benes network exhibits a symmetric topological structure [26]. As shown in Fig. 5.1, the Benes network can be considered as a cascaded combination of an Banyan network (a baseline network in specific) and a reverse Banyan network with the two joint stages overlapping. Since there is only a unique path between any input and any output in both of the

baseline and the reverse baseline networks, each path in the Benes network actually consists of two subpaths, one in the baseline network and the other in the reverse baseline network. These two subpaths can be joined at any one of the central modules to form a complete path. For any complete path in the Benes network, the forward subpath from the input, and the reverse subpath from the output, to the middle stage must have the same binary "destination address," which indicates a particular central module for the connection.

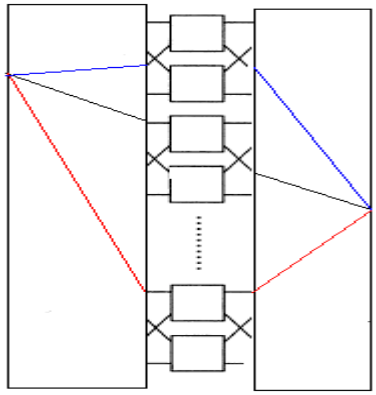

Figure. 5.1 Benes network exhibits symmetry respect to the center stage

In [6] [20], a parallel routing algorithm is proposed, which recursively routes the Benes network from outer layer to inner with N/2 processors. But in optical network, crosstalk elimination or reduction is another important issue besides admissibility. This dissertation proposes a way to reduce contentions between connections at the routes setup phase.

5.1.1 Motivation

Benes network is a rearrangeably non-blocking network, which implies that Benes network can route any permutation provided that the rearrangement of the existing connections is allowed. In other words, Benes network can route off-line permutation.

It would be interesting to know how many different possible routs a connection can take on.

Theorem 5.1 In N x N Benes network, there are N/2 different paths for each connection.

Proof: Because a Benes network exhibits a symmetric topological structure. As shown in Fig. 5.1, the Benes network can be considered as a cascaded combination of a Banyan network (a baseline network in specific) and a reverse Banyan network with the two joint stages overlapping. Therefore, each connection can choose to from view point of input and output can choose to meet at one of the N/2 center stage SEs.

Corollary 5.1. In N x N Benes network, there are at most $((N/2)!)^2$ connection patterns for each permutation.

Proof: Because each connection can choose to meet at any of the available center stage, each center stage SE can accommodate 2 connections. For the first connection, there are N/2 such options; for the second connection, there are also N/2 such options; for the third connection, there are N/2 – 1 options, and for the fourth connection, there are N/2-1 option, and so on. Therefore, the total possible combinations are N/2 * N/2 * (N/2 – 1) * (N/2 – 1)*…*1*1 = $((N/2)!)^2$. ◘

It should be noted that the actual admissible connections are much less than what Corollary 5.1 provides, due to the following reason.

There are total N/2 * (2 logN –1) SEs in a Benes network and each SEs can be in one of the two states, therefore, the total number of different switch settings for the Benes network would be $2^{N/2 * (2 \log N - 1)}$, according to Sterling's approximation [61], this is much less than the upper bound Corollary 5.1 provides.

Nonetheless, the fact that there are multiple routes available motivates us to pick a better route in terms of less cross talk.

Wavelength assignment problem can often be formulated as a graph-coloring problem. In the conflict graph [32], the number of the nodes is the size of the network. Each node represents a connection. If two nodes have conflict during routing, they are connected using an edge.

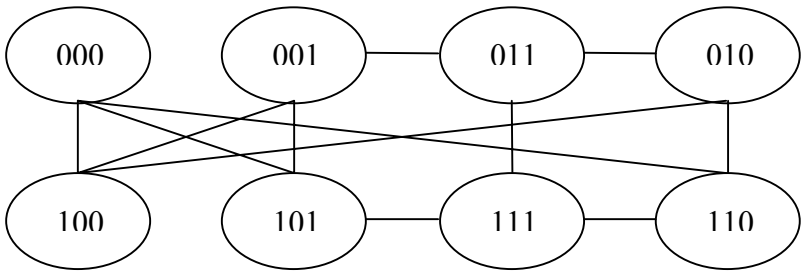

Figure5.2. Conflict graph for a given source-destination permutation

Wavelength assignment problem now becomes the problem of coloring the conflict graph such that no adjacent nodes share the same color. [32][49] The chromatic number of a graph *G* is the smallest number of colors $\chi(G)$ needed to color the vertices of *G* so that no two adjacent vertices share the same color [60]. But to find the chromatic number is NP-Complete [7] [9].

Consider the following permutation

$$\begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 4 & 5 & 1 & 6 & 0 & 2 & 7 & 3 \end{pmatrix}$$

This permutation can be realized by several different SE settings. We provide two of them here.

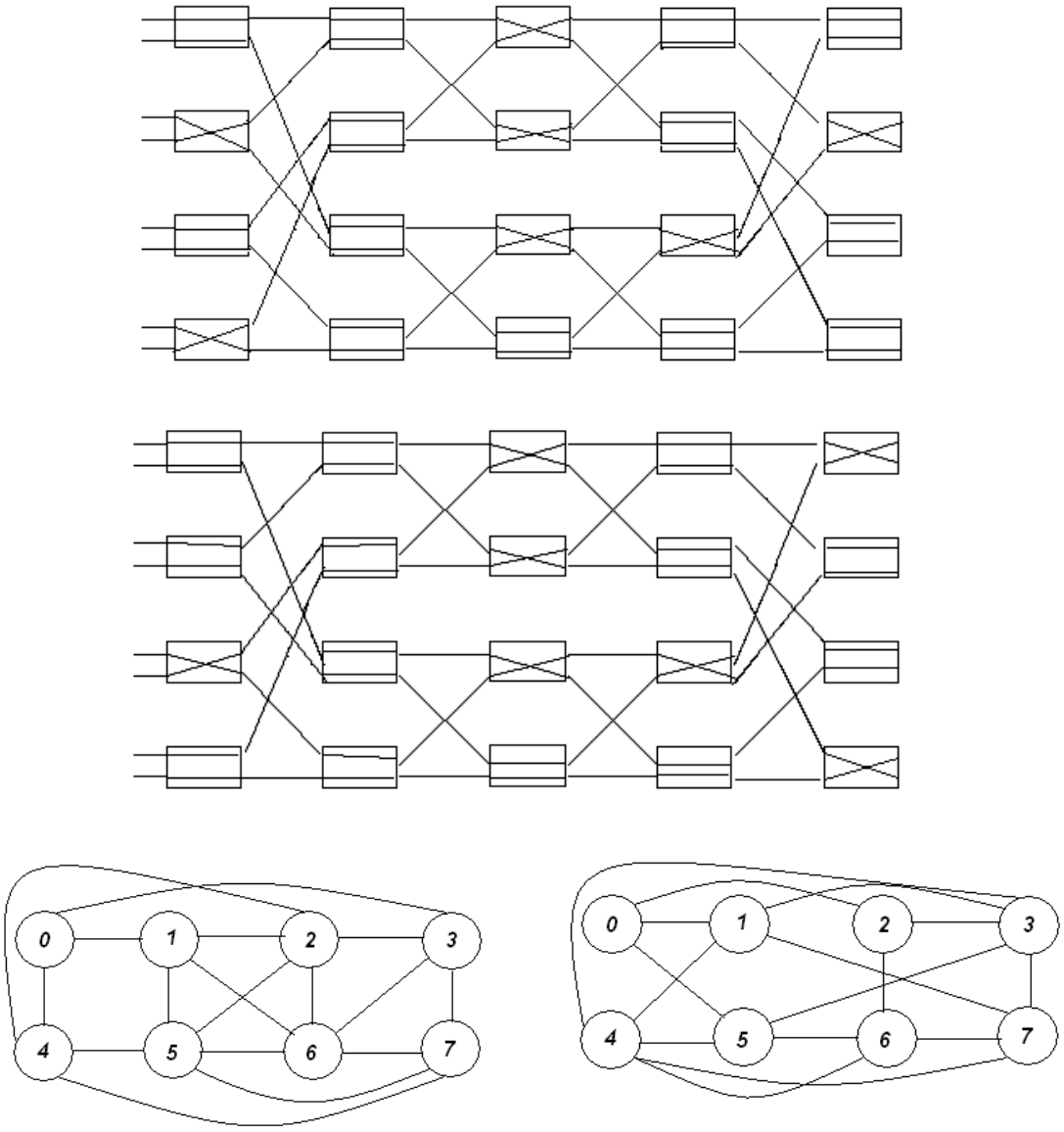

Figure 5.3. Two of the possible SE settings for a permutation and their conflict graphs

As can be seen, the two settings have two different conflict graphs, and therefore have the potential of deriving different graph coloring.

52 Wavelength Awareness Routing

According to Brooks Theorem [7], the chromatic number of a graph is at most the maximum vertex degree Δ, unless the graph is complete or an odd cycle. In [20], it is proved that the chromatic number of any conflict graph with node size N is less than 2logN-1.

In this research, we propose a greedy algorithm, in hopes that in the route setup phase, the conflict graph is so constructed that the number of colors used can be reduced. The idea of the greedy algorithm is illustrated as follows.

Suppose C_1 is conflicting with C_2 at this moment.

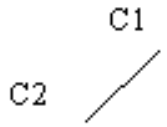

Figure 5.4. C_1 is conflicting with C_2

Suppose a new connection C3 can be connected with C1 or C2, since the contention number of C1 and C2 are both 1, C3 can choose to conflict with either of them.

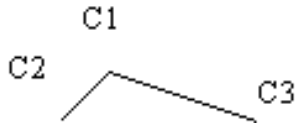

Figure 5.5. C_3 is chosen to be conflicting with C_1

Suppose a new connection C4 can be connected with C1 or C2, since the degree of C1is 2 and that of C2 is 1, C4 will choose to conflict with C2.

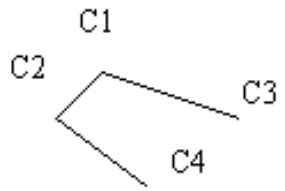

Figure 5.6. C_4 is chosen to be conflicting with C_2

Suppose in a later SE connection C4 will be connected with C1 or C3, since the degree of C1is 2 and that of C3 is 1, C4 will choose to conflict with C3, instead of C1, therefore prevented a clique of size 3 from emerging.

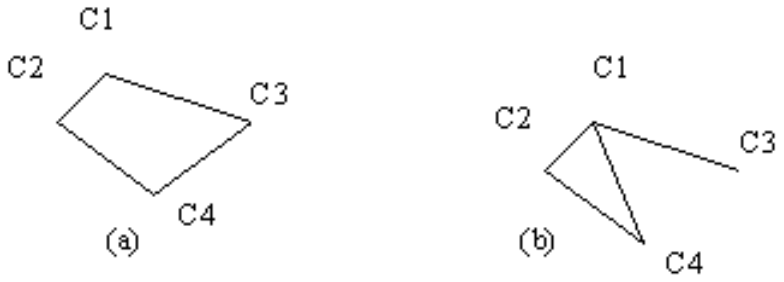

Figure 5.7. C_4 is chosen to be conflicting with C_3 instead of C_1

It should be pointed out that the greedy algorithm is not optimal, as the following example shows:

Suppose in (a), C5 can be chosen to conflict with either C1 or C2, greedy algorithms will always choose C1 with the result being (b) other than C2 with the result being (C) because the degree of C1 is l while that of C2 is 2. However, the chromatic number of (b) is 3 since it is a odd circle while that of (c) is 2.

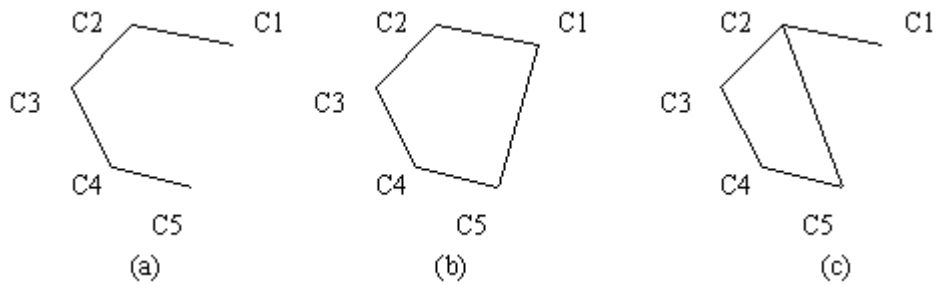

Figure 5.8. Greedy algorithms is not optimal

In [6], the connections are partitioned into several equivalent classes. It is interesting to know how many such equivalent classes could there be.

Theorem 6.2 There are at most N/2 equivalent classes in a N x N Benes network.

Proof: Given a N x N Benes network, the following permutation

$$\begin{pmatrix} 0 & 1 & 2 & \dots & N\text{-}1 \\ 0 & 1 & 2 & \dots & N\text{-}1 \end{pmatrix}$$

Will provide the biggest number of equivalent classes. And this permutation corresponds to $a_0 = b_0$, $a_1 = b_1$, $a_2 = b_2 \dots a_{N/2-1} = b_{N/2-1}$, therefore the equivalent class will be $\{b_0, b_1, b_2 \dots b_{N/2-1}\}$.

The idea of the wavelength-aware routing is that the algorithm anticipates the number of conflicts in the next stage, and set up the states of SEs in the current stage.

The algorithm is described as follows. Since each path in Benes network consists of 2logN-1 SEs, therefore, each connection can conflict with at most 2logN-1 other connections. Connections are denoted by their originating inputs.

Given a permutation π, each connection maintains an array C[ ] of size 2logN-1, which records the connections with which it conflicts. Since there are two connections going through each SE, each SE maintains an array SE[ ] of size 2, which records the connections going through this SE. Each SE also maintains a variable SE_state that records the state of that SE.

Input: A permutation

Output: A set up of the Benes network

- Derive the equations discussed in Chapter 4 according to symmetric routing constraint and internally conflict-free constraint
- Sort the equivalent classes into descending order, using the number of members within each equivalent class as index.
- For each of the *k* equivalent classes bi from 0 to k-1

  Do

```
        Pick a representative from the equivalent class
        If i = 0,    //the first equivalent class
            set the state of the representative to be either 0 or 1,
            for each of the members of the equivalent class
                set SE_state for the current SE
                update SE [ ] for the current SE
                update SE [ ] for the SE the connection leads to
                update C[ ] for each connection within this equivalent class
```

```
else
    set the state of the representative to be 0
        for each of the members of the equivalent class
        update SE [ ] for the current SE,
        update SE [ ] for the SE the connection leads to
        update C[ ] for each connection within this equivalent class
        record the largest number N1 of conflicts from C[ ]
    set the state of the representative to be 1
        for each of the members of the equivalent class
        update SE [ ] for the current SE,
        update SE [ ] for the SE the connection leads to
        update C[ ] for each connection within this equivalent class
        record the largest number N2 of conflicts from C[ ]
compare N1 with N2, pick the state which provides the smaller conflicts and
set SE_state to that state
```

- After that the outer layer of SEs are so set. Then the algorithm will set the inner layers recursively.
- At the last stage, which is the center of Benes network, each connection will also record the SE index it passes.

5.3 Wavelength assignment

Once the routes are decided, which means the status of the SEs are decided, the next step is to assign the routes with different wavelengths. This is equivalent to a vertex cover problem and is known to be NP-Complete [7][9].

To construct the conflict graph, we propose a modified version of the original windows method. Our windows method works as follows.

Take the permutation in Figure 5.9 as an example, and suppose the resulting SEs setting is as in Figure 5.10

$$\begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 4 & 5 & 1 & 6 & 0 & 2 & 7 & 3 \end{pmatrix}$$

Figure 5.9. A permutation

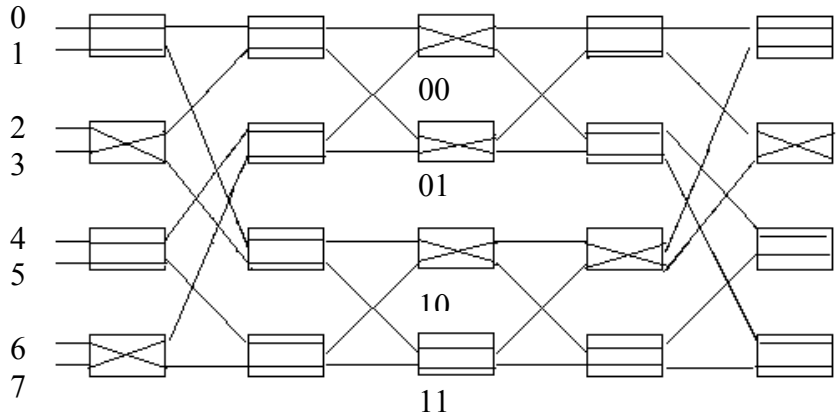

Figure 5.10. The network setting for the permutation in Figure 5.9

From the above diagram, we know that connection 0 records 00 as the center SE index, connection 1 record 10 as the center SE index, connection 2 records 11 as the center SE index, connection 3 records 01 as the center SE index, connection 4 records 00 as the center SE index,

connection 5 records 10 as the center SE index, connection 6 records 11 as the center SE index, connection 7 records 01 as the center SE index.

