Resource Allocation in Multigranular Optical Networks

by Jingxin Wu

B.S. in School of Information Science and Engineering, July 2010, Harbin Institute of Technology
M.S. in School of Information Science and Engineering, July 2012, Harbin Institute of Technology

A Dissertation submitted to

The Faculty of
The School of Engineering and Applied Science
of The George Washington University
in partial satisfaction of the requirements
for the degree of Doctor of Philosophy

August 31, 2018

Dissertation directed by

Suresh Subramaniam
Professor of Engineering and Applied Science

The School of Engineering and Applied Science of The George Washington University certifies that Jingxin Wu has passed the Final Examination for the degree of Doctor of Philosophy as of August 13, 2018. This is the final and approved form of the dissertation.

Resource Allocation in Multigranular Optical Networks

Jingxin Wu

Dissertation Research Committee:

Suresh Subramaniam, Professor of Engineering and Applied Science, Dissertation Director

Tian Lan, Associate Professor of Engineering and Applied Science, Committee Member

Miloš Doroslovački, Associate Professor of Engineering and Applied Science, Committee Member

Guru Venkataramani, Associate Professor of Engineering and Applied Science, Committee Member

Hiroshi Hasegawa, Associate Professor of The Department of Electrical Engineering and Computer Science, Nagoya University, Committee Member

Dedication

In memory of the departed loved ones.

To my family, Lab634 and SEH 5390.

Acknowledgements

I would first like to thank my advisor, Professor Suresh Subramaniam, for his continuous support throughout the past six years. His expertise and immense knowledge on the networking area have always led me to the appropriate direction of my research work. His patience, motivation and advice have been essential for my research papers. He has also provided extensive professional and personal guidance and helped me a great deal. I feel privileged to have him as my advisor.

I wish to thank Professors Hiroshi Hasegawa, Tian Lan, Milos Doroslovacki and Guru Venkataramani for generously offering their time and guidance as my Ph.D. dissertation committee members. I would especially like to thank Professor Hasegawa and Professor Venkataramani, coauthors of my research papers, who are always willing to share their insightful comments.

I gratefully acknowledge the support of NSF (through grant CNS-1406971), which sponsored my research.

I sincerely thank my friends, Yu Xiang, Juzi Zhao, Yifan Chen, Fan Yao, Chong Liu, Yongbo Li, Maotong Xu, Hang Liu, Xiaolu Song, Ming Yao, Shijing Li, Jie Chen, Bingqian Lu, Sultan Alamro and all others, for their friendship and for supporting me through the hard times of my life and making my stay in DC wonderful.

Last, but not the least, I would like to thank my family to whom I owe a lot. Thanks to my beloved parents, grandmother for the continuous and selfless love. I am also grateful to my brother for the moral support and constant encouragement.

Abstract

Resource Allocation in Multigranular Optical Networks

Thesis Statement: Cost-effective switching and spectrum utilization efficiency have become critical design considerations in optical networks. This dissertation provides in-depth exploration of these important aspects, and proposes effective techniques for low-cost switching architectures and resource allocation algorithms to facilitate the adoption of optical networks in the near future.

The dramatic growth of Internet traffic brings challenges for optical network designers. The increasing traffic and bandwidth requirements mean that various resource allocation schemes to achieve different network design goals assume great importance. The general problem of resource allocation to lightpath requests is a challenging problem.

An emerging technology of flexible and more fine-grained grid through the use of Optical Orthogonal Frequency Division Multiplexing (OOFDM) allows fiber bandwidth to be more suitably matched up with application requirements, thereby making the network more elastic than the conventional Wavelength Division Multiplexing (WDM) optical networks. Despite the advances of employing OOFDM technology in elastic optical networks (EONs), imminent fiber capacity exhaustion due to the ever-increasing demands means that multiple fibers per link will be inevitable. While increasing the number of fibers boosts the capacity of networks, there is a price to pay for it in the form of increased number of switch ports / complexity of switches. The huge amount of traffic demands and thus high hardware requirements motivate multigranularity (such as wavebanding) to save costs in optical networks. This dissertation aims to tackle several types of resource allocation challenges in multi-granular optical networks to either improve the spectrum utilization or provide cost-effective switching techniques.

The first part of this dissertation presents our work on an important joint scheduling (or *co-scheduling* of computational and network resources) problem in today's

applications such as cloud computing. Such applications involve the processing of complex jobs consisting of several inter-dependent tasks executing on heterogeneous clusters of computing resources, which are interconnected by high-speed optical networks. Our co-scheduling algorithms take the constraints introduced by elastic optical networks into account and aim at either minimizing the makespan of a set of jobs in the static case, or minimizing the job blocking when jobs arrive dynamically.

The second part of the dissertation addresses problems on waveband switching in WDM networks. Grouping together a set of consecutive wavelengths in a WDM network and switching them together as a single waveband could achieve savings in switching costs of an optical cross-connect (OXC). A specific problem that we consider is the optimal Band Minimization Problem in WDM mesh networks to minimize the required number of switching elements while accommodating a set of traffic demands. Then we evaluate OXC node architectures in WDM networks with multiple parallel fibers. A hierarchical architecture that utilizes wavebanding and has lower complexity is compared with the conventional architecture in terms of the cost of the OXC node and the power consumption. Analytical models for computing the blocking probability of connection requests are proposed and validated.

In an effort to reduce the complexity of optical OXCs, a flexible wavebanding OXC architecture has been proposed recently. Elastic networking and flexible wavebanding introduce a new problem, namely, the routing, fiber, waveband, and spectrum assignment (RFBSA) problem. In the third part of the dissertation, a framework for solving the RFBSA problem in networks with multi-fiber links that can accommodate non-contiguous and non-uniform wavebanding is proposed. Then a joint banding-node placement and RFBSA problem to meet the network budget while maintaining good network performance is addressed.

The final part of the dissertation addresses dynamic Routing and Spectrum Assignment problem in multi-fiber EONs to further improve spectrum efficiency. We propose and evaluate novel schemes to minimize the demand blocking ratio of requests.

Table of Contents

Dedication iii

Acknowledgements iv

Abstract v

List of Figures x

List of Tables xiii

Chapter 1: Introduction 1

1.1 Optical Networks . . . 1

1.1.1 WDM Optical Networks . . . 2

1.1.2 OOFDM-based Optical Networks . . . 2

1.1.3 Multi-fiber Links . . . 3

1.1.4 Multi-granular Optical Networks . . . 3

1.2 Resource Allocation Problems . . . 4

1.2.1 Co-Scheduling Problem . . . 4

1.2.2 Routing and Spectrum Assignment Problem . . . 4

1.2.3 Routing, Fiber, Band, and Spectrum Assignment (RFBSA) Problem . . . 5

1.3 Contributions . . . 5

Chapter 2: Co-scheduling Computational and Networking Resources in Elastic Optical Networks 7

2.1 Related Work . . . 7

2.2 Network Model and Problem Statement . . . 8

2.3 Scheduling Algorithms . . . 10

2.3.1 ILP Formulation . . . 10

2.3.2 Static Case . . . 14

2.3.2.1 First Fit (FF) . . . 15

2.3.2.2 Children-Aware (CA) . . . 15
2.3.3 Dynamic Requests . . . 18
2.4 Simulation Results . . . 19
2.4.1 Static Requests . . . 19
2.4.2 Dynamic Requests . . . 20
2.5 Conclusions . . . 21

Chapter 3: Optimal Nonuniform Wavebanding in WDM Mesh Networks 23

3.1 Related Work . . . 23
3.2 Network Model and Definitions . . . 23
3.3 Band Minimization ILP And Algorithms . . . 26
3.3.1 ILP Formulation . . . 28
3.3.2 Heuristic Solutions . . . 30
3.3.2.1 Nearest Neighbor (NN) . . . 30
3.3.2.2 ReassignDC (RDC) . . . 30
3.3.2.3 SortTraffic (ST) . . . 32
3.4 Simulation Results . . . 33
3.5 Application to Dynamic Stochastic Traffic . . . 35
3.6 Conclusion . . . 39

Chapter 4: Evaluation and Performance Modeling of OXC Architectures 40

4.1 Related Work . . . 40
4.2 OXC Node Architectures . . . 41
4.3 Analytical Models . . . 43
4.3.1 Conventional OXC Architecture . . . 43
4.3.2 Hierarchical OXC Architecture . . . 46
4.4 Simulation Results . . . 49
4.5 Conclusion . . . 52

Chapter 5: Routing, Fiber, Band, and Spectrum Assignment (RFBSA) for Multi-granular Elastic Optical Networks 53

5.1 Background and Problem Statement . . . 53
5.1.1 "FLEX" Architecture . . . 53
5.1.2 "FLEX" Waveband Architecture . . . 53
5.1.3 Comparison Between Architecture . . . 54
5.1.4 Motivation and Problem Definition . . . 55
5.2 The Proposed RFBSA Algorithm . . . 57
5.2.1 Auxiliary Layered-graph Framework . . . 57
5.2.2 Resource Assignment Algorithm . . . 58
5.3 Simulation Results . . . 63
5.4 Conclusions . . . 67

Chapter 6: Joint Banding-node Placement and Resource Allocation for Multi-granular Elastic Optical Networks 68

6.1 Background and Problem Statement . . . 68
6.2 The Integer Linear Programming Model . . . 69
6.2.1 Notations . . . 69
6.2.2 Formulations . . . 69
6.3 Heuristic Solutions . . . 73
6.3.1 Route, Fiber, Band and Slot Assignment Framework . . . 73
6.3.2 Node Placement Schemes . . . 78
6.3.2.1 Random Node Placement . . . 78
6.3.2.2 Traffic-aware Node Placement . . . 79
6.3.3 Extension to Dynamic Instances . . . 80
6.4 Simulation Results . . . 80
6.4.1 Results for a Small Network . . . 81
6.4.2 Results for Larger Topologies . . . 82
6.4.3 Results for Dynamic Requests . . . 85
6.5 Conclusion . . . 87

Chapter 7: Dynamic Routing and Spectrum Assignment for Multi-fiber Elastic Optical Networks **89**

7.1 Related Work 89

7.2 Contributions 90

7.3 Background and Motivatoin 91

7.4 Multi-Path Selection Scheme 92

7.4.1 Notations 93

7.4.2 Mixed Integer Linear Programming for Path Selection 94

7.4.3 Evaluation 94

7.5 Spectrum Assignment with Partitioning 95

7.5.1 Spectrum Management 95

7.5.2 Next State Aware Spectrum Assignment 96

7.5.3 Resource sharing among partitions 102

7.6 Numerical Results 104

7.7 Conclusions 108

Chapter 8: Conclusions and Future Directions **110**

8.1 Conclusions 110

8.2 Future Directions 111

List of Figures

2.1 Examples of a network graph and job DAG. 9
2.2 Example of subgraph. 18
2.3 NSF network. 19
2.4 Number of blocked jobs for dynamic traffic. 21

3.1 A 4-node mesh network example. 27
3.2 A routing and wavelength assignment example for given traffic. 27
3.3 RWA matrix example. 27
3.4 Graph generated from $\overline{\Delta}$ with first row chosen as the first node. . . . 28
3.5 Graph generated from $\overline{\Delta}$ with second row chosen as the first node. . . 28
3.6 A small 5-node network. 33
3.7 28-Node Pan-European network. 33
3.8 A comparison of heuristic results for NSF network. 36
3.9 Request blocking probability versus load in the NSF network for $W_s = 160$. 37
3.10 Request blocking probability versus load in the pan-European network for $W_s = 160$. 37
3.11 Request blocking probability versus number of wavelengths in the NSF network at a load of 800 Erlangs. 38
3.12 Request blocking probability versus number of wavelengths in the pan-European network at a load of 200 Erlangs. 38

4.1 An $N \times N$ WSS-based conventional OXC node architecture ("FLEX"). 41
4.2 Node architecture based on wavelength grouping and fiber selection ("HIER"). 42
4.3 The validation results for "FLEX". 50
4.4 The validation results for "HIER" ($k = 1$). 50
4.5 Comparison between "FLEX" and "HIER" ($k = 1$). 51
4.6 Power consumption comparison. 52
4.7 CapEx comparison. 52

5.1 Flexible waveband node architecture. 54
5.2 Illustrative example. 56
5.3 Performance comparison results for NSF network (F=2). 65
5.4 Performance comparison results for NSF network (F=[5,10]). 65
5.5 Comparison between KSPFF and RFBSA for NSF network (F=[5,10]). 66

6.1 Small network. The numbers are the number of fibers on the links. . 81
6.2 NSF network. The numbers are the number of fibers on the links. . . 82
6.3 Network performance vs. budget for NSF network (B=4). 83
6.4 Network performance vs. budget for NSF network (B=9). 83
6.5 Network performance vs. budget for pan-European network (B=4). . 84
6.6 Network performance vs. budget for pan-European network (B=9). . 84
6.7 Network performance vs. budget for NSF network with uniform traffic pattern (L=6900). 86
6.8 Network performance vs. budget for NSF network with uniform traffic pattern (L=7400). 86
6.9 Network performance vs. budget for NSF network with non-uniform traffic pattern (L=7100). 86
6.10 Network performance vs. budget for NSF network with non-uniform traffic pattern (L=7400). 86
6.11 Sensitivity test for traffic pattern with a perturbation of 0.1. 87

7.1 Slot usages without spectrum management. 92
7.2 Slot usages with spectrum management. 92
7.3 Effectiveness of the path selection scheme. 95
7.4 Effectiveness of spectrum partition. 97
7.5 A small 5-node network topology. 99
7.6 Network state example – link capacities of partial network on bin x_1, x_2, x_3, x_4. 101
7.7 Effectiveness of NSA algorithm in a 20-node unidirectional ring network. 101
7.8 Effectiveness of NSA algorithm in a 14-node NSF network. 102

7.9 Effectiveness of spectrum partition and NSA algorithm. 102
7.10 Effectiveness of resource sharing among partitions. 103
7.11 Comparison of spectrum allocation schemes in 14-Node NSF network with uniform traffic pattern. 105
7.12 Joint comparison in 14-Node NSF network with uniform traffic pattern. 106
7.13 Joint comparison in 28-Node Pan-European network with uniform traffic pattern. 106
7.14 Joint comparison in 14-Node NSF network with non-uniform traffic pattern. 107
7.15 Joint comparison in 28-Node Pan-European network with non-uniform traffic pattern. 107
7.16 Sensitivity test for demand size distribution in NSF network. 108
7.17 Sensitivity test for traffic pattern in NSF network. 109

List of Tables

2.1 Notation for the *co-scheduling* problem 11
2.2 Performance of ILP and algorithms for 5-node network: Static requests 20
2.3 Performance of algorithms for NSFNET: Static requests 21
2.4 Execution time in minutes for NSFNET: Dynamic requests 22

3.1 RWA Matrix Generation. 27
3.2 Performance of ILP and algorithms for 5-node network. 34
3.3 Performance of algorithms for NSF network. 34
3.4 Performance of algorithms for pan-European network. 34
3.5 Performance of algorithms for 5×5 regular mesh network. 35

4.1 Traffic demands for the small example node. 46
4.2 Power consumption and cost of the optical components in "FLEX" and "HIER" architectures. 51

5.1 Power consumption and cost of the optical components in "FLEX" and "FLEX" architectures. 55

6.1 Notation for the banding-node placement problem 70
6.2 Average MSU results for the small network. 82
6.3 Average number of FLEX nodes for NSF network (B=4). 82

7.1 Notation for dynamic RSA problem 93
7.2 Path selection probabilies for the small network. 100

Chapter 1 Introduction

The continuous growth of Internet traffic demands brings challenges for optical network designers. As traffic demands continue to grow and optical networks become even more pervasive, cost-efficient networking techniques become even more essential. Developing cost-efficient switching architectures and algorithms is necessary to reduce switching costs in the network and achieve faster adoption of optical infrastructures so as to facilitate the availability of high-speed networking. Given the limited network bandwidth capacity and ever-increasing traffic demands, the other important problem is to provision the requests with high spectrum efficiency. Thus the resource management problems in optical networks are nontrivial. Different resource allocation schemes would lead to different network performance. How to efficiently utilize various resources (such as wavelengths/subcarriers) in the system as well as meet different constraints (such as demand completion time, hardware requirements, blocking probability of connections) is indispensable in network design.

This dissertation aims at tackling these important aspects, and developing effective techniques for low-cost switching architectures and resource allocation algorithms for the next generation optical networks.

In this chapter, we first briefly review the technologies used in this work, and outline our contributions.

1.1 Optical Networks

Optical networking technologies have been widely introduced and deployed at the Internet's core networks. In the foreseeable future, optical networks will form the underlying physical infrastructure. There is a dramatic growth of Internet traffic, which results from emerging applications, such as livestreams and social networking. According to [3], the global Internet traffic is expected to be more than 60 TBps in 2020. The need for optical-electrical-optical conversions can be eliminated by reconfigurable optical add/drop multiplexers (ROADMs) or optical cross-connects

(OXCs). The enormous bandwidth demands can be carried by such all-optical paths between their sources nodes and destination nodes.

1.1.1 WDM Optical Networks

Wavelength Division Multiplexing (WDM) technology multiplexes information to be transmitted onto a large number of wavelengths. By utilizing the dense wavelength division multiplexing, multiple wavelengths/optical paths can be accommodated within a fiber. Those wavelengths are located on a fixed spacing defined by ITU-T. OXCs are used as the switching nodes in current optical networks [4]. As the traffic grows, more wavelengths are needed in today's WDM networks. To cope with the growing number of wavelengths, large-scale OXCs are needed. In a wavelength-switched optical node, each wavelength is switched individually, that is, one switching element corresponds to one wavelength. In this case, the switching cost can be very high.

1.1.2 OOFDM-based Optical Networks

Optical orthogonal frequency division multiplexing (OOFDM) technology is proposed as an approach in [5] to achieve high spectral efficiency and a flexible rate. A novel programmable mechanism for OOFDM which utilizes advanced digital signal processing, parallel signal detection, and flexible resource management schemes for subwavelength level multiplexing and grooming is presented in [6]. In OOFDM, the fiber bandwidth is carved up into subcarriers spaced only a few GHz apart with subcarrier bit rates of a few Gbps. This is to be contrasted with WDM where the typical wavelength spacing is 25 GHz, and wavelength bit-rates are 10, 40, or 100 Gbps. An attractive feature of OOFDM is that flexible bands of subcarriers may be assigned to a service as needed. In this way, allocated network resources can be matched up with service requirements in a much more flexible manner than in WDM-based networks. Such optical networks have therefore been called *elastic optical networks* [7].

1.1.3 Multi-fiber Links

The explosive use of video-based services, huge data transmission between data centers and so on causes continuous Internet traffic expansion. Imminent fiber capacity exhaustion means that multiple fibers per link will be inevitable. To accommodate increasing traffic demands, deploying multiple fibers on a physical link is a promising approach. Each link contains multiple fibers, and the number of fibers on each link may be different. All fibers consist of the same number of wavelengths or subcarriers (also called frequency slots or FSs) in WDM networks / EONs. At each OXC, a wavelength / FS on an input fiber can be switched to the same wavelength / FS on an output fiber.

1.1.4 Multi-granular Optical Networks

Grouping a bundle of wavelengths and switching them together as a single waveband ([8, 9]) will relieve the hardware requirements. It has been verified that such grouping of wavelengths can be realized by PLC technology ([10, 11]) and the reduced number of switches enables us to monolithically integrate all the necessary de-multiplexing and switching functions on a single chip [12]. Savings in switching costs can be achieved by this waveband switching. For example, each mirror in a 3-D micro-electro-mechanical systems (MEMS) based wavelength selective switch (WSS) [13] is dedicated to switching a wavelength path. If multiple wavelength paths are grouped and switched together, they will share the same mirror so that the number of mirrors needed in one OXC node will be decreased.

Such kind of multi-granularity could also be utilized in optical networks with multi-fiber links. Even though increasing the number of fibers improves the capacity of the network, it also increases the number of ports in OXCs, and thus increasing the complexity and hardware requirements of OXCs. By utilizing the wavebanding feature, the complexity of OXC architecture can be reduced with less hardware requirement. In EONs, the waveband switching corresponds to aggregating *a set* of optical spectral ranges and routing them with a single waveband port [14]. Along with the benefits of

wavebanding technique, the switching constraints are also introduced. The resource allocation problems with different objectives in such multi-granular optical networks need to be carefully addressed.

1.2 Resource Allocation Problems

This dissertation aims to tackle multiple types of resource allocation challenges in multi-granular optical networks to either improve the spectrum utilization or provide cost-effective switching techniques.

1.2.1 Co-Scheduling Problem

Current cloud computing [15–24], e-science, and data center applications [25–30] consist of job requests that usually have multiple inter-dependent tasks, with each task having certain computational needs [31–38]. At the same time, the inter-task communication may not be negligible. Therefore, an important challenge is to efficiently schedule both the computational and networking resources, which is called *co-scheduling*. Co-scheduling in a dynamic, heterogeneous, distributed computing environment has been extensively studied for about two decades, e.g., [39–43]. While co-scheduling problem in WDM networks has been widely explored, it hasn't been studied in the elastic optical networks.

1.2.2 Routing and Spectrum Assignment Problem

One important problem in EONs is the Routing and Spectrum Assignment / Allocation (RSA) problem, which is evolved from the routing and wavelength assignment (RWA) problem in WDM networks. The objective of RSA is to find a number of unoccupied frequency slots (FSs) to meet traffic demands and establish lightpaths [44–46]. The main constraints of the RSA problem are spectrum contiguity, spectrum continuity, and spectrum non-overlapping. The spectrum contiguity constraint ensures that the allocated FSs to a lightpath are contiguous. The spectrum continuity constraint ensures that the same FSs are allocated on every fiber along the route. The

spectrum non-overlapping constraint ensures that any FS in any fiber is allocated to at most one lightpath. The RSA problem can be divided into two subproblems: routing problem and spectrum allocation (SA). For static RSA problem, the objective is to minimize the maximum slot index (MS) while provisioning all traffic requests. For dynamic RSA problem, the objective is to minimize the blocking ratio and demand blocking ratio of traffic requests.

1.2.3 Routing, Fiber, Band, and Spectrum Assignment (RFBSA) Problem

In a multi-granular elastic optical network with multi-fiber links, the RSA problem evolves into a routing, fiber, band, and spectrum assignment problem. The goal of RFBSA problem is to assign the route, fibers, wavebands and the contiguous slot set to accommodate each request. RSA is known to be an NP-complete problem in EONs. Adding the fiber and flexible waveband selection further increases the problem complexity in another two dimensions [47].

1.3 Contributions

In this dissertation, we first address the co-scheduling problem of computing and networking resources in a single-granular elastic optical networks for multi-task jobs. Both static and dynamic versions of the multi-job multi-task co-scheduling problem in OOFDM-based elastic optical networks are considered. Each job request includes a set of tasks, each task has a workload requirement, and there are communication requirements between tasks. The objective is to minimize the makespan (finishing time of all jobs) of the set of jobs. This scheduling problem is clearly NP-hard, as even very restricted versions of the problem have been proved to be so [48]. We present an integer linear programming (ILP) formulation suitable for only very small instances, and two heuristic algorithms (applicable to larger problem instances) to allocate computational resources (such as virtual machines) and networking resources (such as subcarriers) to each job.

In the second part of the dissertation, we study multiple resource allocation problems in multi-granular optical networks. We first consider the optimal waveband design problem in WDM mesh networks, without the restriction of uniform wavebanding. The band minimization problem is considered from the entire network's point of view, as opposed to a single node's. An Integer Linear Programming (ILP) formulation and effective heuristics are proposed to solve the problem. Our framework helps to achieve a great amount of hardware requirements. Then we compare two architectures for an OXC - a conventional architecture and the hierarchical architecture - with multiple fibers per link in WDM networks, and present heuristics for resource (i.e., fiber and wavelength) assignment and analytical models to compute their blocking performance. After that, we address the RFBSA problem in EONs with multi-fiber links that can accommodate non-contiguous and non-uniform wavebanding. We propose an auxiliary layered-graph framework with pluggable cost functions and develop cost functions to minimize the maximum spectral usage for a given set of traffic demands. To achieve good network performance, as well as saving considerable hardware costs, we propose heuristics to solve the joint wavebanding node placement and RFBSA problem, given the total number of available WSSs for the network as a budget.

In the last part of the dissertation, we try to further improve the spectrum efficiency in multi-fiber EONs. We first propose an ILP model for the network planning based on topology and traffic pattern information. For each arriving request, one of the candidate paths for each source-destination node pair will be selected according to probabilities precomputed from the ILP. Then we utilize a dedicated partition scheme which provides a particular spectrum range for each request size. A next state aware spectrum assignment algorithm with resource sharing among partitions is proposed. Each scheme performs well in improving the spectrum efficiency and shows a huge performance enhancement jointly.

Chapter 2 Co-scheduling Computational and Networking Resources in Elastic Optical Networks

In this chapter, we address the problem of *co-scheduling* in elastic optical networks. We consider the multi-job multi-task *co-scheduling* problem to minimize the makespan of traffic demands in OOFDM-based elastic optical networks [49].

2.1 Related Work

To the best of our knowledge, this is the first study on this topic. Work exists in the literature on *co-scheduling* in WDM-based optical networks. The authors of [50] study the feasibility of virtualized optical network (VON) service for workflow applications. They propose a computation and communication delay-aware rescheduling (C^2DAR) scheme based on a new defined Scheduled Result Graph (SRG) concept to allocate computing resources and the shared VON resource. An availability-driven scheduling scheme is proposed in [51], which improves the directed acyclic graph (DAG) applications' availability iteratively by allocating two copies of one communication task to two disjoint lightpaths for data transfer while satisfying application deadline requirements. In [52], a Genetic Algorithm (GA) based approach is presented to co-schedule resources with the objective of minimizing application completion time. The authors of [53] propose a service-oriented energy-efficient Internet architecture. It introduces an energy-aware analytical model and algorithms that cooperatively optimize the selection and scheduling of resources such that the overall power consumption by both the network and IT resources is minimized. In [54], the *co-scheduling* problem in lambda-grids for advance reservation requests is studied, with the aim to minimize the job blocking probability. Our problem is different from these existing work because of the multiple subcarrier allocation feature in elastic optical networks. Specifically, the multiple subcarriers (a subcarrier *band*) for a service are typically allocated in a contiguous manner, as blocks allocated to two different services must be separated by a guardband in order to avoid interference [7].

2.2 Network Model and Problem Statement

We model the system as follows. Informally speaking, there is a network of nodes (with collocated and computing (VMs) and switching resources) interconnected by optical links. A job consists of multiple inter-dependent tasks and each task is assigned a number of VMs on a node. After a task finishes processing, it communicates with its "child" tasks and transfers its results over the network's links. A task cannot start executing until all of its "parent" tasks have finished executing and their results have been transferred to the node assigned to the task. A job is considered to be finished if all of its tasks finish execution. The problem is to allocate resources (VMs and subcarriers) so that an objective is met. We formalize this with some notation below.

The physical network topology is $G(V, E)$, where V is the set of nodes, E is the set of links, and $N = |V|$ and $L = |E|$. Each physical node $v \in V$ has H_v virtual machines (VMs). Each link has two fibers with opposite directions, and each fiber has multiple subcarriers (F), with each subcarrier having bit rate C Gbps (a fixed modulation scheme is assumed). A guardband consisting of multiple subcarriers (G) is placed between two adjacent subcarrier bands assigned to different connections. The shortest path $p_{s,d}$ (based on hop) for each pair of nodes (s, d) is precomputed. We assume a time-slotted system, i.e., each computation or communication takes up an integer number of time slots (which is variable, depending on the number of allocated VMs and subcarriers).[1]

Each job j is represented as a directed acyclic graph (DAG) $G^j(I^j, Q^j)$, where I^j is the set of tasks of job j. If there is a link from task i to task r, it means the following: task i^j must finish and transfer its results to task r^j before the task r^j can start processing. Each task $i^j \in I^j$ has a workload requirement (w_i^j), which is represented as the needed processing time slots of the task when assigned *one* VM. There is a data transfer size D_q^j associated with each link of Q^j; this represents the number of time slots needed to transfer the result over link q (from the task at the head of the DAG link to the task on the tail of the link) when assigned *one* subcarrier.

[1]We use a time-slotted system to facilitate the development of the ILP in Section III.

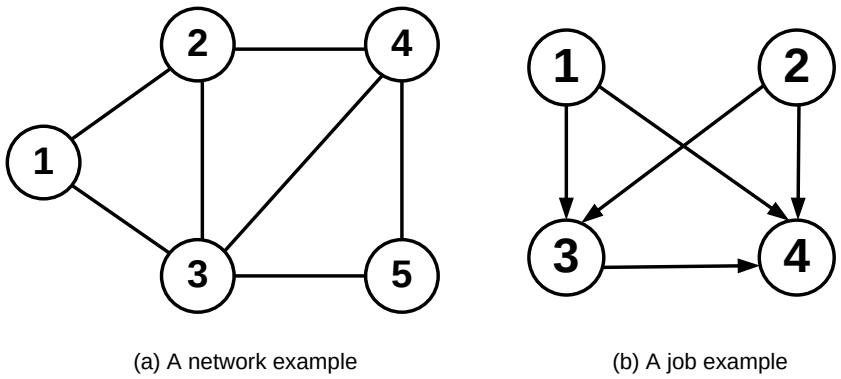

Figure 2.1: Examples of a network graph and job DAG.

Fig. 2.1 shows a network example and a job example, respectively.

We consider the problem of allocating computational resources (VMs) and communicational resources (subcarriers) for each task. Consider a job; for each of its tasks, a physical node v must be assigned to it, and a number of VMs (say, k) on that physical node should be allocated to the task during its execution. The processing time is then calculated as $\left\lceil \frac{w_i^j}{k} \right\rceil$ slots (since k VMs are allocated). Further, a physical path p should be assigned to each link in Q^j, and a band (of size B, say) of contiguous subcarriers on each physical link of the path should be allocated to transfer the results; the transfer time is then calculated as $\left\lceil \frac{D_q^j}{B} \right\rceil$. For two dependent tasks $i^j, r^j \in I^j$ (i.e., there is a link in Q_j between these tasks in the DAG), if the two tasks are assigned to the same physical node, there is no need to allocate subcarriers to transfer the result of task i^j to r^j, and the communication time is 0. Note that after a task finishes processing at a physical node, the results can be stored temporarily at the node for some time before being transmitted to its successor tasks, i.e., data transfer does not have to immediately follow the completion of a task's execution. Similarly, after the results from parent tasks (of a given task) have arrived at the assigned physical node of the task, they can be stored temporarily for some time before the current task starts processing. In this work, we do not take into account the setup delay for each transmission, which could be viewed as a constant.

For the static case, a set of jobs is given, and the objective is to minimize the makespan (completion time of the job that finishes last) [55–63]. For the dynamic

case, the jobs arrive in a random manner, and each job has a deadline requirement. If a job cannot finish before the deadline, it is blocked. The objective is to minimize the *job blocking*.

2.3 Scheduling Algorithms

In this section we introduce our algorithms for job *co-scheduling*. We first present an ILP formulation for static requests that can be used to solve small problem instances. We then present two heuristics for the static case, and later adapt them to the dynamic case.

2.3.1 ILP Formulation

In the ILP formulation, the input parameters are shown in Table 2.1.

$$\text{Objective: Minimize } \max_{i,j} \sum_{x=1}^{\Omega} x g_{i,x}^{j} + \sum_{v=1}^{N} \sum_{k=1}^{H} (\left\lceil \frac{w_i^j}{k} \right\rceil - 1) a_{i,v,k}^{j}$$

Variables:

a)

$$a_{i,v,k}^{j} = \begin{cases} 1, & \text{if task } i \text{ of job } j \text{ is assigned to node } v \text{ and allocated } k \text{ VMs;} \\ 0, & \text{otherwise} \end{cases}$$

b)

$$g_{i,x}^{j} = \begin{cases} 1, & \text{if task } i \text{ of job } j \text{ start processing time slot is } x; \\ 0, & \text{otherwise} \end{cases}$$

c)

$$A_{i,v,t,k}^{j} = \begin{cases} 1, & \text{if task } i \text{ of job } j \text{ use } k \text{ VMs on node } v \text{ at time slot } t; \\ 0, & \text{otherwise} \end{cases}$$

Table 2.1: Notation for the *co-scheduling* problem

Symbol	**Meaning**
N	number of nodes in the network
L	number of links in the network
Ω	estimated upper bound of make span
J	number of jobs
F	number of subcarriers per fiber
j	an arbitrary job
I^j	the set of tasks in job j
i^j	an arbitrary task in job j, $\in I^j$
v	an arbitrary network node
e	an arbitrary network link
p	an arbitrary path ($p_{s,d}$ is the path from node s to node d)
$\xi^e_{p_{s,d}}$	$= 1$ if link e is on path $p_{s,d}$; $= 0$, otherwise
H_v	the maximum number of VMs at node v
H	$\max_v H_v$
w^j_i	workload of task i of job j
S^j_i	the start processing time slot of task i of job j
T^j_j	the finish processing time slot of task i of job j
$D^j_{i,r}$	the data transfer requirement from task i to task r of job j, $= 0$ if there is no data transfer from i to r
$\bar{D}^j_{i,r}$	$= 1$, if the results of task i need to be sent to task r of job j
$B^j_{i,r}$	the number of subcarriers allocated to transfer results from task i to task r of job j
$E^j_{i,r}$	start time slot for transferring results from task i to task r of job j
$X^j_{i,r}$	finish time slot for transferring results from task i to task r of job j

d)

$$\phi^j_{i,r,y,m} = \begin{cases} 1, & \text{if the first subcarrier index allocated for communication} \\ & \text{from task } i \text{ to task } r \text{ of job } j \text{ is } y \text{ and the number of} \\ & \text{allocated subcarriers is } m; \\ 0, & \text{otherwise} \end{cases}$$

e)

$$u^j_{i,r,l} = \begin{cases} 1, & \text{if the start transmission time slot for communication} \\ & \text{from task } i \text{ to task } r \text{ of job } j \text{ is } l; \\ 0, & \text{otherwise} \end{cases}$$

f)

$$o^j_{i,r,c,t,e} = \begin{cases} 1, & \text{if the result transmission from task } i \text{ to task } r \text{ of job } j \\ & \text{use subcarrier } c \text{ on link } e\text{at time slot } t; \\ 0, & \text{otherwise} \end{cases}$$

g)

$$z^j_{i,r,v} = \begin{cases} 1, & \text{if the task } i \text{ and its child task } r \text{ of job } j \\ & \text{are both allocated to node } v; \\ 0, & \text{otherwise} \end{cases}$$

Constraints:

a) Each task of each job can be assigned to only one node and with fixed VM allocation. For all j, i

$$\sum_{v=1}^{N} \sum_{k=1}^{H_v} a^j_{i,v,k} = 1$$

b) Constraint to find the value of $z^j_{i,r,v}$. For all j, i, r, v,

$$\sum_{k=1}^{H_v} a^j_{i,v,k} + \sum_{k=1}^{H_v} a^j_{r,v,k} \leq 1 + z^j_{i,r,v}$$

$$\sum_{k=1}^{H_v} a_{i,v,k}^j + \sum_{k=1}^{H_v} a_{r,v,k}^j \geq 2z_{i,r,v}^j$$

c) Each task of each job can start processing at only one time slot. For all i, j,

$$\sum_{x=1}^{\Omega} g_{i,x}^j = 1$$

d) The constraint to get value of $A_{i,v,t,k}^j$ based on the value of $g_{i,x}^j$ and $a_{i,v,k}^j$. For all $j, i, v, k, t, t - \lceil \frac{w_i^j}{k} \rceil + 1 \leq x \leq t$,

$$g_{i,x}^j + a_{i,v,k}^j \leq 1 + A_{i,v,t,k}^j$$

$$g_{i,x}^j + a_{i,v,k}^j \geq 2A_{i,v,t,k}^j$$

e) Each node's VM capacity cannot be exceeded for any time slot. For all v, t,

$$\sum_{j=1}^{J} \sum_{i \in I^j} \sum_{k=1}^{H} k A_{i,v,t,k}^j \leq H_v$$

f) Each transmission from task i to r can start at only one time slot. For all i, r, j,

$$\sum_{l=1}^{\Omega} u_{i,r,l}^j = 1$$

g) Each transmission has only one start subcarrier index and has a certain number of allocated subcarriers. For all i, r, j,

$$\sum_{y=1}^{F} \sum_{m=0}^{F} \phi_{i,r,y,m}^j = 1$$

h) For each task, its communication start time slot cannot be earlier than its finish processing time slot. For all i, r, j,

$$\sum_{x=1}^{\Omega} x g_{i,x}^j + \sum_{v=1}^{N} \sum_{k=1}^{H} (\lceil \frac{w_i^j}{k} \rceil - 1) a_{i,v,k}^j < \sum_{l=1}^{\Omega} l u_{i,r,l}^j$$

i) A task should start after the data transfer from its preceding/parent tasks is completed. For all i, r, j,

$$\sum_{l=1}^{\Omega} l u_{i,r,l}^j + \sum_{m=1}^{F} \sum_{y=1}^{F} (\lceil \frac{D_{ir}^j}{m} \rceil - 1) \phi_{i,r,y,m}^j - \sum_{x=1}^{\Omega} x g_{r,x}^j < \Lambda(1 - \bar{D}_{ir}^j + \sum_{v=1}^{N} z_{i,r,v}^j)$$

where Λ is a very large number ($> \max_{i,r,j} D_{ir}^j$).

j) Constraint to get the value of $o^j_{i,r,c,t,e}$. For all $j, i, r, e, c, t, c-m+1 \leq y \leq c, t - \lceil \frac{D^j_{ir}}{m} \rceil + 1 \leq l \leq t$,

$$\xi^e_{p_{n_1,n_2}} + u^j_{i,r,l} + \phi^j_{i,r,y,m} + \sum_{k=1}^{H} a^j_{i,n_1,k} + \sum_{k=1}^{H} a^j_{r,n_2,k} \leq 4 + o^j_{i,r,c,t,e}$$

$$\xi^e_{p_{n_1,n_2}} + u^j_{i,r,l} + \phi^j_{i,r,y,m} + \sum_{k=1}^{H} a^j_{i,n_1,k} + \sum_{k=1}^{H} a^j_{r,n_2,k} \geq 5o^j_{i,r,c,t,e}$$

k) Each link's subcarrier c can be allocated to at most one data transfer. For all e, t, c,

$$\sum_{j=1}^{J} \sum_{i \in I^j} \sum_{r \in I^j} o^j_{i,r,c,t,e} \leq 1.$$

Current formulation needs the collaboration between networks and data centers. Exact information must be interchanged between networks and data centers. This could be justified for some large cloud service companies while more constraints should be imposed for small distributed data centers connected by different network operators. To simplify the problem, we just assume that there is a global scheduler which collects information from both the network and data center side.

2.3.2 Static Case

Two heuristics based on List Scheduling are proposed. Both of them allocate computational and communication resources to jobs one by one. The first algorithm (First Fit) schedules the jobs and tasks in each job by the order of their index. The second algorithm (Children Aware) schedules them by decreasing order of their weight (the method to find a job's weight and a task's weight will be shown later). Two global states are maintained: one is $h_v(t)$, the number of free VMs on physical node v at time slot t; and the other is $s^c_e(t)$, which equals 1 if subcarrier c on physical link e is available at time slot t and is 0 otherwise.

Before describing the algorithm details, we introduce the notion of *layer*. Each job's tasks are categorized into several layers – the first layer consists of tasks that have no preceding tasks. The tasks of layer two are those not in layer one and all of their preceding tasks are in layer one. In general, the tasks of layer η are the ones not in layers $\eta' < \eta$, and at least one of their preceding tasks is in layer $\eta - 1$.

2.3.2.1 First Fit (FF)

In the First Fit algorithm, the jobs are numbered (in the order they are generated) and considered from smallest index to largest index. For each layer of a job, from the lowest layer to the highest layer, the tasks are also considered by increasing order of index. Let I_η denote the set of tasks in Layer η.

Layer One: For each start processing time slot S_i^j of task i, the network nodes with $h(S_i^j) > 0$ are considered one by one. For each node n, the number of potentially allocatable VMs k_i^j are checked one by one, from $\min(h_n(S_i), w_i^j)$ to 1. After this scheduling, the corresponding $h_v(t)$ states are updated, as shown in Algorithm 1.

Higher Layers: For each task, the algorithm considers the task's start processing time slot S_i^j, and the nodes with $h(S_i^j) > 0$ are considered one by one. Let U_i^j be the set of preceding tasks of task i^j, then $S_i^j \geq \max_{i' \in U_i^j} T_{i'}^j$; they are equal when task i is assigned to the same node as that preceding task (and there is no waiting time before task i's processing). For each possible S_i^j and n_i^j, we check the condition that all of its preceding tasks (note that all tasks in U_i^j have been assigned already) can finish their data transfer to task i no later than S_i^j. For $q_{i',i}^j$, the number of allocated subcarriers is selected as the one that achieves the minimum transfer finishing time $X_{i',i}^j$. The allocation method we use is First-Fit, i.e., the first available contiguous subcarrier band on the corresponding path is allocated. If the condition is satisfied, then consider the number of allocated VMs k_i^j. After this scheduling, the corresponding $h_v(t)$ and $s_e^c(t)$ states are updated, as shown in Algorithm 2.

2.3.2.2 Children-Aware (CA)

In the Children-Aware algorithm, each task and each job is assigned a weight. To find the weight of each task $i \in I^j$, a directed subgraph of the job j is considered; the top node of the subgraph is task i, and all tasks of higher layers which are children of i are included the subgraph. For example, the subgraph with task 1 of the job in Fig. 2.1 (b) is shown in Fig. 2.2. Let R_i denote the set of tasks of the subgraph which are directly connected to task i (in the example, task 3 is the only task of R_1). The

Algorithm 1 First Fit: Allocate Layer One

for Task $i = 1, 2, \ldots, |I_1|$ **do**
  $T_i' = \infty$
  for Start processing time $S_i = 1, 2, \ldots, \Omega$ **do**
    if $T_i' < \infty$ **then**
      Break
    end if
    Keep the physical nodes with $h_v(S_i) > 0$, get set V'
    for Node $n_i = 1, 2, \ldots, |V'|$ **do**
      if $T_i' < \infty$ **then**
        Break
      end if
      for VM $k_i = \min(h_n(S_i), w_i^j), \ldots, 1$ **do**
        Calculate $T_i = S_i + \lceil \frac{w_i}{k_i} \rceil - 1$
        if $T_i < T_i'$, and $h_n(t) \geq k_i$ for $S_i \leq t \leq T_i$ **then**
          $T_i' = T_i$
          Break
        end if
      end for
    end for
  end for
  Assign task i to the n_i, k_i, S_i that achieves T_i' and update $h_n(t) = h_n(t) - k_i$ for $S_i \leq t \leq T_i'$
end for

Algorithm 2 First Fit: Allocate Layer η

for Task $i = 1, 2, \ldots, |I_\eta|$ **do**
  $T_i' = \infty$
  for Start processing time $S_i = \max_{i' \in U_i} T_{i'}, \ldots, \Omega$ **do**
    if $T_i' < \infty$ **then**
      Break
    end if
    Keep the physical nodes with $h_v(S_i) > 0$, get set V'
    for Node $n_i = 1, 2, \ldots, |V'|$ **do**
      if $T_i' < \infty$ **then**
        Break
      end if
      $U_i' = U_i$
      for $i' = 1, 2, \ldots, |U_i'|$ **do**
        Minimum finish transferring time $\bar{X}_{i',i} = \infty$
        for Start transferring time $E_{i',i} = T_{i'}, \ldots, \Omega$ **do**
          for Number of subcarriers $B_{i',i} = 1, 2, \ldots, F$ **do**
            Finish transferring time $X_{i',i} = E_{i',i} + \lceil \frac{D_{i',i}}{B_{i',i}} \rceil - 1$
            if $X_{i',i} < \bar{X}_{i',i}$, $X_{i',i} \leq S_i$, and there is contiguous available subcarrier band with size $\geq B_{i',i}$ on path from $n_{i'}$ to n_i **then**
              $\bar{X}_{i',i} = X_{i',i}$
            end if
          end for
        end for
      end for
      if For all i', $\bar{X}_{i',i} < \infty$ **then**
        for VM $k_i = \min(h_n(S_i), w_i^j), \ldots, 1$ **do**
          Calculate $T_i = S_i + \lceil \frac{w_i}{k_i} \rceil - 1$
          if $T_i < T_i'$, and $h_n(t) \geq k_i$ for $S_i \leq t \leq T_i$ **then**
            $T_i' = T_i$, Break
          end if
        end for
      end if
    end for
  end for
  Assign task i to the n_i, k_i, S_i that achieves T_i', update $h_n(t) = h_n(t) - k_i$ for $S_i \leq t \leq T_i'$
  Allocate First-Fit subcarriers that achieves $\bar{X}_{i',i}$ and update corresponding $s_e^c(t)$
end for

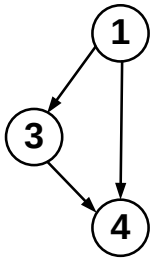

Figure 2.2: Example of subgraph.

weight of task i is defined as,

$$\theta_i = \max_{r \in R_i}(\alpha w_i + \beta D_{i,r} + \theta_r). \tag{2.1}$$

where α and β are suitable constants, and the other variables are as defined in Table I. Parameters α and β are chosen according to the ratio of computation resources to communication resources. The weight of job j is set as $\theta^j = \max_{i \in I_0} \theta_i$ (recall that I_0 is set of tasks of layer 0).

The first step of the CA algorithm is to sort the jobs by decreasing order of their weights θ^j. Then the jobs are considered one by one by this order. For each layer of a job, the tasks of the current layer are sorted by decreasing order of weight θ_i, and the allocation is done in this order.

Two changes are made to Algorithm 1 and Algorithm 2: (a) The Break part in lines 4-6 and 9-11 (both Algorithm 1 and Algorithm 2) are removed, and (b) In line 12 of Algorithm 2, U_i is first sorted in decreasing order of data transfer size to task i; and this ordered set is U_i'.

The worst case time complexity for both algorithms are the same: $J(IwNH_v + I^2D^2NF)$. Here, I denotes the maximum number of tasks in one job, w denotes the maximum task workload, and D denotes the maximum data transfer requirement.

2.3.3 Dynamic Requests

For the dynamic case, we assume the following model. At each time slot there is a probability p_{new} that a new job will arrive during this time slot. Thus the tasks of the job can start processing from the next time slot. Besides the workload and transfer size requirements, each job has a deadline requirement. If the algorithm determines

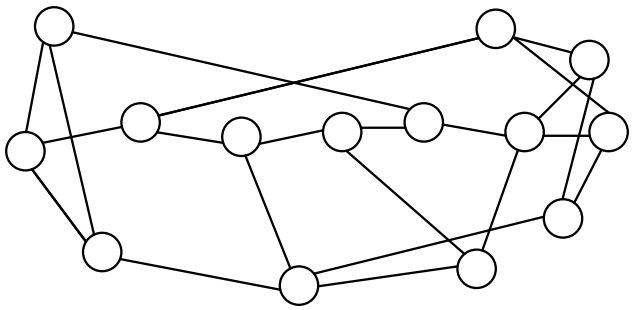

Figure 2.3: NSF network.

that some of the tasks of the job cannot finish before the deadline, the job is blocked and no resources are allocated to it. We adapt the previous two heuristics to the dynamic case.

At each time slot, the algorithms check if a new job arrives during this time slot; if so, schedule it; otherwise, go on to consider the next time slot. Note that for a job that arrives at time slot τ, its tasks may start execution at a later time slot τ', as long as all of its tasks can finish before the deadline.

2.4 Simulation Results

We present results for two network topologies, a small 5-node network shown in Fig. 2.1 (a) and the larger NSFNET network, shown in Fig. 2.3. Each subcarrier being 12.5 GHz (according to standardization work related to the flexible grid specification is on-going in ITU-T [64]), the data rate for QPSK is 12.5 Gbps per subcarrier. In addition, the guardband is assumed to be $G = 2$ subcarriers. For each pair of network nodes, the shortest path (in terms of hops) is used.

2.4.1 Static Requests

We first present results for the static case. Due to the high complexity of the ILP, we are only able to obtain results for the small network. There are 5-10 VMs per network node. Each fiber has 12 subcarriers. Job requests are generated as follows. Each job has between 2 and 4 tasks (uniformly random), and a result transfer requirement exists from task i to another task r, $r > i$ (of the same job) with probability 0.8 (we discard any task graph that is not connected). Each task's workload is a random

Table 2.2: Performance of ILP and algorithms for 5-node network: Static requests

No. of Jobs	**First Fit (FF)**	**Children Aware (CA)**	**ILP**
2	5	4	4
3	7	7	6
4	9	8	7
5	11	10	7
6	11	8	7
7	11	10	8

number between 5 and 15. Each transfer size is uniformly distributed over the range 15-20. In our case, since subcarrier resources are more abundant than VM resources, we choose $\alpha = 0.5$ and $\beta = 1$ for the Children-Aware algorithm. Results for 2-7 job requests are shown in Table 2.2.

From these results, we observe that, in general, the CA algorithm performs better than FF. In addition, their results are quite close to those achieved by the ILP. For the NSFNET network, there are 50-100 VMs per network node [65], and each fiber has 320 subcarriers. Job requests are generated as follows. Each job has between 5 and 10 tasks, and a result transfer requirement exists from task i to another task r with probability 0.5. Each task's workload is a random number between 50 and 150. Each transfer size is uniformly distributed over the range 250-350. Results for the two heuristics are presented in Table 2.3. These results confirm our earlier observations. The execution times (ET) in minutes for the two heuristics are also given in the same table. We see that CA performs better for large workloads with reasonable time complexity.

2.4.2 Dynamic Requests

For the dynamic case, at each time slot, there is a new job arrival with probability 0.5. Each job's deadline Γ is set as $\Gamma = \max_{i \in I^j} \lceil \theta_i \rceil$, where θ_i is calculated by Equation (2.1), but with $\alpha = 0.01$, $\beta = 0.01$ (different from the values used to calculate the task's weight), which estimates the job makespan by assuming 100 VMs

Table 2.3: Performance of algorithms for NSFNET: Static requests

No. of Jobs	FF	ET (FF)	CA	ET (CA)
50	159	8	42	20
100	159	18	82	53
150	204	34	123	93
200	204	50	162	209
250	268	63	202	400
300	324	77	244	427

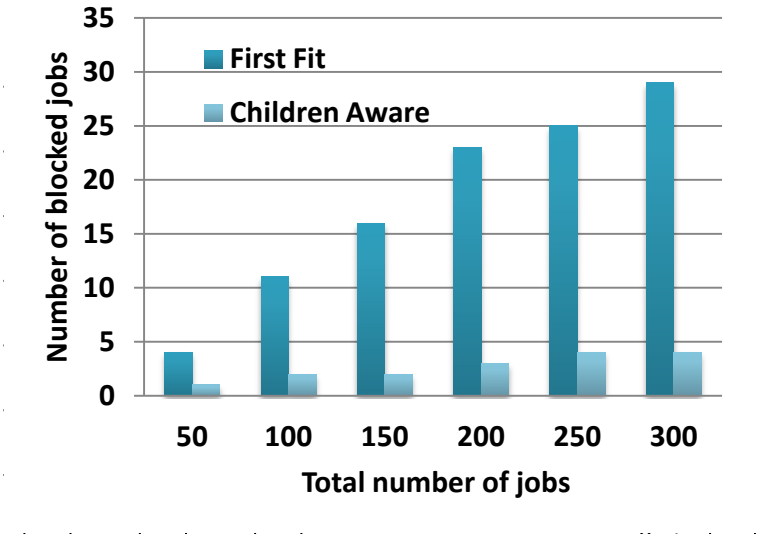

Figure 2.4: Number of blocked jobs for dynamic traffic.

to each task and 100 subcarriers to transfer results between tasks. The parameters of the network and jobs are the same as in the static case. Results for the heuristics are shown in Fig. 2.4. The execution times for the heuristics are shown in Table 2.4. Once again, the superior performance of CA can be seen.

2.5 Conclusions

In this chapter, we investigated the problem of *co-scheduling* computational and networking resources for both static and dynamic multi-task jobs in elastic optical networks. We proposed two heuristics, and presented representative simulation results.

Table 2.4: Execution time in minutes for NSFNET: Dynamic requests

No. of Jobs	**Execution time (FF)**	**Execution time (CA)**
50	7	25
100	14	56
150	23	78
200	32	115
250	34	142
300	49	210

According to the results, for the static case, in general, the Children-Aware algorithm has better makespan performance than the First Fit algorithm. For the dynamic case, the adapted Children-Aware algorithm has good blocking performance. Since we give weights to tasks according to the task dependencies, tasks which have more computation requirement and initiate more transmission requirements are expected to schedule first.

Chapter 3 Optimal Nonuniform Wavebanding in WDM Mesh Networks

This chapter investigates the multi-granularity in WDM networks. The scheduling of wavelengths into wavebands without uniform wavebanding constraint to save switching costs in WDM mesh networks is studied.

3.1 Related Work

There are two types of waveband switching – uniform waveband switching and nonuniform waveband switching [66]. For uniform waveband switching, the sizes of wavebands at a node are all the same and this common waveband size is used for all network nodes. Optical port-saving under uniform waveband switching is studied in [67]. A uniform waveband design problem in mesh networks is studied in [68]. In nonuniform waveband switching, the sizes of bands could be different for each node of the network. Utilizing nonuniform waveband switching is more advantageous in saving switching elements for WDM networks [69]. The authors in [69] first consider a star topology with a single source node and then extend their results to ring topologies.

Previous work [66] which investigated nonuniform wavebanding focusing on a single switching node has been demonstrated to be not suitable for an entire network [70]. The authors in [70] focus on optimal waveband design in ring networks, and propose a novel framework for band minimization. In this chapter, we study a more complicated case, which is the optimal waveband design problem in mesh networks, without the restriction of uniform wavebanding.

3.2 Network Model and Definitions

Consider a mesh network with N nodes. The nodes are numbered as $1, 2, \cdots, N$. Given a traffic as a set of lightpaths, a Routing and Wavelength Assignment (RWA) algorithm is used to provision the lightpaths. Suppose W wavelengths are assigned to the given traffic after routing and wavelength assignment. The node degree (excluding local add/drop) for node n is d_n. We can construct a "code" for each node n and

concatenate the codes for all nodes to generate a code for each wavelength.

The principles for a code are as follows. There are $\binom{d_n}{2} + d_n = \frac{d_n(d_n+1)}{2}$ bits in the code for node n. We assume the connection requests are bidirectional, and the code permutations for I-O switching are not considered here. For every input-output pair in a node, $(I_i, O_j), i, j = 1, 2, ..., d_n$, we have one bit that is 1 if the wavelength needs to be switched between I_i and O_j, and 0 otherwise. There are $\binom{d_n}{2}$ such pairs for each node. d_n more bits are appended to the code – the ith bit in the d_n bits is 1 if the wavelength for port i is added or dropped. Thus, a code of length $\frac{d_n(d_n+1)}{2}$ fully encodes the behavior of a single wavelength at node n. Concatenating such codes for a wavelength at nodes 1 through N makes up the row for that wavelength in the RWA matrix. Let $R = \sum_{n=1}^{N} \frac{d_n(d_n+1)}{2}$. Thus the RWA matrix has W rows and R columns. Let r_w denote the row vector of the RWA matrix corresponding to wavelength w, and let c_v denote the column vector corresponding to bit v in the concatenated code.

We observe that the runs of 1's in a specific column, say c_n, represent opportunities for wavebanding at a specific input-output pair or at an add/drop port at a node. If the location of c_n corresponds to an input-output pair (I_i, O_j) at a node, a string of consecutive 1's in the column means that all of those consecutive wavelengths are bypassed between port i and port j, and hence can be switched as a single band. Similarly, if the location of c_n corresponds to the ith add/drop port at a node, a string of consecutive 1's in this column means that all of those consecutive wavelengths are added/dropped at port i. Thus, the number of bands at a node can be obtained by counting the strings of consecutive 1's in each column which corresponds to a bit of the node's code and adding them together. The total number of bands in the network can be obtained by counting the strings of consecutive 1's in each column and summing them up through all columns.

For each column of the RWA matrix, we observe that when the transition between the first row and the second row is (0→1), (1→0) or (1→1), the number of bands is one; in the case of (0→0), the number of bands is zero. From the second row on, if the transition between two adjacent rows is (0→1), the number of bands increases by 1 which means that a new band begins; otherwise, the number of bands remains

unchanged. It is obvious that in the case of (0→0), the number of bands remains unchanged. For (1→0) and (1→1) cases, the 1's belong to the previous bands which have already been counted, thus the number of bands remains unchanged. In this case, we can define the distance between two rows of one column as below. For each column c, we have

$$d_{1,j}^{c} = \begin{cases} 1 & \text{if row 1 to row } j \text{ is } (0\to1),\ (1\to0) \text{ or } (1\to1); \\ 0 & \text{if row 1 to row } j \text{ is } (0\to0), \end{cases}$$

where $j = 2, \cdots, W$.

$$d_{i,j}^{c} = \begin{cases} 1 & \text{if row } i \text{ to row } j \text{ is } (0\to1); \\ 0 & \text{if row } i \text{ to row } j \text{ is } (0\to0),\ (1\to0) \text{ or } (1\to1), \end{cases}$$

where $i = 2, \cdots, W; j = 2, \cdots, W; i \neq j$. Then the number of bands for a column is $\beta_c = \sum_{i=1}^{W-1} d_{i,(i+1)}^{c}$. For example, the number of bands for the column $\begin{pmatrix} 1 & 0 & 0 & 1 & 1 \end{pmatrix}^{\mathrm{T}}$ is 2.

The total number of bands in the RWA matrix is equal to the sum of the number of bands over all columns. We use B to denote the total number of bands in the RWA matrix.

$$B = \sum_{c=1}^{R} \beta_c = \sum_{c=1}^{R} \sum_{i=1}^{W-1} d_{i,(i+1)}^{c} = \sum_{i=1}^{W-1} \sum_{c=1}^{R} d_{i,(i+1)}^{c}. \tag{3.1}$$

Since $d_{i,j} = \sum_{c=1}^{R} d_{i,j}^{c}$ is the distance between two rows i and j, we have $B = \sum_{i=1}^{W-1} d_{i,(i+1)}$, which means that the total number of bands is equal to the summation of the distances between adjacent rows (where distances are defined as above). The Band Minimization Problem (BMP) is to minimize this number for a given RWA by re-ordering the rows of the RWA matrix (equivalent to re-assigning the wavelengths to lightpaths).

We can then transform this problem into a Minimum Weight Hamiltonian Path Problem ([71, 72]) with a constraint that the first row is chosen. We can construct a graph $G = (V, E)$ consisting of W nodes, each corresponding to one row/wavelength of the RWA matrix. Every node of this graph is connected to every other node with an edge weighted as the distance between the corresponding rows. Hence, the edge

(v_i, v_j) has the weight $d_{i,j}$ for every i and j. $d_{i,j}$ could be different from $d_{j,i}$ according to the definition of distance above. The graph in this case is a directed graph with edge weights being integers. Then, the Hamiltonian path that visits every node in this graph with the minimum total weight gives the minimum total number of bands of the RWA matrix. Due to the definition of distance clarified above, different first rows/nodes give different link weights in the graph. The Hamiltonian path should start from the node corresponding to the chosen first row for that graph. Every row in the RWA matrix could be chosen as the first row, so we need to construct W different graphs corresponding to W different first rows and get solutions for each graph. The minimum solution among the W solutions is the desired one.

3.3 Band Minimization ILP And Algorithms

In this section, we first present an ILP formulation that can be used to solve small problem instances, and then present several heuristics that can be used to solve larger problem instances.

Given a set of traffic requests, we first use Dijkstra's algorithm to assign shortest paths and the First-Fit algorithm to assign wavelengths to the given traffic requests. For each connection request, the first available wavelength is assigned to it; if no existing wavelength can serve this request, a new wavelength is used. The RWA matrix (Δ) is then generated according to assignment results and the rules of code generation in the above section. Assume that W wavelengths are used. The RWA matrix may include several empty elements (*don't cares*). To complete the matrix, each *don't care* is assigned the same value as the one immediately above it in the same column. If the *don't care* occurs in the first row, it is assigned the value of 0. The new matrix after assigning *don't cares* is denoted as $\overline{\Delta}$. Assume that there are two lightpaths between every pair of nodes in the small network shown in Figure 3.1. The port ids are shown around each node in the network. There are 4 nodes in the network with degrees 3, 2, 2, 3, respectively. Correspondingly, there are 6, 3, 3, 6 bits for each node resulting in 18 bits for each wavelength. Figure 3.2 shows an RWA example for the given traffic. The corresponding RWA matrix is generated as in Table 3.1. The entry (i, j) for a

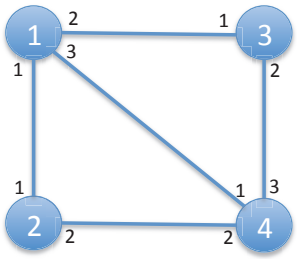

Figure 3.1: A 4-node mesh network example.

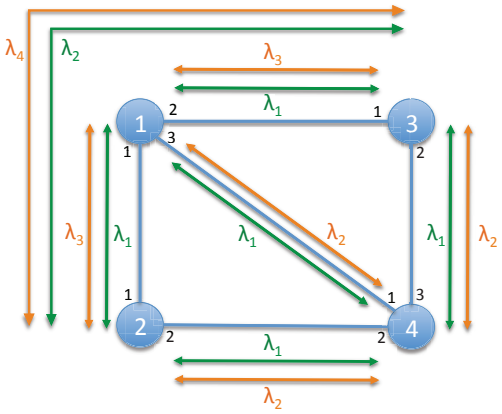

Figure 3.2: A routing and wavelength assignment example for given traffic.

Table 3.1: RWA Matrix Generation.

	Node 1						**Node 2**			**Node 3**			**Node 4**					
	(1,2)	(1,3)	(2,3)	1	2	3	(1,2)	1	2	(1,2)	1	2	(1,2)	(1,3)	(2,3)	1	2	3
λ_1	0	0	0	1	1	1	0	1	1	0	1	1	0	0	0	1	1	1
λ_2	1	0	0	0	0	1	0	1	1	0	1	1	0	0	0	1	1	1
λ_3	0	0	0	1	1	×	0	1	×	0	1	×	×	×	×	×	×	×
λ_4	1	0	0	0	0	×	0	1	×	0	1	×	×	×	×	×	×	×

node denotes the routing behavior between port i and port j of that node, while the entry k for a node denotes the add/drop behavior of a wavelength at port k of that node. The ×s in the matrix denote the *don't cares*. The generated RWA matrix after *don't cares* assignment is shown in Figure 3.3.

The second step is to construct W directed graphs according to the RWA matrix with each row being the first node to be visited in the Hamiltonian Path. For each graph, a Minimum Weight Hamiltonian Path Problem [73–81] with given first node is to be solved and the desired result is the minimum result among those W solutions.

$$\bar{\Delta}=\begin{bmatrix}0&0&0&1&1&1&0&1&1&0&1&1&0&0&0&1&1&1\\1&0&0&0&0&1&0&1&1&0&1&1&0&0&0&1&1&1\\0&0&0&1&1&1&0&1&1&0&1&1&0&0&0&1&1&1\\1&0&0&0&0&1&0&1&1&0&1&1&0&0&0&1&1&1\end{bmatrix}$$

Figure 3.3: RWA matrix example.

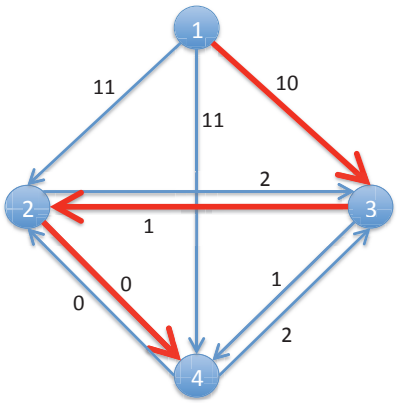

Figure 3.4: Graph generated from $\overline{\Delta}$ with first row chosen as the first node.

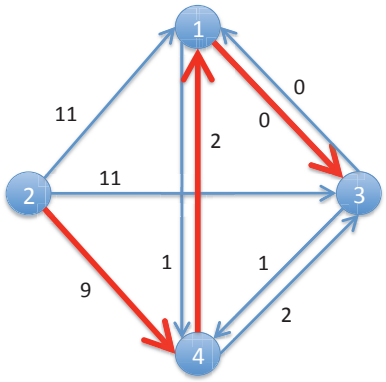

Figure 3.5: Graph generated from $\overline{\Delta}$ with second row chosen as the first node.