The modified windows method has two parts, one part is from the input to the center stage, and the second part is from the output back to the center stage. Notice that the topology of Omega network and Baseline network is different. The Omega network has a shuffle exchange property while Baseline network does not. The first part of the modified window method is as follows.

In step 1, we first list the input port number, which is also the connection index, concatenated with its corresponding center SE index.

Conn 0:	0 0	0
Conn 1:	0 0	1
Conn 2:	0 1	0
Conn 3:	0 1	1
Conn 4:	1 0	0
Conn 5:	1 0	1
Conn 6:	1 1	0
Conn 7:	1 1	1

Figure 5.11 Step 1 in the first part

In the above example connection 0 and 1 in step 1 (window 0) have the same bit pattern of “00” inside the window and hence have a conflict. The bit patterns in the above example can be any of the four combinations of “00”, “01”, “10”, “11”, and hence are shaded using different colors.

Step 2: The new input port address in stage 2 is decided as substituting the least significant bit of the input port number from the previous input port number by the XOR of the state of the SE and the least significant digit of the previous input port and then right cycle-shift it. For example, consider connection 2 in the following Benes network, the input port address of the connection in stage 1 is 010, and the stage state is 1. In order to derive the input port number at the second stage, we substitute the least significant bit of 010, which is 0, with the state of the XOR of 0 and SE, which is 1, and we will get 011. We then right cyclically shift the new address we then get the address of connection 2 in stage 2, which is 101.

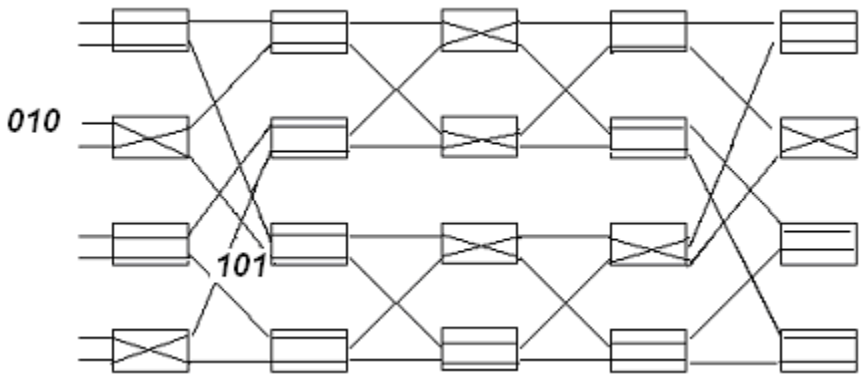

Figure 5.12. Address change in Baseline network

Therefore, the window for the connections in stage 2 is shown below as

Conn 0:	0 0	0
Conn 1:	1 0	0
Conn 2:	1 0	1
Conn 3:	0 0	1
Conn 4:	0 1	0
Conn 5:	1 1	1
Conn 6:	1 1	1
Conn 7:	0 1	1

Figure 5.13 Step 2 in the first part

Different from the original window method, where the window moves from left to right; in our modified window method, the window will not move and it always covers the first two bits of the input address of the connection at each stage.

Step 3: Similarly, the window for the connections in stage 3 is shown below as

Conn 0:	0 0	0
Conn 1:	1 0	0
Conn 2:	1 1	0
Conn 3:	0 1	0
Conn 4:	0 0	1
Conn 5:	1 0	1
Conn 6:	1 1	1
Conn 7:	0 1	1

Figure 5.14 Step 3 in the first part

In the second part, we list the output ports number followed by the connection they belong but with all the digits reversed, for example 001 will be reversed as 100, as follows. And in this second part, stage1 refers to the first stage from output to input, stage 2 refers to the second stage from output to input, and so on.

Step 1:

0	0 0	Conn 4
1	0 0	Conn 2
0	1 0	Conn 5
1	1 0	Conn 7
0	0 1	Conn 0
1	0 1	Conn 1
0	1 1	Conn 3
1	1 1	Conn 6

Figure 5.15 Step 1 in Second Part

Step 2: The new input port address in stage 2 is decided as substituting the least significant bit of the output port number from the previous output port number by the XOR of the state the current SE and the least significant bit of the previous output port number and then right cycle-shift it. And the window for step 2 is as below,

0	0 0	Conn 4
0	0 1	Conn 2
1	0 1	Conn 5
1	0 0	Conn 7
0	1 0	Conn 0
1	1 1	Conn 1
1	1 0	Conn 3
1	1 1	Conn 6

Figure 5.16 Step 2 in Second Part

Step 3: Similarly, the window for step 3 is as follows,

0	0 0	Conn 4
0	1 1	Conn 2
0	0 1	Conn 5
0	1 0	Conn 7
1	0 0	Conn 0
1	0 1	Conn 1
1	1 0	Conn 3
1	1 1	Conn 6

Figure 5.17 Step 3 in Second Part

And eventually the conflict graph will be as follows

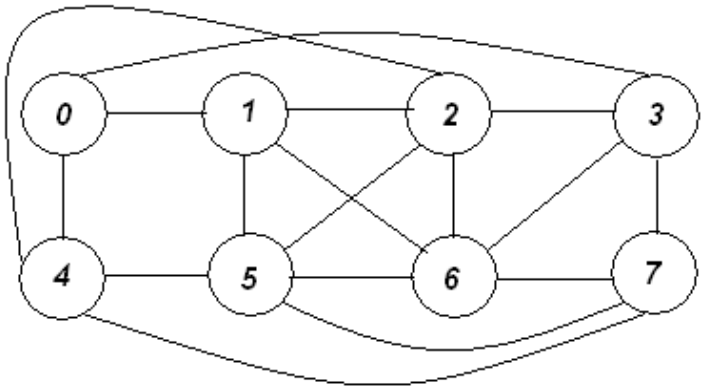

Figure 5.18 Conflict graph – result of the modified windows method

Chapter 6

Genetic Algorithms

Since Wavelength Assignment is an NP-Complete problem, heuristic solutions are often desirable. Among other AI techniques, genetic algorithms are applied to search heuristic solution to many problems, such as Traveling Salesman Problem. In this dissertation, we apply genetic algorithms in wavelength assignment.

6.1 Introduction to Genetic Algorithms

Looking at the world around us, we see a staggering diversity of life. Millions of species, each with its own unique behavior patterns and characteristics, abound. Yet, all of these plants and creatures have evolved, and continue evolving, over millions of years. They have adapted themselves to a constantly shifting and changing environment in order to survive. Those weaker members of a species tend to die away, leaving the stronger and fitter to mate, create offspring and ensure the continuing survival of the species. Their lives are dictated by the laws of natural selection and Darwinian evolution. And it is upon these ideas that genetic algorithms are based.

A genetic algorithm starts with a set of solutions, which are represented by chromosomes. Those solutions are called populations. Solutions from one population are taken and used to form a new population. This is motivated by a hope, that the new population will be better than the old one. The new solutions (offspring's) to form a new population are selected based on the fitness of the solution, i.e. the more suitable they are the more chances they get to reproduce. This is repeated generation by generation until some condition is satisfied such as the population size gets to the limit or the improvement of the best solution is good enough for the research or no further improvement is possible.

The idea of survival of the fittest is of great importance to genetic algorithms. GAs use what is termed as a fitness function in order to select the fittest string that will be used to create new, and conceivably better, populations of strings. The fitness function takes a string and assigns a relative fitness value to the string. The method by which it does this and the nature of the fitness value does not matter. The only thing that the fitness function must do is to rank the strings in some way by producing the fitness value. These values are then used to select the fittest strings. The concept of a fitness function is, in fact, a particular instance of a more general AI concept, the objective function.

The following is a flow chat of the GAs[27].

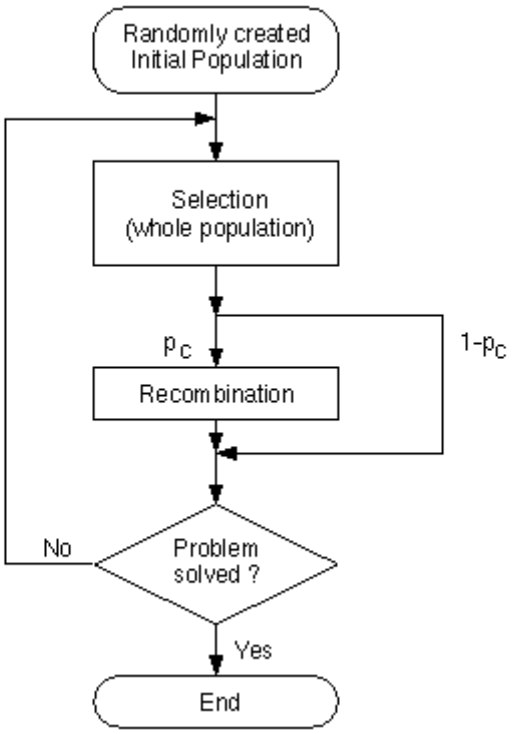

Figure 6.1 Genetic Algorithms

6.2 Operators of Genetic Algorithm

There are three major operations in genetic algorithm, namely crossover, mutation and selection.

Crossover

Crossover selects genes from parent chromosomes and creates a new offspring. There are many ways to do crossover such as single-point crossover, double point crossover, etc. The simplest way is the single-point crossover, which is to choose some crossover point randomly and everything before this point is copied from a first parent and then everything after that point is copied from the second parent. It looks like this:

Chromosome 1: 00001111111

Chromosome 2: 1 1111111000

The arrow shows the selected point for crossover. So the new offspring are:

Chromosome 1: 00001111000

Chromosome 2: 1 1 111111111

Different crossover methods can be used on different problems. The specific crossover made for a specific problem may improve the performance of the Genetic Algorithm.

Mutation

Mutation takes place after the crossover is performed. The reason to use mutation is to prevent from falling all solutions in population into a local optimum of solved problem [14]. Mutation changes randomly the new offspring. In binary encoding, we can switch a few randomly chosen a few bits from 1 to 0, or from 0 to 1.

Offspring from the crossover:

Chromosome 1: 00001111000

Chromosome 2: 1 1 111111111

After Mutation:

Chromosome 1: 01001111000

Chromosome 2: 1 1 111110111

Selection

When we have many solutions, we need to select some of them to produce the offspring in pairs. The selection is to try the parents on the fitness function to get the better fitting parents. Then use the selected parents to be the parents of the next generation.

6.3 Parameter of Genetic Algorithm

There are two basic parameters in Genetic Algorithm, which are crossover probability and mutation probability.

Crossover probability is the parameter to say how often the crossover will be performed. If the crossover probability is 0%, this means no crossover is performed, the offspring is exact the copy of the parent. If the crossover probability is 100%, then all the offspring is made by crossover.

Mutation probability is the parameter to control how often the parts of the chromosome will be mutated. If there is no mutation, offspring is taken after crossover without any change. If mutation is performed, part of the chromosome is changed. If mutation probability is 100%, the whole chromosome is changed. Mutation is used to prevent falling Genetic Algorithm falling into local extreme, but it should not occur very often, because it will change the Genetic Algorithm to random search. This is also approved by the results of our testing in this research.

Population size is the parameter to determine how many chromosomes are in population for one generation. The population size cannot be too small or too large because too small size of

population give Genetic Algorithm less search space, the result will not be improved much while too big population size will slow down the execution. Even if it will get better result, if the performance is too bad, it is not suitable. Researches show that after some limit (which depends mainly on encoding and the problem), it is not useful to increase the population size because it makes solving the problem much slower [15].

Generation [15] is an important parameter in genetic algorithm. The more generations a test case has the better chance that the genetic algorithm will find a better solution. In each generation, we use the fitness function to select the better solutions to be the members of the next generation. Then, this new generation will produce its next possible better offspring. With more generations a genetic algorithm can produce a better solution. However, as the generation size gets bigger and bigger, the test cases take more and more time to execute. The performance is related to the number of generations.

6.4. A Modified GAs

As observed by Whitley, " It can be argued that there are only two primary factors (and perhaps only two factors) in genetic search: population diversity and selective pressure. These two factors are inversely related. Increasing selective pressure results in a faster loss of population diversity. Maintaining population diversity offsets the effect of increasing selective pressure. In some sense this is just another variation on the idea of exploration versus exploitation that has been discussed by Holland and others."[52] In John Holland's canonical generic algorithms, fitness is defined by fi/f, where fi is the evaluation associated with string i and f is the average evaluation of all the strings in the population. This is known as fitness proportional reproduction. There can be a couple of problems with fitness proportional

reproductions. First selection can be too strong in the first few generations: too many duplicates are sometimes allocated to very good individuals found early in the search. Second, as individuals in the population improve over time, there tends to be less variation in fitness, with more individuals being close to the population average. As the population average fitness increases, the fitness variance decreases and the corresponding uniformity in fitness values causes selective pressure to go down. In this case, the search begins to stagnate.

Other methods to sample a population are based on introducing artificial weights: chromosomes are selected proportionally to their rank rather than actual evaluation values [51] [52] [53]. These methods are based on a belief that the common cause of rapid (premature) convergence is the presence of super individuals, which are much better than the average fitness of the population. Such super individuals have a large number of offspring and (due to the constant size of the population) prevent other individuals from contributing any offspring tin the next generations. In a few generations a super individual can eliminate desirable chromosomal material and cause a rapid convergence to (possibly local) optimum. There are many methods to assign a number of offspring based on ranking. In [51], a linear function and a parameter are defined:

$$\text{Prob (rank)} = q - (\text{rank} - 1)r,$$

Or a non-liner function,

$$\text{Prob (rank)} = q(1-q)^{\text{rank}-1}$$

In this research, GAs is further modified based on their relative fitness and their relative distribution [54]. Both functions return the probability of an individual ranked in position rank (rank = 1 means the best individual, rank = pop_size the worst one) to be selected in a single

selection. Such approaches, though shown to improve genetic algorithm behavior in some cases, have some apparent drawbacks. The major problem is they ignore the information about the relative evaluations of different chromosomes.

6.4.1 Take Relative Fitness Information into Account

The modified GAs [63] takes the chromosome's relative fit into account, and the selective pressure is modified by the following,

$Prob' = Prob * 2/(1+\exp(f-f_i))$

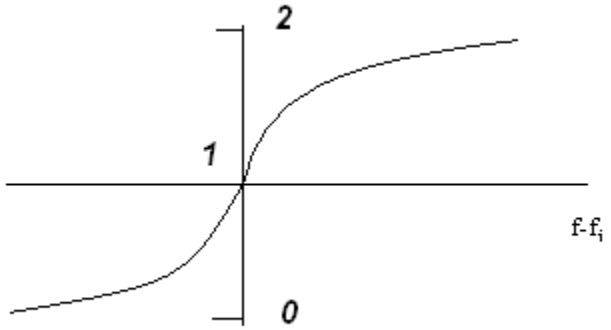

Figure 6.2 Adjust the selective pressure according to relative fitness

The idea of this adjustment is to keep the selective pressure of those chromosomes which have the average fitness intact; to increase the selective pressure of those chromosomes which have the above average fitness and to decrease the selective pressure of those chromosomes which have the below average fitness.

6.4.2 Take Fitness Distribution Information into Account

The population is divided into K intervals, the number of chromosomes falling into the respecting interval is counted. Then the selective pressure from regular ranking function **Prob** (or from **Prob'**) is adjusted according to their distribution.

Probj'' = Prob * 2/(1+exp(d-dj)) in which j is from 1 to K; d is the average NO of chromosomes for the whole range, dj is the average chromosome count for interval j.

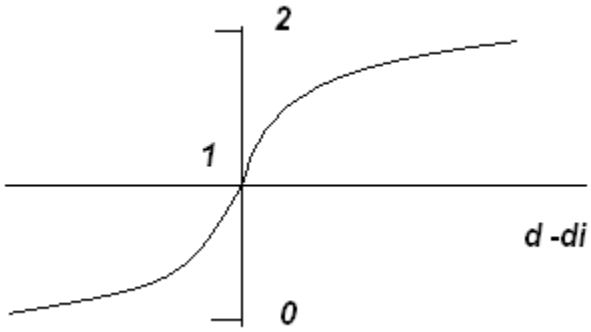

Figure 6.3 Adjust the selective pressure according to chromosome distribution

The idea of this adjustment is to keep the selective pressure of those chromosomes which have the average distribution intact and to increase the selective pressure of those chromosomes which have the above average distribution and to decrease the selective pressure of those chromosomes which have the below average distribution.