Figure 3.4 gives an example of the constructed graph when row 1 in $\overline{\Delta}$ of Figure 3.3 is chosen as the first node. Each node corresponds to a row in $\overline{\Delta}$. The number on each link denotes the link weight, which is the distance between the corresponding two rows. For this graph, a Hamiltonian Path with total weight 11 is chosen as the minimum weight Hamiltonian Path. The path is shown as red lines in the figure. The constructed graph when row 2 in $\overline{\Delta}$ is chosen as the first node and the resulting Hamiltonian Path is shown in Figure 3.5. For each of the other two graphs generated by choosing other rows in $\overline{\Delta}$ as the first node, a Hamiltonian Path can also be chosen. Then the minimum among the four weights is the final result for this example.

We now present an ILP to solve the MWHPP for each of the W constructed graphs.

3.3.1 ILP Formulation

Given a constructed graph with chosen first node and corresponding link weights calculated from the complete RWA matrix, the goal is to find a Minimum Weight Hamiltonian Path Problem [82–89] for this graph. The ILP formulation is as follows.

$$\text{Objective: Minimize } \sum_{i=1}^{W} \sum_{j=1, j\neq i}^{W} x_{i,j} \cdot d_{i,j},$$

where $d_{i,j}$ is the link weight between node i and j.

Variables:

a)

$$x_{i,j} = \begin{cases} 1, & \text{if node } j \text{ is visited immediately following node } i; \\ 0, & \text{otherwise} \end{cases}$$

b)

$$a_{k,i} = \begin{cases} 1, & \text{if node } i \text{ is visited at step } k; \\ 0, & \text{otherwise} \end{cases}$$

Constraints:

a) Each node must be visited exactly once.

$$\sum_{k=1}^{W} a_{k,i} = 1, \forall i = 1, 2, \ldots, W.$$

b) Only one node is visited at each step.

$$\sum_{i=1}^{W} a_{k,i} = 1, \forall k = 1, 2, \ldots, W.$$

c) If node j is visited immediately following node i, then node j is one step further than node i.

$$\text{if } x_{i,j} = 1, \text{then } \sum_{k=1}^{W} k \cdot a_{k,j} - \sum_{k=1}^{W} k \cdot a_{k,i} = 1, \forall i, j = 1, 2, \ldots, W.$$

d) When a node is visited at the first step, which means that this node is the starting node of the Hamiltonian Path, there should be no node visited before this node. Otherwise, for each node visited at later steps, there should be exactly one node visited just before this node.

$$\sum_{j=1, j \neq i}^{W} x_{j,i} + a_{1,i} = 1, \forall i = 1, 2, \ldots, W.$$

e) When a node is visited at the last step, which is the Wth step, there should be no node visited after this node. Otherwise, there should be exactly one node visited immediately following this node.

$$\sum_{j=1, j \neq i}^{W} x_{i,j} + a_{W,i} = 1, \forall i = 1, 2, \ldots, W.$$

3.3.2 Heuristic Solutions

We now present three heuristic algorithms to solve the MWHPP.

3.3.2.1 Nearest Neighbor (NN)

The RWA matrix with all *don't cares* assigned ($\overline{\Delta}$) is the input to this algorithm. Each row in $\overline{\Delta}$ can be visited as the first node in the path. For each chosen first row, a distance matrix D can be generated according to the definitions in the previous section. The first step is to find the minimum element in the row corresponding to the chosen first row id in the distance matrix. This means that the row having minimum distance from the first row is chosen. For example, if $D_{1,i}$ is the minimum element in the first row of D, then row i is chosen to be the second node in the Hamiltonian path and row i is marked as visited. In the next step, find the minimum element $D_{i,j}$ for unvisited row j's in the ith row of D, then row j is chosen as the next node to be visited in the Hamiltonian path and marked as visited. Repeat the steps until all rows are visited. For each graph generated by each chosen first row, a Hamiltonian path is created. The path with minimum total weight gives the desired order of rows. The outline of this heuristic is shown in Algorithm 3. In Algorithm 3, $BandsCal$ is a function to calculate the total number of bands in a given matrix. $DistanceMatrixGen$ is a function to generate the distance matrix D according to the first row id and the distance calculation rule described in the previous section. The algorithm $NearestNeighbor$ is used to determine new order of the matrix given the $DistanceMatrixGen$ and the first row id.

3.3.2.2 ReassignDC (RDC)

This heuristic is based on the previous Nearest Neighbor heuristic. It is an improvement on NN in terms of reassigning *don't cares* after getting the new order of rows. In this case, the original RWA (Δ) is reordered according to the $NewRowOrder$ and then each *don't care* is assigned the same value as its adjacent row above it in the same column to form Δ'. Then $BandsCal$ is applied to Δ' to give the total number

Algorithm 3 Pseudocode for the Nearest Neighbor heuristic

Require: $\overline{\Delta}$

Ensure: Δ' with reordered rows of $\overline{\Delta}$

Initialization: min_total_bands = BandsCal($\overline{\Delta}$);
for chosen first row id $f = 1, 2, \ldots, W$ **do**
  D = DistanceMatrixGen($\overline{\Delta}$, f);
  [$RowOrder, minTotalWeight$]=NearestNeighbour(D, f);
  if $minTotalWeight < min_total_bands$ **then**
    $min_total_bands = minTotalWeight$;
    $NewRowOrder = RowOrder$;
  end if
end for

Algorithm 4 NearestNeighbor(D, f)

Initialize $OrigRows$ as a set of all row ids;
Insert f into $RowOrder$ as the first row and remove f from $OrigRows$;
$currentRow = f$;
while $OrigRows$ not empty **do**
  $mindis = \infty$;
  for all j in $OrigRows$ **do**
    if $D[currentRow][j] < mindis$ **then**
      $mindis = D[currentRow][j]$;
      $nextRow = j$;
    end if
  end for
  remove $nextRow$ from $OrigRows$ and insert it into $RowOrder$;
  $minTotalWeight = minTotalWeight + mindis$;
  $currentRow = nextRow$;
end while
return $RowOrder, minTotalWeight$;

of bands in this new matrix. The Pseudocode for ReassignDC is in Algorithm 5.

Algorithm 5 Pseudocode for the ReassignDC heuristic

Require: $\overline{\Delta}$

Ensure: Δ' with reordered rows of Δ and reassigned *don't cares*

1: *Initialization*: min_total_bands = BandsCal($\overline{\Delta}$);
2: **for** chosen first row id $f = 1, 2, \ldots, W$ **do**
3: D = DistanceMatrixGen($\overline{\Delta}$, f);
4: $[RowOrder, minTotalWeight]$=NearestNeighbor($D$, f);
5: **if** $minTotalWeight < min_total_bands$ **then**
6: $min_total_bands = minTotalWeight$;
7: $NewRowOrder = RowOrder$;
8: **end if**
9: **end for**
10: Δ' equals to the reordering of Δ according to $NewRowOrder$;
11: Assign *don't cares* in Δ';

3.3.2.3 SortTraffic (ST)

In this heuristic, the traffic requests are first sorted according to their path length in descending order. Then the FirstFit wavelength assignment algorithm is applied to generate the RWA matrix. Thus the RWA matrix and the original number of bands are different from those of the previous two heuristics. After sorting traffic and performing wavelength assignment, the ReassignDC heuristic is applied to this RWA matrix.

The worst case time complexity of the $NearestNeighbor$ algorithm is $O(W^2)$, while that of the ILP is $O(2^{W^2})$. Since we need to run the $NearestNeighbor$ algorithm or the ILP W times for W different first rows, the worst-case time complexity is $O(W^3)$ for all heuristics and $O(W \cdot 2^{W^2})$ for the ILP.

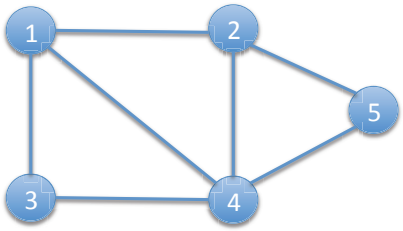

Figure 3.6: A small 5-node network.

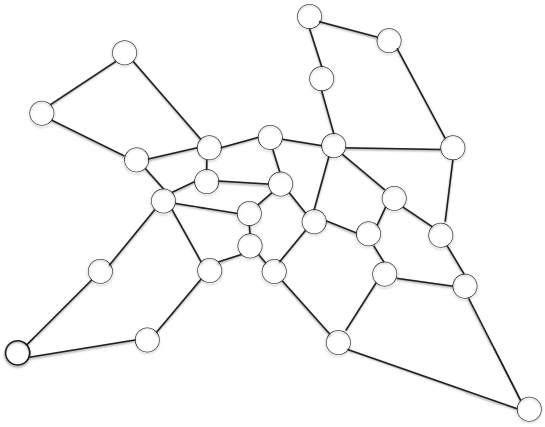

Figure 3.7: 28-Node Pan-European network.

3.4 Simulation Results

We present results for a small 5-node network (Figure 3.6), two real network topologies - the NSFNET network (Figure 2.3) and the pan-European network (Figure 3.7), and a 5×5 regular mesh network. The NSFNET network consists of 14 nodes and 21 links, while the pan-European network consists of 28 nodes and 43 links. For the small 5-node network topology, we compare the total number of nonuniform wavebands calculated by the ILP and heuristics to the total number of switching elements in a wavelength switching architecture. Due to the high complexity of the ILP, we are only able to obtain results for the small network. Therefore, only the results in terms of the total number of nonuniform wavebands achieved from the heuristics are compared to the total number of switching elements in a wavelength switching architecture for the NSF, pan-European and 5×5 regular mesh networks.

The traffic requests are generated as follows. Each request represents a connection between a pair of nodes in the mesh network. The source node and the destination node are uniformly randomly chosen from the node set. Dijkstra's algorithm is applied to find the shortest path between the source and destination nodes.

Results for the small 5-node mesh network are shown in Table 3.2. SWN in the table refers to the number of switching elements used in the original wavelength switching network without banding. For each heuristic, the original number of bands

Table 3.2: Performance of ILP and algorithms for 5-node network.

# of Requests	SWN	NN		RDC		ST		ILP
		Orig. # of bands	Optimized # of bands	Orig. # of bands	Optimized # of bands	Orig. # of bands	Optimized # of bands	
20	43	20	16	20	16	20	16	16
30	64	27	17	27	17	22	18	16
40	87	34	17	34	17	28	18	16

Table 3.3: Performance of algorithms for NSF network.

# of Requests	SWN	NN		RDC		ST	
		Orig. # of bands	Optimized # of bands	Orig. # of bands	Optimized # of bands	Orig. # of bands	Optimized # of bands
50	170	117	115	117	103	120	101
500	1579	865	655	865	596	844	452
5000	15606	8360	4607	8360	3924	7884	2538

(before applying the MWHPP algorithm) and the new number of bands achieved by the heuristic are presented. Since NN and RDC apply the routing and wavelength assignment to the requests in the order that they are generated, the original number of bands are the same for these two heuristics. ST first sorts the requests in descending order of their path length, thus the RWA matrix, and hence the original number of bands, for ST differs from others. From the table, we can see that the results of heuristics are very close to the ILP results, thus demonstrating the effectiveness of the heuristics. Besides, the reduction in bands is significant when comparing SWN with the number of bands achieved by the heuristics.

Table 3.4: Performance of algorithms for pan-European network.

# of Requests	SWN	NN		RDC		ST	
		Orig. # of bands	Optimized # of bands	Orig. # of bands	Optimized # of bands	Orig. # of bands	Optimized # of bands
50	186	131	130	131	129	123	120
500	2061	1141	1083	1141	940	1060	823
5000	20384	10734	8980	10734	7419	9635	5070

Table 3.5: Performance of algorithms for 5×5 regular mesh network.

# of Requests	SWN	NN		RDC		ST	
		Orig. # of bands	Optimized # of bands	Orig. # of bands	Optimized # of bands	Orig. # of bands	Optimized # of bands
50	211	145	138	145	136	143	142
500	2156	1211	1070	1211	957	1041	737
5000	21706	11367	8328	11367	7028	9498	3943

Table 3.3, 3.4 and 3.5 show the numerical results for the NSF, pan-European and 5×5 regular mesh network, respectively. We can see that the SWN and the number of bands for pan-European and 5×5 regular mesh network are larger than that of the NSF network under the same traffic demand. This is caused by the longer average path length and more traversing switches for the connection requests in larger networks. From all the tables, we can see that heuristic ST works well in the large networks and for a large number of requests. For a better visualization, a comparison among the heuristics for NSF network is shown in Figure 3.8. Sorting the requests in descending order of the path length and then assigning wavelengths first to longer paths gives more opportunity for wavebanding. The heuristic itself achieves a large further reduction in the number of bands. There are also significant reductions in the number of bands when compared with SWN. This demonstrates that the number of switching elements can be reduced by a large amount using waveband switching compared to wavelength switching.

3.5 Application to Dynamic Stochastic Traffic

In this section, we apply the proposed band framework to the case of dynamic stochastic traffic. We first calculate the nonuniform bands for static all-to-all traffic (one lightpath between every pair of nodes) for the given network, and then assume that the wavelengths are banded accordingly. The minimum number of wavelengths required by the all-to-all traffic is denoted by W_m. The total number of available wavelengths in the system, which is denoted by W_s, could be different from W_m. We assume that $W_s > W_m$. In this case, the bands calculated for the deterministic

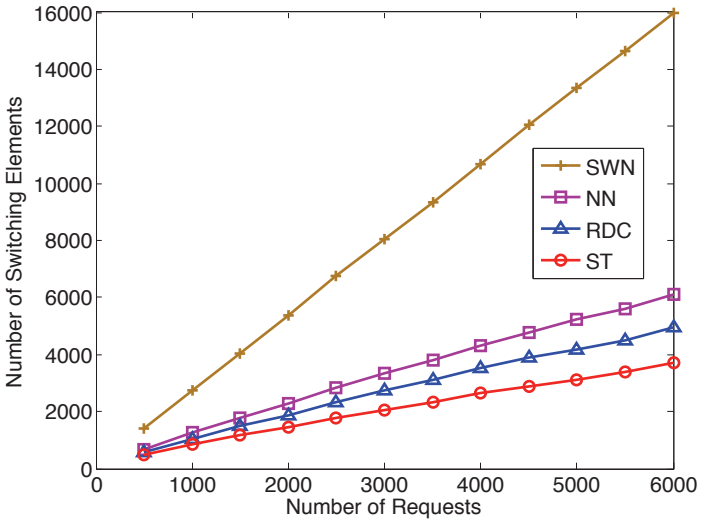

Figure 3.8: A comparison of heuristic results for NSF network.

traffic can be proportionally expanded according to the ratio of W_s to W_m. Assume $r = W_s \text{mod} W_m$ and $n = \lfloor W_s/W_m \rfloor$. Given the reordered RWA matrix Δ' for all-to-all traffic, r rows of Δ' are duplicated n times, while the remaining $W_m - r$ rows need to be duplicated $n - 1$ times.

The dynamic traffic requests arriving to the network are generated according to a Poisson process in our study. Each request has exponential holding time. The source and destination nodes for each request are uniformly randomly chosen from the node set. To accommodate a connection request, a wavelength with available waveband switches on all nodes along the path is needed.

The First-Fit algorithm is adapted to make wavelength assignment decisions for incoming connection requests. For each source-destination node pair, a wavelength set is pre-calculated based on the designated wavelength in Δ' for the all-to-all traffic and the expansion metrics r and n. For each incoming connection request, we first check the designated wavelength set for this node pair starting from the first wavelength in the set. If no wavelength in this set is available for the request, the availabilities of other remaining wavelengths are examined one by one and the first available wavelength is chosen to accommodate this request. If no wavelength is available at the time the request arrives, the connection request is blocked.

We conduct performance evaluation through simulations, and compare the blocking

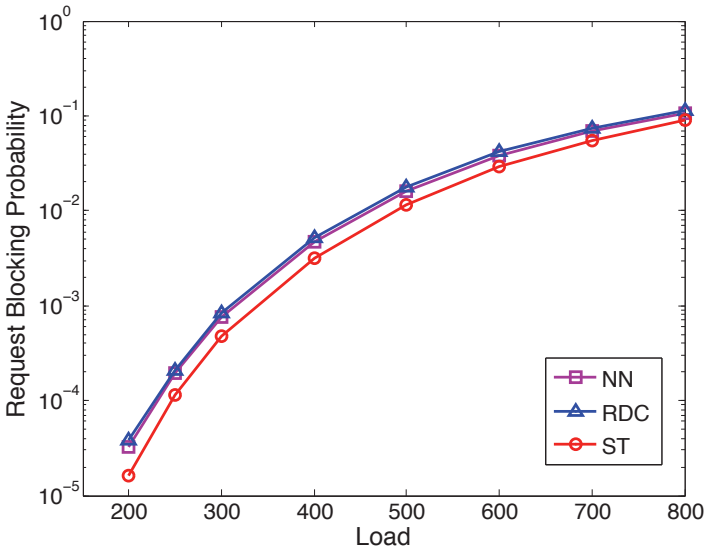

Figure 3.9: Request blocking probability versus load in the NSF network for $W_s = 160$.

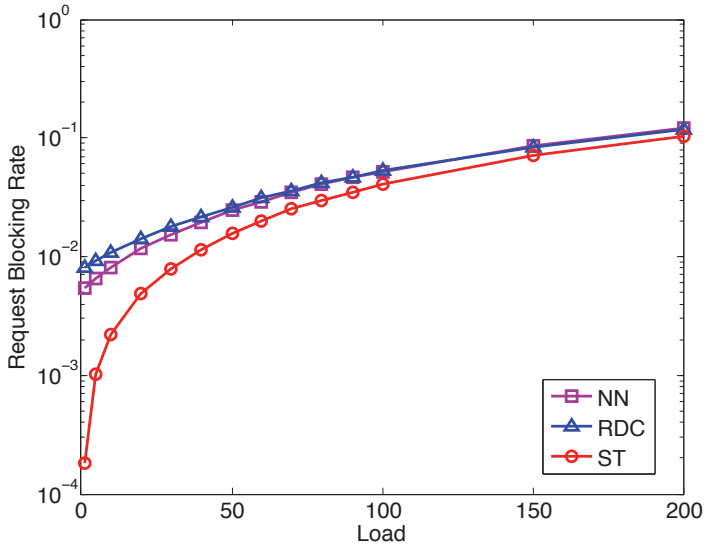

Figure 3.10: Request blocking probability versus load in the pan-European network for $W_s = 160$.

probability of dynamic traffic demands for the proposed heuristics. In our simulation, each request has a mean holding time of 1 (arbitrary time unit), and the arrival rate of traffic requests is varied in order to examine the performance under varying offered loads. Wavelength conversion is not considered here. For the networks, we assume that each link has two fibers, each with 80 wavelengths. Thus $W_s = 160$.

Figures 3.9 and 3.10 show the blocking probability versus load for the NSF network and the pan-European network, respectively. Each point in the figures is obtained

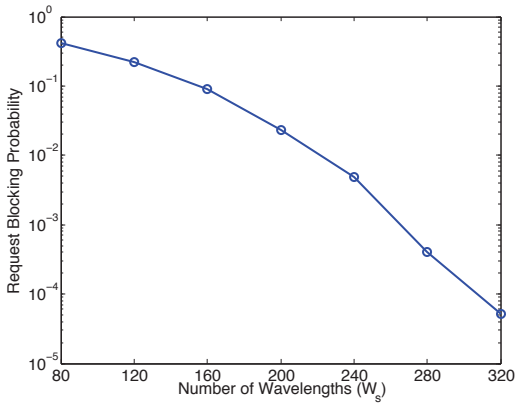

Figure 3.11: Request blocking probability versus number of wavelengths in the NSF network at a load of 800 Erlangs.

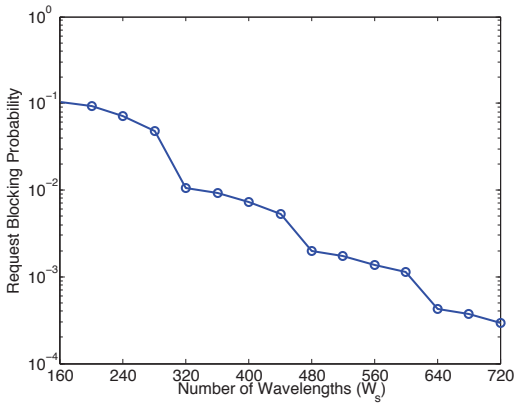

Figure 3.12: Request blocking probability versus number of wavelengths in the pan-European network at a load of 200 Erlangs.

by simulating one million requests after 10000 network warm-up requests in order to obtain steady-state blocking rates. In Figure 3.9, the request arrival rate varies from 200 to 800, while in Figure 3.10, the request arrival rate varies from 5 to 200. These ranges were chosen to obtain blocking rates in the desired range of approximately 0.1 to 0.0001. From the figures, we can see that ST works better than NN and RDC in terms of blocking rate of traffic demands. By using ST in the NSF network when $W_s = 160$, the pre-calculated $W_m = 28$, the $SWN = 568$ and the number of required wavebands is 257. After the expansion described above, the number of required wavebands for this network is still 257, which is much less than the number of required switches if using wavelength switching. For the pan-European network, the pre-calculated $W_m = 160$, the $SWN = 3066$, and the number of required non-uniform wavebands for this network is 1253. We see that the performance for the pan-European network is worse than that of the NSF network when $W_s = 160$. This is because $W_s = W_m$, and the set of available wavelengths for any connection request is limited in the pan-European network. Figure 3.11 and Figure 3.12, respectively, show the request blocking probability versus number of wavelengths in the system (W_s) when using ST for NSF network at a load of 800, and the pan-European network at a load of 200.

3.6 Conclusion

In this chapter, we proposed nonuniform wavebanding strategies to serve a given traffic in mesh networks [90]. We formulated a band minimization problem and transformed it into a modified version of Minimum Weight Hamiltonian Path Problem. We presented an ILP formulation for this problem, and proposed heuristic algorithms for the problem. The numerical simulation results show that heuristic results are very close to ILP results for a small network. Numerical results for the comparison of number of wavebands and number of switching elements for three different larger networks were also presented. Significant reduction in terms of switching elements was achieved through wavebanding. We also applied our nonuniform waveband minimization framework to the dynamic traffic case, and the performance of the network in terms of request blocking probability with dynamic stochastic traffic was evaluated.

Chapter 4 Evaluation and Performance Modeling of OXC Architectures

Despite the static configuration of wavebands in the previous chapter, this chapter introduces Optical Cross-Connect (OXC) node architectures built on top of wavelength-selective switches (WSSs) which facilitate dynamic configuration in WDM networks. An evaluation of Two Optical Cross-Connect (OXC) node architectures with multiple fibers per link – one, a conventional architecture, and the second, a hierarchical architecture that has lower complexity than the first one is presented. Resource (fiber and wavelength) assignment and analytical models for computing the blocking probability of connection requests are studied.

4.1 Related Work

Current OXCs are built using WSSs with a single input and multiple outputs [13, 91]. WSSs have the capability of demultiplexing, multiplexing, and switching. In a $1 \times n$ WSS, each wavelength from the input can be independently switched to any of the n outputs. Large port counts (i.e., large values of n) are required for large-scale OXCs. However, current technology limits the port count to around 20 in commercially available WSSs [92].

A conventional way to build large port-count OXCs is to cascade small port-count WSSs. However, the nature of cascading leads to a square-order increment of the required hardware. To relieve the hardware requirement as well as costs, a novel OXC architecture utilizing hierarchical routing techniques was proposed in [1]. This architecture can accommodate almost as much traffic as the conventional one, while using much fewer WSSs [46]. For WDM networks, fixed-grid WSSs are utilized. Each optical path in the WDM network corresponds to a wavelength. The wavelengths of each fiber are located in a fixed channel frequency spacing, generally 50/100 GHz.

One of the key performance metrics in optical networks is blocking probability, which is the probability that a connection request cannot be accommodated due to a lack of resources or certain constraints. There have been numerous models for studying

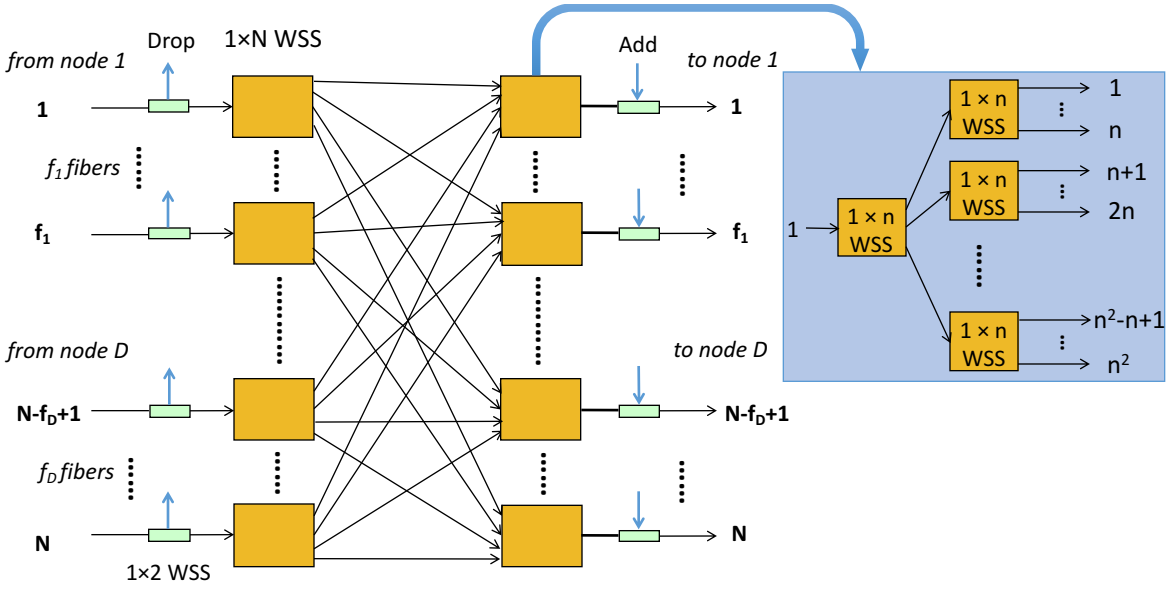

Figure 4.1: An $N \times N$ WSS-based conventional OXC node architecture ("FLEX"). the blocking performance of optical networks both at the network level [93–97] and the node level [98].

4.2 OXC Node Architectures

Suppose the OXC node has a physical degree of D, i.e., D is the number of connected neighbor nodes or the number of input/output links. The number of parallel fibers on link i is denoted as f_i $(i = 1, 2, \ldots, D)$, and $N = \sum_{i=1}^{D} f_i$ represents the total number of input/output fibers. N is usually larger than the limited port count of WSSs. We will analyze an $N \times N$ OXC node with different implementations utilizing port-count-limited WSSs.

The conventional OXC node architecture ("FLEX") implemented by cascading WSSs is shown in Figure 4.1. Using this architecture, $2N$ of $1 \times N$ WSSs are required, and the 1×2 WSSs are used to add or drop local traffic. Suppose 1×4 WSSs are utilized to build "FLEX" nodes. For a 4×4 OXC node with $D = 4$ and a single fiber per link, 8 1×4 WSSs are needed, whereas for a 16×16 OXC node with $D = 4$ and 4 parallel fibers per link, 32 1×16 WSSs are needed. Each 1×16 WSS could be realized by cascading 5 1×4 WSSs. Thus, in total 160 1×4 WSSs are needed for the 16×16 OXC node. We can see that the required number of WSSs to realize

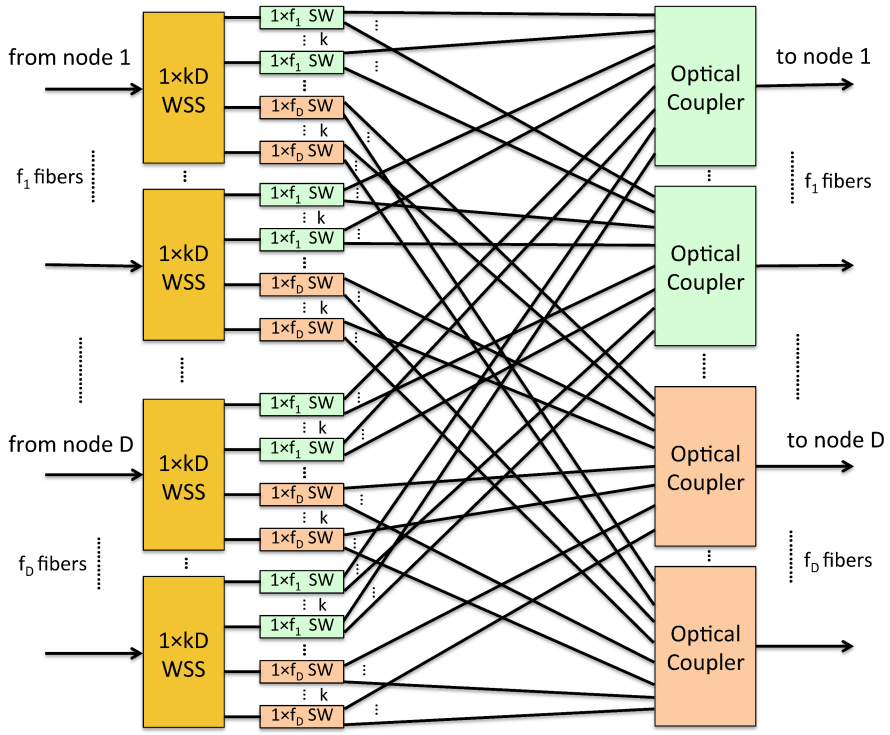

Figure 4.2: Node architecture based on wavelength grouping and fiber selection ("HIER").

an OXC node suffers from a square-order increment. The number of cascaded 1×4 WSSs required by a large $1 \times N$ WSS can be approximately calculated by

$$S(N) \approx \frac{N}{4} + \frac{N}{4^2} + \ldots + 1 = \frac{N}{3}\left(1 - (1/4)^{\log_4 N}\right). \tag{4.1}$$

In this architecture there is no switching constraint, i.e., *any* incoming request can be switched to *any* output fiber at any time, if there is a common available wavelength.

Figure 4.2 shows another implementation of an OXC node via wavelength grouping and fiber selection [1]. For each input fiber, the number of selectable fibers on each output link is k, where k is a parameter of the OXC. The incoming traffic request is switched to one of k parallel selectable fibers on the desired output link through the $1 \times kD$ WSS. The input WSS partitions the incoming traffic requests into kD groups, k for each of the D output links. Each group j $(j = 1, \ldots, kD)$ is then switched to one of the f_i parallel fibers on the output link i via a $1 \times f_i$ switch (SW). Even though the traffic requests can be switched to any of the f_i parallel output fibers, the set of requests to an adjacent node i can only be assigned to up to k different fibers at

the same time. N $1 \times kD$ WSSs, $k{\cdot}D{\cdot}N$ SWs, and N optical couplers are required to implement this architecture, which is referred to as "HIER". For example, $k = 1$ implies that all the optical paths intended for the same adjacent node from an input fiber are routed to a single fiber on the link at a given time (the single fiber may be different at different times).

4.3 Analytical Models

In this section, we propose the resource allocation (wavelength assignment and fiber selection) algorithms and analytical models for evaluating the blocking performance of the OXC node architectures. To simplify the analysis, we make the following assumptions:

(1) There are W wavelengths per fiber. Each input/output link has F fibers; however, this can be easily generalized to varying numbers of fibers per link. There are N fibers in total, i.e., $N = D{\cdot}F$. Let $C = W{\cdot}F$ be the total number of *channels* on each link.

(2) Wavelength conversion is not used.

(3) A static traffic demand model in which the total number of requests at the input fibers is binomially distributed with p being the probability that there is a request for an input channel. Thus, the mean number of requests from all input fibers is NWp. The output link of each request is chosen uniformly randomly from the D output links.

(4) A wavelength is *free* on a link if it is idle on at least one fiber of the link. If there is no *free* wavelength along the route from input to output link, the request is blocked.

4.3.1 Conventional OXC Architecture

We first show that the only reason for blocking in this case is the lack of available resource on the requested output link, i.e., a request can be assigned a fiber and

wavelength (note that the same wavelength must be assigned on the input fiber as well) as long as there is at least one idle channel on that output link.

The basic idea is to show that we can assign at most C requests to any output link, while making assigned requests to any output fiber to be no larger than W.

Theorem 1 *Given a set of requests on input fibers in "FLEX", the only reason for blocking is the lack of resource in the requested output links.*

Proof 1 *Define $Q_{f,d}$ as the number of requests destined to output link d from input fiber f, and suppose that $\sum_{f=1}^{N} Q_{f,d} \leq C$, $\forall d$. We will show that all requests can be assigned a fiber and wavelength through the following algorithm. First, we select a fiber on the desired output link for each traffic request. We consider the request assignments from input fiber 1 to input fiber N in order. For requests from input fiber f, $f \in [1, N]$, we do assignments in the order of destined output links. Let $\widetilde{Q}_{f,d}$ denote the remaining number of requests that need to be assigned from input fiber f to output link d. Let r_{ε_d} denote the number of requests that have been already assigned to output fiber ε_d of link d. ε_d is set to 1 initially. If the remaining capacity of fiber ε_d is not less than $\widetilde{Q}_{f,d}$, we assign $\widetilde{Q}_{f,d}$ requests to ε_d. Otherwise, we partition $\widetilde{Q}_{f,d}$ into two parts to guarantee that the number of requests assigned to any output fiber is no larger than W. Also, since the number of requests from any input fiber is no larger than W, the requests from any input fiber to any output link is assigned to at most two different output fibers. The number of requests of the first part equals $W - r_{\varepsilon_d}$ and we assign them to ε_d. The remaining requests are assigned to fiber $\varepsilon_d + 1$, which does not have any assigned request yet. This guarantees that at most one output fiber of any output link is partially occupied (having larger than 0 and less than W assigned requests). Since there is no routing constraint within the OXC, we can assign C requests to any output link. The pseudocode is shown in Algorithm 6. Based on the algorithm, we can give three conclusions and summarize them in Lemma 1.*

Lemma 1 *The requests assignment algorithm guarantees that*

1. *The number of requests assigned to any output fiber is no larger than W;*

Algorithm 6 Requests assignment algorithm for "FLEX"

1: **for** $f = 1 : N$ **do**
2: **for** $d = 1 : D$ **do**
3: **if** $\widetilde{Q}_{f,d} > 0$ and $W - r_{\varepsilon_d} \geq \widetilde{Q}_{f,d}$ **then**
4: assign $\widetilde{Q}_{f,d}$ requests to output fiber ε_d
5: r_{ε_d} += $\widetilde{Q}_{f,d}, \widetilde{Q}_{f,d} = 0$
6: **else if** $\widetilde{Q}_{f,d} > 0$ and $W - r_{\varepsilon_d} < \widetilde{Q}_{f,d}$ **then**
7: assign $W - r_{\varepsilon_d}$ requests to output fiber ε_d
8: $\widetilde{Q}_{f,d} = \widetilde{Q}_{f,d} - (W - r_{\varepsilon_d})$
9: $r_{\varepsilon_d} = W$, ε_d += 1
10: assign $\widetilde{Q}_{f,d}$ requests to output fiber ε_d
11: r_{ε_d} += $\widetilde{Q}_{f,d}, \widetilde{Q}_{f,d} = 0$
12: **end if**
13: **end for**
14: **end for**
15: use edge coloring algorithm for wavelength assignment

2. *At most one output fiber of any output link is partially occupied; and*

3. *Requests from any input fiber to any output link are assigned to at most two different output fibers.*

Now, wavelength assignment can be done simply as follows. We generate a bipartite multigraph by denoting input fibers as left-hand vertices r_a $(a = 1, \cdots, N)$ *and output fibers as right-hand vertices* c_b $(b = 1, \cdots, N)$*; there are* q_{ab} *parallel edges between* r_a *and* c_b*, where* q_{ab} *is the number of requests from input fiber* a *to output fiber* b *as assigned by the algorithm above. From the given constraint, we know that the maximum degree of this bipartite multigraph is* W*, which is equal to the fiber capacity. The wavelength assignment problem is thus the same as coloring the edges of the bipartite multigraph, which can be done in polynomial time [99]. Thus, blocking occurs only if the number of traffic requests to an output link exceeds the link's capacity.*

We illustrate the above algorithm with a small example. Suppose we have a small

Table 4.1: Traffic demands for the small example node.

Output link \ Input fiber	1	2	3	4	5	6
1	3	2	1	3	0	3
2	1	2	3	1	4	1

node with $D = 2$, $F = 3$, and $W = 4$. Let $r_{i,d}$ denote the set of requests from input fiber i to output link d. Let o_j^d denote the output fiber j in link d. The traffic demands ($|r_{i,d}|$) for this node are listed in Table 4.1. Taking output link 1 as an example, using Algorithm 12, we have $r_{1,1}$ and one of $r_{2,1}$ assigned to o_1^1; the remaining request of $r_{2,1}$, $r_{3,1}$, and two requests of $r_{4,1}$ assigned to o_2^1; the remaining request of $r_{4,1}$, and $r_{6,1}$ assigned to o_3^1. Thus, all the requests destined to output link 1 can be accommadated.

We now present the analytical model for computing the blocking probability.

Let R_t denote the total number of requests from all input fibers. Then

$$P(R_t = g) = \binom{NW}{g} p^g (1-p)^{NW-g}, \quad g = 0, 1, \ldots, NW. \tag{4.2}$$

Let R_j denote the number of requests destined to output link j from all input fibers. Then we have

$$P(R_j = n | R_t = g) = \binom{g}{n} \left(\frac{1}{D}\right)^n \left(1 - \frac{1}{D}\right)^{g-n}, \quad n = 0, 1, \ldots, g. \tag{4.3}$$

Blocking occurs only when the number of requests destined to an output link exceeds the link's capacity. Thus, the blocking probability for output link j is:

$$P(B_j) = \sum_{g=C+1}^{NW} \sum_{n=C+1}^{g} \frac{n-C}{g} \cdot P(R_j = n,\ R_t = g). \tag{4.4}$$

The total blocking probability for the node is

$$P(B) = D \cdot P(B_j). \tag{4.5}$$

4.3.2 Hierarchical OXC Architecture

We next analyze the hierarchical OXC architecture ("HIER") separately for the following two cases: $k = 1$, and $k \geq 2$.

Case 1: $k = 1$. In order to minimize blocking, the total number of requests routed to any fiber exceeding its capacity must be minimized. In other words, we have to minimize

$$\sum_{f'=1}^{N} \max\{0, \sum_{f=1}^{N} q_{f,f'} - W\}, \tag{4.6}$$

where $q_{f,f'}$ is the number of requests assigned from input fiber f to output fiber f'. We will present a heuristic later to minimize the above expression.

In order to facilitate the analysis, we assume that the output fiber for all requests from an input fiber to a particular output link is chosen uniformly randomly from the F possible fibers. (Note that since $k = 1$, all requests from an input fiber to a given output link must be switched to a single fiber on that output link.) We call this assignment method as HIER Random Fiber Selection Algorithm (*HRFS*).

Let X_f^j denote the number of input fibers which choose the output fiber f for output link j. We have

$$P(X_f^j = m) = \binom{N}{m} \left(\frac{1}{F}\right)^m \left(1 - \frac{1}{F}\right)^{N-m}, \quad m = 0, 1, \ldots N. \tag{4.7}$$

Let R_x denote the number of requests from the X_f^j input fibers. Then we have

$$P(R_x = n | X_f^j = m) = \binom{Wm}{n} p^n (1-p)^{Wm-n}, \quad n = 0, 1, \ldots Wm. \tag{4.8}$$

Let R_f^j denote the number of requests assigned to fiber f of output link j.

$$\begin{aligned} P(R_f^j = e) &= \sum_{m=0}^{N} \sum_{n=e}^{Wm} P(R_f^j = e | R_x = n, X_f^j = m) \cdot P(R_x = n | X_f^j = m) \cdot P(X_f^j = m) \\ &= \sum_{m=0}^{N} \sum_{n=e}^{Wm} \binom{n}{e} \left(\frac{1}{D}\right)^e \left(1 - \frac{1}{D}\right)^{n-e} \cdot P(R_x = n | X_f^j = m) \cdot P(X_f^j = m). \end{aligned} \tag{4.9}$$

The blocking probability for one output fiber is:

$$P(B_f) = \sum_{g=W+1}^{NW} \sum_{e=W+1}^{g} P(B_f | R_f^j = e, R_t = g) \cdot P(R_f^j = e) P(R_t = g), \tag{4.10}$$

where

$$P(B_f | R_f^j = e, R_t = g) = \frac{e - W}{g}. \tag{4.11}$$

Algorithm 7 *HSA* algorithm for "HIER" with $k = 1$

for $d = 1 : D$ **do**

group all requests to link d based on the input fiber indexes

sort all groups in non-increasing order of the number requests and the order is indexed by s

while not all groups have been assigned **do**

if output fiber ε_d has the largest remaining capacity **then**

if $W - r_{\varepsilon_d} \geq |g_s|$ **then**

assign g_s to output fiber ε_d

r_{ε_d} += $|g_s|$

else

assign $W - r_{\varepsilon_d}$ requests chosen from g_s to output fiber ε_d

the remaining $|g_s| - (W - r_{\varepsilon_d})$ requests are blocked

$r_{\varepsilon_d} = W$

end if

end if

$|g_s| = 0$, $s = s + 1$

end while

end for

use edge coloring algorithm for wavelength assignment

Recall that R_t was defined before 4.2. Then the total blocking probability is

$$B = N \cdot P(B_f). \tag{4.12}$$

The *HRFS* assignment algorithm lends itself to analysis, but performs poorly (as expected and as shown later) because it assigns fibers randomly and independently of other requests. We therefore propose the HIER Sort Assignment Algorithm (*HSA*) to solve the problem. We consider output links one by one. For each output link, we first group all requests based on the input fiber index. Then, we sort the groups by the number of requests in the group in non-increasing order. Based on the sorted order, we assign each group to the output fiber with the largest remaining capacity, while making the number of requests assigned to any output fiber to be no larger than W.

Finally, we use edge coloring algorithm for wavelength assignment. The pseudocode is shown in Algorithm 7. Here, g_s represents the sth largest group of requests.

As an example, if *HRFS* is used to assign requests destined to output link 1 in Table 4.1, one possible assignment can be as follows. Input fiber 1 and 6 select o_1^1, input fiber 2 and 4 select o_2^1, and input fiber 3 selects o_3^1. In this case, 3 requests are blocked. If *HSA* is used instead, input fiber 1 and 2 select o_1^1, input fiber 3 and 4 select o_2^1, and input fiber 6 selects o_3^1. Thus, the number of blocked requests can be reduced to just one in this example.

Case 2: $k \geq 2$. We show that in "HIER" with $k = 2$, we can achieve the same blocking performance as that of "FLEX" by using Algorithm 6.

Theorem 2 *Given the requests on input fibers in "HIER" with $k = 2$, the only reason for blocking is the lack of resource in the desired output links.*

Proof 2 *The only difference between "FLEX" and "HIER" with $k = 2$ is that in "FLEX", we can assign requests from a fiber to any of the F fibers on a link, while in "HIER" with $k = 2$, we can assign requests from a fiber to at most 2 fibers on a link. However, from Lemma 1, we can see that Algorithm 6 assigns requests from any input fiber to a given output link to at most two different output fibers. Thus, there is no difference in blocking between "FLEX" and "HIER" with $k = 2$.*

4.4 Simulation Results

We first validate our analytical models by comparing analytical results with those from simulations. Connection requests for each input fiber are generated according to a binomial distribution with parameters W and p, and the requested output link is chosen randomly from 1 to D. For each data point of simulation results in the graphs, we simulated 1000 instances and obtained the average results.

We present results for the OXC nodes with node degree $D = 4$ and the number of wavelengths per fiber $W = 32$. Note that our analytical model for the two architectures can be easily extended to the case of different numbers of fibers on different links. For

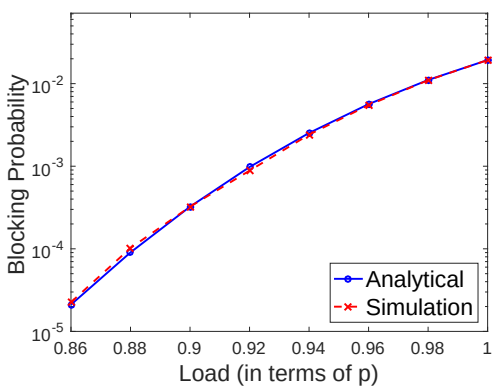

Figure 4.3: The validation results for "FLEX".

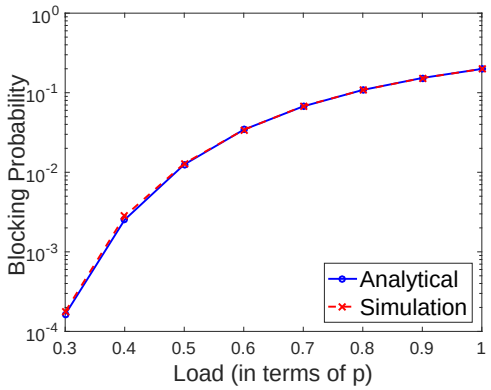

Figure 4.4: The validation results for "HIER" ($k = 1$).

simplicity, in the evaluation we choose the number of fibers to be 10 for all links (i.e., a 40×40 OXC node).

Figures 4.3 and 4.4 show the comparison of analytical and simulation results for "FLEX" and "HIER" ($k = 1$) using *HRFS*, respectively. Analytical estimations match simulation results very well for both architectures. As shown in Figure 4.5, the blocking probability of "HIER" ($k = 1$) using *HRFS* is much higher than that of "FLEX". This is due to the routing constraint caused by $k = 1$ and the random fiber selection. Figure 4.5 also shows the simulation results when we use *HSA* for request assignments. Obviously, we can see that this algorithm gives much better blocking performance. Also, we can see that the blocking probabilities of "HIER" ($k = 1$) are only a little bit higher than those of "FLEX" when using *HSA*.

We next examine the cost and power consumption of the two OXC architectures. The complexity of each architecture is as follows. Since WSS is the most costly element in the architecture, we count the number of WSSs to denote the complexity. For "FLEX", if 1×4 WSS is utilized, the total number of WSSs required for a 40×40 node is 1120, whereas for "HIER" ($k = 2$), N $1 \times kD$ WSSs are required at the node, which corresponds to 120 1×4 WSSs. Thus, "HIER" requires much less hardware than "FLEX" without hurting the blocking performance.

We compare "FLEX" and "HIER" in terms of power consumption and capital expenditure (CapEx). "FLEX" consists of $2N$ $1 \times N$ WSSs. "HIER" consists of N $1 \times kD$ WSSs, $k{\cdot}D{\cdot}N$ $1 \times F$ optical switches based on MEMS, and N $N \times 1$

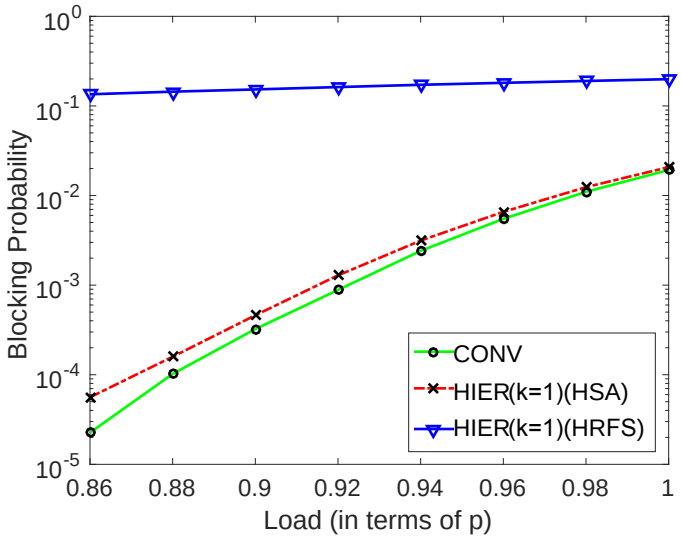

Figure 4.5: Comparison between "FLEX" and "HIER" ($k = 1$).

Table 4.2: Power consumption and cost of the optical components in "FLEX" and "HIER" architectures.

Component	WSS (port)	MEMS switch (port)	Coupler
Power(Watts)	1	0.25	0
Cost(Dollars)	1000	255	195
CONV	$2N{\cdot}Z(N)$	0	0
HIER	$N{\cdot}Z(kD)$	$k{\cdot}N{\cdot}D{\cdot}F$	N

optical couplers. The total number of WSS ports in a $1 \times N$ WSS can be denoted as $Z(N) = 4{\cdot}S(N)$ (see equation 4.1). The power consumption and CapEx are calculated by summing up the consumed power and dollar cost of each component. A summary is shown in Table 4.2 [100][101].

The difference in power consumption between "FLEX" and "HIER" is

$$2N{\cdot}Z(N) - N{\cdot}Z(kD) - 0.25{\cdot}kN^2. \tag{4.13}$$

The CapEx difference between "FLEX" and "HIER" is

$$2000N{\cdot}Z(N) - 1000N{\cdot}Z(kD) - 255kN^2 - 195N. \tag{4.14}$$

The power consumption and CapEx difference between "FLEX" and "HIER" ($k = 2$)

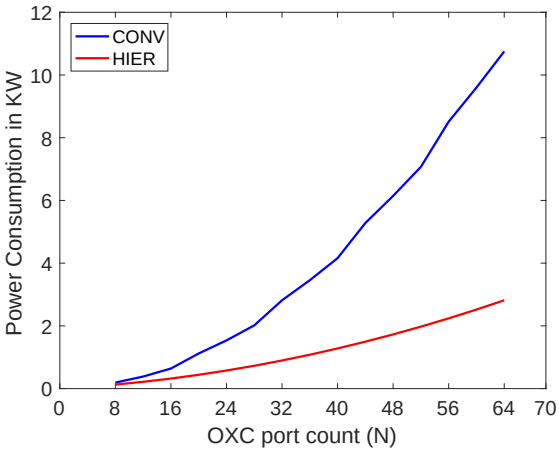

Figure 4.6: Power consumption comparison.

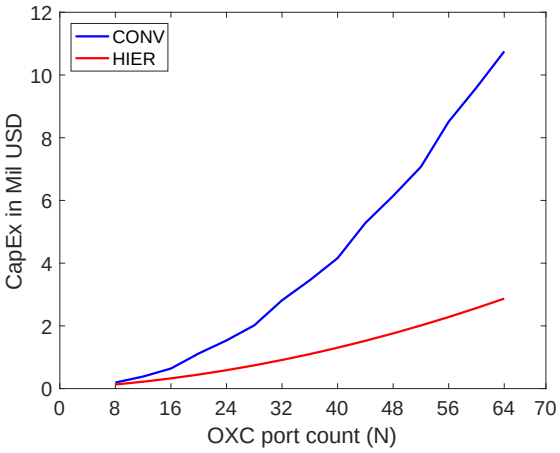

Figure 4.7: CapEx comparison.

for an OXC node with $D = 4$ and varying F are shown in Figures 4.6 and 4.7.

4.5 Conclusion

In this chapter, we investigate the resource assignment problem and analytical models for predicting the blocking performance of multi-fiber OXC nodes using the conventional architecture and a cost-efficient hierarchical architecture [102]. The numerical results validate the accuracy of our proposed analytical models and show that the hierarchical architecture can achieve similar blocking performance as the conventional architecture while using much less hardware. A comparison of the power consumption and cost of the two architectures also reveals the advantages of hierarchical architecture. Our results show that the hierarchical architecture which utilizes wavebanding features exhibits a good balance between performance, cost, and power consumption.

Chapter 5 Routing, Fiber, Band, and Spectrum Assignment (RFBSA) for Multi-granular Elastic Optical Networks

In this chapter, we consider the wavebanding feature in EONs. A flexible waveband ("FLEX") multi-granular architecture to increase waveband path utilization was presented in [2]. This architecture enables non-uniform and non-contiguous flex-grid wavebands [47]. We solve the routing, fiber, waveband, and spectrum assignment (RFBSA) problem introduced by elastic optical networking and flexible wavebanding.

5.1 Background and Problem Statement

We first present a comparison of the conventional optical cross-connect (OXC) architecture and the flexible waveband cross-connect architecture for EONs.

Consider an OXC node with a physical node degree of D, which is the number of physical nodes connected to this node. Each connectivity is represented by an input or output link. Each physical link contains several fibers. $N = \sum_{i=1}^{D} x_i$ denotes the total number of input/output fibers to/from the node, where x_i denotes the number of parallel fibers on link i.

5.1.1 "FLEX" Architecture

The details of this architecture has been described in Section 4.2. In EONs, the WSSs are flex-grid in order to switch frequency slots. Current commercially available flex-grid WSSs typically have a port count limit of 4, 9, or 20, which are not easily scalable.

5.1.2 "FLEX" Waveband Architecture

Figure 5.1 shows the flexible waveband OXC architecture ("FLEX") proposed in [2]. "FLEX" is composed of small-port-count $1\times B$ flex-grid WSSs and B cost-effective matrix switches. Requests from an incoming fiber can be partitioned into B groups and each group is switched as a whole to one of the output fibers of the OXC node.

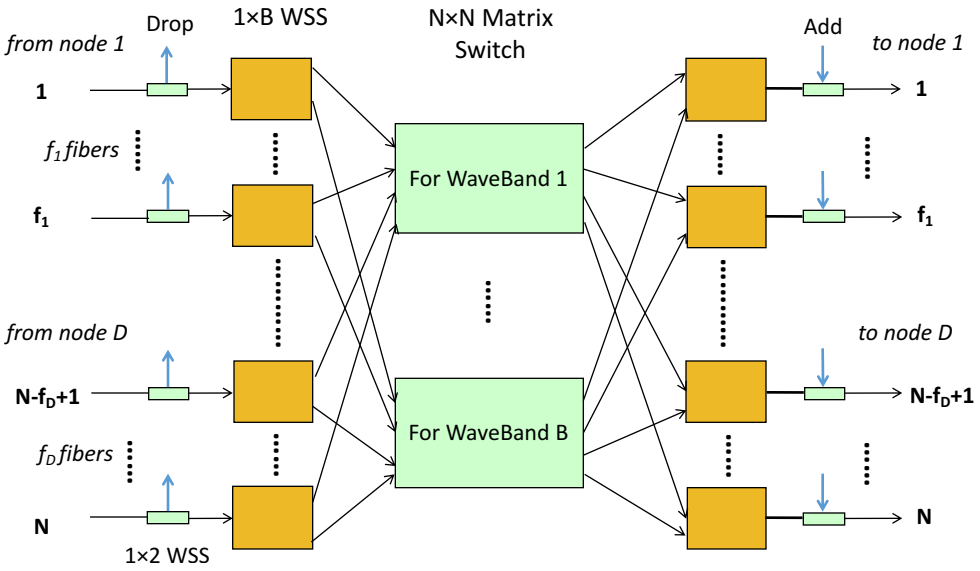

Figure 5.1: Flexible waveband node architecture.

Note that the flex-grid WSSs provide the capability of *independently* switching a variable-sized set of contiguous FSs (corresponding to one lightpath) to *any* of its output fibers. Therefore, even though the FSs that are allocated to each lightpath must be contiguous to satisfy the spectrum contiguity constraint, the sets of lightpaths (different sets of contiguous FSs) that are switched as a band need not occupy a contiguous spectral range in "FLEX". This is in contrast to [14] which requires the wavebands to be of uniform size and each waveband occupies a contiguous spectral range.

Since B is quite small compared to N, the hardware cost can be saved in terms of fewer number of costly WSSs in this architecture. However, the switching capability is reduced, since an entire group of lightpaths needs to be switched as a single entity, and the number of groups that can be switched simultaneously cannot exceed B.

5.1.3 Comparison Between Architecture

We compare the two architectures in terms of power consumption and hardware cost. Suppose the port count of the node is N. A "FLEX" node requires $2N$ $1\times N$ WSSs, each of which is constructed with $S(N)$ 1×4 WSSs. $S(N)$ is presented in Equation 4.1. A "FLEX" node consists of $2N$ $1\times B$ WSSs, as well as B $N\times N$ cost-effective matrix switches. Each $N\times N$ matrix switch can be constructed with N

Table 5.1: Power consumption and cost of the optical components in "FLEX" and "FLEX" architectures.

Component	WSS (port)	MEMS switch (port)	Coupler
Power(Watts) [103]	1	0.25	0
Cost(Dollars) [103]	1000	255	195
CONV	$2N{\cdot}S(N){\cdot}4$	0	0
FLEX	$2N{\cdot}B$	$B{\cdot}N{\cdot}N$	$B{\cdot}N{\cdot}N$

$1{\times}N$ MEMS optical switches and N $N{\times}1$ optical couplers. We calculate the power consumption and hardware cost of an OXC node by summing up the consumed power and dollar cost of each component. Table 5.1 gives a summary of the comparison.

Typically, B equals 4. The power consumption of "FLEX" is higher than that of "FLEX" with a difference of

$$8{\cdot}N{\cdot}S(N) - 8{\cdot}N - N^2 = O(N^2). \tag{5.1}$$

The difference in hardware costs between the two architectures is

$$8000{\cdot}N{\cdot}S(N) - 8000{\cdot}N - 1800{\cdot}N^2 = O(N^2). \tag{5.2}$$

Thus, the power consumption and cost of a "FLEX" node is $O(N^2)$ better than that of "FLEX". We can see that significant savings can be realized when utilizing the "FLEX" node architecture.

5.1.4 Motivation and Problem Definition

Routing and spectrum assignment is known to be an NP-complete problem in EONs [104]. Adding the fiber and flexible waveband selection further increases the problem complexity in another two dimensions. Before we proceed to formally state the problem, we present an example in Figure 5.2 to illustrate the challenge in solving the problem. In this example, there are 4 "FLEX" nodes, 4 parallel fibers per link, and an input fiber can be switched to at most 2 output fibers, i.e., $B = 2$. The green

dashed lines represent already established connections, i.e., existing lightpaths are switched from the indicated input fibers to the corresponding output fibers. The matrix shows the already occupied FSs on the various fibers. We denote by $f_{v_s,v_d,i}$ the i^{th} fiber in the direction from v_s to v_d. Our objective is to find a lightpath from node 1 to node 3 for a request whose bandwidth requirement is 2 FSs, and minimize the maximum spectral usage. Due to spectrum contiguity, spectrum continuity, and spectrum non-overlapping constraints, we cannot establish a lightpath by using any of (a) $f_{1,2,1}$ and $f_{2,3,1}$, (b) $f_{1,2,1}$ and $f_{2,4,1}$, or (c) $f_{1,2,2}$ and $f_{2,3,2}$ without increasing the maximum index of used FSs. Also, we cannot establish a new waveband connection from $f_{1,2,1}$ without exceeding the waveband constraint (since $B = 2$). One possible solution is to establish a new waveband connection from $f_{1,2,2}$ to $f_{2,4,2}$ in node 2 and the lightpath passes through $f_{4,3,1}$. The lightpath uses FS 1 and 2, and this solution does not increase the spectral usage and keeps the maximum number of used FSs to be 4. The example illustrates the point that a judicious choice of fibers, bands, and FSs must be jointly made in order to optimize spectrum usage.

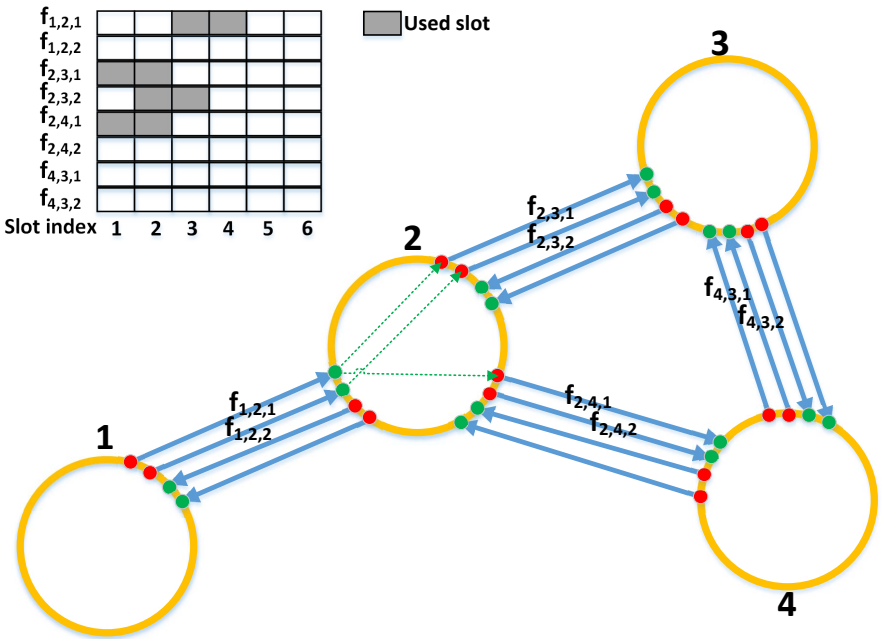

Figure 5.2: Illustrative example.

We formally define the problem as follows. $G = (V, E)$ represents the physical

topology, wherein V denotes the set of OXC nodes and E denotes the set of unidirectional links. For each pair of adjacent nodes, there are two unidirectional links and f_e unidirectional fibers on the unidirectional link e. We are given a set of requests $\mathcal{R}$. A request $r \in \mathcal{R}$ has source node s_r, destination node d_r, and a bandwidth requirement w_r FSs. The resource allocation algorithm assigns a path, which consists of links and fibers, and FSs to the request r. The number of available slots on each fiber is assumed to be unlimited. Our objective is to minimize the maximum spectrum usage (MSU), i.e., the highest FS index used over all fibers to establish all traffic demands.

We propose a new auxiliary layered-graph framework for solving the RFBSA problem. This framework generates an auxiliary graph for each outgoing fiber of the source node of a request, and generates one or more layered graphs for each auxiliary graph. The framework uses pluggable cost functions, and is therefore suitable for handling different network design objectives.

5.2 The Proposed RFBSA Algorithm

We present the details of our proposed RFBSA framework in this section.

5.2.1 Auxiliary Layered-graph Framework

We start by describing the auxiliary layered-graph framework for solving the RFBSA problem. In the following, we denote $|f_{v_s,v_d}|$ as the number of fibers from v_s to v_d.

For each request r, we generate $\sum_{v_d}|f_{s_r,v_d}|$ auxiliary graphs. Each auxiliary graph involves one outgoing fiber from s_r and includes all remaining nodes and fibers in the network. By fixing one outgoing fiber from the source node, we only need to consider the available FSs on that outgoing fiber, thereby reducing the complexity of searching for a lightpath. For a specific FS starting index SI in an auxiliary graph $G_a = (V_a, E_a)$, we define a layered graph $G_{a,SI} = (V_{a,SI}, E_{a,SI})$, which consists of all fibers of G_a and FSs ranging from SI to $SI + w_r - 1$. The range of SI is from 1 to the current MSU.

Since layered graph is at the FS level, it helps us to include spectrum contiguity, spectrum continuity, and spectrum non-overlapping constraints in each auxiliary graph. In the illustrative example, we can build two auxiliary graphs, each including one of the two outgoing fibers from node 1. In the first auxiliary graph, the only edge between node 1 and node 2 is $f_{1,2,1}$, while the only edge between node 1 and node 2 is $f_{1,2,2}$ in the second auxiliary graph. Each auxiliary graph has all edges among nodes 2, 3, and 4. Also, each auxiliary graph has layered graphs with contiguous FSs from SI to $SI + w_r - 1$, where w_r equals 2 in the example.

5.2.2 Resource Assignment Algorithm

In this subsection, we describe the algorithm for solving the RFBSA problem based on the above auxiliary layered-graph framework. The pseudocode is shown in Algorithm 8.

We first sort all requests based on their bandwidth requirements in non-increasing order and store them in an ordered list $\mathcal{R}^*$. We consider lightpath assignments from the first request in the list, i.e., the one with the highest bandwidth demand. For each request, we first use the algorithm Switching Cost Update for updating costs of each incoming fiber in each "FLEX" node.