6.5 Genetic Algorithms in This Thesis

The chromosomes in this thesis are a list of digits, each digit represent a connection and a color. For example, in a 16 x 16 Benes network, one of the chromosomes could be 2015304201234513, each digit represents a connection, and the different digit values are used to represent different wavelength assigned to the connection. There are several parameters that have

significant effect on the performance of GA, which are cross over rate, initial population size, number of generations and objective function, and mutation rate.

6.5.1 Crossover Rate

In order for GA to be effective, the crossover rate must be big enough, so that there will be enough offspring, and we have found that the cross over rate should be as large as enough in this research. Therefore, we choose our cross over rate to be 70% and 100%

6.5.2 Initial Population Size

The initial population size should be large enough so that GA can search a wide area of sample space, instead of being confined in small area and quickly falling in local optimal. Through our experiments, we have found that for a network of size 8 x 8, initial population size of 100, for a network of size 16 x 16, initial population size of 500, for a network of size 32 x 32, initial population size of 1000 to be adequate.

6.5.3 Number of Generations

The number of generations should also be large enough so that GA can evolve over time, without that, GA will settle with sub-optimal value quickly. Through our experiments, we have found that for a network of size 8 x 8, the number of generations of 100, for a network of size 16 x 16, number of generations of 500 and for a 32 x 32 network, number of generation of 1000 to be adequate.

6.5.4 Objective Function

There are two factors we should consider when design the objective function. The obvious one is the number of colors, but that alone is not effective. The reason that we cannot consider only the number of colors when evaluate the chromosomes is that since the cross over

and mutation both are random, there will inevitably be many offspring that are wrongly colored. If these offspring are eliminated from the sampling pool all together, we will not have enough population, and the GA will quickly lead to a situation where there is no properly colored offspring at all or the nodes are colored all in different color. Therefore, the object function must also include the measurement of the number of wrongly colored nodes. In this research, we will reward the nodes, which have different color with their neighbors, by increasing their fitness score and penalize those having the same color by decreasing their fitness score. The overall fitness value would be the fitness score derived from taking the wrongly colored nodes into account divided by the number of colors used, therefore, the fitness function penalizes wrongly colored nodes as well as excessive number of colors.

6.5.5. Mutation Rate

In GA, mutation rate is usually preferred to be relatively small, and we choose 0.1, 0.05 and 0.2 for comparison purposes in this research.

6.6. Results and Discussion

The result of genetic algorithm provides better results in terms of the number of wavelengths needed compared with [62]. For example, consider the following permutation in a 8 x 8 Benes network,

$$\begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 0 & 2 & 1 & 4 & 3 & 7 & 5 & 6 \end{pmatrix}$$

Figure 6.4. A permutation

In [62], the resulting conflict graph is as follows, as can be seen in Figure, it needs 6 wavelengths to route the connection.

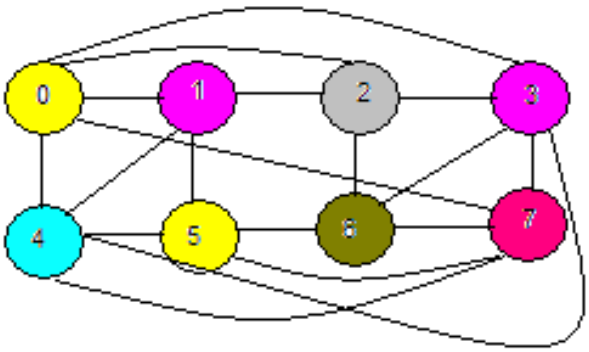

Figure 6.5 The graph coloring of the above permutation in [62]

While the result of Genetic algorithms is as follows, and it needs only 5 wavelengths.

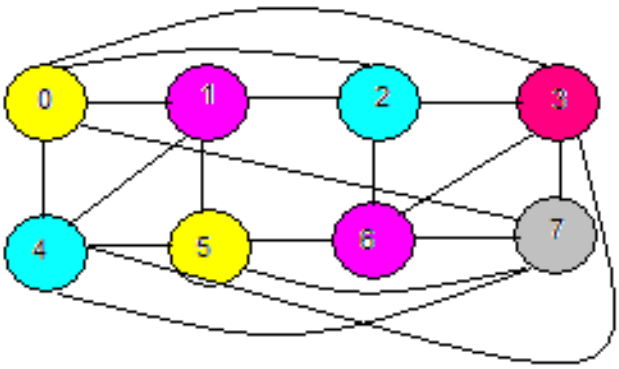

Figure 6.6 The graph coloring of the above permutation using GAs

We then have tested our wavelength awareness routing algorithm and that proposed in [36].

The following tables show the different test cases.

Table 6.1 is the number of wavelength required in 8 x 8 Benes network, with cross rate = 0.7, Figure 6.7. is a comparison between the original algorithm with wavelength-aware routing algorithm.

Table 6.1 Number of wavelength required in 8 x 8 Benes network, cross over rate = 0.7

No. of Tests	mutation	crossover	original algorithm	wavelength aware
100	0.05	0.7	3.74	3.73
100	0.1	0.7	3.74	3.74
100	0.4	0.7	3.74	3.74

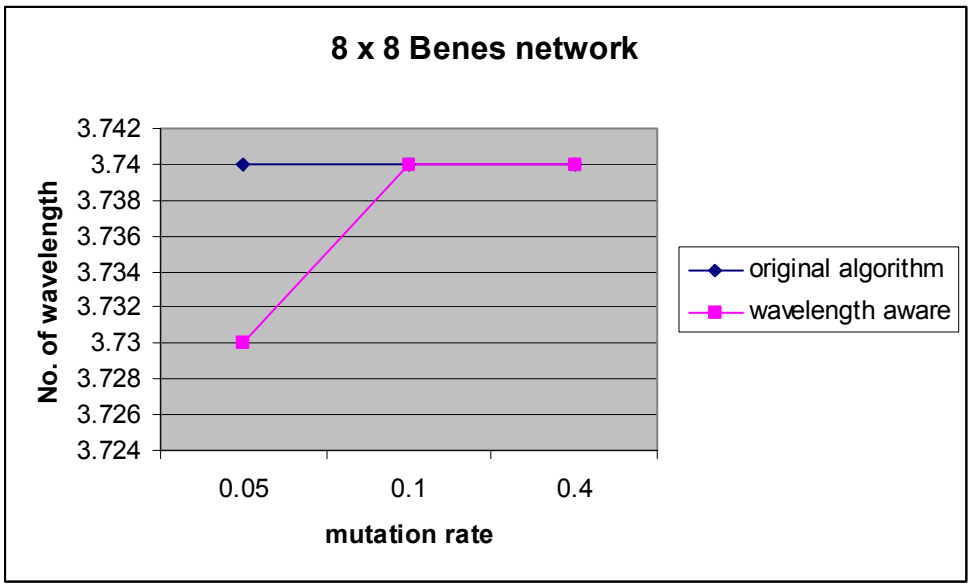

Figure 6.7 Number of wavelength required in 8 x 8 Benes network, cross over rate = 0.7

Table 6.2 is the number of wavelength required in 8 x 8 Benes network, with cross rate = 1, Figure 6.8. is a comparison between the original algorithm with wavelength-aware routing algorithm.

Table 6.2 Number of wavelength required in 8 x 8 Benes network, cross over rate = 1

No. of Tests	mutation	crossover	original algorithm	wavelength aware
100	0.05	1	3.74	3.73
100	0.1	1	3.74	3.73
100	0.4	1	3.74	3.74

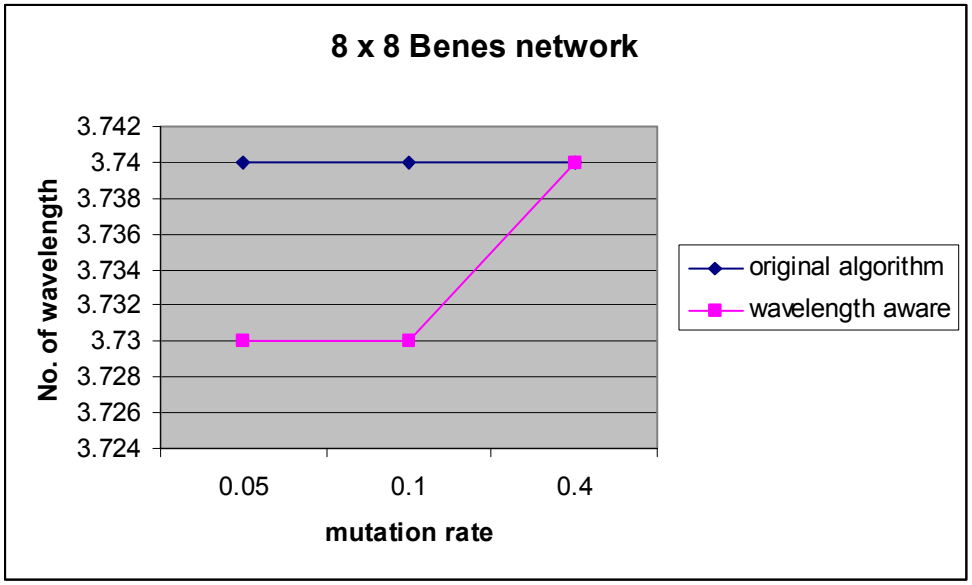

Figure 6.8 Number of wavelength required in 8 x 8 Benes network, cross over rate = 1

Table 6.3 is the number of wavelength required in 16 x 16 Benes network, with cross rate = 0.7, Figure 6.9. is a comparison between the original algorithm with wavelength-aware routing algorithm.

Table 6.3 Number of wavelength required in 16 x 16 Benes network, cross over rate = 0.7

No. of Tests	mutation	crossover	original algorithm	wavelength aware
100	0.05	0.7	5.87	5.71
100	0.1	0.7	5.87	5.71
100	0.4	0.7	5.87	5.71

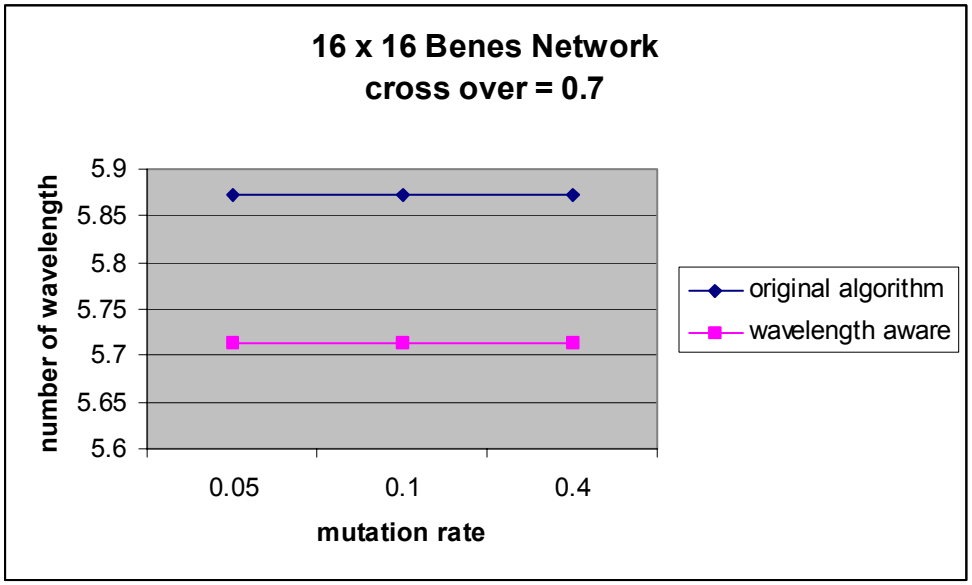

Figure 6.9. No. of wavelength required in 16 x 16 Benes network, cross over rate = 0.7

Table 6.4 is the number of wavelength required in 16 x 16 Benes network, with cross rate = 1, Figure 6.10. is a comparison between the original algorithm with wavelength-aware routing algorithm.

Table 6.4 Number of wavelength required in 16 x 16 Benes network, cross over rate = 1

No. of Tests	mutation	crossover	original algorithm	wavelength aware
100	0.05	1	5.87	5.71
100	0.1	1	5.87	5.71
100	0.4	1	5.87	5.72

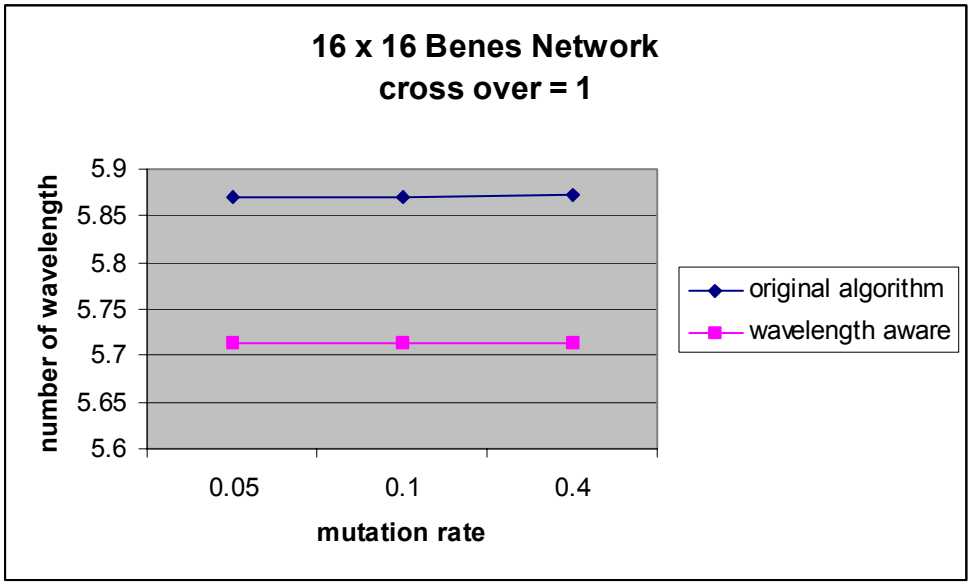

Figure 6.10 No. of wavelength required in 16 x 16 Benes network, cross over rate = 1

Table 6.5 is the number of wavelength required in 32 x 32 Benes network, with cross rate = 0.7, Figure 6.11. is a comparison between the original algorithm with wavelength-aware routing algorithm.

Table 6.5 Number of wavelength required in 32 x 32 Benes network, cross over rate = 0.7

No. of Tests	mutation	crossover	original algorithm	wavelength aware
100	0.05	0.7	7.55	7.21
100	0.1	0.7	7.56	7.21
100	0.4	0.7	7.56	7.22

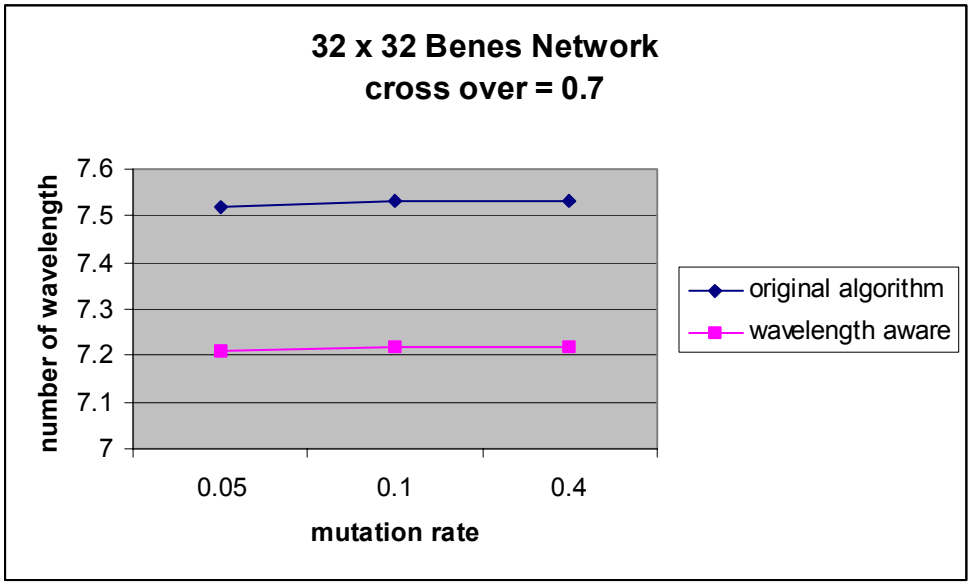

Figure 6.11 No. of wavelength required in 32 x 32 Benes network, cross over rate = 0.7

Table 6.6 is the number of wavelength required in 32 x 32 Benes network, with cross rate = 1, Figure 6.12. is a comparison between the original algorithm with wavelength-aware routing algorithm.