The basic idea of Switching Cost Update is that we prefer to use already established wavebands (if any) and leave more waveband choices for subsequent requests. For the input fiber f_{in} in "FLEX" node v, we compute the switching cost for the outgoing fiber f_{out} of node v. If the waveband from f_{in} to f_{out} is already established, the switching cost from f_{in} to f_{out} is 0. If the waveband from f_{in} to f_{out} has not been established and the number of established wavebands from f_{in} to all outgoing fibers of node v is $b < B$, the switching cost between the two fibers is $\alpha \cdot b$. Here, α is a tradeoff parameter. If the number of established wavebands from f_i to all outgoing fibers of node v already equals B, the switching cost from f_{in} to f_{out} is set to ∞ because we cannot establish any more wavebands in "FLEX". The pseudocode is shown in Algorithm 9, and the cost function is shown in Equation 6.1. Denote $C_{v,f_{in},f_{out}}$ as the switching cost from f_{in} to f_{out} in "FLEX" node v.

$$C_{v,f_{in},f_{out}} = \begin{cases} 0, & \text{if the waveband is established} \\ \alpha \cdot b, & \text{if } b < B \\ \infty, & \text{otherwise.} \end{cases} \tag{5.3}$$

After updating switching costs, we start to generate auxiliary graphs for each outgoing fiber from s_r for request r. For auxiliary graph AG_κ of outgoing fiber κ, we need to update spectrum costs of all connected fibers. Define $C_{r,\kappa}$ as the spectrum cost of output fiber κ for request r. We consider contiguous FS sets on κ one by one. If the number of FSs in FS set $\widetilde{s}$ is less than w_r, we set $C_{r,\kappa}$ to infinity. We only consider the case that the number of FSs in $\widetilde{s}$ is no less than w_r in the following. Denote $SI_{\widetilde{s}}^{start}$ and $SI_{\widetilde{s}}^{end}$ as the first and the last FS index in $\widetilde{s}$, respectively. Further, denote SI as the FS index and denote $\widetilde{ss}$ as the FS set ranging from SI to $SI + w_r - 1$. We increase SI starting from $SI_{\widetilde{s}}^{start}$ and stopping at $(SI_{\widetilde{s}}^{end} - w_r + 2)$. For each SI, Spectrum Cost Update is used to compute the spectrum cost of $\widetilde{ss}$ of all fibers in AG_κ.

The basic idea of Spectrum Cost Update is to minimize the maximum FS index on each fiber after the current request is set up. We denote f_{sl} as the largest FS index of fiber f and denote $C_{r,\kappa,\widetilde{ss},f}$ as the spectrum cost on fiber f of $\widetilde{ss}$ in AG_κ for request r. If $\widetilde{ss}$ is not available on fiber f, $C_{r,\kappa,\widetilde{ss},f}$ is set to ∞. This means that we cannot establish a lightpath by using fiber f. If $\widetilde{ss}$ is available and $SI + w_r - 1$ is no more than f_{sl}, $C_{r,\kappa,\widetilde{ss},f}$ is set to 1. In this case, the maximum FS index on fiber f would not increase, and the spectrum cost works as a counter for the number of hops on the path. Otherwise, $C_{r,\kappa,\widetilde{ss},f}$ is $SI + w_r - 1$. In this case, the spectrum cost is proportional to the maximum FS index after the request is set up. The pseudocode is shown in Algorithm 10 and $C_{r,\kappa,\widetilde{ss},f}$ is shown in Equation 5.4.

$$C_{r,\kappa,\widetilde{ss},f} = \begin{cases} 1, & \text{if } SI + w_r - 1 \leq f_{sl} \\ \infty, & \text{if } \widetilde{ss} \text{ is not available on } f \\ SI + w_r - 1, & \text{otherwise.} \end{cases} \tag{5.4}$$

For a lightpath l, we denote $C_{total,r,\kappa,\widetilde{ss}}$ as the total cost combining Switching Cost $C_{v,f_in,f_{out}}$ of the "FLEX" node set V_l and Spectrum Cost $C_{r,\kappa,\widetilde{ss},f}$ of the fiber set F_l.

Algorithm 8 RSBFA Algorithm

Input: $\mathcal{R}$, $G = (V, E)$

Output: $Path$

Sort all requests based on bandwidth requirements in non-increasing order and store in $\mathcal{R}^*$

for $r : \mathcal{R}^*$ **do**

  for κ : outgoing fibers connecting to s_r **do**

    Create auxiliary graph AG_κ

    for $\widetilde{s}$: available FS sets on κ **do**

      if $|\widetilde{s}| < w_r$ **then**

        $C_{r,\kappa} = \infty$

        CONTINUE

      end if

      $SI_{\widetilde{s}}^{start}$ = the start FS index of $\widetilde{s}$

      $SI_{\widetilde{s}}^{end}$ = the last FS index of $\widetilde{s}$

      $SI = SI_{\widetilde{s}}^{start}$

      while $SI \leq SI_{\widetilde{s}}^{end} - w_r + 1$ **do**

        $\widetilde{ss}$ = FS set from SI to $SI + w_r - 1$

        Spectrum Cost Update

        Use shortest path algorithm to find a lightpath with the smallest $C_{total,r,\kappa,\widetilde{ss}}$

        if $C_{total,r,\kappa,\widetilde{ss}} < \infty$ **then**

          $C_{total,r,\kappa} = C_{total,r,\kappa,\widetilde{ss}}$

          BREAK

        end if

        $SI = SI + 1$

      end while

      if $C_{total,r,\kappa} < \infty$ **then**

        BREAK

      end if

    end for

  end for

  $path_r$ = a lightpath with the smallest $C_{total,r,\kappa}, \forall \kappa$

  $Path.add(path_r)$

  Switching Cost Update along the $path_r$

end for

Algorithm 9 SWITCHING COST UPDATE

Input:v, f_{in}, f_{out}

Output:$C_{v,f_{in},f_{out}}$

for f_{in} : all incoming fibers in node v **do**

  for f_{out} : all outgoing fibers in node v **do**

    b = the number of established wavebands from f_{in}

    if the waveband is established **then**

      $C_{v,f_{in},f_{out}} = 0$

    else if $b < B$ **then**

      $C_{v,f_{in},f_{out}} = \alpha \cdot b$

    else

      $C_{v,f_{in},f_{out}} = \infty$

    end if

  end for

end for

Algorithm 10 SPECTRUM COST UPDATE

Input:$\widetilde{ss}$, f, SI, w_r

Output:$C_{r,\kappa,\widetilde{ss}}$

for f : all fibers in the network **do**

  f_{sl} = the largest FS index on f

  if $\widetilde{ss}$ is not available on f **then**

    $C_{r,\kappa,\widetilde{ss},f} = \infty$

  else if $SI + w_r - 1 \leq f_{sl}$ **then**

    $C_{r,\kappa,\widetilde{ss},f} = 1$

  else

    $C_{r,\kappa,\widetilde{ss},f} = SI + w_r - 1$

  end if

end for

This is defined in Equation 5.5.

$$C_{total,r,\kappa,\widetilde{ss}} = \sum_{f_{in},f_{out} \in V_l} C_{v,f_{in},f_{out}} + \beta \cdot \sum_{f \in F_l} C_{r,\kappa,\widetilde{ss},f}, \tag{5.5}$$

where β is a tradeoff parameter between Switching Cost and Spectrum Cost.

Once the costs for the edges in the layered graph are assigned, we use Dijkstra's algorithm to find the shortest paths from the source node output fiber to different input fibers of destination node, and choose the path p with smallest total cost. If the cost of path p is less than infinity, in order to keep the algorithm's complexity low, we stop generating further layered graphs for this auxiliary graph, and consider p as a *candidate path*. We then proceed to find a candidate path on the next auxiliary graph, and so on, until one (or zero, if no paths have finite cost) candidate path is found for each auxiliary graph. Finally, we compare the total costs of all the candidate paths and choose a path with the smallest total cost. Assume N is total number of nodes in the network with max node degree D, L is total number of links with max number of fibers F per link, and η is the estimated upper bound of the maximum spectral usage. For each request, for each output fiber of the source node, $O(\eta)$ auxiliary graphs are created. Each graph consists of $2LF$ vertices and $LF + N(DF)^2$ edges. The dominant part is to find the shortest path in the auxiliary graphs, thus the time complexity of this algorithm is $O(|\mathcal{R}|DF\eta N(DF)^2 \log(LF))$.

In the illustrative example, we first consider auxiliary graph with $f_{1,2,1}$ and start to generate layered graphs from the first available contiguous FSs. In this auxiliary graph, we cannot build new waveband connections without making the switching cost to be infinity. In the first layered graph, which consists of FSs 1 and 2, the spectrum costs of $f_{2,3,1}$ and $f_{2,4,1}$ are both infinity. Then, we consider the second layered graph, which consists of FSs 5 and 6, and SI equals 5. In this layered graph, the spectrum cost on each fiber is $SI + w_r - 1 = 6$ and the spectrum cost of a lightpath is the number of fibers of the lightpath multiplied by the spectrum cost on each fiber. Thus, the spectrum cost by using $f_{1,2,1}$ and $f_{2,3,1}$ is $2(SI + w_r - 1) = 12$, and the spectrum cost by using $f_{1,2,1}$, $f_{2,4,1}$, and $f_{4,3,1}$ (or $f_{4,3,2}$) is $3(SI + w_r - 1) = 18$. Thus, the best path of the first auxiliary graph is the path consisting of $f_{1,2,1}$ and $f_{2,3,1}$ and the total cost

is $\beta \cdot 12$. Here, the switching cost is 0. The second auxiliary graph starts from $f_{1,2,2}$. The spectrum cost of the path consisting of $f_{1,2,2}$ and $f_{2,3,2}$ is $2(SI + w_r - 1) = 10$, where $SI = 4$, and switching cost is 0. The spectrum cost of the path consisting of $f_{1,2,2}$, $f_{2,4,2}$, and $f_{4,3,1}$ is $3(SI + w_r - 1) = 6$, where $SI = 1$ and switching cost is α. Suppose we set $\alpha = 0.1$ and $\beta = 1$, the second path is the best path in the second auxiliary graph. Finally, we compare total costs of the best path from each auxiliary graph and the path consisting of $f_{1,2,2}$, $f_{2,4,2}$, and $f_{4,3,1}$ is chosen as the best path. The lightpath for the request is thus established on the fibers and FSs that correspond to this best path.

5.3 Simulation Results

In this section, we present simulation results to demonstrate the effectiveness of our proposed RFBSA algorithm and to compare the performances of the "FLEX" and "FLEX" architectures. The network topology used for simulations is the NSFNET network topology as shown in Figure 2.3, which has 14 physical nodes and 22 bidirectional links [46]. Each link has multiple parallel fibers. We apply two different settings: 1) all links have the same number of parallel fibers ($F = 2$ per link); 2) the number of fibers on each link is randomly selected according to a uniform distribution between 5 and 10 ($F = [5, 10]$ per link).

We use a static traffic demand model – a given set of traffic requests is to be allocated resources in the network. Each request denotes a connection between a pair of nodes in the network. The source and destination nodes for each connection request are uniformly randomly selected from the physical nodes of the network. We assume three different types of demands with different number of desired frequency slots [2]. The number of frequency slots required by each request is chosen from the following distribution: 3 slots with probability 0.2, 4 slots with probability 0.5, and 7 slots with probability 0.3. For the "FLEX" node architecture, the limited number of wavebands B is assumed to be 4, which is equal to the port count of commercially available 1×4 WSS.

For each given set of static traffic demands, we record the MSU, which is the

largest used slot index over all fibers of the network. The results from our proposed algorithm are compared with those from a commonly-used baseline algorithm for both "FLEX" and "FLEX" architectures. The performances are also compared between the two architectures.

The baseline algorithm we adapt to our multi-fiber network is the Shortest Path First Fit (SPFF) algorithm. In this algorithm, a single shortest path (i.e., smallest number of hops) is pre-defined for each source-destination pair. For each request, we try to find the first fit (i.e., a set of FSs with the lowest starting index) slot set (number of contiguous slots equal to the request demand) on the first fit fibers (considering any switching constraints) along the pre-calculated route. If multiple shortest paths are pre-defined, the baseline algorithm could be adapted to a K-Shortest Path First Fit (KSPFF) algorithm.

Since the conventional node architecture has no switching constraints, any input fiber can be switched to any output fiber on each node. The selection of fibers on previous links does not affect the fiber selection on other links. For all links along the route, if a block of slots is available on at least one fiber on each link, this block of slots is allocated to the request. The first fiber containing available resources on each link is selected to accommodate the request. We use $CONV_{SPFF}$ to denote this baseline algorithm for the conventional architecture.

For the "FLEX" architecture, there is a switching constraint defined by B – each input fiber can be switched to up to B output fibers for each node. The fiber selection on the previous link of the route will affect the decision on the current link, and then all the following links on the route. If an input fiber to a node is already switched to B different output fibers, with a subset belonging to the desired output link, then the fiber selection on the next link can be chosen only from that subset. Taking this switching constraint into consideration, we first filter out all unreachable fibers along the route, and then find the first fit slot set on the first fit fibers. We call this baseline algorithm for the "FLEX" architecture as $FLEX_{SPFF}$.

Figure 5.3 and 5.4 show the performance results when utilizing our proposed algorithm, given different number of static traffic requests for the simulation settings

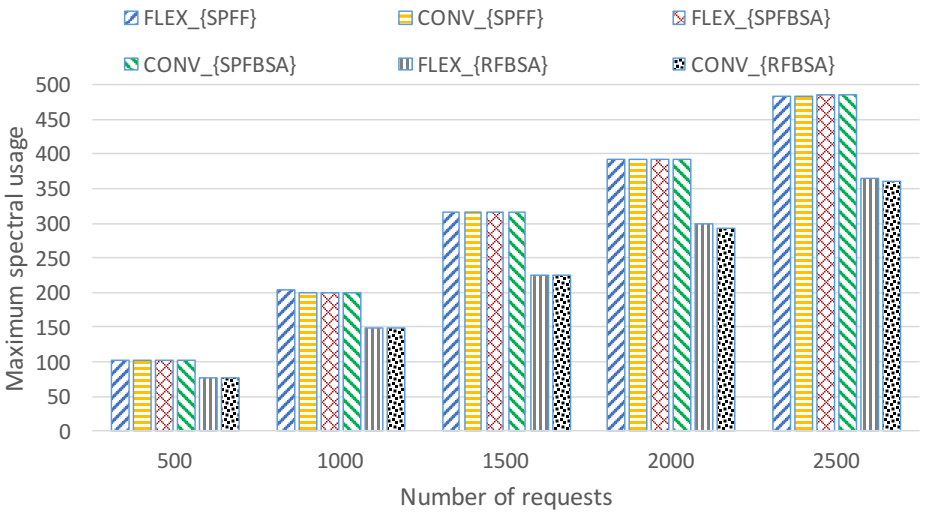

Figure 5.3: Performance comparison results for NSF network (F=2).

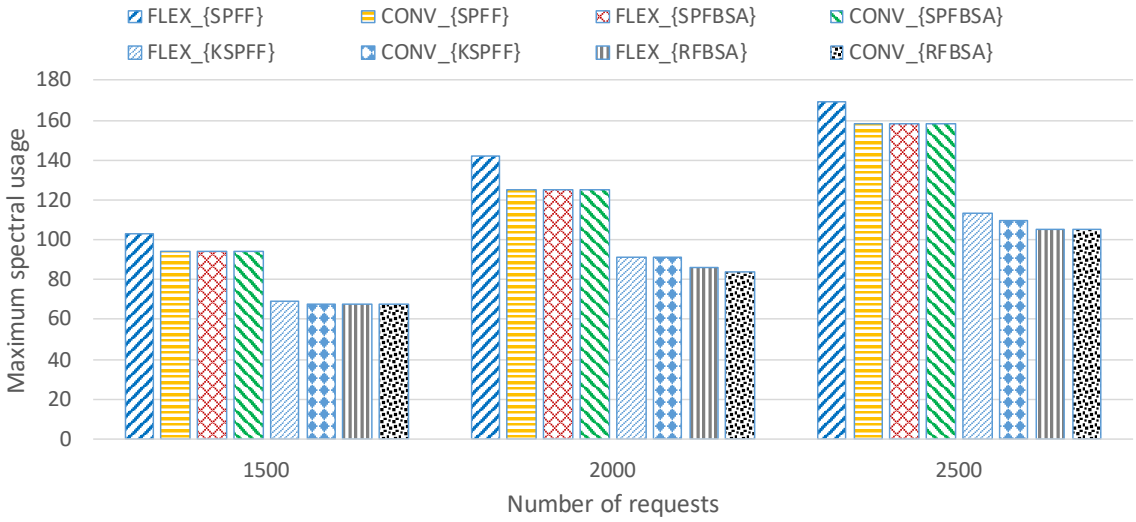

Figure 5.4: Performance comparison results for NSF network (F=[5,10]).

$F = 2$ and $F = [5, 10]$, respectively. The parameter settings for the above results are: $\alpha = 1$, and $\beta = 1$. This parameter set works well because both the waveband and slot status are jointly considered with the same weight when making allocation decisions. The performance results are compared with those of $CONV_{SPFF}$ and $FLEX_{SPFF}$. We also compare the performance of our joint RFBSA algorithm with that of an algorithm in which the routing is fixed (smallest number of hops) but the frequency slots are selected using our auxiliary layered-graph approach. We call this algorithm $FLEX_{SPFBSA}$ and $CONV_{SPFBSA}$ for the two architectures. In Figure 5.4, we also

Figure 5.5: Comparison between KSPFF and RFBSA for NSF network (F=[5,10]).

compare the performance of our joint RFBSA algorithm with that of KSPFF for both architectures (denoted as $FLEX_{KSPFF}$ and $CONV_{KSPFF}$).

We make several observations from the results. First, we note that SPFBSA and SPFF perform very similarly for both the "FLEX" and the "FLEX" architectures when $F = 2$. This implies that the first fit slot assignment is sufficient when SP routing is used in networks with a small number of fibers. Further, when $F = 2$, the performance of the "CONV" architecture and the "FLEX" architecture are almost the same using any algorithm. The waveband constraint does not have much influence because the number of fibers per link is small. For the $F = [5, 10]$ case, due to the waveband switching constraints, the MSU of $FLEX_{SPFF}$ is slightly higher than that of $CONV_{SPFF}$. In this case, $FLEX_{SPFBSA}$ can find better results (almost the same as $CONV_{SPFF}$ and $CONV_{SPFBSA}$) than $FLEX_{SPFF}$.

Our RFBSA algorithm gives much better results than the baseline algorithms with shortest path routing for both the "CONV" architecture and "FLEX" architecture. Note that there is an improvement of $25 - 33\%$ in slot usage. Since MSU may be limited by a single fiber in the network, we conduct another set of comparison between the KSPFF and our RFBSA algorithm by using the average MSU across the network as performance indicator. Figure 5.5 shows that RFBSA outperforms KSPFF.

This demonstrates the effectiveness of joint routing, fiber, band, and spectrum

selection. We also note that the "FLEX" architecture, even with a small $B(= 4)$, can achieve a performance comparable to the "CONV" architecture through judicious resource allocation. Recall that the "FLEX" architecture is much less expensive than the "CONV" architecture.

5.4 Conclusions

In this chapter, the joint routing, fiber, waveband, and spectrum allocation (RFBSA) problem for EONs with a novel cross-connect architecture and multi-fiber links is investigated. Our main contributions can be summarized as follows:

- We formulate a new and practical problem called the RFBSA problem. To the best of our knowledge, this is the first work that considers the RFBSA problem.
- We propose an auxiliary layered-graph framework for solving the RFBSA problem. The framework is suitable for solving different objective functions, and is cost-function pluggable. We also propose cost functions for minimizing the maximum used FS.
- Simulation results show the effectiveness of our framework by comparing its performance results with the results from a standard baseline algorithm.

Chapter 6 Joint Banding-node Placement and Resource Allocation for Multi-granular Elastic Optical Networks

In this chapter, we continue our work based on the previous chapter. The following problem is addressed. Given the total number of available WSSs for the network as a budget for network design, determine how many FLEX nodes to deploy and where to deploy them, and solve the RFBSA problem jointly to optimize the network performance.

6.1 Background and Problem Statement

Due to the switching constraint in the FLEX architecture, deploying FLEX nodes in the network may cause worse performance than utilizing CONV nodes. To achieve both cost efficiency and good performance, the deployment of CONV and FLEX nodes should be carefully addressed. Different node placements may need different numbers of WSSs. If a budget in terms of number of available WSSs is given for network planning, we would like to determine how many FLEX nodes should be deployed and where to place them.

Each request can be represented by a source node, a destination node, and a bandwidth requirement in terms of number of required FSs. A path (including links and fibers) that satisfies routing constraints caused by limited wavebands, and a set of contiguous FSs should be assigned to accommodate the request. The node placement will have a direct impact on the resource assignment and performance of the network, as different FLEX node placements will cause different switching constraints. Thus, we should jointly consider the *RFBSA* and node placement problem.

For a given set of traffic requests and a network planning budget in terms of the number of WSSs of a given fixed port count, our objective is to find the number and locations of FLEX nodes as well as the resource allocation (FSs on fibers) of requests to minimize the total maximum spectrum usage (MSU), which is the sum of the maximum slot indices used on all fibers in the network. In the case of dynamically

arriving and departing traffic requests, given a planning budget, our objective is to determine node placements and perform resource allocation to minimize the demand blocking ratio for dynamic traffic requests. The demand blocking ratio is defined as the ratio of the sum of bandwidths of blocked requests to the sum of bandwidths of all requests. Since requests can have different bandwidth requirements, we use demand blocking ratio instead of lightpath blocking ratio as the performance indicator.

6.2 The Integer Linear Programming Model

We now present an ILP formulation for the joint problem. The ILP model can be used to solve small problem instances for a set of static connection requests.

6.2.1 Notations

The input parameters to the ILP formulation are shown in Table 6.1. Consider a network $\mathcal{G} = (\mathcal{V}, \mathcal{L})$, where $\mathcal{V}$ denotes the set of OXC nodes, and $\mathcal{L}$ denotes the set of unidirectional links.

6.2.2 Formulations

We use $\mathcal{J}$ to denote a given set of requests. Request $j \in \mathcal{J}$ requires d^j FSs with source node s^j and destination node t^j. The K shortest paths for each request are precomputed. T denotes the network planning target, which is the total number of available WSSs. Our goal is to determine the placement of FLEX nodes and resource allocation for requests. The MSU for a fiber is defined as the highest slot index utilized to accommodate demands on that fiber.

Variables:

a)

$$a_s^j = \begin{cases} 1, & \text{if starting slot of request } j \text{ is } s; \\ 0, & \text{otherwise} \end{cases}$$

Table 6.1: Notation for the banding-node placement problem

Symbol	Meaning
G	the network topology
$\mathcal{V}$	the set of OXC nodes
$\mathcal{L}$	the set of unidirectional links
f_l	the number of parallel fibers on link $l \in \mathcal{L}$
M	number of physical nodes in the network
L	number of physical links in the network
F	total number of fibers in the network
B	the band limit of FLEX nodes
T	the budget, the maximum available number of WSSs
K	the number of predetermined paths between each node pair
v	an arbitrary network node
e	an arbitrary network link
f	an arbitrary network fiber
s	an arbitrary slot
p	an arbitrary path ($p_{s,d,r}$ is the rth shortest path from node s to node d)
$P^e_{s,d,r}$	$= 1$ if link e is on path $p_{s,d,r}$; $= 0$, otherwise
ξ^f_e	$= 1$ if fiber f is on link e; $= 0$, otherwise
IN^f_v	$= 1$ if fiber f is an input fiber of node v; $= 0$, otherwise
OUT^f_v	$= 1$ if fiber f is an output fiber of node v; $= 0$, otherwise
ζ_v	the number of WSSs needed if node v is CONV
Γ_v	the number of WSSs needed if node v is FLEX
Ω	estimated upper bound of maximum slot usage
$\mathcal{J}$	a given set of requests
J	number of requests
j	an arbitrary request
s^j	source node of request j, $s^j \in \mathcal{V}$
t^j	destination node of request j, $t^j \in \mathcal{V}$
d^j	the required number of slots for request j

b)

$$z_s^j = \begin{cases} 1, & \text{if request } j \text{ uses slot } s; \\ 0, & \text{otherwise} \end{cases}$$

c)

$$y_f^j = \begin{cases} 1, & \text{if fiber } f \text{ is allocated to accommodate request } j; \\ 0, & \text{otherwise} \end{cases}$$

d)

$$x_{f,s}^j = \begin{cases} 1, & \text{if slot } s \text{ on fiber } f \text{ is allocated to request } j; \\ 0, & \text{otherwise} \end{cases}$$

e)

$$w_{f_a,f_b}^{j,v} = \begin{cases} 1, & \text{if there is a switching from fiber } f_a \text{ to} \\ & \text{fiber } f_b \text{ at node } v \text{ caused by request } j; \\ 0, & \text{otherwise} \end{cases}$$

f)

$$W_{f_a,f_b}^{v} = \begin{cases} 1, & \text{if there is a switching from fiber } f_a \\ & \text{to fiber } f_b \text{ at node } v; \\ 0, & \text{otherwise} \end{cases}$$

g)

$$C_v = \begin{cases} 1, & \text{if node } v \text{ is chosen to use CONV architecture;} \\ 0, & \text{if node } v \text{ is chosen to use FLEX architecture} \end{cases}$$

h)

$$\lambda_r^j = \begin{cases} 1, & \text{if request } j \text{ uses path } r \text{ of the } K \text{ shortest paths;} \\ 0, & \text{otherwise} \end{cases}$$

Our objective is to minimize the average MSU, or, equivalently, the total MSU over all fibers, as the number of fibers for a single network is constant:

$$\text{Minimize} \quad \textstyle\sum_{f=1}^{F} \max_s (s \cdot \sum_{j=1}^{J} x_{f,s}^j)$$

Here, $(s \cdot \sum_{j=1}^{J} x_{f,s}^j)$ denotes the index of a slot that is used by some request on fiber f.

Constraints:

a) There is only one starting slot index for each request. For all j,

$$\sum_{s=1}^{\Omega-d^j+1} a_s^j = 1, \quad \sum_{s=\Omega-d^j}^{\Omega} a_s^j = 0$$

b) The spectrum contiguity should be met for each request. The slots assigned to accommodate a request must use consecutive slots from its starting slot. For all j, s,

$$\sum_{i=0}^{d^j-1} z_{s+i}^j \geq d^j \cdot a_s^j, \quad \sum_{s=1}^{\Omega} z_s^j = d^j$$

c) Only one of the K precomputed shortest paths should be chosen for a request. For all j,

$$\sum_{r=1}^{K} \lambda_r^j = 1$$

d) Only one fiber on each link along the chosen path is selected to route each request. For all j, e,

$$\sum_{f=1}^{F} \xi_e^f \cdot y_f^j = \sum_{r=1}^{K} P_{s,d,r}^e \cdot \lambda_r^j$$

e) The value of $x_{f,s}^j$ should be based on both y_f^j and z_s^j. Slot s on fiber f is used by request j if and only if slot s is assigned to j, and fiber f is allocated on the path. For all j, f, s,

$$y_f^j + z_s^j \leq 1 + x_{f,s}^j, \quad y_f^j + z_s^j \geq 2x_{f,s}^j$$

f) Any slot on any fiber can accommodate at most one request. For all f, s,

$$\sum_{j=1}^{J} x_{f,s}^j \leq 1$$

g) The value of $w_{f_a,f_b}^{j,v}$ is based on the value of $y_{f_a}^j$ and $y_{f_b}^j$. For any two fibers f_a and f_b, there is a switching between them to route request j if f_a is an input fiber and f_b is an output fiber of a node, and both f_a and f_b are allocated to the request. For all j, v, f_a, f_b,

$$y_{f_a}^j \cdot IN_v^{f_a} + y_{f_b}^j \cdot OUT_v^{f_b} \leq 1 + w_{f_a,f_b}^{j,v}$$

$$y_{f_a}^j \cdot IN_v^{f_a} + y_{f_b}^j \cdot OUT_v^{f_b} \geq 2w_{f_a,f_b}^{j,v}$$

h) There is a waveband from fiber f_a to fiber f_b if at least one request is routed by this switching. For all v, f_a, f_b,

$$\sum_{j=1}^{J} w_{f_a,f_b}^{j,v} \geq W_{f_a,f_b}^v, \quad \sum_{j=1}^{J} w_{f_a,f_b}^{j,v} \leq J \cdot W_{f_a,f_b}^v$$

i) The total required number of WSSs for the whole network should be no larger than the given budget.

$$\sum_{v=1}^{M} C_v\zeta_v + (1 - C_v)\Gamma_v \leq T$$

j) The routing capacity limit should be satisfied for each input fiber on any FLEX node. Here, Λ is a large value to loosen the constraint for CONV nodes. For all v, f_a,

$$\sum_{f_b=1}^{F} W^v_{f_a,f_b} \leq B + \Lambda C_v$$

6.3 Heuristic Solutions

Our proposed heuristics to solve the problem are presented in detail in this section. We first describe our framework to solve the *RFBSA* problem for static instances. Then we propose the node placement scheme which is based on both topology information and results of *RFBSA*. Finally we apply the proposed schemes to accommodate dynamic traffic requests.

6.3.1 Route, Fiber, Band and Slot Assignment Framework

For a given set of requests, we first sort them based on the bandwidth requirements in non-increasing order. The requests are stored in an ordered list $\mathcal{J}$. We consider resource assignments in order from the list, i.e., starting with the request having highest bandwidth demand.

We start by describing the auxiliary layered-graph framework for solving the assignment problem. The vertexes in an auxiliary graph represent all input and output fibers (denoted by I vertices and O vertices). The edges between vertexes are classified into two types. The first type of edges is from an I vertex to an O vertex, representing the switching inside a corresponding physical node. Since there is a band limit on FLEX nodes in the network, we define a switching cost between each input fiber and each output fiber on a physical node. The costs of this type of edges are related to the wavebanding status of the node. The other type of edges from an O vertex to an I vertex represents the physical link connection. The costs of such edges are related to the spectrum usage on the particular fiber.

In the following, we briefly describe the algorithm for route, fiber, band, and slot assignment. The pseudocode is shown in Algorithm 11. We modify the algorithm in our previous work [105] to reduce the time complexity.

For each request j, we examine the K shortest paths. For each path k, we generate a basic auxiliary graph AG^k based on the links and fibers on this particular path. The algorithm SWITCHING COST UPDATE is used to update the costs of the first type of edges in AG^k.

The switching cost function is shown in Equation (6.1). We denote $C_{v,f_{in},f_{out}}$ as the switching cost or cost of the edge from vertex f_{in} to vertex f_{out} in physical FLEX node v.

$$C_{v,f_{in},f_{out}} = \begin{cases} 0, & \text{if waveband is established} \\ \alpha \cdot \frac{b}{B}, & \text{if } b < B \\ \infty, & \text{otherwise} \end{cases} \tag{6.1}$$

The basic idea of Switching Cost Update (Equation (6.1)) is to give preference to existing wavebands (if any) so as to leave more waveband choices for later requests. When computing the switching cost from an input fiber f_{in} to an outgoing fiber f_{out} of a FLEX node v, we use the following rules. If the waveband from f_{in} to f_{out} already exists, the switching cost from f_{in} to f_{out} is set to 0. Otherwise, if the number of already established wavebands from f_{in} to other outgoing fibers of node v is $b < B$, the switching cost is set to $\alpha \cdot \frac{b}{B}$, where α is a tuning parameter. If the number of established wavebands corresponding to f_i equals B, the switching cost is set to be ∞, which means that no new waveband can be established from f_{in}. If v is a CONV node, the switching costs from any input fiber to any output fiber of that node are set to 0.

Then we copy AG^k to generate an auxiliary graph for each outgoing fiber from s^j in the first link of that path for the request. Each auxiliary graph involves one outgoing fiber from source node s^j, and includes all remaining nodes and fibers in the kth shortest path. Let $|f_{l_0}^k|$ denote the number of fibers on the first link of the kth shortest path. In total, we generate $\sum_k |f_{l_0}^k|$ auxiliary graphs.

By fixing one outgoing fiber a in the first link of the path, we only need to consider the available FSs on that fiber, thereby reducing the complexity of searching for a

Algorithm 11 *RFBSA* Algorithm

Input: $G = (V, E)$, request j, K shortest paths

Output: $path_j$ — Fiber and slot assignment for j

for k : K shortest paths of request j **do**

Create auxiliary graph AG_k

Switching Cost Update along path k

$C^k_{total} = \infty$

for a : outgoing fibers from s^j **do**

Copy AG_k to create auxiliary graph AG^k_a

$C^k_a = \infty$

for $\widetilde{ss}$: available FS sets of size d^j on a **do**

SI = the start FS index of $\widetilde{ss}$

Create layered graph $AG^k_{a,SI}$

Spectrum Cost Update

Use Dijkstra's algorithm to find a lightpath with the smallest cost $C^k_{a,\widetilde{ss}}$

if $C^k_{a,\widetilde{ss}} < \infty$ **then**

$C^k_a = C^k_{a,\widetilde{ss}}$

BREAK

end if

end for

if $C^k_a < C^k_{total}$ **then**

$C^k_{total} = C^k_a$

end if

end for

end for

$path_r$ = a lightpath p with the smallest $C^k_{total}, \forall k$

valid lightpath. In an auxiliary graph $AG_a^k = (V_a^k, E_a^k)$, we define a layered graph on FS level $AG_{a,SI}^k = (V_{a,SI}^k, E_{a,SI}^k)$ for a specific spectrum range from FS index SI to $SI + d^j - 1$. Thus, in the layered graphs, spectrum contiguity, spectrum continuity, and spectrum non-overlapping constraints are automatically satisfied. For auxiliary layered graph $AG_{a,SI}^k$ corresponding to the outgoing fiber a of s^j, we need to do Spectrum Cost Update for the second type of edges, which correspond to all fibers on the physical links along the path regarding an available slot set $\widetilde{ss}$ on fiber a.

Algorithm 12 SWITCHING COST UPDATE

Input: v, f_{in}, f_{out}

Output: $C_{v,f_{in},f_{out}}$

for f_{in} : all incoming fibers in node v **do**

  for f_{out} : all outgoing fibers in node v **do**

    b = the number of established wavebands from f_{in}

    if the waveband is established **then**

      $C_{v,f_{in},f_{out}} = 0$

    else if $b < B$ **then**

      $C_{v,f_{in},f_{out}} = \alpha \cdot b/B$

    else

      $C_{v,f_{in},f_{out}} = \infty$

    end if

  end for

end for

For each contiguous slot set $\widetilde{ss}$ ranging from SI to $SI + d^j - 1$ that is available on source ouput fiber a of path k, the spectrum cost is defined in Equation (6.2).

$$C_{a,\widetilde{ss},f}^k = \begin{cases} 1, & \text{if } SI + d^j \leq m_f \\ \infty, & \text{if } \widetilde{ss} \text{ is not available on } f \\ (SI + d^j - m_f)/\Omega + 1, & \text{otherwise} \end{cases} \tag{6.2}$$

The basic idea of Spectrum Cost Update (Equation (6.2)) is to try to not increase MSU on each fiber after the request is established. We denote m_f as the largest FS

Algorithm 13 SPECTRUM COST UPDATE

Input: f, $\widetilde{ss}$, SI, d^j

Output: $C^k_{a,\widetilde{ss},f}$

m_f = the largest FS index on f

if $\widetilde{\widetilde{ss}}$ is not available on f **then**

$C^k_{a,\widetilde{ss},f} = \infty$

else if $SI + d^j \leq m_f$ **then**

$C^k_{a,\widetilde{ss},f} = 1$

else

$C^k_{a,\widetilde{ss},f} = (SI + d^j - m_f)/\Omega + 1$

end if

index on fiber f and $C^k_{a,\widetilde{ss},f}$ as the spectrum cost on fiber f for the slot set $\widetilde{ss}$ in AG^k_a. If $\widetilde{ss}$ is not available on fiber f, $C^k_{a,\widetilde{ss},f}$ is set to ∞. This means that we cannot establish a lightpath by using the slot set $\widetilde{ss}$ on fiber f. If $\widetilde{ss}$ is available and the ending FS index is no more than m_f, $C^k_{a,\widetilde{ss},f}$ is set to 1. In this case, the local MSU on fiber f would not increase, and the spectrum cost works as a hop count. Otherwise, $C^k_{a,\widetilde{ss},f}$ is set to be $(SI + d^j - m_f)/\Omega + 1$, which is proportional to the number of slots by which the local MSU increases on that fiber, while taking the hop count into consideration.

For a lightpath l in the kth path, we denote $C^k_{a,\widetilde{ss}}$ as the total cost combining Spectrum Costs $\sum_f C^k_{a,\widetilde{ss},f}$ of the fiber set F_l and Switching Costs $\sum_v C_{v,f_{in},f_{out}}$ of the node set V_l which the lightpath is going through. Once the costs of all edges in the layered graph are determined, we use Dijkstra's algorithm to find the shortest paths from one output fiber of the source node to different input fibers of the destination node, and choose the valid lightpath p with smallest path cost. In order to decrease the algorithm's complexity, we stop generating further layered graphs for this auxiliary graph, and consider p as a *candidate lightpath*. We compare the path costs of all the candidate lightpaths and choose a path with the smallest total cost.

Suppose the total number of links in the network is L, the maximum number of fibers per link is ε, and Ω is an estimated upper bound of the MSU. For each request,

the number of hops in the path of a mesh network, denoted as h, is usually small compared to L. For each output fiber of the source node, $O(\Omega)$ auxiliary graphs are created. Since each path is precalculated, each auxiliary graph consists of $O(h\varepsilon)$ vertex and $O(h\varepsilon^2)$ edges ($O(\varepsilon^2)$ first type of edges on each intermediate node of the path and $O(\varepsilon)$ second type of edges on each link of the path). The dominant part is to find the shortest path in the auxiliary graphs, thus the time complexity of the algorithm is $O(\Omega h\varepsilon^3 \log(h\varepsilon))$, which is much less than the time complexity of our previous algorithm $O(\Omega L\varepsilon^3 \log(L\varepsilon))$, that generates the auxiliary graphs based on the whole network.

6.3.2 Node Placement Schemes

Given only the budget, i.e., the total number of available small-port-count WSSs for the network, the number and location of FLEX nodes should be determined. Assume all nodes are initially CONV nodes; by replacing CONV nodes with FLEX nodes, the required number of WSSs will be decreased. Then the problem transforms to how many and which CONV nodes should be replaced to meet the budget. Of course, the performance of the network after replacement should be taken into consideration when making the replacements.

Indeed, how many CONV nodes need to be replaced also depends on where they are. Replacing a node with a larger port count may have the same reduction in number of WSSs as replacing two nodes with small port counts. Therefore, we divide the joint placement and assignment problem into two stages. The first stage is to decide the location of FLEX nodes. Then a routing path, fibers on links along the path, waveband, and a slot set are selected for each request by our *RFBSA* algorithm.

6.3.2.1 Random Node Placement

We use a random node placement scheme as the baseline. We first try to randomly select a CONV node to replace. The number of required WSSs is reduced accordingly. The procedure of randomly replacing the remaining CONV nodes and updating the required number of WSSs is performed repeatedly until the budget constraint is met.

6.3.2.2 Traffic-aware Node Placement

For a better node placement scheme, the network topology and the traffic requests should be taken into consideration. The first intuition is that if fewer CONV nodes are replaced with FLEX, the performance degradation would be less. In this case, we would like to replace the physical nodes with larger port count first. The physical nodes are sorted by their port counts in descending order and replaced one by one to meet the budget.

On the other hand, physical nodes with large port counts usually have more traffic going through them. Replacing these nodes might affect more connection requests and add routing constraints to them, leading to worse performance. Accordingly, the node placement should consider the traffic pattern, as well as the port counts of physical nodes.

We assume all nodes are CONVs initially, and the *RFBSA* is applied to accommodate all requests in the network. To facilitate node placement, the band usage information and the total number of requests going through each physical node are recorded. A cost function related to the traffic for each node v is defined in Equation (6.3). Let $maxb_v$ denote the maximum number of bands originated from all input fibers of node v. $maxb_v$ is related to the waveband usage on each node. Δ_v denotes the number of traffic demands going through node v (excluding requests originating or terminating at this node). Let D_v denote the port count of node v. The larger D_v is, the more reduction in required number of WSSs that is achieved. When $maxb_v$ is not larger than the band limit B, we set the cost to 0, otherwise we set the cost to the traffic density (Δ_v/D_v) that would be affected by the limited banding. The larger C_v a node has, the more traffic it will affect when replaced by FLEX. Nodes with a small cost C_v are preferred to be replaced. We sort the nodes in ascending order of the costs C_v, and replace them one by one until the budget constraint is met.

$$C_v = \begin{cases} 0, & \text{if } maxb_v \leq B \\ \Delta_v/D_v, & \text{otherwise} \end{cases} \tag{6.3}$$

When all the FLEX nodes are placed, the route, fiber, band and slot assignment

for each request should be determined in the mixed network. The *RFBSA* algorithm described in the previous subsection is utilized to assign the route, fibers, wavebands and a contiguous slot set to accommodate each request. We finally select the node placement by comparing the performance of the above two strategies for each budget.

6.3.3 Extension to Dynamic Instances

To accommodate dynamically arriving traffic requests, we should determine the node placements first. Given the information on network topology and traffic (request size distribution and traffic pattern), we first generate a large number of static requests according to the given traffic pattern, and use the Traffic Aware Placement introduced in the previous subsection to find appropriate FLEX node deployments. Then the *RFBSA* algorithm is applied to the mixed network for the arriving requests. In the dynamic case, the goal is to minimize demand blocking ratio. We adapt the spectrum cost function in the *RFBSA* algorithm to achieve good performance. $C^k_{a,\widetilde{ss},a} = SI$ denotes the spectrum cost on one outgoing fiber a from the source node for its available slot set $\widetilde{ss}$ and $C^k_{a,\widetilde{ss},f} = 1$ if $\widetilde{ss}$ is available on fiber f along the path k. Here we try to compact the total utilized network resources to leave more resources for later requests. The cost function emphasizes both starting slot and hop count of the path.

6.4 Simulation Results

In this section, we present performance evaluation results for both static and dynamic instances to demonstrate the effectiveness of our proposed method. ILP results for small network instances and simulation results for large network instances are generated to examine the effects of changing budget on the performance of the network.

Given a set of connection requests and a budget for the network, the number and locations of FLEX nodes should be determined, and networking resources need to be allocated to the requests. Each request represents a connection from a source node to a destination node with a demand size requirement. A uniform traffic pattern

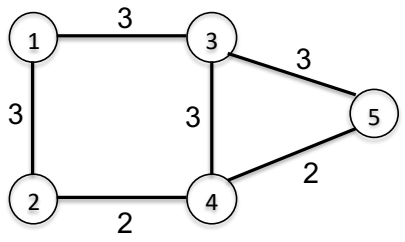

Figure 6.1: Small network. The numbers are the number of fibers on the links.

means the source/destination nodes are randomly selected from the physical nodes of the network. We assume three types of demands in terms of the number of required FSs [2], with the following distribution: 3 slots (40 Gbps) with probability 0.2, 4 slots (100 Gbps) with probability 0.5, and 7 slots (400 Gbps) with probability 0.3. For the FLEX nodes, the maximum number of wavebands B is assumed to be 4 or 9.

For a given set of static traffic requests, we record the total MSU. The average MSU over the total number of fibers is used as the performance measure. The parameter α is set to 1 for RFBSA.[1] In the node selection scheme, both band usage and the traffic distribution are considered. RP is used to denote the random node placement strategy, while TAP represents the traffic aware node placement strategy in the following results.

6.4.1 Results for a Small Network

We get single-run ILP results for a 5-node small network as shown in Figure 6.1. The number of parallel fibers, x, on each link is randomly distributed between 2 and 3 ($x = [2, 3]$ per link). The band limit $B = 4$.

Table 6.2 shows the average MSUs over all fibers achieved by the ILP and our proposed heuristics for different numbers of requests in the 5-node small network. The given budget is 89 WSSs (which is between the required number of WSSs for all FLEX nodes - 64, and all CONV nodes - 125). We only simulate the $K = 1$ case here due to the complexity of the ILP. For example, the execution time of the heuristics is a few seconds, while that of ILP is more than 6 hours. We can see that the results from TAP are better than for RP, and are not far from the ILP's results.

[1]This was chosen based on observed performance for a large set of parameters.

Table 6.2: Average MSU results for the small network.

Number of Requests	RP	TAP	ILP
40	10.25	9.5	9.03125
50	13.53125	12.46875	12.125
60	15.8125	14.96875	14
100	24.75	23.0625	22.21875

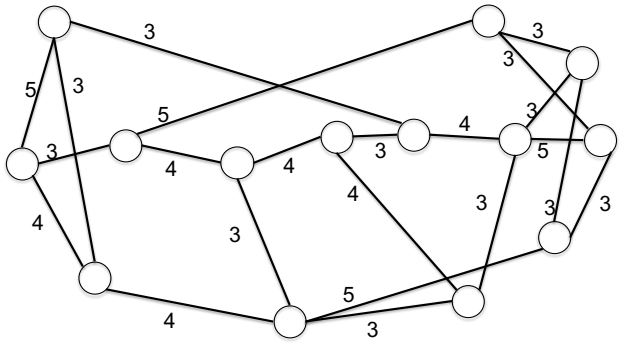

Figure 6.2: NSF network. The numbers are the number of fibers on the links.

6.4.2 Results for Larger Topologies

The larger topologies used for simulations are the NSFNET network and pan-European network. The NSFNET as shown in Figure 6.2 has 14 physical nodes and 22 bidirectional links [46], while the pan-European network as shown in Figure 3.7 consists of 28 physical nodes and 43 bidirectional links [106]. The number of fibers, x, on each link is randomly distributed between 3 and 5 ($x = [3, 5]$ per link). The network topology is fixed for every run of the simulation.

Figures 6.3 and 6.4 show the performance change in the NSF network as the given

Table 6.3: Average number of FLEX nodes for NSF network (B=4).

Schemes \ Budget	320	434	548	700	891	1000	1142
TAP	14	11.66	9.52	6.36	3.98	2.68	0
RP	14	12.436	10.52	8.03	4.78	2.838	0

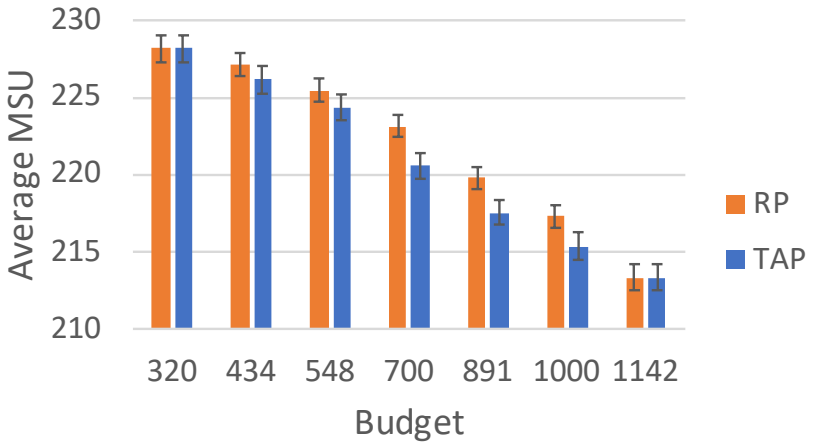

Figure 6.3: Network performance vs. budget for NSF network (B=4).

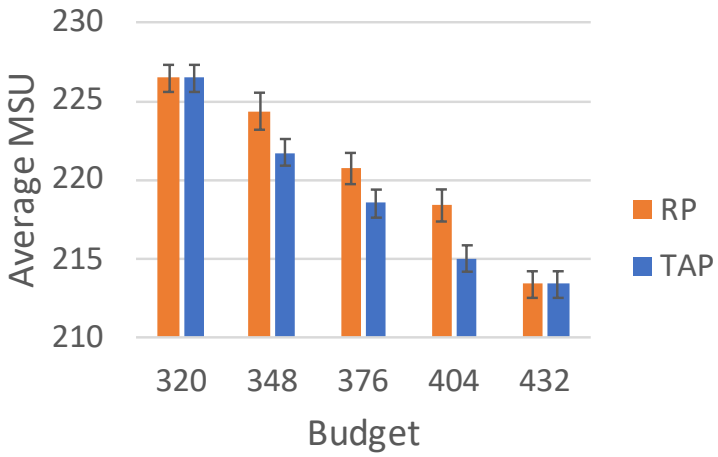

Figure 6.4: Network performance vs. budget for NSF network (B=9).

budget changes. We conducted 50 trials of simulation, each consisting of 3000 traffic requests, and show the average results with 95% confidence intervals. Figure 6.3 shows the comparison results when $B = 4$, while Figure 6.4 is for $B = 9$. We use $K = 3$ for all simulations. The range of the budget (number of available WSSs) for the network varies from 320 (all nodes are FLEX) to 1142 (all nodes are CONV) when $B = 4$, and to 432 when $B = 9$. Table 6.3 shows the illustrative results depicting the average number of FLEX nodes for each budget in the 14-node NSF network when $B = 4$.

Similarly, Figures 6.5 and 6.6 show the performance change in the pan-European network as the given budget changes. We use similar settings for the simulations. Figure 6.5 shows the comparison results when $B = 4$, while Figure 6.6 is for $B = 9$. The range of the budget (number of available WSSs) for the network varies from 652

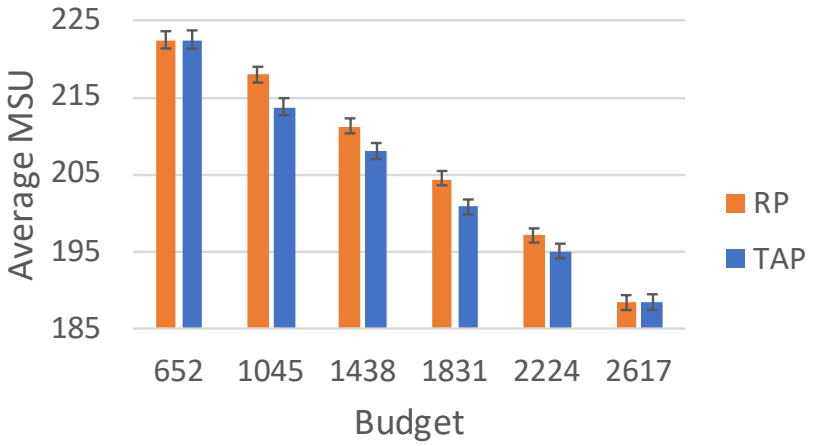

Figure 6.5: Network performance vs. budget for pan-European network (B=4).

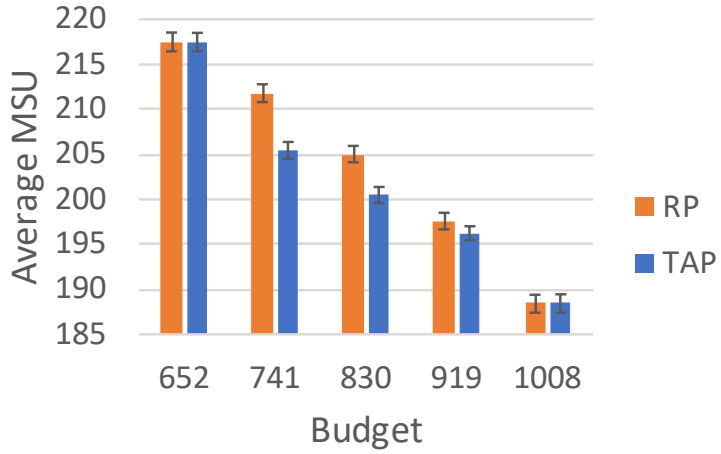

Figure 6.6: Network performance vs. budget for pan-European network (B=9).

(all nodes are FLEX) to 2617 (all nodes are CONV) when $B = 4$, and to 1008 when $B = 9$.

As expected, the average MSU increases, which means network performance degrades, as the budget decreases (more FLEX nodes and fewer CONV nodes). The comparison of the two node placement strategies is shown in the figures. When all nodes are FLEX or CONV, there is no difference between RP and TAP as the network is the same for both policies. In general, TAP performs better than RP in the sense that fewer slots are required to accommodate all requests. However, even in the worst case, the degradation of network performance is relatively small, which means that it may be possible to replace most CONV nodes with FLEX nodes in the network without a significant penalty. We note that the MSU is increased by 10% to 15% for

the two topologies when all CONV nodes are replaced by FLEX nodes. However, the number of WSSs needed can be reduced by a factor of 2 or more in many cases. These results suggest that the slight increase in spectrum cost is more than offset by the savings in WSSs.

6.4.3 Results for Dynamic Requests

The traffic demand is a set of dynamically arriving connection requests. Connection requests arrive to the network according to a Poisson process. Each request has a mean holding time of 1 (arbitrary time unit), and the arrival rate of traffic requests is varied in order to examine the network performance under varying offered loads (denoted by L). We use the demand blocking ratio of dynamic traffic requests to indicate the network performance. The parameter α is set to 100[2] for *RFBSA*. For each simulation, the results of 200,000 dynamic requests excluding 10,000 warm-up requests are recorded.

The NSNET topology is used for this set of simulations. Each link has a random number of fibers, x, that are uniformly distributed between 5 and 10 fibers ($x = [5, 10]$ per link). We assume the fiber capacity of 352 frequency slots, with each slot having a bandwidth of 12.5 GHz. The banding limit of FLEX nodes is assumed to be 4 (i.e., $B = 4$). The range of the budget for the network varies from 660 (all nodes are FLEX) to 5137 (all nodes are CONV).

We first show a set of simulation results for the uniform traffic pattern. A set of static connection requests is first generated according to this traffic pattern in order to determine the node placement to satisfy a given budget of the network. Then the demand blocking ratio of dynamic requests is evaluated. Figures 6.7 and 6.8 show the performance change in the NSF network as the given budget changes for different traffic loads. We can see that the network performance degrades (demand blocking ratio increases) as the budget decreases. In general, TAP performs better than RP (lower demand blocking ratio under each budget). When the traffic load is small, the difference between TAP and RP is more obvious.

[2]This was chosen based on observed performance for a large set of parameters.

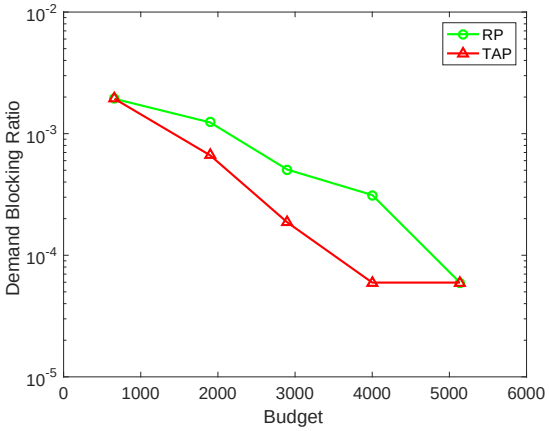

Figure 6.7: Network performance vs. budget for NSF network with uniform traffic pattern (L=6900).

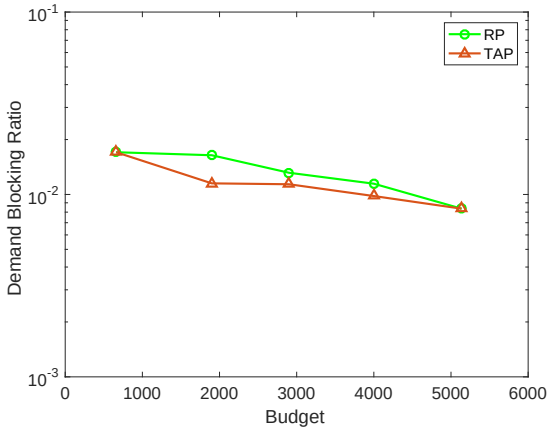

Figure 6.8: Network performance vs. budget for NSF network with uniform traffic pattern (L=7400).

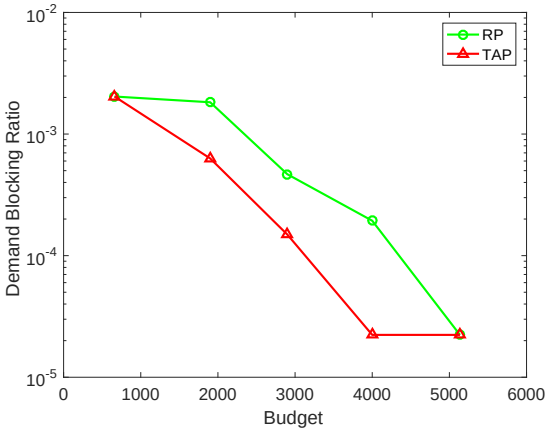

Figure 6.9: Network performance vs. budget for NSF network with non-uniform traffic pattern (L=7100).

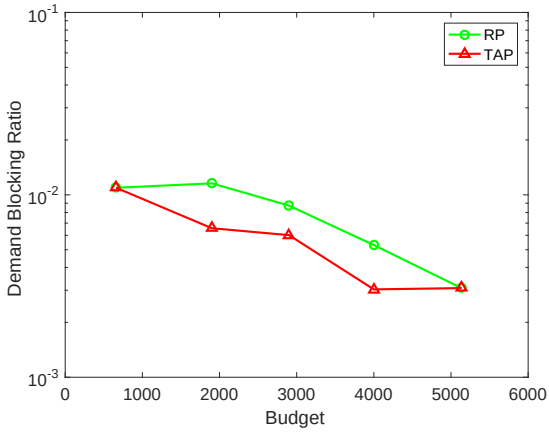

Figure 6.10: Network performance vs. budget for NSF network with non-uniform traffic pattern (L=7400).

We also conduct simulations for a non-uniform traffic pattern. We assume that nodes with higher connectivity have larger opportunity to send and receive traffic, and choose the probability that a node v is selected as a source or destination, u_v, in proportion to the node's physical degree. Again, a set of static connection requests of this traffic pattern is generated in order to determine the node placement for each budget. The comparison results for different network loads are shown in Figures 6.9 and 6.10. We can see similar trends as in previous results.

We also conduct sensitivity tests to see how well our TAP scheme performs when the

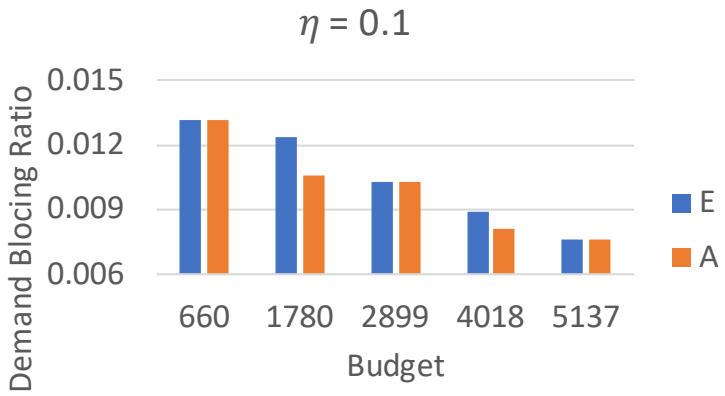

Figure 6.11: Sensitivity test for traffic pattern with a perturbation of 0.1.

actual traffic pattern is slightly different from what the network was planned for. Here, we perturb the source/destination selection probabilities u_v by a factor η_v (η_v could be either positive or negative, with the average of $|\eta_v|$ being η). In other words, we set the node selection probabilities to be $(u_1(1+\eta_1), u_2(1+\eta_2), \cdots, 1-\sum_{v=1}^{N-1} u_v(1+\eta_v))$. Then, we compare the performance obtained by applying the original node placement to the changed traffic pattern with the performance obtained by applying the node placement for the changed traffic pattern. Figure 6.11 shows results for $\eta = 0.1$. The results are represented by “A" (with placement based on exact information) and “E" (without exact information). There is no performance difference between “E" and “A" when all nodes are FLEX or CONV because the node placements are the same. For other budgets, the difference in performance between “E" and “A" is relatively small, indicating that the algorithm is not sensitive to small changes in traffic pattern.

6.5 Conclusion

Flexible waveband OXCs require much less hardware cost than the conventional OXCs, with the penalty of some switching constraints. In this work, we jointly consider RFBSA and FLEX node placement to satisfy a network planning budget in terms of the total number of available WSSs [107]. In addition to an integer linear programming formulation, we present node placement schemes and extend the cost-function-pluggable auxiliary layered-graph framework in our previous work to solve this problem. The simulation results demonstrate that our heuristic solution

saves network resources and achieves good network performance indicated by the average maximum spectrum usage. The framework is also demonstrated to have good performance for dynamic traffic requests.

Chapter 7 Dynamic Routing and Spectrum Assignment for Multi-fiber Elastic Optical Networks

In this chapter, we study dynamic Routing and Spectrum Assignment problem in elastic optical networks with multiple fibers per link to further improve network performance. We use the demand blocking ratio for dynamically arriving requests as the performance indicator.

7.1 Related Work

The first comprehensive study on the RSA problem is [44], which formally defined the problem and proved its NP-hardness. In order to solve the RSA problem efficiently, many heuristics have been proposed in the literature. Dealing with routing and spectrum allocation problems jointly usually requires high complexity [47, 105, 108]. Many of the literature works solve this problem by decomposing it and solving the two subproblems in sequence. The shortest path with maximum spectrum reuse algorithm and balanced load spectrum allocation algorithm (which determines the routing by balancing the load in the network) are proposed in [44] to solve the static RSA problem. In [109], given a set of traffic requests, the authors try to find disjoint paths to route requests in order to increase slot reuse in SA. For the dynamic RSA problem, bandwidth fragmentation and spectrum misalignment caused by dynamic set up and tear down of traffic requests hurt the network performance. Many RSA schemes have been proposed in order to overcome the bandwidth fragmentation issue [110–116]. Defragmentation algorithms which reroute connections are developed in [117, 118]. Another scheme to eliminate bandwidth fragmentation without rerouting connections is to partition the spectrum for heterogeneous bandwidth demands. In [119, 120], different partition schemes are investigated, and the well-known First Fit spectrum allocation is used. In [121], the spectrum is partitioned by classifying connection groups. All the above RSA schemes are proposed for EONs with a single fiber per link.

To accommodate increasing traffic demands, deploying multiple fibers on a physical link is needed. The wavelength / slot assignment in networks with multi-fiber links differ from prior schemes of single-fiber networks in that multi-fiber links provide more flexibility in switching wavelengths / frequency slots. It's more complicated to design WA / SA algorithms to fully utilize this flexibility. The WA problem in multi-fiber WDM networks has been explored in the literature [122–128]. [122, 123] take the network state information into consideration and show good performance. Different cost functions based on the network state are utilized in the literature. Due to the heterogeneity of demands in EONs, which fiber to use on each link matters and will cause unnecessary fragmentation if not carefully addressed. In order to fully utilize the flexibility in multi-fiber links while satisfying constraints introduced by EONs, an efficient RSA scheme is necessary.

7.2 Contributions

In this study, we propose a novel and efficient solution for the dynamic RSA problem in multi-fiber elastic optical networks. Each link in the network contains multiple fibers. A network planning formulation based on the topology information is proposed so that the candidate paths for each source-destination node pair can be selected according to certain predetermined probabilities. Due to the heterogeneous request bandwidths in the network, a partition scheme is applied so that each request size can use a particular spectrum range. In this case, each fiber can be viewed the same and fiber selection on each link can also be avoided. Given the spectrum partitioning, an SA algorithm based on both the network state information and the path selection probabilities is proposed. The Next State Aware SA algorithm is further improved by considering the resource sharing among different partitions. For each arriving request, a routing path is first selected by the precomputed probabilities, then the next state aware SA is performed to assign contiguous FSs to that request. If there are no available FSs, the request will be blocked. We use the demand blocking ratio (ratio of the sum of bandwidths of blocked requests to total bandwidths of all requests) to indicate network performance.

Our contributions are: (1) we propose a multi-path selection method for the routing problem; (2) we utilize a spectrum management scheme to alleviate fragmentation caused by heterogeneous request bandwidths; (3) we adopt a spectrum assignment method that optimizes the state of network after the assignment. We demonstrate that each scheme is effective in improving the spectrum efficiency and network performance.

7.3 Background and Motivatoin

Consider a network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ denotes a set of optical cross-connects (OXCs), and $\mathcal{E}$ denotes a set of physical links. We assume that end nodes are connected to each OXC so that every OXC can be the origin or destination of connection requests. Each link contains multiple fibers, and the number of fibers on each link may be different. The spectrum resource on each fiber is carved up into frequency slots, with bandwidth of 12.5 GHz each. All fibers consist of the same number of FSs. At each OXC, assume that there are no switching constraints from input fibers to output fibers. A FS on an input fiber can be switched to the *same* FS on any output fiber.

A connection request to the network is an end-to-end lightpath with a source, a destination node, and a data rate requirement. The number of FSs assigned to a request depends on the data rate requirement and the modulation format. In this work, the same modulation format is assumed. Thus the bandwidth demand for each connection only depends on the data rate requirement. In general, a connection request can be represented by a three-tuple $(source, destination, b)$: a source node, a destination node, and a request size in terms of number of contiguous FSs. Heterogeneous traffic requests with different data rate requirements will consume different number of FSs, which cause the fragmentation issue.

The routing and spectrum assignment (RSA) problem in multi-fiber EONs is to find a path and a set of contiguous FSs on *some* fiber on links along the path for an arriving request.