Table 6.6 Number of wavelength required in 32 x 32 Benes network, cross over rate = 1

No. of Tests	mutation	crossover	original algorithm	wavelength aware
100	0.05	1	7.52	7.19
100	0.1	1	7.53	7.2
100	0.4	1	7.53	7.2

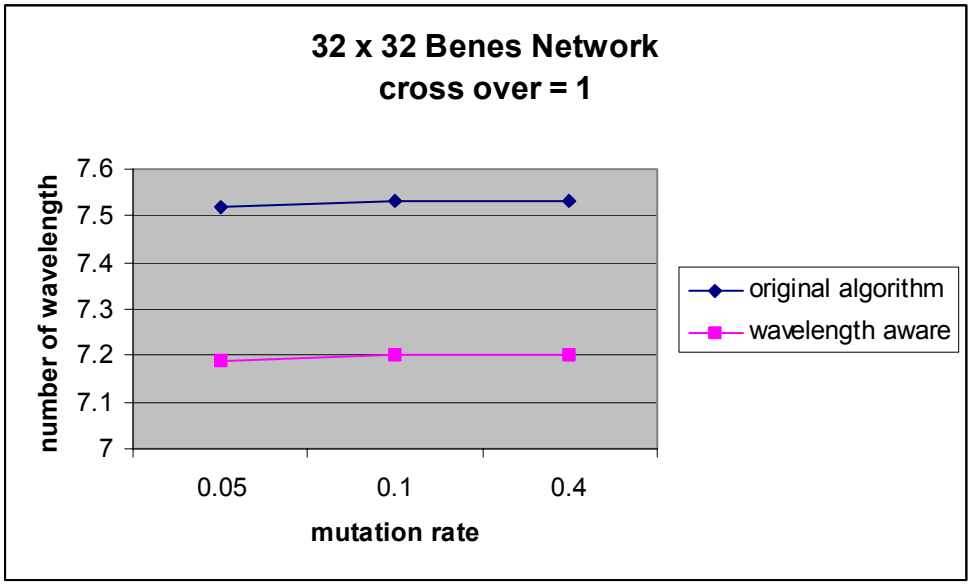

Figure 6.12 Number of wavelength required in 32 x 32 Benes network, cross over rate = 1

As can be seen from the tables, changing mutation rate does not change the result much, while wavelength awareness routing outperforms the original algorithm in every case, in terms of number of wavelengths needed. And it also can be seen that with the network size increasing, the improvement of wavelength awareness routing over original algorithm gets more significant. We list the improvement in a separate figure as follows,

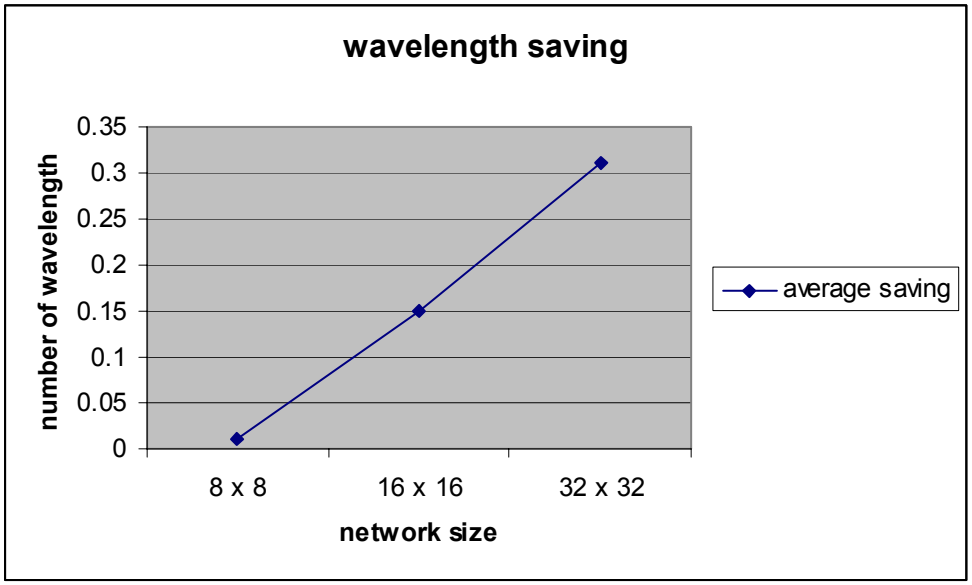

Figure 6.13 Average saving of No. of wavelength over the original algorithm

We think that the performance improves with the network size because with the size increases, there is more room for a connection to choose between different paths; while in a smaller network such options are very limited.

Chapter 7

Simulated Annealing

Simulated annealing is a generalization of a Monte Carlo method for examining the equations of state and frozen states of n-body systems [63]. The concept is based on the manner in which liquids freeze or metals recrystalize in the process of annealing. In an annealing process a melt, initially at high temperature and disordered, is slowly cooled so that the system at any time is approximately in thermodynamic equilibrium. As cooling proceeds, the system becomes more ordered and approaches a "frozen" ground state at T=0. Hence the process can be thought of as an adiabatic approach to the lowest energy state. If the initial temperature of the system is too low or cooling is done insufficiently slowly the system may become quenched forming defects or freezing out in metastable states (ie. trapped in a local minimum energy state).

The original Metropolis scheme was that an initial state of a thermodynamic system was chosen at energy E and temperature T, holding T constant the initial configuration is perturbed and the change in energy dE is computed. If the change in energy is negative the new configuration is accepted. If the change in energy is positive it is accepted with a probability given by the Boltzmann factor $exp\ -(dE/T)$. This processes is then repeated sufficient times to give good sampling statistics for the current temperature, and then the temperature is decremented and the entire process repeated until a frozen state is achieved at T=0.

By analogy the generalization of this Monte Carlo approach to combinatorial problems is straightforward [64]. The current state of the thermodynamic system is analogous to the current solution to the combinatorial problem, the energy equation for the thermodynamic system is analogous to at the objective function, and ground state is analogous to the global minimum. The major difficulty (art) in implementation of the algorithm is that there is no obvious analogy for

the temperature T with respect to a free parameter in the combinatorial problem. Furthermore, avoidance of entrainment in local minima (quenching) is dependent on the "annealing schedule", the choice of initial temperature, how many iterations are performed at each temperature, and how much the temperature is decremented at each step as cooling proceeds.

In SA, the value of an objective function [65] that we want to minimize is analogous to the energy in a thermodynamic system. At high temperatures, SA allows function evaluations at faraway points and it is likely to accept a new point with higher energy. This corresponds to the situation in which high-mobility atoms are trying to orient themselves with other non-local atoms and the energy state can occasionally go up. At low temperatures, SA evaluates the objective function only at local points and the likelihood of it accepting a new point with higher energy is much lower. This is analogous to the situation in which the low-mobility atoms can only orient themselves with local atoms and the energy state is not likely to go up again. Obviously, the most important feature of SA is the so-called annealing schedule [65] or cooling schedule, which specifies how rapidly the temperature is lowered from high to low values. This is application specific and requires some experimentation by trial-and-error.

SA's major advantage over other methods is an ability to avoid becoming trapped in local minima. The algorithm employs a random search that not only accepts changes that decrease the objective function f (assuming a minimization problem), but also some changes that increase it. The latter are accepted with a probability

$$P = \exp(-\delta f/T)$$

where δf is the increase in f and T is a control parameter, which by analogy with the original application is known as the system *"temperature"* irrespective of the objective function involved. Figure 7.1 shows the structure of a SA.

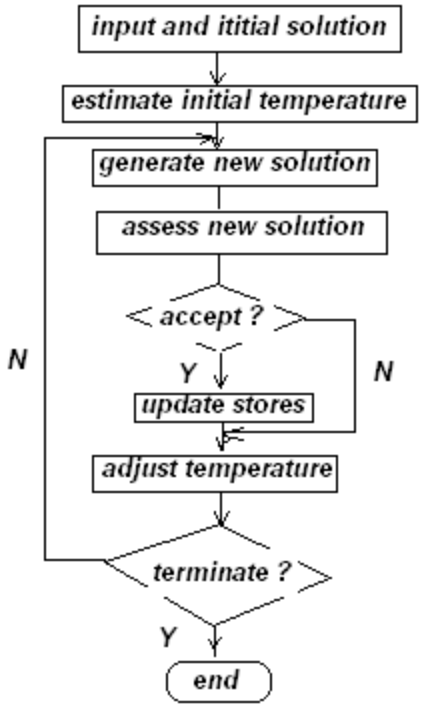

Figure 7.1 A SA flow chart

If the new solution were randomly generated, it would be possible that this new solution is not a legal solution; in order to solve this problem, another concept move set [65] is proposed. A move set {m}, is the set of all operations that generate one solution from another by moving a component of the solution. A popular moving set is inversion, translation and switching [66].

7.1 Inversion

Inversion is to reverse the order of a segment in a string. For example, if the original list is 0 1 2 3 4 5 6 7, and after randomly choose two points in this list, for example 3 and 5, the inversion of that list according to these two points would be 0 1 2 5 4 3 6 7.

7.2 Translation

Translation is to randomly generate two points and then translate that section of the list to be stored in two randomly generated points. For example, if the list is 0 1 2 5 4 3 6 7 and if we

generate 1 and 3 for the section to be replaced, and then 6 and 7, the section 1-2-5 is placed in between 6 and 7 as 0 4 3 6 1 2 5 7.

7.3 Switching

Two points are randomly chosen and then the points are switched at those positions. Suppose if we generate 0 and 2, then 0 and 2 are interchanged in the above list as 2 0 4 3 6 1 0 5 7.

Clearly, these three operations will not generate illegal result, because all the information is derived within the system, and because our case, the information refer to the wavelength, or color, it must be legal results.

7.4 The Experiment and Analysis of The Test Result

The experiment follows the setting as in [66], the parameters used in the algorithm are the starting temperature, the final temperature, the temperature-cooling rate (α), the number of iterations M for a particular temperature value and the stopping value ts. The starting temperature is used for the Algorithm to start at a particular temperature. The cooling rate (α) is chosen between 0 and 1. Generally it is chosen between 0.8 and 1. The ts is chosen as 10, and the number of iterations M vary as indicated in [66].

The result of genetic algorithm provides better results in terms of the number of wavelengths needed compared with [62]. For example, consider the following permutation in a 8 x 8 Benes network as in Figure 8.2,

$$\begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 0 & 2 & 1 & 4 & 3 & 7 & 5 & 6 \end{pmatrix}$$

Figure 7.2 A permutation

In [62], the resulting conflict graph is as follows, as can be seen in Figure 7.3, it needs 6 wavelengths to route the connection

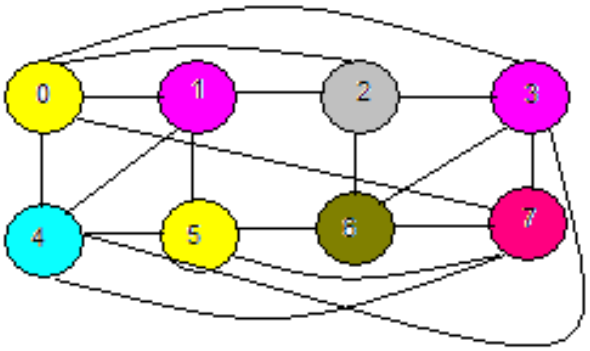

Figure 7.3 The graph coloring of the above permutation in [62]

While the result of Genetic algorithms is as follows, and it needs only 5 wavelengths.

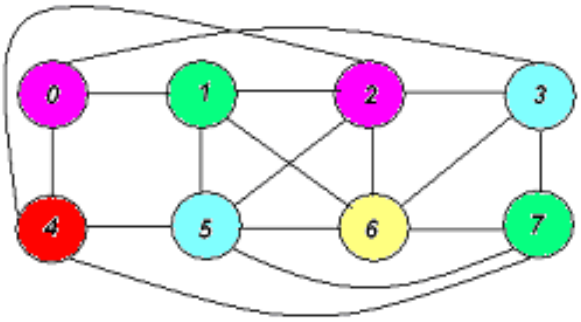

Figure 7.4 The graph coloring of the above permutation using SA

Then we test several cases with Simulated Annealing.

First, we test the effect of different move sets on the results of simulated annealing.

Table 7.1 is the comparison of number of wavelength required in 8 x 8 Benes network, using different move sets, Figure 7.5. is a comparison between the original algorithm with wavelength-aware routing algorithm.

Table 7.1. No. of wavelengths in 8 x 8 Benes network under different move set (avg. of 1000 tests)

starting temp	end temp	cool rate	No. Of iterations	orig inv	orig trans	orig switch	nw inv	nw trans	new switch
100	10	0.8	2	3.814	4.131	4.107	3.814	4.131	4.107
100	10	0.8	4	3.814	4.112	4.112	3.814	4.112	4.112
100	10	0.8	6	3.814	3.973	4.103	3.814	3.973	4.103
100	10	0.8	8	3.814	3.994	3.997	3.813	3.992	3.997
100	10	0.8	10	3.814	3.986	3.985	3.813	3.986	3.985
100	10	0.8	12	3.813	3.991	3.987	3.812	3.986	3.985
100	10	0.8	14	3.813	3.989	3.985	3.812	3.989	3.978
100	10	0.8	16	3.812	3.99	3.978	3.812	3.989	3.978
100	10	0.8	18	3.812	3.978	3.976	3.812	3.978	3.976
100	10	0.8	20	3.812	3.978	3.976	3.812	3.978	3.976

From above table, we can see that the inversion move set outperforms other move sets, and it conforms to the result in [66]; therefore, we choose inversion as the representative of the our wavelength aware routing and the original routing scheme.

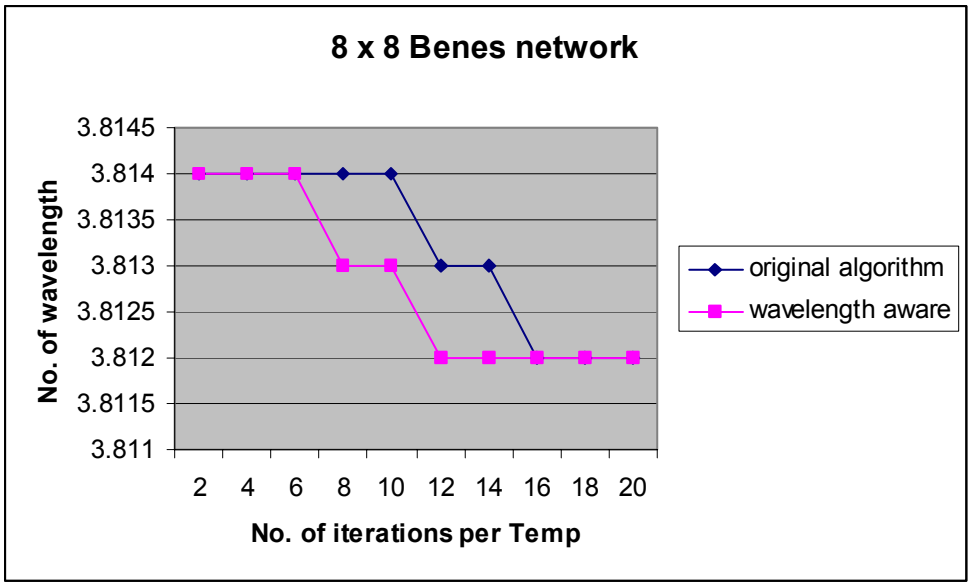

Figure 7.5 No. of wavelengths in 8 x 8 Benes network under different move set

Table 7.2 is the comparison of number of wavelength required in 16 x 16 Benes network, using different move sets, Figure 7.6. is a comparison between the original algorithm with wavelength-aware routing algorithm.

Table 7.2. No. of wavelengths in 16 x 16 Benes network under different move set (avg. of 1000 tests)

starting temp	end temp	cool rate	No. of iterations	orig inv	orig trans	orig switch	nw inv	nw trans	new switch
100	10	0.8	2	5.923	5.923	5.92	5.805	5.851	5.851
100	10	0.8	4	5.919	5.925	5.918	5.8	5.852	5.848
100	10	0.8	6	5.917	5.922	5.921	5.799	5.851	5.849
100	10	0.8	8	5.917	5.923	5.92	5.793	5.848	5.85
100	10	0.8	10	5.914	5.921	5.921	5.798	5.849	5.848
100	10	0.8	12	5.912	5.918	5.918	5.799	5.848	5.85
100	10	0.8	14	5.911	5.919	5.917	5.799	5.848	5.848
100	10	0.8	16	5.911	5.917	5.918	5.8	5.842	5.845
100	10	0.8	18	5.911	5.918	5.916	5.798	5.848	5.845
100	10	0.8	20	5.91	5.917	5.916	5.797	5.848	5.845

From the above table, also it can be seen that inversion move set out performs other move sets; therefore, we choose inversion move set as the representative to compare the original algorithm with the wavelength aware routing algorithm.