Spectrum management such as partitioning the spectrum into dedicated ranges is motivated by the heterogeneous request sizes. Let us take a look at a simple example with request sizes of 2 and 3 slots. If there is no spectrum management based on

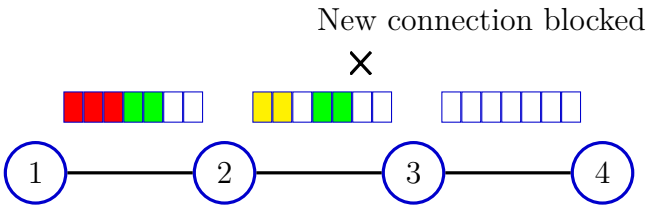

Figure 7.1: Slot usages without spectrum management.

Figure 7.2: Slot usages with spectrum management.

request classification and First Fit Spectrum/Slot Assignment algorithm is used, for the requests, the network status is as shown in Figure 7.1. When a new request of size 3 from node 2 to node 4 arrives, there are no available FSs and this request will be blocked. If we do spectrum management, such as dedicating the first 4 slots for requests of size 2 and the last 3 slots for requests of size 3, for the same set of requests, the slot assignments will change and the new arriving request can be accommodated. Figure 7.2 shows the new assignments when utilizing spectrum management.

7.4 Multi-Path Selection Scheme

A path between the source and destination nodes should be determined to accommodate the request. Commonly used routing schemes such as using a precomputed fixed single shortest path or one of many paths selected dynamically (based on the network state each time a request arrives) suffer from either poor performance (in the former case) or high computation complexity (in the latter case). Thus we would like to fully utilize multiple paths between each node pair efficiently. We intend to select a routing path based on some predetermined rules, instead of calculating the best path each time for a new arrival request.

Given the traffic loads, we propose to compute a set of candidate paths and the path selection probabilities for each node pair *offline*. This approach fully utilizes the multiple paths between each node pair while keeping computational complexity quite

Table 7.1: Notation for dynamic RSA problem

Symbol	Meaning
N	number of nodes in the network
L	number of links in the network
e	an arbitrary network link
F^e	the number of fibers on link e, $e \in \mathcal{E}$
R	number of routes in the networks
r	an arbitrary route
W_r	the load for route r; depends on the traffic pattern; $W_r = 1, \quad \forall r$ for uniform traffic
K_r	number of candidate paths for route r
k	an arbitrary candidate path
$A^e_{r,k}$	$= 1$ if link e is on the k^{th} candidate path of route r; $= 0$, otherwise

low. Given the network topology and traffic loads for each node pair, the selection probabilities for all candidate paths are computed offline via a Mixed Integer Linear Program to span loads over fibers. The result of this is that we obtain the probability distribution of the path to be selected for a request from s to d, $p^k_{(s,d)}$, the probability that the k^{th} path should be selected for node pair (s, d). Each time when a request arrives at the network, a path is selected to accommodate the request according to those predetermined probabilities.

7.4.1 Notations

The input parameters are shown in Table 7.1, including the detailed network topology information and the expected load for each route. Load for each route is provided based on the traffic pattern of the network. For example, if the source and destination of requests follow a uniform distribution, the load for each route can be set as 1.

7.4.2 Mixed Integer Linear Programming for Path Selection

We have derived Mixed Integer Linear Programming formulations to compute the optimal probabilities of candidate paths for each route (source-destination node pair). The objective is to minimize average and maximum traffic load over all fibers (as opposed to links, since different links may have different numbers of fibers) to balance the loads.

Objective: Minimize

$$\frac{1}{L}\sum_{e=1}^{L}\frac{\sum_{r=1}^{R}W_r y_r^e}{F^e} + \max_e \frac{\sum_{r=1}^{R}W_r y_r^e}{F^e}$$

Variables:

a) The probability of selecting the k^th candidate path of route r:

$$0 \leq p_r^k \leq 1.$$

b) The percentage of load for route r that traverses link e:

$$0 \leq y_r^e \leq 1.$$

Constraints:

a) The total probability of selecting candidate paths for a route should be 1.0. For all r,

$$\sum_{k=1}^{K_r} p_r^k = 1,$$

b) The percentage of load for a route r on a link depends on the candidate path selection probabilities. For all r, e,

$$y_r^e = \sum_{k=1}^{K_r} A_{r,k}^e p_r^k$$

7.4.3 Evaluation

Figure 7.3 shows the comparison between the performance (demand blocking ratio) of First-Fit slot assignment algorithm with a fixed single shortest path routing (SSP) and that of FF with our multi-path selection scheme (MPS) for 1 million dynamic

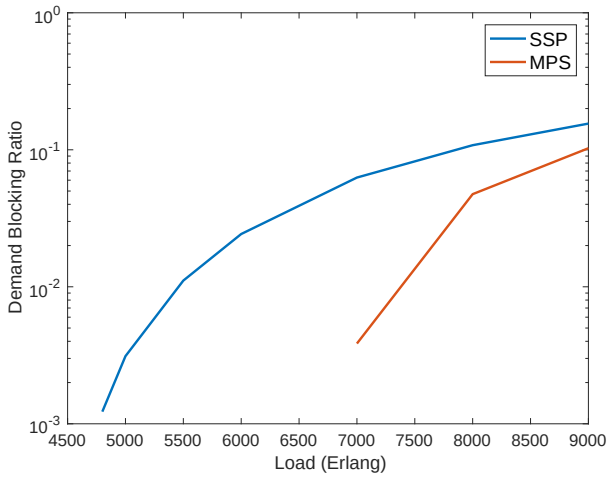

Figure 7.3: Effectiveness of the path selection scheme.

requests. The detailed simulation settings are described in Section 7.6. We can see that our routing scheme works much better than the fixed single shortest path scheme. Since the selection probabilities can be precomputed via the ILP model, there will be no increment in computation overhead.

7.5 Spectrum Assignment with Partitioning

We look at the spectrum assignment problem in this section. When a request arrives, a path is selected to accommodate the request by simply using the probability distribution computed above. After a path is chosen for the arriving request, SA is performed to assign contiguous FSs to that request. If there are no available slots on the selected path, the request will be blocked.

7.5.1 Spectrum Management

To eliminate the fragmentation caused by the mismatch of heterogeneous request bandwidths during the dynamic set-up and tear-down of traffic, a dedicated partition is utilized. The spectrum is partitioned into different segments, each dedicated to traffic requests with the same bandwidth.

We use the number of contiguous slots to represent the size of a request. Suppose there are M sets of connections with different request sizes b_j, $j = 1 \cdots M$. Then the whole spectrum can be partitioned into M segments. The number of contiguous

FSs dedicated to each segment is denoted by $P_j,\ j = 1 \cdots M$. Let S be the size of the whole spectrum, that is, the total number of FSs in a fiber. Then the sizes of segments should meet the constraint: $\sum_{j=1}^{M} P_j = S$. In the segment for request set j, we have P_j/b_j bins (a bin is a consecutive set of FSs, with bin size of b_j) in each fiber.

Assume that the traffic distribution is known, with ρ_j being the probability for connection requests of size b_j. The total probability of the traffic distribution should be 1.0, $\sum_{j=1}^{M} \rho_j = 1.0$. The distribution information is then used to calculate the sizes of different segments.

$$P_j = S \cdot \frac{\rho_j \cdot b_j}{\sum_{j=1}^{M} \rho_j \cdot b_j} \tag{7.1}$$

Through this partition, spectrum fairness can be achieved to some extent. For example, suppose there are 352 FSs on each fiber, there are 3 sets of requests with sizes $(b_1, b_2, b_3) = (3, 4, 7)$ and densities $(\rho_1, \rho_2, \rho_3) = (0.2, 0.5, 0.3)$. Then according to Equation 7.1, we get the segment sizes $(P_1, P_2, P_3) = (45, 152, 154)$, which correspond to 15, 38, and 22 bins for requests of size 3, 4, and 7, respectively, and 75 bins in total. The distribution of the number of bins for each request size are almost the same with the traffic distribution.

Figure 7.4 shows the spectrum efficiency improvement caused by dedicated spectrum partition. The baseline – FF without partitioning is compared with the result of dedicated partition (PD) with FF spectrum assignment. Both use the single shortest path routing (SSP). We can see that even without resource sharing among partitions, there is a steady improvement in spectrum efficiency. The spectrum underutilization for small workloads can be improved by resource sharing among different spectrum segments.

7.5.2 Next State Aware Spectrum Assignment

In this section, we propose a Next State Aware (NSA) Spectrum Assignment algorithm based on both the next state (i.e., the network state *after* the request is set up) and the path selection probabilities for requests, given the spectrum partitioning.

Since the whole spectrum has been partitioned into dedicated segments, each providing consecutive slots for a particular request size, the spectrum assignment

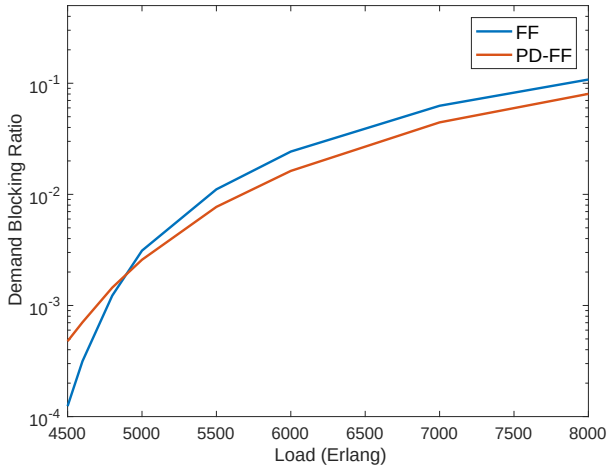

Figure 7.4: Effectiveness of spectrum partition.

problem can be addressed by scheduling a bin in the specific segment to a request. In the jth dedicated segment of the spectrum, we have P_j/b_j FS bins (a bin is a set of b_j consecutive FSs) in each fiber. Suppose the request has a bandwidth requirement of b_j FSs; the SA problem is to assign a bin of size b_j for the request. The spectrum assignment problem is then transformed to select an available bin in the dedicated segment to accommodate a request without disrupting existing connections so that the blocking probability is minimized.

For each new request, we only need to consider the network state corresponding to the specific spectrum segment which this request size belongs to, based on the partitioning in the previous section. The network state denotes the current status of the network, such as the slot usage in fibers. Since dedicated partitioning has been determined, we only need to look at the status of each bin in that segment. In order to reduce the blocking probability, we should select a bin that can provide good network state after the request is established, e.g. the path capacities are least reduced.

In the rest of this section, we assume X to be the number of bins in the segment the request belongs to. We first define the link capacity c_e^x of link e on bin x in one network state as the number of fibers on which x is unused on the link. Initially, in an empty network, $c_e^x = F^e, \forall x$. The path capacity is determined based on capacities of links along the path. For a path k, the path capacity on bin x is defined as the least link capacity on bin x along the path (which is also the link capacity of the most

congested link along the path),

$$C_k^x = \min_{e \in \Xi(k)} c_e^x. \tag{7.2}$$

where $\Xi(k)$ is the set of links in path k. Then the path capacity C_k is the total capacity across all bins,

$$C_k = \sum_{x=1}^{X} C_k^x. \tag{7.3}$$

Since for each route (source-destination node pair) r in the network, we've already determined the selection probabilities p_r^k for each path candidate k, the capacity for that route is then defined as

$$C_r = \sum_{k=1}^{K_r} p_r^k C_k. \tag{7.4}$$

Then the network state can be represented as the total capacity over all routes $\sum_{r=1}^{R} C_r$.

Suppose a request is assigned to a candidate path κ. After the request is established on a bin in the spectrum segment, there may be capacity loss for other routes in the network. We should choose a bin that causes the least capacity loss among all available bins to accommodate the request. In order to calculate the capacity loss, we utilize the definition of conflict graph $\mathcal{G}' = (\mathcal{V}', \mathcal{E}')$, where each vertex $k \in \mathcal{V}'$ represents a path k in the network, and an undirected edge $(k_1, k_2) \in \mathcal{E}'$ denotes that paths k_1 and k_2 shares at least one link. The capacity loss will occur only in the paths which contain common links with κ. So when calculating the total capacity loss, we only need to look at paths that are connected to κ in the conflict graph $\mathcal{G}'$. We use ψ^κ to denote this set of paths.

Let's consider the total capacity loss ζ_x when bin x is chosen for path κ. If at least one of the common links between κ and a path $k' \in \psi^\kappa$ have the minimum capacity on bin x along path k', $C_{k'}^x$ will be decreased by 1 after the establishment of the request. Otherwise, when all common links between κ and k' have capacity larger than the current path capacity $C_{k'}^x$, the link capacity decrement of the common links caused by the request establishment will not affect $C_{k'}^x$ and there will be no capacity loss for path k'. We use $\upsilon_{k'}$ to denote the capacity loss in path k'. Taking the path selection probability into account, if path k' belongs to route r and the probability of selecting

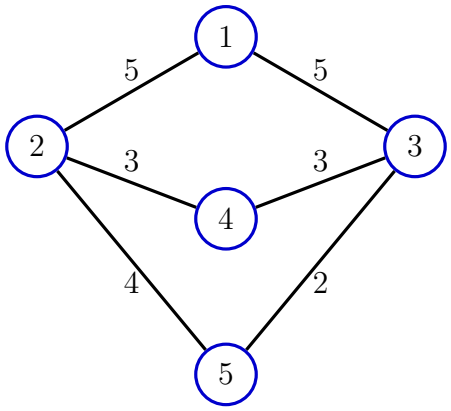

Figure 7.5: A small 5-node network topology.

k' for r is $p_r^{k'}$, then $\upsilon_{k'}$ equals to either $p_r^{k'}$ or 0, depending on the network state. The total capacity loss after bin x is chosen will be $\zeta_x = \sum_{k' \in \psi^\kappa} \upsilon_{k'}$. We should assign the bin $x*$ which causes least capacity loss by the request, e.g.,

$$x* = \arg \min_{x \in \Omega(\kappa)} \zeta_x \tag{7.5}$$

where $\Omega(\kappa)$ is the set of available bins in path κ for the request.

Let's take a small 5-node network as Figure 7.5 as an example. There are 5 physical nodes and 6 physical links in the network. The number of fibers are $5, 5, 3, 4, 3, 2$ on bidirectional links $(1,2), (1,3), (2,4), (2,5), (3,4), (3,5)$, respectively. Given the path candidates for each route, based on the multi-path selection formulations, we got the probabilities as in Table 7.2.

When a new request from source node 2 to destination node 5 arrives, from Table 7.2, path $2-5$ is selected. Then we should check the network state of the spectrum segment this request belongs to. Based on the conflict graph, we can see that only paths $1-2-5$, $4-2-5$, $2-5-3$ have common links with path $2-5$. So we only need to check the state of partial network. Assume there are 4 bins x_1, x_2, x_3, x_4 in the segment. Current network states are shown as Figure 7.6. The number attached to each link denotes current link capacity on a bin, which is the number of fibers the bin is unused on.

Let's take bin x_2 as an example to see how the total capacity loss is calculated. Since link $(2,5)$ has the least link capacity along both path $1-2-5$ and $4-2-5$, if x_2 is allocated to the new request, both path capacities will be decreased. According

Table 7.2: Path selection probabilies for the small network.

Route	**Candidate Paths**	**Selection Probability**
$1 \sim 2$	1-2	1.0
$1 \sim 3$	1-3	1.0
$1 \sim 4$	1-2-4	1/3
	1-3-4	2/3
$1 \sim 5$	1-2-5	1.0
	1-3-5	0.0
$2 \sim 3$	2-1-3	1.0
	2-4-3	0.0
	2-5-3	0.0
$2 \sim 4$	2-4	1.0
$2 \sim 5$	2-5	1.0
$3 \sim 4$	3-4	1.0
$3 \sim 5$	3-5	1.0
$4 \sim 5$	4-2-5	2/3
	4-3-5	1/3

to the selection probabilities of these two paths, capacity loss is 1.0 for path $1-2-5$, and $\frac{2}{3}$ for path $4-2-5$. For path $2-5-3$, the selection probability is 0, so we don't need to take this path capacity into account. In total, the capacity loss is $\frac{5}{3}$ if bin x_2 is used. Similarly, capacity losses for x_1, x_3, x_4 are $1.0, 0, \frac{2}{3}$, respectively. Bin x_3 will be allocated to accommodate this request.

To evaluate the efficiency of our NSA algorithm, we run simulations for traffic requests with a single size. The performance of NSA is compared with that of baseline – First Fit spectrum assignment. Both use the single shortest path routing (SSP). The first set of comparison is on a 20-node unidirectional ring with 10 fibers per link. Figure 7.7 shows a huge spectrum efficiency improvement by utilizing NSA in the unidirectional ring network. The second set of comparison is on the 14-node NSF network, with different number of fibers on each link. In Figure 7.8, the

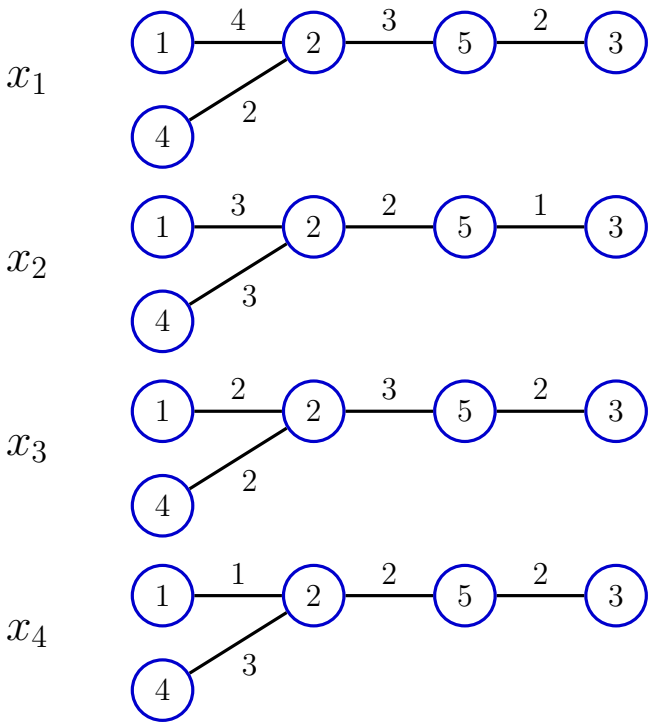

Figure 7.6: Network state example – link capacities of partial network on bin x_1, x_2, x_3, x_4.

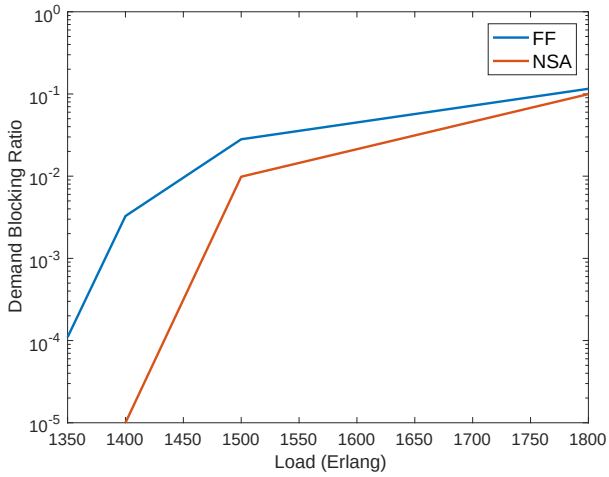

Figure 7.7: Effectiveness of NSA algorithm in a 20-node unidirectional ring network. spectrum efficiency improvement in the mesh network is not as obvious as that in the unidirectional ring network. It indicates that when there are more paths overlapping (more common links), our algorithm works better. But in Figure 7.8, we can still see $9\% - 31\%$ performance improvement for the mesh network.

Figure 7.9 shows the spectrum efficiency improvement caused by both dedicated

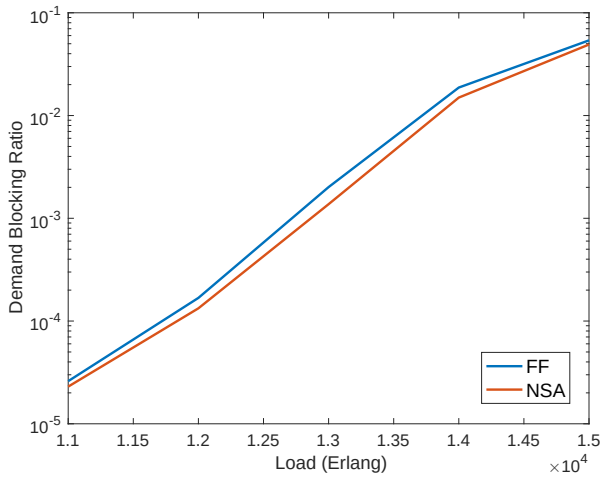

Figure 7.8: Effectiveness of NSA algorithm in a 14-node NSF network.

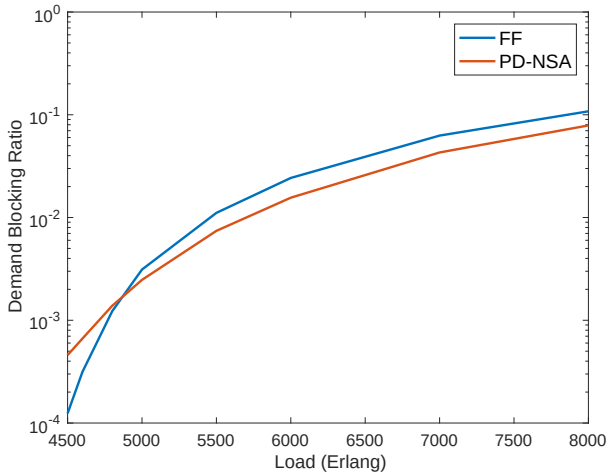

Figure 7.9: Effectiveness of spectrum partition and NSA algorithm.

spectrum partition and our proposed NSA algorithm. The baseline – FF is compared with the result of dedicated partition (PD) with NSA spectrum assignment. Both use single shortest path routing (SSP). We can see that even without resource sharing among different partitions, there is a steady improvement in spectrum efficiency. The underutilization of spectrum in the small load can be eliminated by resource sharing among spectrum segments.

7.5.3 Resource sharing among partitions

According to the observation of previous results, there are spectrum underutilizations for low loads caused by the dedicated partitioning. So we consider to further

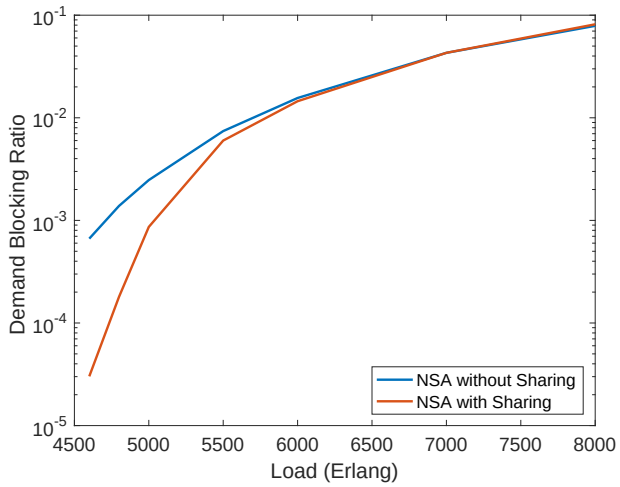

Figure 7.10: Effectiveness of resource sharing among partitions.

improve spectrum efficiency by resource sharing among partitions. For each request, we first check whether there is any available bin in the dedicated spectrum segment this request belongs to. If no available bin exists in this segment, FSs in other partitions will be examined. In addition to the bin states in each segment, the occupancy states for each FS also need to be recorded, which are used to determine the candidate FS sets. We use the same assignment scheme as the previous section, with slight modifications in the calculation of capacity loss. Take a candidate FS set χ as an example, it may be across multiple bins, and only partial of the first or last bins are in χ. For the bins where all FSs are within χ, the capacity loss calculation is the same as that in the previous section. For the first or last bin, if it has already been marked as occupied (which means some slots in the bin while outside χ are in use) because of preceding requests, the link capacity loss for this bin is set to 0. The total capacity loss will be the sum of capacity loss for each bin in this candidate FS set. Still, the FS set which causes minimum capacity loss will be assigned to the request. If χ is the selection, all bins χ crosses will be marked as occupied. Figure 7.10 shows the spectral efficiency improvement by resource sharing among partitions. We can see a huge improvement in performance for low traffic loads in the network.

7.6 Numerical Results

We present performance evaluation results for two real network topologies, the NSF network (Figure 2.3) and the pan-European network (Figure 3.7). Each link has a random number of fibers which are uniformly distributed between 5 and 10 fibers. For the elastic optical network, we assume the fiber capacity of 352 frequency slots, with each slot having a bandwidth of 12.5 GHz.

The traffic demand is a set of dynamically arriving connection requests. Each request represents a connection between a pair of nodes in the network. Connection requests arrive to the network according to a Poisson process. Each request has a mean holding time of 1 (arbitrary time unit), and the arrival rate of traffic requests is varied in order to examine the network performance under varying offered loads. Frequency conversion is not considered in this work. There are three types of demands with 40/100/400 Gbps, requesting 3, 4, 7 frequency slots, respectively [46]. The distribution is $(\rho_1, \rho_2, \rho_3) = (0.2, 0.5, 0.3)$. Based on this information, the spectrum partitioning can be determined. We use the demand blocking ratio of dynamic traffic requests to indicate the performance in multi-fiber elastic optical networks. For each simulation, the results of 1 million dynamic requests excluding 10000 warm-up requests are recorded.

We use "R" to denote random slot set assignment, "FF" as first fit SA, "FLF" as first last fit SA, "MK" as the scheme proposed in the literature [120] and "NSA" as our network state aware algorithm. In the FLF SA scheme [45], the whole spectrum is divided into several partitions, requests attempt to use the lowest indexed FSs in the odd number partitions and highest indexed FSs in the even number partitions to create chance for more contiguous FSs. We adapt MK to our multi-fiber link model. In MK, the same partitioning with NSA and first fit slot assignment in each partition is used, while the resource sharing in MK is different, each partition is shared by only higher bandwidth requests.

We first conduct a comparison among several spectrum assignment schemes with the fixed single shortest path routing. Figure 7.11 shows the results for the uniform

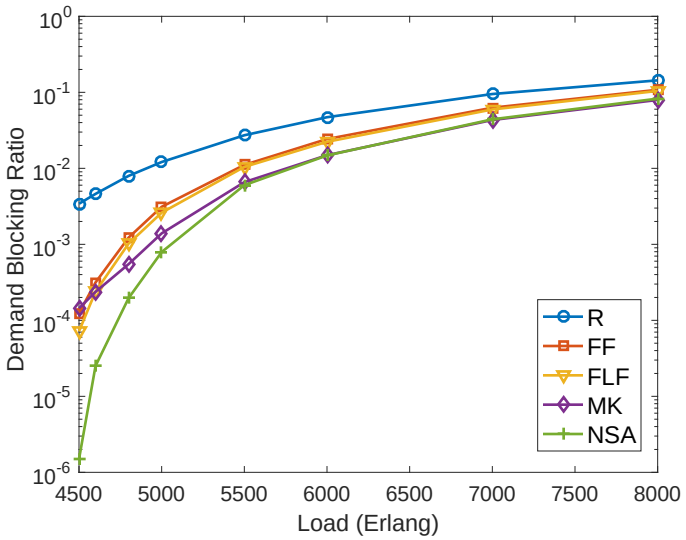

Figure 7.11: Comparison of spectrum allocation schemes in 14-Node NSF network with uniform traffic pattern.

traffic pattern in NSF network. The source and destination nodes for each connection request are uniformly randomly selected. As expected, in schemes without dedicated partitioning, R always performs worst and FLF performs better than FF by giving more contiguous aligned FSs. In low load cases, MK has higher demand blocking probability than FF and FLF due to its limited resource sharing and therefore resource underutilization. Then the effect of spectrum management (decreased fragmentation) dominates in higher load cases, which leads to better performance of MK than FF and FLF. Our scheme takes all these aspects into account, and performs spectrum allocation based on the global network states, therefore it shows best performances.

Then we conduct two sets of simulations to evaluate the joint effect of the multi-path selection and the spectrum assignment schemes. According to the traffic pattern, a set of path selection probabilities is precomputed. We apply this multi-path selection jointly with all spectrum assignment schemes in our evaluation.

We first show a set of simulation results for the uniform traffic pattern. Figure 7.12 and 7.13 show the performance comparison among our scheme and other baselines jointly with multi-path selection for both NSF and Pan-European networks. With the joint scheme, our algorithm still performs best and the same conclusion can be drawn. By comparing results in Figure 7.11 and 7.12, we can see that by jointly applying

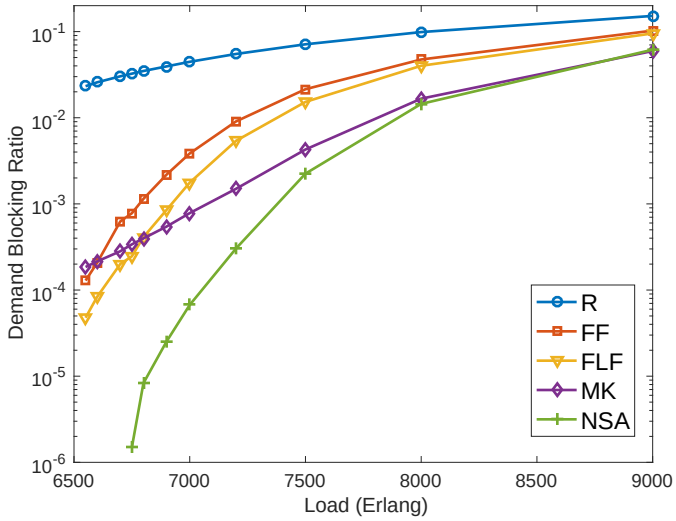

Figure 7.12: Joint comparison in 14-Node NSF network with uniform traffic pattern.

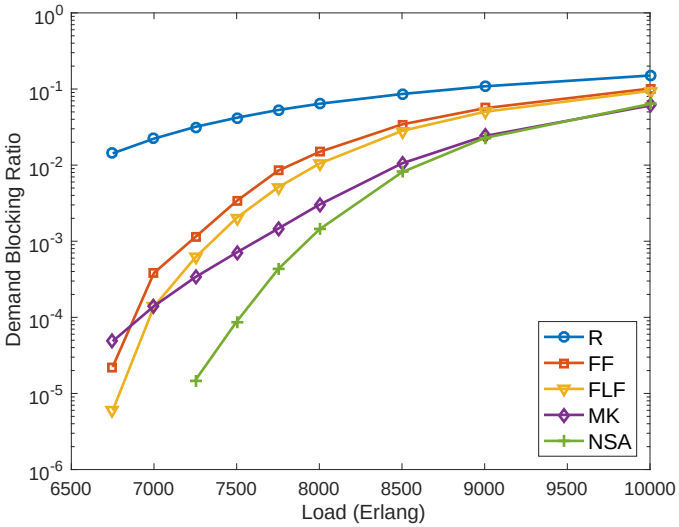

Figure 7.13: Joint comparison in 28-Node Pan-European network with uniform traffic pattern.

multi-path selection with each spectrum assignment scheme, the demand blocking ratio greatly improves and more load can be accommodated in the network.

The other set of simulations is conducted for a non-uniform traffic pattern, where the probability that each node is selected as source or destination $u_v, v \in \mathcal{V}$ can be different from each other. $\sum_{v \in \mathcal{V}} u_v = 1.0$. We assume nodes with higher connectivities have larger chance to send and receive traffic, and the probabilities are in proportional to the node degrees. The comparison results for both NSF and Pan-European network

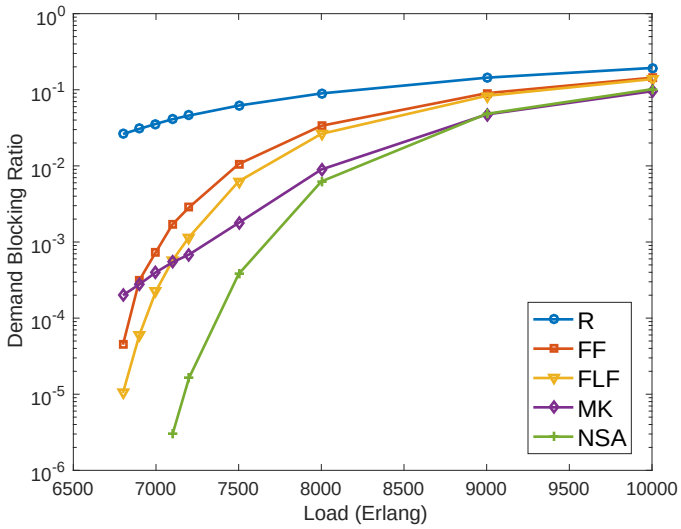

Figure 7.14: Joint comparison in 14-Node NSF network with non-uniform traffic pattern.