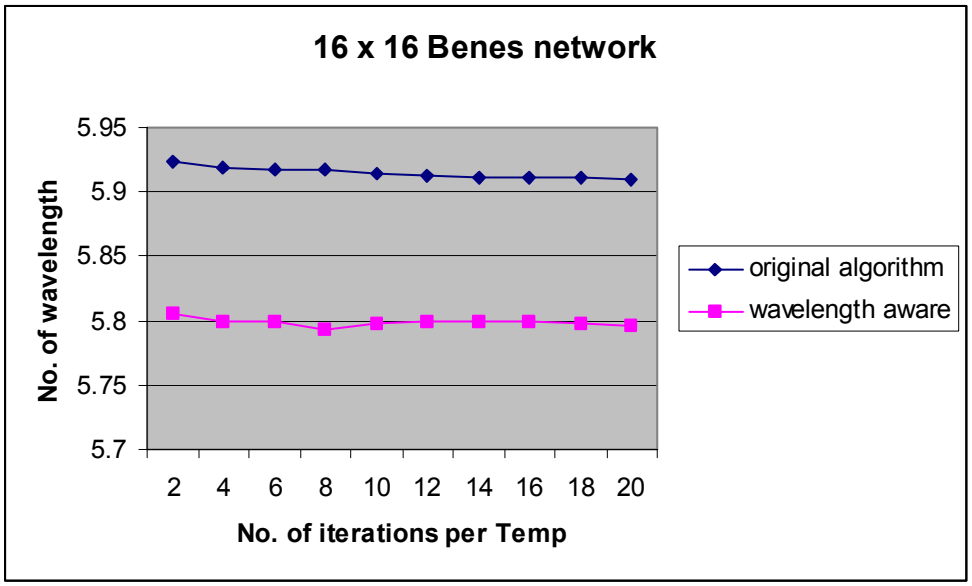

Figure 7.6. No. of wavelengths in 16 x 16 Benes network under different move set

Table 7.3 is the comparison of number of wavelength required in 16 x 16 Benes network, using different move sets, Figure 7.7. is a comparison between the original algorithm with wavelength-aware routing algorithm.

Table 7.3. No. of wavelengths in 32 x 32 Benes network under different move set (avg. of 1000 tests)

starting temp	end temp	cool rate	No. of iterations	orig inv	nw inv
100	10	0.8	2	7.661	7.59
100	10	0.8	4	7.647	7.579
100	10	0.8	6	7.628	7.572
100	10	0.8	8	7.626	7.568
100	10	0.8	10	7.59	7.565
100	10	0.8	12	7.59	7.563
100	10	0.8	14	7.58	7.551
100	10	0.8	16	7.566	7.49
100	10	0.8	18	7.562	7.425
100	10	0.8	20	7.559	7.382

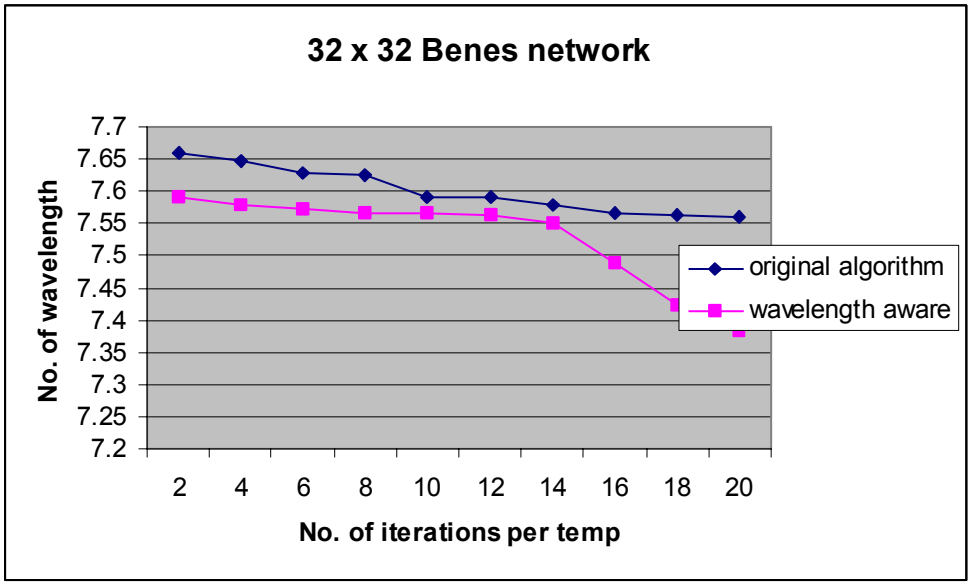

Figure 7.7. No. of wavelengths in 32 x 32 Benes network under different move set (avg. of 1000 tests)

We then investigate the effect of starting temperature and ending temperature on the performance of SA on Benes routing algorithms.

Table 7.4 is the number of wavelength under SA with changing starting temperature in 8 x 8 Benes network, and Figure 7.8 is a comparison between the original algorithms with the wavelength-aware routing.

Table 7.4. Number of wavelength in Benes network under SA with changing starting temperature in 8 x 8 Benes network (avg. of 1000 tests)

starting temp	end temp	cool rate	No. of iterations	orig inv	nw inv
100	10	0.8	20	3.813	3.812
200	10	0.8	20	3.813	3.812
300	10	0.8	20	3.812	3.812
400	10	0.8	20	3.812	3.812
500	10	0.8	20	3.811	3.81
600	10	0.8	20	3.809	3.808
700	10	0.8	20	3.809	3.807
800	10	0.8	20	3.809	3.807
900	10	0.8	20	3.808	3.806
1000	10	0.8	20	3.808	3.806

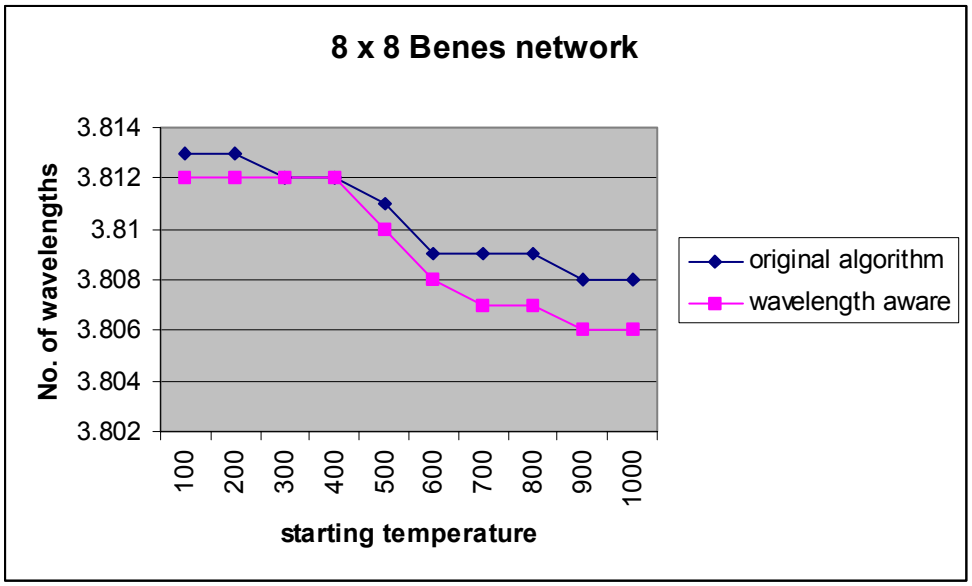

Figure 7.8. Number of wavelength in Benes network under SA with changing starting temperature in 8 x 8 Benes network

Table 7.5 is the number of wavelength under SA with changing starting temperature in 16 x 16 Benes network, and Figure 7.9 is a comparison between the original algorithms with the wavelength-aware routing.

Table 7.5. Number of wavelength in Benes network under SA with changing starting temperature in 16 x 16 Benes network (avg. of 1000 tests)

starting temp	end temp	cool rate	No. of iterations	orig inv	nw inv
100	20	0.8	20	5.91	5.8
200	20	0.8	20	5.899	5.79
300	20	0.8	20	5.895	5.783
400	20	0.8	20	5.896	5.78
500	20	0.8	20	5.883	5.779
600	20	0.8	20	5.882	5.775
700	20	0.8	20	5.88	5.77
800	20	0.8	20	5.879	5.768
900	20	0.8	20	5.879	5.764
1000	20	0.8	20	5.877	5.75

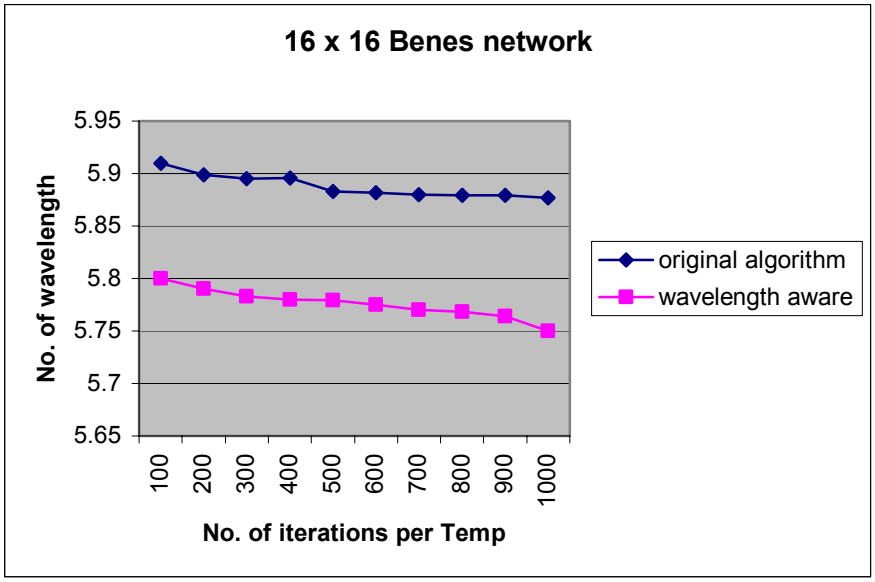

Figure 7.9 Number of wavelength in Benes network under SA with changing starting temperature in 16 x 16 Benes network

Table 7.6 is the number of wavelength under SA with changing starting temperature in 32 x 32 Benes network, and Figure 7.10 is a comparison between the original algorithms with the wavelength-aware routing.

Table 7.6. Number of wavelength in Benes network under SA with changing starting temperature in 32 x 32 Benes network (avg. of 1000 tests)

starting temp	end temp	cool rate	No. of iterations	orig inv	nw inv
100	20	0.8	20	7.559	7.382
200	20	0.8	20	7.559	7.382
300	20	0.8	20	7.557	7.38
400	20	0.8	20	7.555	7.382
500	20	0.8	20	7.557	7.377
600	20	0.8	20	7.549	7.375
700	20	0.8	20	7.544	7.375
800	20	0.8	20	7.539	7.374
900	20	0.8	20	7.539	7.371
1000	20	0.8	20	7.539	7.371

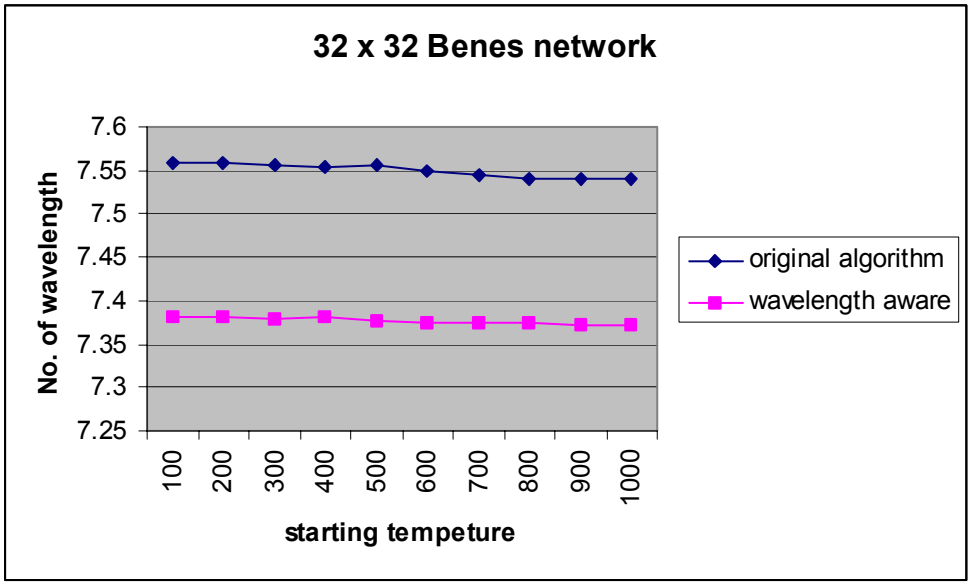

Figure 7.10 Number of wavelength in Benes network under SA with changing starting temperature in 32 x 32 Benes network

As can be seen, the number of wavelength required reduces as the starting temperature increases. This conforms to our speculation. When the temperature increase, the simulated annealing has more space to search for and therefore, it will more likely provide better solution than when the temperature is low. And in any case, the wavelength awareness routing outperforms the original one.

Table 7.7 is the number of wavelength under SA with changing ending temperature in 8 x 8 Benes network, and Figure 7.11 is a comparison between the original algorithms with the wavelength-aware routing.

Table 7.7. Number of wavelength in Benes network under SA with changing ending temperature in 8 x 8 Benes network (avg. of 1000 tests)

starting temp	end temp	cool rate	No. of iterions	orig inv	nw inv
1000	10	0.8	20	3.808	3.806
1000	5	0.8	20	3.806	3.806
1000	2.5	0.8	20	3.801	3.801
1000	0.125	0.8	20	3.797	3.795
1000	0.0625	0.8	20	3.794	3.794
1000	0.03125	0.8	20	3.794	3.794
1000	0.015625	0.8	20	3.794	3.792
1000	0.007813	0.8	20	3.794	3.794
1000	0.00391	0.8	20	3.794	3.788
1000	0.00196	0.8	20	3.794	3.788

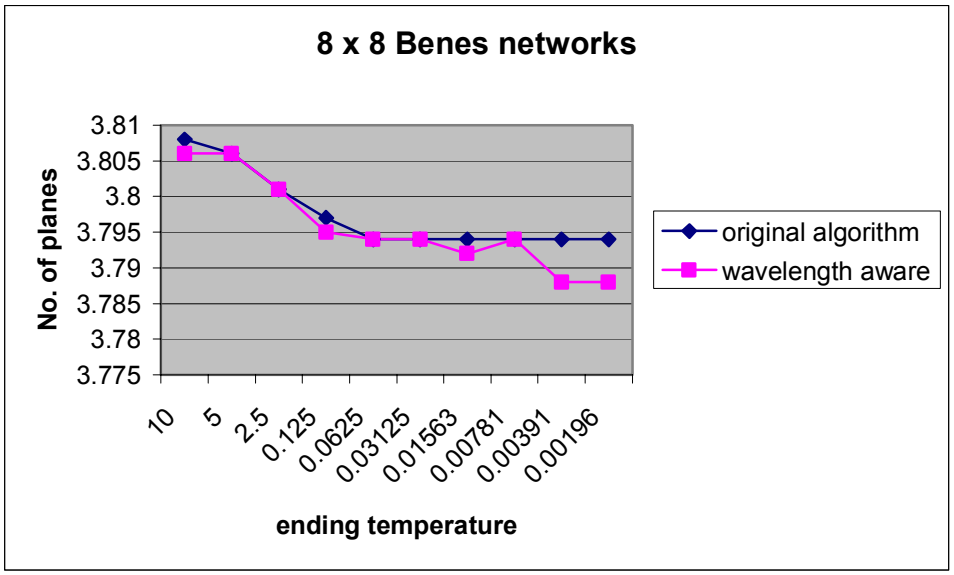

Figure 7.11 Number of wavelength in Benes network under SA with changing ending temperature in 8 x 8 Benes network

Table 7.8 is the number of wavelength under SA with changing ending temperature in 16 x 16 Benes network, and Figure 7.12 is a comparison between the original algorithms with the wavelength-aware routing.

Table 7.8. Number of wavelength in Benes network under SA with changing ending temperature in 16 x 16 Benes network (avg. of 1000 tests)

starting temp	end temp	cool rate	No. of iterions	orig inv	nw inv
1000	10	0.8	20	5.872	5.75
1000	5	0.8	20	5.866	5.746
1000	2.5	0.8	20	5.861	5.746
1000	0.125	0.8	20	5.861	5.744
1000	0.0625	0.8	20	5.855	5.738
1000	0.03125	0.8	20	5.85	5.74
1000	0.015625	0.8	20	5.843	5.735
1000	0.007813	0.8	20	5.841	5.731
1000	0.00391	0.8	20	5.832	5.729
1000	0.00196	0.8	20	5.831	5.729

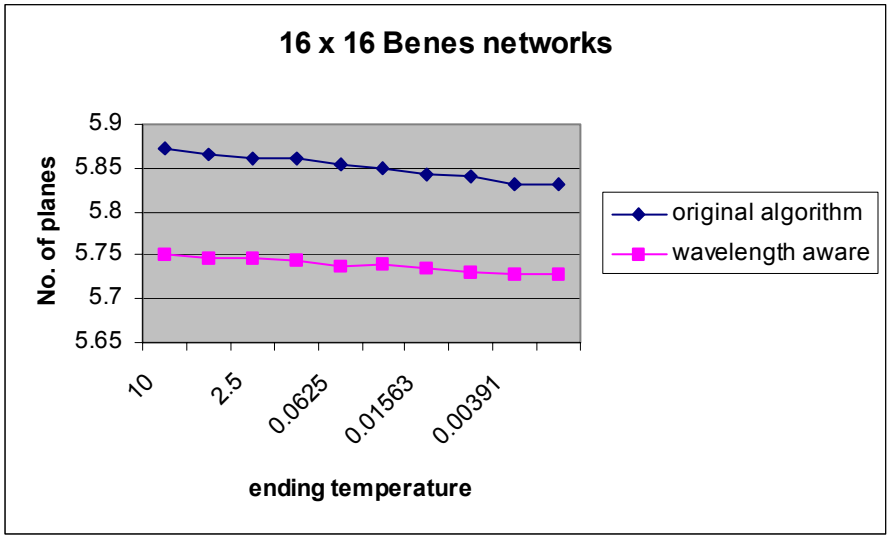

Figure 7.12 Number of wavelength in Benes network under SA with changing ending temperature in 16 x 16 Benes network

Table 7.9 is the number of wavelength under SA with changing ending temperature in 32 x 32 Benes network, and Figure 7.13 is a comparison between the original algorithm with the wavelength-aware routing.

Table 7.9. Number of wavelength in Benes network under SA with changing ending temperature in 32 x 32 Benes network (avg. of 1000 tests)

starting temp	end temp	cool rate	No. of iterations	orig inv	nw inv
100	20	0.8	20	7.559	7.382
200	20	0.8	20	7.559	7.382
300	20	0.8	20	7.557	7.38
400	20	0.8	20	7.555	7.382
500	20	0.8	20	7.557	7.377
600	20	0.8	20	7.549	7.375
700	20	0.8	20	7.544	7.375
800	20	0.8	20	7.539	7.374
900	20	0.8	20	7.539	7.371
1000	20	0.8	20	7.539	7.371

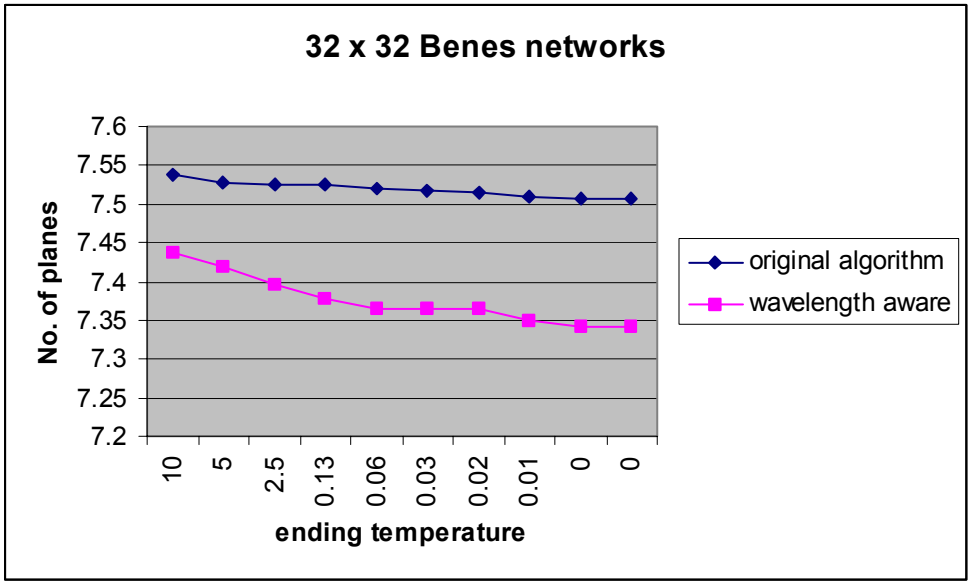

Figure 7.13 Number of wavelength in Benes network under SA with changing ending temperature in 32 x 32 Benes network

As can be seen from the above tables, the number of wavelengths required decease as the temperature decreases. This is because when we evaluate exp (-δf / T) and compare it with the value generated by the random number generator, as the temperature is getting low, exp (-δf/T) evaluates to a very less value which in many cases is less than the random number generated. So,

at this point only solutions, which are strictly better than the original solution, are accepted and the others are rejected. And in any case, the wavelength awareness routing outperforms the original one.

We then investigate the effect of different cooling rate on the number of wavelengths. Table 7.10 is the number of wavelength under SA with changing ending temperature in 8 x 8 Benes network, and Figure 7.14 is a comparison between the original algorithms with the wavelength-aware routing.

Table 7.10. Number of wavelength in Benes network under SA with changing cooling rate in 8 x 8 Benes network (ave. of 1000 tests)

starting temp	end temp	cool rate	No. of iterations	orig inv	nw inv
1000	10	0.81	20	3.808	3.806
1000	10	0.82	20	3.808	3.806
1000	10	0.83	20	3.805	3.805
1000	10	0.84	20	3.808	3.805
1000	10	0.85	20	3.805	3.805
1000	10	0.86	20	3.805	3.802
1000	10	0.87	20	3.801	3.802
1000	10	0.88	20	3.801	3.801
1000	10	0.89	20	3.801	3.801
1000	10	0.9	20	3.801	3.801

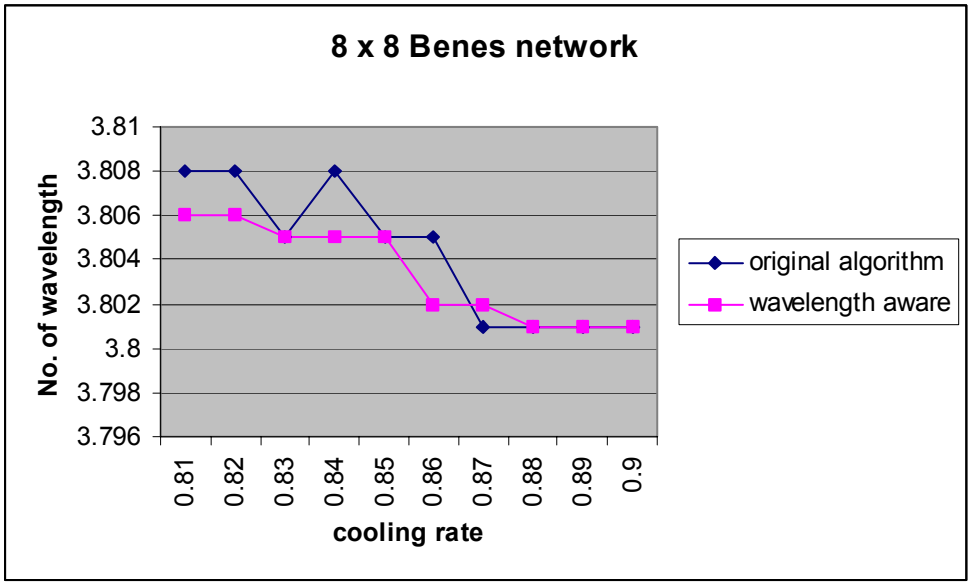

Figure 7.14. . Number of wavelength in Benes network under SA with changing cooling rate in 8 x 8 Benes network

Table 7.11 is the number of wavelength under SA with changing ending temperature in 16 x 16 Benes network, and Figure 7.15 is a comparison between the original algorithms with the wavelength-aware routing.

Table 7.11. Number of wavelength in Benes network under SA with changing cooling rate in 16 x 16 Benes network (ave. of 1000 tests)

starting temp	end temp	cool rate	No. of iterations	orig inv	nw inv
1000	10	0.81	20	5.881	5.8
1000	10	0.82	20	5.881	5.79
1000	10	0.83	20	5.879	5.782
1000	10	0.84	20	5.879	5.779
1000	10	0.85	20	5.877	5.779
1000	10	0.86	20	5.876	5.778
1000	10	0.87	20	5.876	5.773
1000	10	0.88	20	5.874	5.759
1000	10	0.89	20	5.873	5.751
1000	10	0.9	20	5.872	5.744

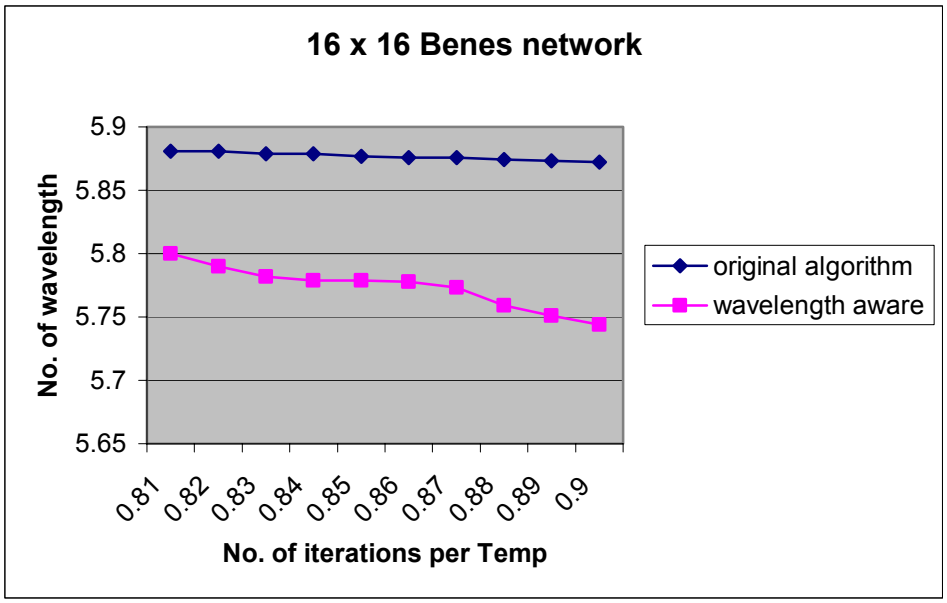

Figure 7.15. Number of wavelength in Benes network under SA with changing cooling rate in 16 x 16 Benes network

Table 7.12 is the number of wavelength under SA with changing ending temperature in 32 x 32 Benes network, and Figure 7.16 is a comparison between the original algorithms with the wavelength-aware routing.

Table 7.12. Number of wavelength in Benes network under SA with changing cooling rate in 32 x 32 Benes network (avg. of 1000 tests)

starting temp	end temp	cool rate	No. of iterations	orig inv	nw inv
1000	10	0.81	20	7.54	7.371
1000	10	0.82	20	7.54	7.37
1000	10	0.83	20	7.539	7.371
1000	10	0.84	20	7.539	7.365
1000	10	0.85	20	7.537	7.362
1000	10	0.86	20	7.537	7.355
1000	10	0.87	20	7.537	7.355
1000	10	0.88	20	7.537	7.351
1000	10	0.89	20	7.535	7.349
1000	10	0.9	20	7.535	7.349

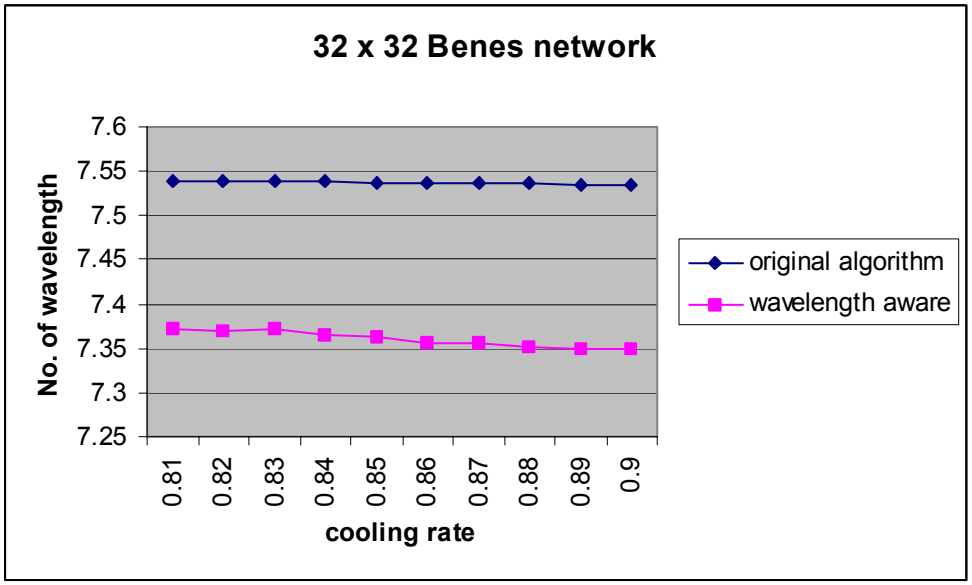

Figure 7.16. Number of wavelength in Benes network under SA with changing cooling rate in 32 x 32 Benes network

As can be seen from the above figure, the performance of SA gets better when the cooling rate slows down, which conforms to the theory [65]. And in any case, the wavelength awareness routing outperforms the original one.

Chapter 8

Ant Colony

Ant Colony Algorithms are inspired by the behavior of natural ant colonies, in the sense that they solve their problems by multi agent cooperation using indirect communication through modifications in the environment.

8.1 Introduction to Ant Colony

Natural, or real, ants release a certain amount of pheromone while walking, and each ant prefers (probabilistically) to follow a direction that is rich of pheromone. This simple behavior explains why ants are able to adjust to changes in the environment, such as new obstacles interrupting the currently shortest path [67]. For example, consider the following scenario, where ants are searching for food.

Figure 8.1 Ants are looking for food

If at some point, an obstacle is put on the ants' way to food,

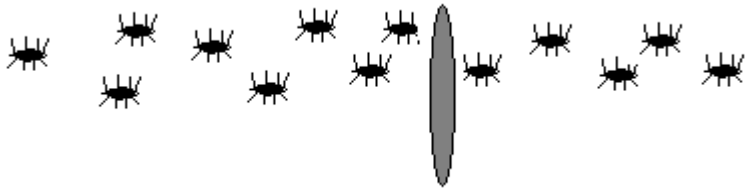

Figure 8.2 An obstacle is in the way

When the ants reach the obstacle they will randomly choose some way around it (right, left, over or under).

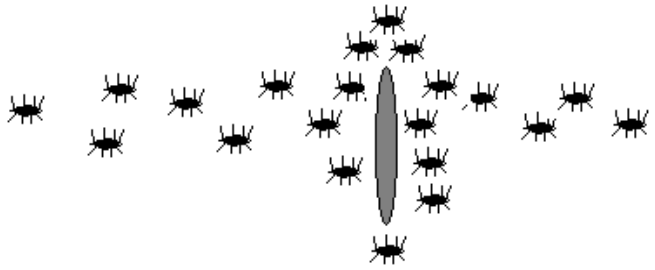

Figure 8.3 Ants randomly choose a route

From the figure, we can see that the upper half rout is shorter than the lower half out. But the ants do not know which is short, they just randomly pick one of the route. Apparently, those ants who choose the upper route will get to the food faster than those who choose the lower half, and therefore those who choose the upper half will build a stronger trail than the ones taking the lower part. This, in turn, will attract more ants to the upper route (since ants have the tendency of following trails) and at the same time, the pheromone left on the lower route will evaporate over time. Eventually, there is only one route left, which is the upper router, the shortest one in this case.

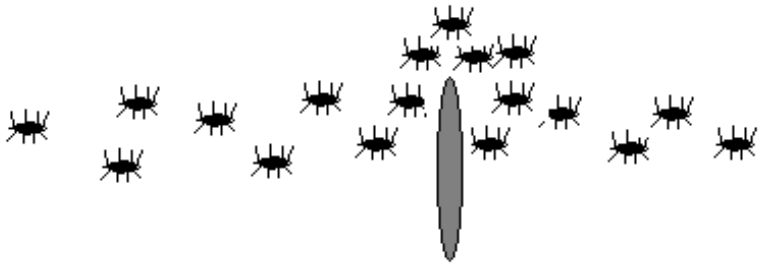

Figure 8.4 Ants have found the shortest path

8.2 Application of Ant Colony Optimization to Benes network routing

The routing in Benes network has been formulated as a graph-coloring problem. The connections in the Benes network are mapped to the nodes in the graph. A routing is non-blocking if and only if its corresponding graph can be properly colored.[68] In our research, we are trying to find the smallest possible number which is sufficient to color the conflict graph. The graph-coloring algorithm ANTCOL is designed by Costa and Hertz [68] and this algorithm is an implementation of the Ant Colony technique and it can properly color the graph.

The ANTCOL algorithm is briefly described as follows,

There are n ants, and their functionality is to travel from nodes to nodes and color them in a constructive way. When they color the nodes, they try to minimize the conflicts among the nodes. Their experience is stored in a N x N array; each element of this array represents the quality between two non-adjacent, the quality will evaporate over time, unless reinforced by the coloring process. And the quality is updated by the following formula, $M_{rs} := \rho . M_{rs} + \sum_{s_a \in S_{rs}} 1/q_a$,

where M is the quality, (1-ρ) is the evaporation rate, therefore ρ is the preserving rate, the second part of the formula is the reinforcement part, that is to increase the quality, and q is the number of colors in the previous cycle.

The quality of the resulting coloring strongly depends on the order in which the vertices are scanned. The ordering of the vertices can be either static or dynamic [68], An ordering of the vertices is called static if it can be completely determined before any color has been assigned. An ordering is called dynamic if the choice of the next vertex to be colored is based on previous color assignments. There are three major static ordering methods, namely *RANDOM, LF (Largest First), and SL (Smallest Last);* there are two major dynamic ordering methods, namely

DSATUR, RLF (Recursive Largest First). In [66], it shows that among these ordering method, RLF performs the best. We therefore will test our algorithm only with RLF.

8.3 Experiment Analysis

Table 8.1. The number of wavelength needed with respect to different number of ants in 16 x 16 Benes network, with cycle = 100

No. of Ants	evaporation rate	old algorithm	wavelength aware
2	0.05	3.785	3.785
4	0.05	3.761	3.76
8	0.05	3.761	3.759
16	0.05	3.763	3.762
32	0.05	3.771	3.769
64	0.05	3.78	3.775
128	0.05	3.785	3.782
256	0.05	3.788	3.785
512	0.05	3.793	3.789

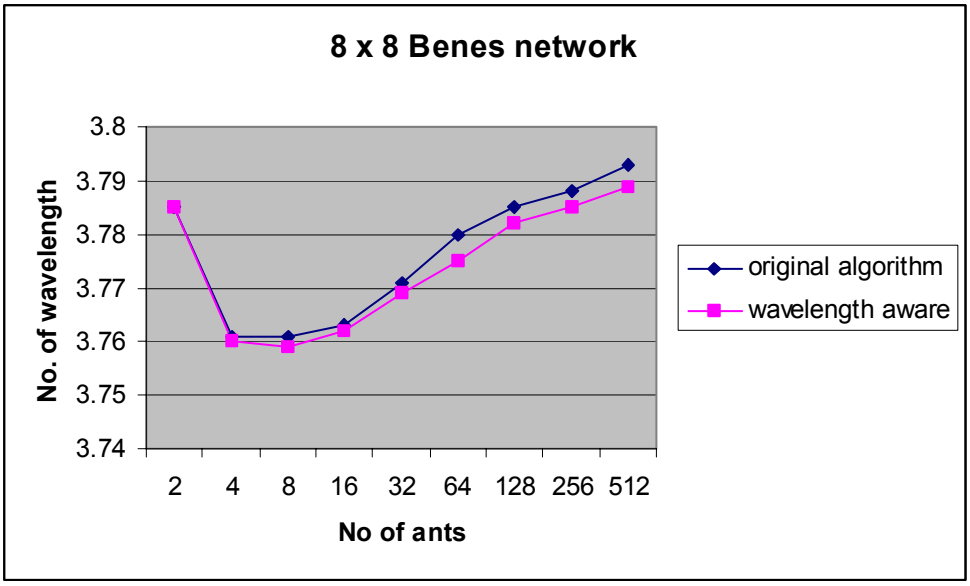

Figure 8.5. Number of wavelength with respect to number of ants in 8 x 8 Benes Network

Table 8.2, the number of wavelength needed with respect to different number of ants in 16 x 16 Benes network, with cycle = 100

No. of Ants	evaporation rate	old algorithm	wavelength aware
2	0.05	5.96	5.85
4	0.05	5.92	5.81
8	0.05	5.91	5.793
16	0.05	5.901	5.749
32	0.05	5.873	5.728
64	0.05	5.868	5.719
128	0.05	5.87	5.719
256	0.05	5.879	5.725
512	0.05	5.883	5.76

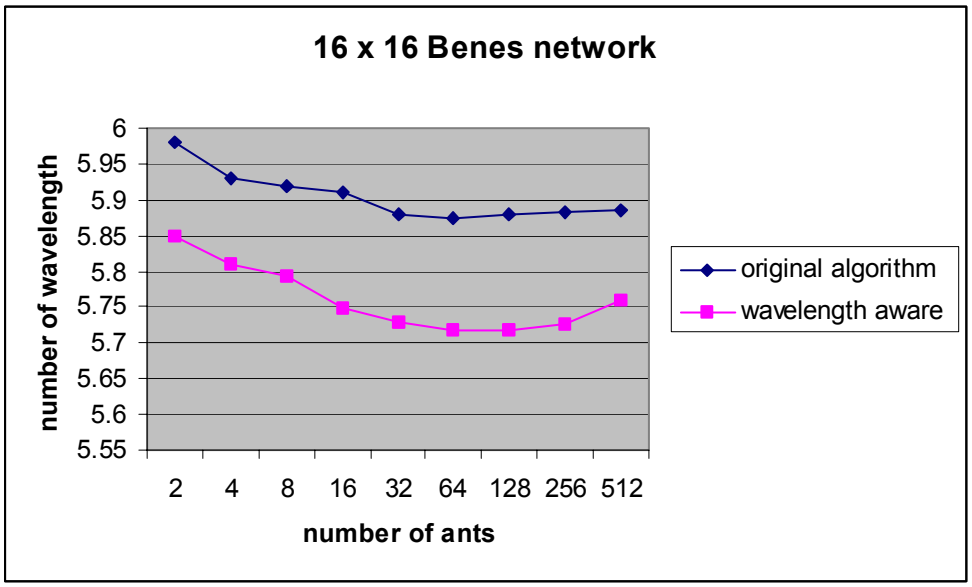

Figure 8.6 Number of wavelength with respect to number of ants in 16 x 16 Benes network

Table 8.3. The number of wavelength needed with respect to different number of ants in 16 x 16 Benes network, with cycle = 100

No. of Ants	evaporation rate	old algorithm	wavelength aware
2	0.05	7.79	7.442
4	0.05	7.765	7.413
8	0.05	7.73	7.401
16	0.05	7.71	7.385
32	0.05	7.651	7.377
64	0.05	7.628	7.363
128	0.05	7.603	7.315
256	0.05	7.611	7.322
512	0.05	7.615	7.327

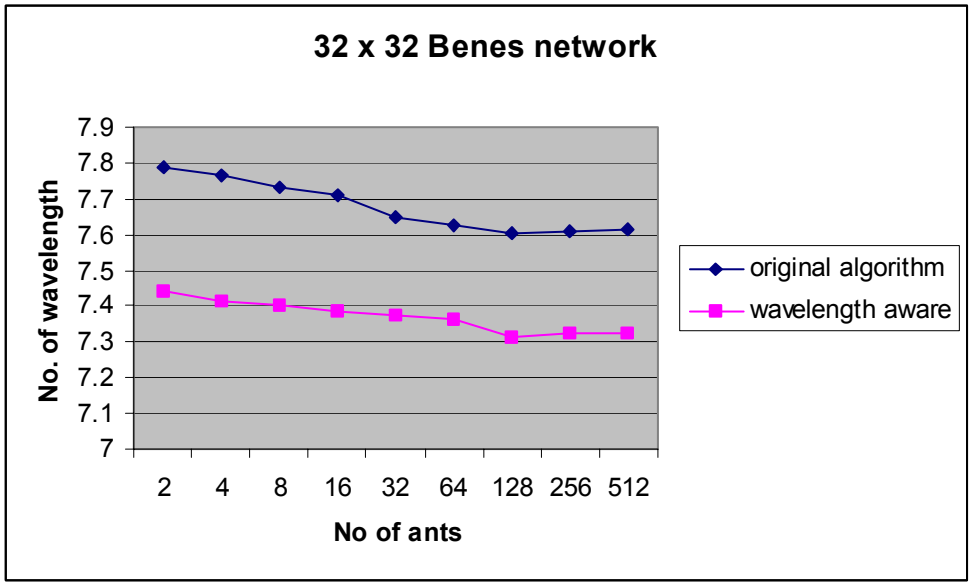

Figure 8.7 The number of wavelength with respect to number of ants in 32 x 32 Benes network

As can be seen from the figures, the Ant Colony performs the best in 8 x 8 network when the number of ants is 8, in 16 x 16 network when the number of ants is 64, and in 32 x 32 network when the number of ants is 128. It is interesting to notice that when the number of ants increase passes that mark, the performance goes down. The reason perhaps is that when the number of ants goes above some threshold value, the overhead of controlling these ants has offset the computational power gain.

8.4. Comparisons Among GA, SA and ACO

From the discussion of each the GA, SA and ACO, it is clear that the proposed algorithm is better than the original Benes network routing algorithm in terms of the number of wavelength required. And it is also interesting to compare these three different technologies.

We will compare these three technologies and the original heuristics in a 3 different network sizes.

Table 8.4. Comparison among GA, SA and ACO and Heuristics

Network size	GA	SA	ACO	GA-new	SA-new	ACO-new	original heuristic
8 x 8	3.74	3.828	3.77	3.74	3.811	3.78	4.17
16 x 16	5.87	5.896	5.88	5.71	5.753	5.742	6.31
32 x32	7.56	7.678	7.623	7.21	7.391	7.35	8.29

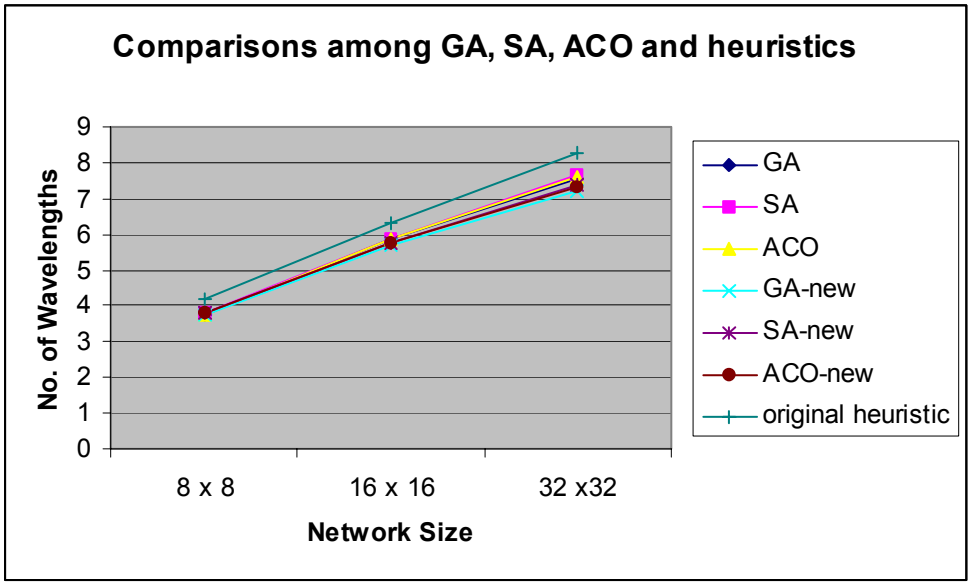

Figure 8.8 Comparisons among GA, SA and ACO and Heuristics

As can be seen from Figure 8.8 the average number of wavelength of the wavelength awareness routing is are better than the corresponding original routing algorithms; and among them GA has the best performance in terms of number of wavelength.

We now compare the running time for each of the three AI techniques and that of the original heuristics.

Table 8.5 Comparison of running time among the AI techniques and the heuristic.

Network size	GA	SA	ACO	GA-new	SA-new	ACO-new	original heuristic
8 x 8	221sec	0.05sec	0.07	475sec	5sec	10sec	0.01sec
16 x 16	58minutes30Sec	0.33sec	2sec	1hr30minutes	8sec	17sec	0.01sec
32 x32	10hours	3sec	5sec	15hours	15sec	39sec	0.01sec

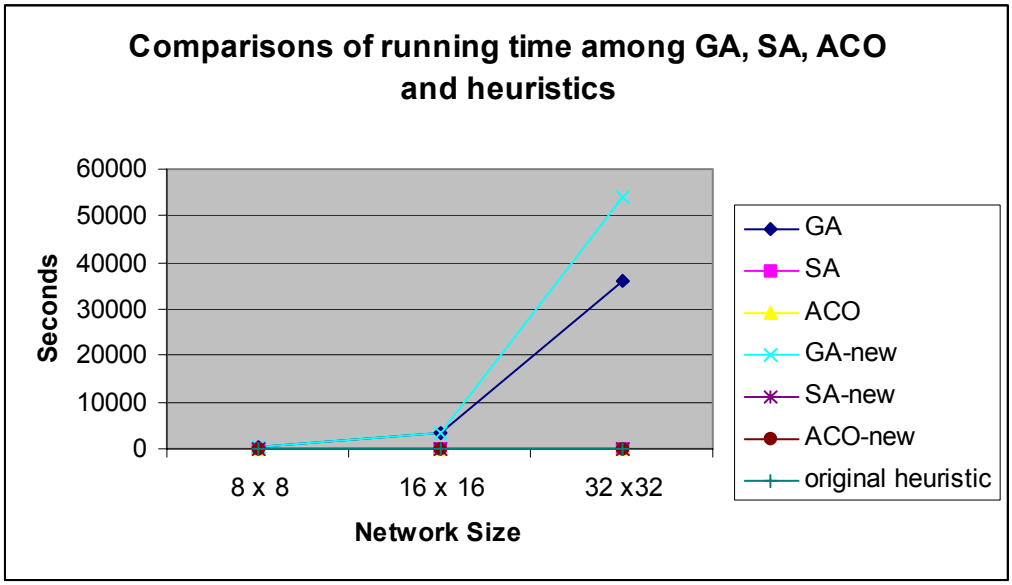

Figure 8.9 Comparisons of running time among GA, SA and ACO and Heuristics

It is clear from Figure 8.9. that the running time of Genetic Algorithms is much higher that the rest of the algorithms, even Genetic Algorithms provide the best result in terms of the No. of Wavelengths needed to routing the network.

Because the gap between Genetic Algorithms and the rest of the algorithms are so huge, it is necessary to illustrate the other algorithms except GA in a separate figure 8.10.

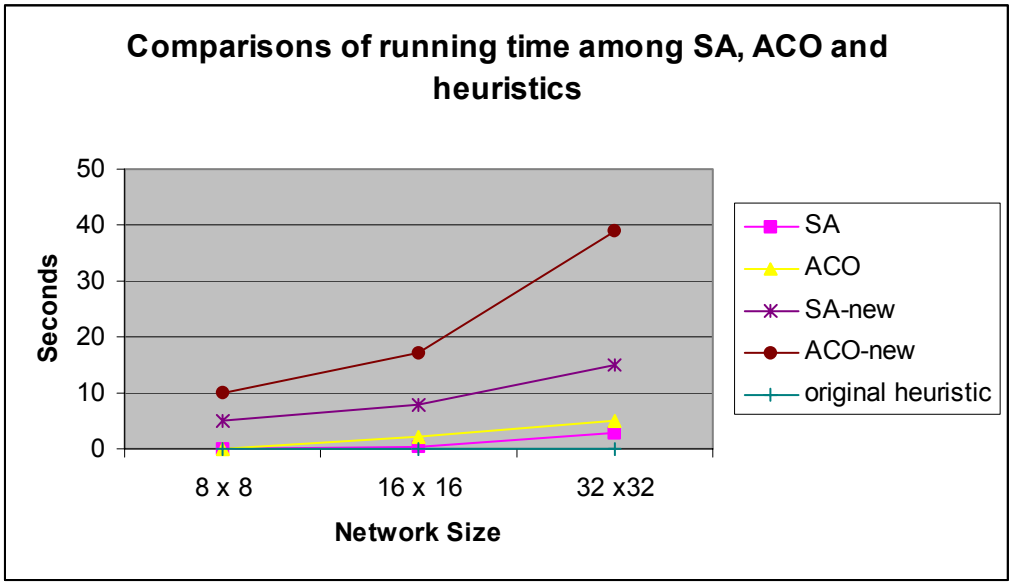

Figure 8.10 Comparisons of running time among SA and ACO and Heuristics

It can be seen from figure 8.9 that the running time of heuristic is the best and that of Ant Colony algorithm is the worst among the three. But in terms of the number of wavelengths needed to routing the network, heuristic provides the worst result and Simulated Annealing provides the best.

Chapter 9

Conclusion and Future Work

This final chapter concludes the thesis and identifies some future research topics.

9.1 Conclusion

In this dissertation, we propose a mathematical model to analyze the blocking probability of stacked Benes network. We find that it is not cost effective to meet non-blocking requirements in most cases. We further derive an upper bound of the blocking probability with respect to the number of planes in stacked Benes network.

We propose a new algorithm in WDM Benes network routing that reduces the number of wavelengths needed we further adapt the windows method to baseline network.

We develops AI techniques, such as the genetic algorithm (GA), simulated annealing (SA) algorithm and the ant colony optimization (ACO) technique are applied to the problem to calculate the number of wavelength required for routing a given permutation. We then compare the results from these three techniques and we find that genetic algorithms yield the best result in terms of the number of wavelength; but in terms of the running time, Genetic Algorithms performs the worst while Simulated Annealing provides the best performance.

9.2 Future Work

We then identify some of the possible future research topics.

Lower Bound Estimation

We will find the clique of a conflict graph [49], which is obtained after applying the window method to a given network. We find the number of cliques [57] and use this value as a lower bound on the number of wavelengths. A clique in an undirected graph $G = (V, E)$ is a

subset $V' \subset V$ of vertices, each pair of which is connected by an edge in E. The size of the clique is the number of vertices it contains. The clique problem is the optimization problem of finding a clique of maximum size in a graph [57]. An algorithm for determining whether a graph $G = (V, E)$ with $|V|$ vertices has a clique of size k is to list all k-subsets of V, and check each one to see whether it forms a clique. The running time of the algorithm is polynomial if k is a constant. k could be proportional to $|V|$, in which case the algorithm runs in super polynomial time. We can find the clique by using an algorithm which tests for a particular clique size [57] by traversing through all the nodes in the conflict graph. For example a clique of size 4 on a graph having 8 nodes can be found using four loops as shown below.

```
for ( i = 0; i < 5; i ++ ) {
        for ( j = i + 1; j < 6; j ++ ) {
                for ( k = j + 1; k < 7; k ++ ) {
                        for ( l = k + 1; l < 8; l ++ ) {
                                check if vertices V_i, V_j, V_k, and V_l form a clique of size 4
                        }
                }
        }
}
```

Similarly we can find the clique of different sizes. As we can see this algorithm has a very high time complexity because finding the clique of a graph is a NP-complete problem [57]. The number of possible combinations for finding a clique of size 'k' with 'n' nodes can be found using the combination formula $n\mathrm{c}_{\mathrm{k}}$. This method of finding the clique may not be an efficient implementation, but we are not concerned with the clique problem but the problem of finding a lower bound estimate on the number of passes required for routing the messages.

Since finding the clique has a very high time complexity, we will avoid finding all the cliques equal to the number of nodes in the graph. This can be accomplished by finding the maximum degree of the graph as the upper bound. Then using this value as the upper bound we

can find the clique of a given conflict graph. For any conflict graph the minimum clique value will be 2. So starting from a clique value of 2 we will find higher clique values until the upper bound value. If we obtain a new clique value then we update the value of the clique. This is done until the upper bound is reached. We stop at this point because there is no point in searching for a higher clique value as we have already found the maximum number of passes required. If a higher clique value exists then the maximum number of passes value would have been different. Hence the clique value found is the correct value. If the clique value found for a given permutation is equal to the number of passes obtained using any algorithm discussed so far then the algorithm has worked well in routing that permutation.

The clique value calculated is used as a lower bound on the number of wavelengths required in routing a given permutation. In other words the maximum number of wavelengths can never be less than the clique size.

Parallel Routing

It is well known that wavelength assignment is NP-Complete, therefore parallelization is a possible research field. As in [20], A permutation can be correctly set up in O (NlogN) time using a completely connected multiprocessor system of N Processors.

We can further improve this by combining our refined route setup algorithms with this parallel wavelength assignment algorithm.

One to Many Casting

In the Benes network we discussed, we assume the traffic is one-to-one; another possible research is to allow one-to-many traffic, the so-called multicasting. Because the connection within one multicasting group can share the same wavelength, the question is how to set up the connection such that those connections within one multicasting group share the same

wavelength while connections belonging to different group do not.

More Order of Worst Cases

In this dissertation, we analyze the blocking probability of VSOBN with respect to the number of planes. In particular, we analyze the non-blocking requirement when the several worst cases do not happen. In our dissertation, we stop at the third worst case, because neglecting 3rd worst case alone will not save a plane. But it is interesting to further investigate the situation in which the first n cases, for example the first 100 worst cases or first 200 worst cases do not happen. Maybe the aggregate effect from 3rd worst case to the nth case will save a plane. Further, we are interested in deriving a function of the nth worst cases, namely, if up to the nth worst cases are neglected, how many planes are required to guarantee non-blockingness?

Bibliography

[1] D. Banerjee and B. Mukherjee, “A practical approach for routing and wavelength assignment in large wavelength-routed optical networks,” *IEEE J. Select. Areas Commun.*, vol. 14, pp. 903-908, June 1996.

[2] Regis J. Bates, “Optical Switching and Networking Handbook,” McGraw-Hill, February 2001.

[3] Laxmi N. Bhuyan, Dharma P. Agrawal, “Design and Performance of Generalized Interconnection Networks,” IEEE Transactions on Computers 32(12): 1081-1090, December 1983.

[4] Uyless D. Black, “Optical Networks: Third Generation Transport Systems,” Prentice Hall PTR, February 2002.

[5] M. Brenner, D. Tutsch, G. Hommel, “Measuring Transient Performance of a Multistage Interconnection Network Using Ethernet Networking Equipment,” Proceedings of the International Conference on Communications in Computing 2002 (CIC'02). Las Vegas, USA, 2002, pp. 211-216.

[6] Tony T. Lee, Soung Y. Liew, “Parallel Routing Algorithms in Benes–Clos Networks” IEEE TRANSACTIONS ON COMMUNICATIONS, VOL. 50, NO. 11, NOVEMBER 2002

[7] Skiena, S. S. "Clique and Independent Set" and "Clique." §6.2.3 and 8.5.1 in The Algorithm Design Manual. New York: Springer-Verlag, pp. 144 and 312-314, 1997.

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

[9] Cormen, T.; Leiserson, C.; and Rivest, R. Introduction to Algorithms. Cambridge, MA: MIT Press, 1990.

[10] N. Das, B.B. Bhattacharya, R. Menon, A. Sarkar, “Permutation Routing in Optical MIN's with Minimum Number of Stages,” Journal of Systems Architecture, vol. 48, 2003, 311-323.

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

[12] M. M. Vaez and C.-T. Lea, “Strictly nonblocking directional-coupler-based switching networks under crosstalk constraint,” *IEEE Trans. Commun.*, vol. 48, pp. 316–323, Feb. 2000.

[13] M. M. Vaez and C.-T. Lea, “Wide sense nonblocking banyan-type switching

systems based on directional couplers," *IEEE J. Select. Areas Commun.*, vol. 16, pp. 1327–1332, Sept. 1998.

[14] J. Duato, S. Yalamanchili, L. Ni, "Interconnection Networks," Morgan Kaufman, 2002.

[15] A.E. Eiben, J.K. van der Hauw, and J.I. van Hemert, "Graph coloring with adaptive evolutionary algorithms," Journal of Heuristics, 4(1):25-46, 1998.

[16] L.R. Goke and G.J. Lipovski, "Banyan networks for partitioning multiprocessing systems," Proceedings of the First International Symposium on Computer Architecture, pp. 21-28, 1973.

[17] David Greenfield, "The Essential Guide to Optical Networks," Prentice Hall, December 2001.

[18] C. Ji, "Routing in Optical Multistage Networks using Genetic Algorithms," A Master thesis, Computer Science Department, Georgia State University, May 2001.

[19] X Jiang, H Shen, M Khandker, S Horiguchi, "Blocking Behaviors of Crosstalk-Free Optical Banyan Networks on Vertical Stacking" *IEEE/ACM Trans. Networking.*, vol. 11, No. 6, Dec. 2003.

[20] Enyue Lu and S. Q. Zheng "Parallel Routing and Wavelength Assignment forOptical Multistage Interconnection Networks", Proceedings of the 2004 International Conference on Parallel Processing (ICPP'04)

[21] M. Dorigo and L M Gambardella, "Ant colony system: A cooperative learning approach to the traveling salesman problem," IEEE Transactions on Systems, 1(1):34--47, (1997).

[22] Clyde P. Kruskal, Marc Snir, "The Performance of Multistage Interconnection Networks for Multiprocessors," IEEE Transactions on Computers 32(12): 1091-1098, December 1983.

[23] Manoj Kumar, J. Robert Jump, "Performance of Unbuffered Shuffle-Exchange Networks," IEEE Transactions on Computers 35(6): 573-578, June 1986.

[24] W. Leland, M. Taqqu, W. Willinger and D. Wilson, "On the self-similar nature of ethernet traffic (extended version)," IEEE/ACM Transactions on Networking, 2(1):1–15, February 1994.

[25] M. Li, Weijia Jia, W. Zhao, "A Whole Correlation Structure of Asymptotically Self-Similar Traffic in Communication Networks," WISE 2000: 480-485.

[26] V.E. Benes, Mathematical Theory of Connecting Networks and Telephone Traffic,

Academic Press, New York, 1965

[27] A Comparison of Selection Schemes used in Genetic algorithms, Blickle, T., Thiele, L., TIK_Report Nr. 11, December 1995, Version

[28] Biswanath. Mukherjee, D. Banerjee, S. Ramamurthy and A. Mukherjee, "Some principles for designing a wide-area WDM optical network," IEEE/ACM Trans. Networking, vol. 4, pp. 684-696, Oct. 1996.

[29] R. A. Nordin, W. R. Holland and M. A. Shahid, "Advanced Optical Interconnection Technology in Switching Equipment," *J. Lightwave Technology*, 13(6), pp. 987-994, 1995.

[30] Asuman E. Ozdaglar and D. P. Bertsekas, "Routing and wavelength assignment in optical networks," Lab. Information and Decision Systems, Mass. Inst. Technology, Cambridge, MA, LIDS Rep., 2001.

[31] Yi Pan, C. Qiao, and Y. Yang, "Optical Multistage interconnection Networks: New Challenges and Approaches," IEEE Communications Magazine, Vol 37, No 2, pp 50-56, February 1999.

[32] Y. Pan, C. Qiao, Y. Yang, and J. Wu, "Recent Developments in Optical Multistage Networks," Optical Networks - Recent Advances**,** L. Ruan and D.-Z. Du (Eds), Kluwer Academic Publisher, September 2001, pp. 151-185.

[33] J.H. Patel, "Performance of processor-memory interconnections for multiprocessors," IEEE Transactions on Computers, vol. C-30, pp. 771-780, October 1981.

[34] Qian-Ping Gu, Shietung Peng, "Multihop All-to-All Broadcast on WDM Optical Networks," IEEE Transactions on Parallel and Distributed Systems 14(5): 477-486 (2003).

[35] Qian-Ping Gu, Shietung Peng, "Wavelengths Requirement for Permutation Routing in All-Optical Multistage Interconnection Networks," IPDPS 2000: 761-768.

[36] William H. Press, Brian P. Flannery, Saul A. Teukolsky, William T. Vetterling, "Numerical recipes in C: the art of scientific computing," Cambridge University Press, NY, 1992.

[37] C. Qiao, R. Melhem, D. Chiarulli and S. Levitan, "A Time Domain Approach for Avoiding Crosstalk in Optical Blocking Multistage Interconnection Networks," *Journal of Lightwave Technology,* vol 12, no 10, pp. 1854-1862, (1994).

[38] C. Qiao and L. Zhou, "Scheduling Switching Element Disjoint Connections in

Stage-Controlled Photonic Banyans," IEEE Trans. on Communications, Vol. 47, No. 1, pp. 139-148, 1999.

[39] R. Ramaswami and G. Sasaki, "Multiwavelength optical networks with limited wavelength conversion," IEEE/ACM Trans. Networking, vol. 6, pp. 744-754, Dec. 1998.

[40] R. Ramaswami and K. N. Sivarajan, "Routing and wavelength assignment in all-optical networks," IEEE/ACM Trans. Networking, vol. 3, pp. 489-499, Oct. 1995.

[41] X. Shen, F. Yang, Y. Pan, "Equivalent Permutation Capabilities between Time Division Optical Omega Networks and Non-optical Extra-Stage Omega Networks," IEEE/ACM Transactions on Networking, Vol. 9, No. 4, August 2001, pp. 518 -524.

[42] S. Subramaniam and A. B. Richard, "Wavelength assignment in fixed routing WDM networks," in Proc. IEEE Int. Conf. Communications, 1997, pp. 406-410.

[43] R.A. Thompson, "The dilated slipped banyan switching network architecture for use in an all-optical local-area network," Journal of Lightwave Technology, vol. 9. no. 12, pp. 1780-1787, Dec. 1991.

[44] C. Tocci and H.J. Caulfield, "Optical Interconnection – Foundations and Applications," Artech House Publishers, 1994.

[45] D. Tutsch, "Transient Multicast Traffic Performance of MINs: A Case Study," Workshop Distributed Computing on the Web 1999 (DCW'99). Rostock, Germany, 1999, pp. 103-110.

[46] P. J. M. Van Laarhoven and E. H. L. Aarts, "Simulated Annealing: Theory and Applications," D. Reidel, Dordrecht, The Nethelands, 1987.

[47] Anujan Varma and C.S. Raghavendra, "Interconnection Network for Multiprocessors and Multicomputers, Theory and Practice," IEEE Computer Press, 1994.

[48] C.L. Wu and T.-Y. Feng, "On a class of multistage interconnection networks," IEEE Transactions on Computers, vol. C-29, pp. 694-702, August 1980.

[49] Y. Yang, J. Wang, and Y. Pan, "Permutation Capability of Optical Multistage Interconnection Networks," Journal of Parallel and Distributed Computing (JPDC), Vol. 60, No. 1, pp. 72-91, Jan. 2000.

[50] H. Yoon, K.Y. Lee, and M.T. Liu, "Performance analysis of multibuffered packet–switching networks in multiprocessor systems," IEEE Transactions on Computers, 39(3):319–327, March 1990.

[51] Zbigniew Michalewcz Genetic Algorithms + Data Structures = Evolution Program Springer, 1996.

[52] Whitley, D., The GENITOR Algorithm and Selection Pressure: Why Rank-Based Allocation of Reproductive Trials is Best. Proceedings of the Third International Conference on Genetic Algorithms, Morgan Kaufmann Publishers, San Mateo, CA, 1989.

[53] Baker, J. E. Adaptive selection methods in genetic algorithms. Proceedings of the First International Conference on Genetic algorithms, Lawrence Erlbaum Associates, Hillsdale, NJ, 1985.

[54] J Zhong, Y Zhang, A Relative Fitness Awared Rank-based Selection Procedure in GA**.** 7th International Conference on Computational Intelligence and Natural Computing. Salt Lake city, UT, 2005.

[55] Ajay Kumar Katangur, Routing Algorithms and Performance Evaluation for Optical Multistage Networks with Limited Crosstalk, Ph.D. Dissertation. 2004.

[56] H. Zhijun, T. Pushan, "HEWN: a polynomial algorithm for CLIQUE problem," Journal of Comp. Science and Technology, vol.13 (suppl. issue), pp.33-44, Dec. 1998.

[60] G. Chartrand, L. Lesniak, "Graphs and Digraphs", Chapman & Hall, 3rd edition, 1996.

[61] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, The MIT Press; 2 edition, 2001.

[62] E. Lu and S. Q. Zheng, "Parallel Routing Algorithms for Nonblocking Electronic and Photonic Switching Networks", *IEEE Transactions on Parallel and Distributed Systems*, vol 16, no. 8, pp. 702-713, Aug, 2005.

[63] Metropolis,N., A. Rosenbluth, M. Rosenbluth, A. Teller, E. Teller, "Equation of State Calculations by Fast Computing Machines", *J. Chem. Phys.*,21, 6, 1087-1092, 1953.

[64] Kirkpatrick, S., C. D. Gelatt Jr., M. P. Vecchi, "Optimization by Simulated Annealing",*Science*, 220, 4598, 671-680, 1983

[65] P. J. M. Van Laarhoven and E. H. L. Aarts, "Simulated Annealing: Theory and Applications," D. Reidel, Dordrecht, The Nethelands, 1987.

[66] Ajay K. Katangur, Yi Pan, Martin D. Fraser, Simulated annealing routing and wavelength lower bound estimation on wavelength-division multiplexing optical multistage networks Optical Engineering, Vol. 43 No. 5, May 2004

[67] Leandro N. De Castro, Fernando J. Von Zuben, Recent Developments in Biologically inspired Computing IDEA Group Publishing 2005

[68] D. Costa and A. Hertz, “Ants can colour graphs,” Journal of the Operational Research Society, 48:295--305, 1997.