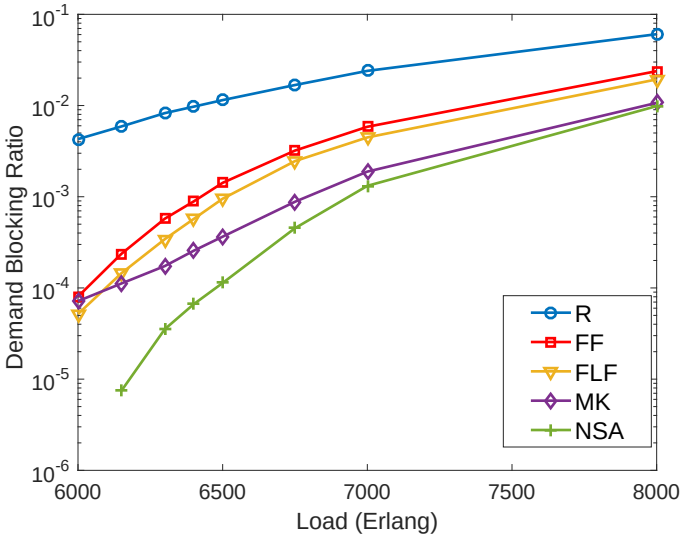

Figure 7.15: Joint comparison in 28-Node Pan-European network with non-uniform traffic pattern.

are shown in Figure 7.14 and 7.15 respectively. We can see similar trends to the previous results. This further validates the effectiveness of our algorithm.

We also conduct sensitivity tests to see how well our scheme performs when there is variation in the traffic pattern and demand size distribution. Figure 7.16 shows the test results regarding to demand size distribution in the NSF network with uniform traffic pattern. The distribution of three demand types is $(\rho_1, \rho_2, \rho_3) = (0.2, 0.5, 0.3)$

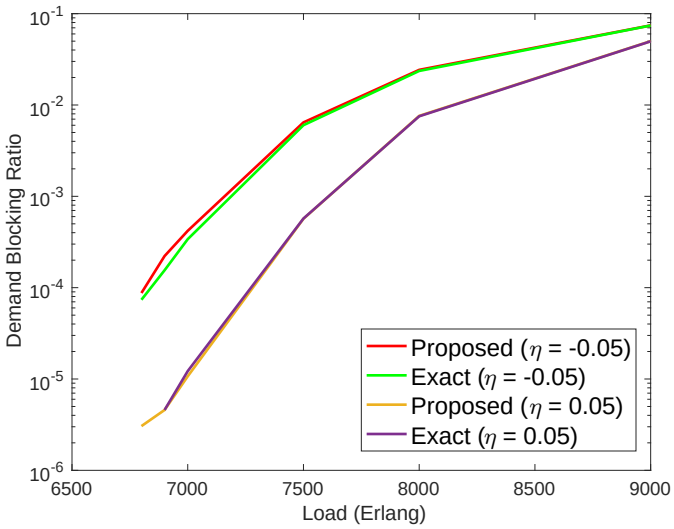

Figure 7.16: Sensitivity test for demand size distribution in NSF network.

originally, and spectrum partitioning is calculated based on this distribution. If there is a small change η in the distribution $(\rho_1 + \eta, \rho_2 + \eta, \rho_3 - 2\eta)$, we compare the performance of applying the original spectrum partition to the changed traffic and that of applying adjusted partition calculated based on exact distribution. The results are represented by "Exact" (with adjustment according to exact information) and "Proposed" (without exact information). The two sets ($\eta = -0.05$ and $\eta = 0.05$) both show little performance degradation.

Figure 7.17 is the test result regarding to traffic pattern in the NSF network. The demand size distribution does not change. The original probabilities that a node is selected as source or destination are in proportional to the node degrees and the multi-path selection has been calculated according to this non-uniform traffic pattern. If there is a small change η in the probabilities $(u_1 + \eta, u_2 + \eta, \cdots, u_N - (N-1)\eta)$, we compare the performance of applying the original path selection to the changed traffic and that of applying adjusted path selection calculated based on exact traffic pattern. The two sets ($\eta = -0.01$ and $\eta = 0.01$) both show little performance variation.

7.7 Conclusions

We study the Routing and Spectrum Assignment problem for dynamic heterogeneous traffic requests in elastic optical networks with multiple fibers per link [129].

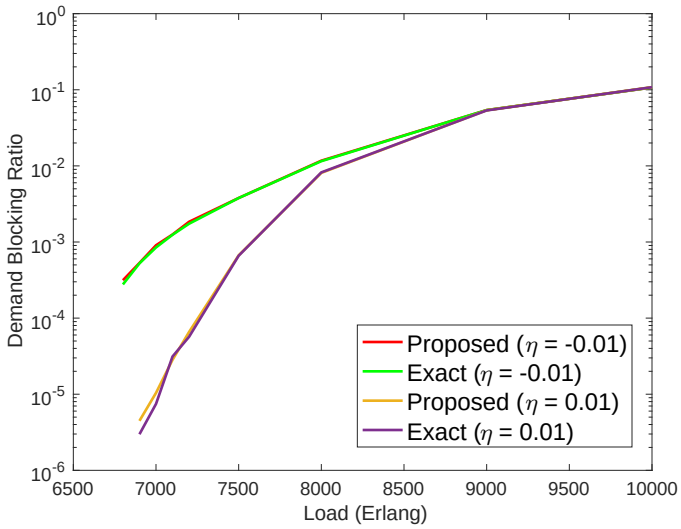

Figure 7.17: Sensitivity test for traffic pattern in NSF network.

The routing problem is solved by a multi-path selection scheme, which is formulated mathematically with parameters given by the topology information and traffic pattern. Once the selection probabilities of each candidate path between a source and a destination are determined, any arriving traffic request for that source-destination pair will choose one path according to the probabilities. Then the slot usage along the path is examined to make the spectrum assignment. To avoid the fragmentation caused by the heterogeneity of demand sizes, we partition the whole spectrum into segments dedicated to different demand sizes. A spectrum assignment algorithm based on the network state within a specific partition is proposed first, and further improved by considering the sharing among all partitions. We validate the effectiveness of each scheme for subproblems. Multi-path selection, appropriate spectrum partitioning, and network state aware spectrum assignment all can improve the spectrum efficiency. The results of combining these schemes show much better performance than the baselines.

Chapter 8 Conclusions and Future Directions

8.1 Conclusions

This dissertation investigated several problems of resource allocation in multi-granular optical networks. The research objective was to increase the resource utilization efficiency under various constraints.

In Chapter 2, we investigated the co-scheduling problem in elastic optical networks for both static and dynamic multi-task jobs. We proposed an Integer Linear Programming model and two heuristics to jointly allocate computational and networking resources to jobs. The simulation results show that our proposed Children-Aware algorithm outperformed the baseline algorithm in terms of makespan for static jobs and blocking probability for dynamic jobs.

In Chapters 3 and 4, we studied resource allocation problems in multi-granular WDM networks. A non-uniform wavebanding framework from the entire network's view was proposed to minimize the total waveband required while satisfying a set of demands so as to reduce hardware requirements. The simulation results suggested that wavebanding is an effective strategy to reduce switching elements. Then we investigated the OXC architectures for WDM networks with multiple fibers per link to accommodate increasing traffic demands. Two architectures were evaluated and compared in terms of hardware requirements, power consumption and costs. Heuristics for resource (i.e., fiber and wavelength) assignment were presented and analytical models were developed to predict the blocking performance.

In Chapter 5 and 6, we studied the routing, fiber, band and spectrum allocation problem in multi-granular EONs with OXC nodes which can accommodate non-contiguous and non-uniform wavebands and multi-fiber links. An auxiliary layered-graph framework with pluggable cost functions was developed to solve the joint RFBSA problem. Cost functions to minimize the maximum spectral usage were proposed for a set of traffic requests. By comparison with other baseline heuristics, our RFBSA framework could achieve less spectral usage. The RFBSA framework was

further improved in Chapter 6 to reduce time complexity. To achieve good network performance, while saving considerable hardware costs, a joint wavebanding node placement and RFBSA problem given the network budget (in terms of total number of available WSSs) was investigated. Heuristics based on the above framework was proposed to solve the problem.

In Chapter 7, we proposed schemes for dynamic Routing and Spectrum Assignment problem in elastic optical networks. We investigate both path selection and spectrum management for heterogeneous bandwidth requests in multi-fiber elastic optical networks (EONs) to improve the spectrum efficiency. An ILP model based on topology and traffic pattern information was proposed to precomputed the selection probabilities of candidate paths for each source-destination node pair. A dedicated spectrum partition scheme and a spectrum assignment algorithm based on network states with resource sharing among partitions were proposed. The simulation results showed each scheme performs well in improving the spectrum efficiency. The joint allocation scheme significantly outperformed other heuristics in the literature.

8.2 Future Directions

Space Division Multiplexing (SDM) is expected to be the next frontier in optical communication by increasing capacity through multiplicity of space channels, such as utilizing multi-core fiber (MCF) or multi-mode fiber (MMF). One direction of my future research is to extend my previous work to resource allocation problem in space division multiplexing elastic optical networks by taking the physical impairment and crosstalk into account. The other problem is the survivable virtual infrastructure embedding over software-defined SDM optical networks. The problem is more complicated than the general virtual network embedding problem given the additional constraints related to the underlying optical networks.

Bibliography

[1] T. Ban, H. Hasegawa, K.-i. Sato, T. Watanabe, and H. Takahashi, "A novel large-scale OXC architecture and an experimental system that utilizes wavelength path switching and fiber selection," *Optics express*, vol. 21, no. 1, pp. 469–477, 2013.

[2] H. Hasegawa, S. Subramaniam, and K.-i. Sato, "Flexible waveband routing optical networks," in *IEEE International Conference on Communications*, 2015.

[3] Available online: "http://www.cisco.com", 2016.

[4] Y. Tanaka, H. Hasegawa, and K.-i. Sato, "Performance evaluations of large-scale OXC that achieves modular and hitless expansion," in *Optical Fiber Communication Conference*, 2014, pp. W2A–50.

[5] I. B. Djordjevic and B. Vasic, "Orthogonal frequency division multiplexing for high-speed optical transmission," *Optics Express*, vol. 14, no. 9, pp. 3767–3775, 2006.

[6] W. Wei et al, "PONIARD: A programmable optical networking infrastructure for advanced research and development of future internet," vol. 27, no. 3, pp. 233 – 242, Feb. 2009.

[7] M. Jinno et al., "Distance-adaptive spectrum resource allocation in spectrum-sliced elastic optical path network," vol. 48, no. 8, pp. 138 – 145, Aug. 2010.

[8] R. Izmailov, S. Ganguly, T. Wang, Y. Suemura, Y. Maeno, and S. Araki, "Hybrid hierarchical optical networks," *Communications Magazine, IEEE*, vol. 40, no. 11, pp. 88–94, 2002.

[9] Y. Wang and X. Cao, "Multi-granular optical switching: A classified overview for the past and future," *Communications Surveys & Tutorials, IEEE*, vol. 14, no. 3, pp. 698–713, 2012.

[10] S. Chandrasekhar, C. Doerr, and L. Buhl, "Flexible waveband optical networking without guard bands using novel 8-skip-0 banding filters," *Photonics Technology Letters, IEEE*, vol. 17, no. 3, pp. 579–581, 2005.

[11] S. Kakehashi, H. Hasegawa, K. Sato, O. Moriwaki, S. Kamei, Y. Jinnouchi, and M. Okuno, "Performance of waveband MUX/DEMUX using concatenated AWGs," *IEEE PHOTONICS TECHNOLOGY LETTERS*, vol. 19, no. 16, p. 1197, 2007.

[12] K. Ishii, H. Hasegawa, K.-I. Sato, M. Okuno, S. Kamei, and H. Takahashi, "An ultra-compact waveband cross-connect switch module to create cost-effective multi-degree reconfigurable optical node," in *Optical Communication, 2009. ECOC'09. 35th European Conference on*, 2009, pp. 1–2.

[13] S. Yuan, N. Madamopoulos, R. Helkey, V. Kaman, J. Klingshirn, and J. Bowers, "Fully integrated NxN MEMS wavelength selective switch with 100% colorless add-drop ports," in *Optical Fiber Communication Conference*, 2008.

[14] A. N. Patel, P. N. Ji, J. P. Jue, and T. Wang, "Hierarchical multi-granular switching in flexible grid WDM networks," in *Optical Fiber Communication Conference*. Optical Society of America, 2012, pp. OTh3B–6.

[15] S. Alamro, M. Xu, T. Lan, and S. Subramaniam, "Cred: Cloud right-sizing to meet execution deadlines and data locality," in *Cloud Computing (CLOUD), 2016 IEEE 9th International Conference on*. IEEE, 2016, pp. 686–693.

[16] N. Limrungsi, J. Zhao, Y. Xiang, T. Lan, H. H. Huang, and S. Subramaniam, "Providing reliability as an elastic service in cloud computing," in *Communications (ICC), 2012 IEEE International Conference on*. IEEE, 2012, pp. 2912–2917.

[17] M. Xu, S. Alamro, T. Lan, and S. Subramaniam, "Optimizing speculative execution of deadline-sensitive jobs in cloud," in *Proceedings of the 2017 ACM*

SIGMETRICS/International Conference on Measurement and Modeling of Computer Systems. ACM, 2017, pp. 17–18.

[18] ——, "Laser: A deep learning approach for speculative execution and replication of deadline-critical jobs in cloud," in *Computer Communication and Networks (ICCCN), 2017 26th International Conference on*. IEEE, 2017, pp. 1–8.

[19] ——, "Cred: Cloud right-sizing with execution deadlines and data locality," *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 12, pp. 3389–3400, 2017.

[20] V. Aggarwal, M. Xu, T. Lan, and S. Subramaniam, "On the optimality of scheduling dependent mapreduce tasks on heterogeneous machines," *arXiv preprint arXiv:1711.09964*, 2017.

[21] J. Zhao, Y. Xiang, T. Lan, H. H. Huang, and S. Subramaniam, "Elastic reliability optimization through peer-to-peer checkpointing in cloud computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 2, pp. 491–502, 2017.

[22] S. Alamro, M. Xu, T. Lan, and S. Subramaniam, "Shed: Optimal dynamic cloning to meet application deadlines in cloud," in *Communications (ICC), 2018 IEEE International Conference on*. IEEE, 2018, pp. 1–6.

[23] M. Xu, S. Alamro, T. Lan, and S. Subramaniam, "Chronos: A unifying optimization framework for speculative execution of deadline-critical mapreduce jobs," in *Distributed Computing Systems (ICDCS), 2018 IEEE 38th International Conference on*. IEEE, 2018.

[24] N. Lu, Y. Wang, F. Yang, and M. Xu, "A novel approach for single-packet ip traceback based on routing path," in *Parallel, Distributed and Network-Based Processing (PDP), 2012 20th Euromicro International Conference on*. IEEE, 2012, pp. 253–260.

[25] F. Yao, J. Wu, G. Venkataramani, and S. Subramaniam, "A comparative analysis of data center network architectures," in *IEEE International Conference on Communications (ICC)*. IEEE, 2014, pp. 3106–3111.

[26] Y. Li, "Pushing the envelope of mobile computing: Improving security, energy, and latency by bridging the gap between analytical modeling and system design," Ph.D. dissertation, The George Washington University, 2018.

[27] F. Yao, Y. Li, Y. Chen, H. Xue, T. Lan, and G. Venkataramani, "Statsym: vulnerable path discovery through statistics-guided symbolic execution," in *Dependable Systems and Networks (DSN), 2017 47th Annual IEEE/IFIP International Conference on*. IEEE, 2017, pp. 109–120.

[28] H. Xue, Y. Chen, F. Yao, Y. Li, T. Lan, and G. Venkataramani, "Simber: Eliminating redundant memory bound checks via statistical inference," in *IFIP International Conference on ICT Systems Security and Privacy Protection*. Springer, 2017, pp. 413–426.

[29] Y. Wang, Y. Li, and T. Lan, "Capitalizing on the promise of ad prefetching in real-world mobile systems," in *2017 IEEE 14th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*. IEEE, 2017, pp. 162–170.

[30] Y. Li, F. Yao, T. Lan, and G. Venkataramani, "Poster: Semantics-aware rule recommendation and enforcement for event paths," in *International Conference on Security and Privacy in Communication Systems*. Springer, 2015, pp. 572–576.

[31] C. Liu, M. Xu, and S. Subramaniam, "A reconfigurable high-performance optical data center architecture," in *Global Communications Conference (GLOBECOM), 2016 IEEE*. IEEE, 2016, pp. 1–6.

[32] Y. Li and T. Lan, "Multichoice games for optimizing task assignment in edge computing," in *2018 IEEE Global Communications Conference*. IEEE, 2018.

[33] F. Yao, J. Wu, G. Venkataramani, and S. Subramaniam, "A dual delay timer strategy for optimizing server farm energy," in *IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*. IEEE, 2015, pp. 258–265.

[34] Y. Li, Y. Chen, T. Lan, and G. Venkataramani, "Mobiqor: Pushing the envelope of mobile edge computing via quality-of-result optimization," in *Distributed Computing Systems (ICDCS), 2017 IEEE 37th International Conference on*. IEEE, 2017, pp. 1261–1270.

[35] F. Yao, J. Wu, S. Subramaniam, and G. Venkataramani, "WASP: Workload Adaptive Energy-Latency Optimization in Server Farms using Server Low-Power States," in *IEEE International Conference on Cloud Computing (CLOUD)*. IEEE, 2017.

[36] Y. Li, F. Yao, T. Lan, and G. Venkataramani, "Sarre: semantics-aware rule recommendation and enforcement for event paths on android," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 12, pp. 2748–2762, 2016.

[37] F. Yao, J. Wu, G. Venkataramani, and S. Subramaniam, "Ts-bat: Leveraging temporal-spatial batching for data center energy optimization," in *IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2017, pp. 1–6.

[38] B. Lu, S. S. Dayapule, F. Yao, J. Wu, G. Venkataramani, and S. Subramaniam, "Popcorns: Power optimization using a cooperative network-server approach for data centers (invited paper)," in *IEEE International Conference on Computer Communication and Networks (ICCCN)*. IEEE, 2018.

[39] S. Selvakumar and C. S. R. Murthy, "Communication contention in task scheduling," *IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS*, vol. 5, no. 3, pp. 328–336, 1994.

[40] A. Riabov and J. Sethuraman, "Scheduling periodic task graphs with communication delays," in *ACM SIGMETRICS*, 2001.

[41] O. Sinnen and L. A. Sousa, "Communication contention in task scheduling," *IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS*, vol. 16, no. 6, pp. 503–515, 2005.

[42] N. Vydyanathan, S. Krishnamoorthy, G. M. Sabin, U. V. Catalyurek, T. Kurc, P. Sadayappan, and J. H. Saltz, "An integrated approach to locality-conscious processor allocation and scheduling of mixed-parallel applications," *IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS*, vol. 20, no. 8, pp. 1158–1172, 2009.

[43] P. Choudhury, P. Chakrabarti, and R. Kuma, "Online scheduling of dynamic task graphs with communication and contention for multiprocessors," *IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS*, vol. 23, no. 1, pp. 126–133, 2012.

[44] Y. Wang, X. Cao, and Y. Pan, "A study of the routing and spectrum allocation in spectrum-sliced elastic optical path networks," in *INFOCOM, 2011 Proceedings IEEE*, 2011, pp. 1503–1511.

[45] B. C. Chatterjee, N. Sarma, and E. Oki, "Routing and spectrum allocation in elastic optical networks: a tutorial," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 3, pp. 1776–1800, 2015.

[46] J. Wu, S. Subramaniam, and H. Hasegawa, "Comparison of oxc node architectures for wdm and flex-grid optical networks," in *Computer Communication and Networks (ICCCN), 2015 24th International Conference on*, 2015, pp. 1–8.

[47] J. Wu, M. Xu, S. Subramaniam, and H. Hasegawa, "Routing, fiber, band, and spectrum assignment (RFBSA) for multi-granular elastic optical networks," in *Proceedings of the IEEE International Conference on Communications (ICC)*, 2017, pp. 1–6.

[48] J. Hoogeveena, S. van de Veldeb, and B. Veltman, "Complexity of scheduling

multiprocessor tasks with prespecified processor allocations," *Discrete Applied Mathematics*, vol. 55, no. 3, pp. 259–272, Dec. 1994.

[49] J. Wu, J. Zhao, and S. Subramaniam, "Co-scheduling computational and networking resources in elastic optical networks," in *Proceedings of the IEEE International Conference on Communications (ICC)*, 2014, pp. 3307–3312.

[50] J. Ding, Y. Wang, J. Le, and Y. Jin, "Dynamic scheduling for workflow applications over virtualized optical networks," in *IEEE Computer Communications Workshops (INFOCOM WKSHPS)*, July 2011.

[51] M. Zhu, W. Guo, S. Xiao, A. Wei, Y. Jin, W. Hu, and B. Geller, "Availability-driven scheduling for real-time directed acyclic graph applications in optical grids," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 2, no. 7, pp. 469–480, 2010.

[52] M. Abouelela and M. El-Darieby, "Co-scheduling computational and networking resources in E-science optical grids," in *Proceedings of the Global Communications Conference (GLOBECOM)*, Dec. 2010.

[53] C. E. Abosi, G. S. Zervas, R. Nejabati, and D. Simeonidou, "Energy-aware service plane co-scheduling of a novel integrated optical network-IT infrastructure," in *Optical Network Design and Modeling (ONDM)*, Feb. 2011.

[54] H. Kulkarni, A. Gadkar, and V. M. Vokkarane, "Deadline-aware co-scheduling using anycast advance reservations in wavelength routed lambda grids," in *International Conference on Computing, Networking and Communications (ICNC) Workshop on Computing, Networking and Communications*, Jan. 2013.

[55] Y. Xiang, B. Balasubramanian, M. Wang, T. Lan, S. Sen, and M. Chiang, "Self-adaptive, deadline-aware resource control in cloud computing," in *Self-Adaptation and Self-Organizing Systems Workshops (SASOW), 2013 IEEE 7th International Conference on*. IEEE, 2013, pp. 41–46.

[56] Y. Xiang, T. Lan, V. Aggarwal, Y.-F. R. Chen, Y. Xiang, T. Lan, V. Aggarwal, and Y.-F. R. Chen, "Joint latency and cost optimization for erasure-coded data center storage," *IEEE/ACM Transactions on Networking (TON)*, vol. 24, no. 4, pp. 2443–2457, 2016.

[57] V. Aggarwal, Y.-F. R. Chen, T. Lan, and Y. Xiang, "Sprout: A functional caching approach to minimize service latency in erasure-coded storage," *IEEE/ACM Transactions on Networking*, vol. 25, no. 6, pp. 3683–3694, 2017.

[58] Y. Xiang, T. Lan, V. Aggarwal, and Y.-F. R. Chen, "Multi-tenant latency optimization in erasure-coded storage with differentiated services," in *2015 IEEE 35th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2015, pp. 790–791.

[59] Y. Xiang, V. Aggarwal, Y.-F. Chen, and T. Lan, "Differentiated latency in data center networks with erasure coded files through traffic engineering," *IEEE Transactions on Cloud Computing*, 2017.

[60] Y. Xiang, T. Lan, V. Aggarwal, and Y.-F. Chen, "Optimizing differentiated latency in multi-tenant, erasure-coded storage," *IEEE Transactions on Network and Service Management*, vol. 14, no. 1, pp. 204–216, 2017.

[61] Y. Xiang, V. Aggarwal, Y.-F. R. Chen, and T. Lan, "Taming latency in data center networking with erasure coded files," in *Cluster, Cloud and Grid Computing (CCGrid), 2015 15th IEEE/ACM International Symposium on*. IEEE, 2015, pp. 241–250.

[62] Y. Xiang, H. Liu, T. Lan, H. Huang, and S. Subramaniam, "Optimizing job reliability through contention-free, distributed checkpoint scheduling," Online technical report available at www. seas. gwu. edu/tlan/papers/ICAC. pdf, Tech. Rep., 2013.

[63] ——, "Optimizing job reliability via contention-free, distributed scheduling of

vm checkpointing," in *Proceedings of the 2014 ACM SIGCOMM workshop on Distributed cloud computing*. ACM, 2014, pp. 59–64.

[64] ITU-T C1284, "Proposal of subjects to be discussed regarding flexible grids," 2011.

[65] M. Yu et al., "Rethinking virtual network embedding: substrate support for path splitting and migration," in *ACM SIGCOMM*, April 2008.

[66] R. Izmailov, S. Ganguly, V. Kleptsyn, and A. C. Varsou, "Nonuniform waveband hierarchy in hybrid optical networks," in *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, vol. 2, 2003, pp. 1344–1354.

[67] P. Torab, V. Hutcheon, D. Walters, and A. Battou, "Waveband switching efficiency in WDM networks: Analysis and case study," in *Optical Fiber Communication Conference*, 2006, p. OTuG3.

[68] M. Alnaimi, O. Turkcu, and S. Subramaniam, "Uniform waveband switching in WDM mesh networks," in *Telecommunications (ICT), 2010 IEEE 17th International Conference on*. IEEE, 2010, pp. 509–515.

[69] L.-W. Chen, P. Saengudomlert, and E. Modiano, "Uniform versus non-uniform band switching in WDM networks," *Computer Networks*, vol. 50, no. 2, pp. 149–167, 2006.

[70] O. Turkcu and S. Subramaniam, "Optimal wavebanding in WDM ring networks," *IEEE/ACM Transactions on Networking (TON)*, vol. 22, no. 1, pp. 179–190, 2014.

[71] C. Berge and E. Minieka, *Graphs and hypergraphs*. North-Holland publishing company Amsterdam, 1973, vol. 7.

[72] M. Libura, "Sensitivity analysis for minimum hamiltonian path and traveling salesman problems," *Discrete Applied Mathematics*, vol. 30, no. 2, pp. 197–211, 1991.

[73] F. Yao, J. Chen, and G. Venkataramani, "Jop-alarm: Detecting jump-oriented programming-based anomalies in applications," in *2013 IEEE 31st International Conference on Computer Design (ICCD)*. IEEE, 2013, pp. 467–470.

[74] A. Moon, J. Kim, J. Zhang, H. Liu, and S. W. Son, "Understanding the impact of lossy compressions on iot smart farm analytics," in *Big Data (Big Data), 2017 IEEE International Conference on*. IEEE, 2017, pp. 4602–4611.

[75] F. Yao, M. Doroslovački, and G. Venkataramani, "Covert timing channels exploiting cache coherence hardware: Characterization and defense," *International Journal of Parallel Programming*, 2018.

[76] H. Liu and H. H. Huang, "Graphene: Fine-grained io management for graph computing," in *15th USENIX Conference on File and Storage Technologies (FAST 17)*. USENIX Association, pp. 285–300.

[77] N. Xia, C. Tian, Y. Luo, H. Liu, and X. Wang, "Uksm: swift memory deduplication via hierarchical and adaptive memory region distilling," in *Proceedings of the 16th USENIX Conference on File and Storage Technologies*. USENIX Association, 2018, pp. 325–339.

[78] F. Yao, M. Doroslovački, and G. Venkataramani, "Are Coherence Protocol States Vulnerable to Information Leakage?" in *IEEE International Symposium on High Performance Computer Architecture (HPCA)*. IEEE, 2018, pp. 168–179.

[79] D. Yan and H. Liu, "Parallel graph processing," *Encyclopedia of Big Data Technologies*, pp. 1–8, 2018.

[80] H. Liu, H. H. Huang, and Y. Hu, "ibfs: Concurrent breadth-first search on gpus," in *Proceedings of the 2016 International Conference on Management of Data (SIGMOD)*, 2016.

[81] H. Liu and H. H. Huang, "Enterprise: Breadth-first graph traversal on gpu servers," in *International Conference for High Performance Computing, Networking, Storage and Analysis (SC)*, 2015.

[82] H. Liu, J.-H. Seo, R. Mittal, and H. H. Huang, "Gpu-accelerated scalable solver for banded linear systems," in *IEEE International Conference on Cluster Computing (CLUSTER)*. IEEE, 2013, pp. 1–8.

[83] H. Fang, S. S. Dayapule, F. Yao, M. Doroslovački, and G. Venkataramani, "Prefetch-guard: Leveraging hardware prefetchers to defend against cache timing channels," in *IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*. IEEE, 2018.

[84] R. Mittal, J. H. Seo, V. Vedula, Y. J. Choi, H. Liu, H. H. Huang, S. Jain, L. Younes, T. Abraham, and R. T. George, "Computational modeling of cardiac hemodynamics: current status and future outlook," *Journal of Computational Physics*, vol. 305, pp. 1065–1082, 2016.

[85] H. Fang, S. S. Dayapule, F. Yao, M. Doroslovački, and G. Venkataramani, "Prodact: Prefetch-obfuscator to defend against cache timing channels," *International Journal of Parallel Programming*, 2018.

[86] H. H. Huang and H. Liu, "Big data machine learning and graph analytics: Current state and future challenges," in *Big Data (Big Data), 2014 IEEE International Conference on*. IEEE, 2014, pp. 16–17.

[87] F. Yao, G. Venkataramani, and M. Doroslovački, "Covert timing channels exploiting non-uniform memory access based architectures," in *Proceedings of ACM Great Lakes Symposium on VLSI (GLSVLSI)*. ACM, 2017, pp. 155–160.

[88] Y. Zhao, Y. Jian, Z. Liu, H. Liu, Q. Liu, C. Chen, Z. Li, L. Wang, H. H. Huang, and C. Zeng, "Network analysis reveals the recognition mechanism for dimer formation of bulb-type lectins," in *Nature Scientific Report*, 2017.

[89] H. Fang, S. S. Dayapule, F. Yao, M. Doroslovački, and G. Venkataramani, "A noise-resilient detection method against advanced cache timing channel attack," in *Asilomar Conference on Signals, Systems, and Computers*, 2018.

[90] J. Wu, S. Subramaniam, and H. Hasegawa, "Optimal nonuniform wavebanding in wdm mesh networks," in *2015 International Conference on Optical Network Design and Modeling (ONDM)*. IEEE, 2015, pp. 86–91.

[91] P. Wall, P. Colbourne, C. Reimer, and S. McLaughlin, "WSS switching engine technologies," in *Optical Fiber Communication Conference*, 2008, p. OWC1.

[92] T. Ban, H.-C. Le, H. Hasegawa, and K.-i. Sato, "Evaluation of hardware requirements for large-scale OXC architecture employing wavelength switching and fiber selection," in *OptoElectronics and Communications Conference and Photonics in Switching*, 2013, p. WT2_2.

[93] S. Subramaniam, M. Azizoglu, and A. K. Soman, "All-optical networks with sparse wavelength conversion," *Networking, IEEE/ACM Transactions on*, vol. 4, no. 4, p. 544.

[94] H. Zang, J. P. Jue, L. Sahasrabuddhe, R. Ramamurthy, and B. Mukherjee, "Dynamic lightpath establishment in wavelength routed wdm networks," *Communications Magazine, IEEE*, vol. 39, no. 9, pp. 100–108, 2001.

[95] K. Lu, G. Xiao, and I. Chlamtac, "Blocking analysis of dynamic lightpath establishment in wavelength-routed networks," in *Proceedings of the IEEE International Conference on Communications (ICC)*, 2002.

[96] K. Lu, G. Xiao, J. P. Jue, T. Zhang, S. Yuan, and I. Chlamtac, "Blocking analysis of multifiber wavelength-routed networks," in *Proceedings of the Global Communications Conference (GLOBECOM)*, vol. 3, 2004, pp. 1958–1962.

[97] A. Sridharan and K. N. Sivarajan, "Blocking in all-optical networks," *Networking, IEEE/ACM Transactions on*, vol. 12, no. 2, pp. 384–397, 2004.

[98] C. Yang, D. Liu, S. Wu, W. Li, and D. Huang, "A node-level blocking probability analysis for wrons," *Photonic Network Communications*, vol. 10, no. 2, pp. 215–223, 2005.

[99] R. Cole, K. Ost, and S. Schirra, "Edge-coloring bipartite multigraphs in O (E log D) time," *Combinatorica*, vol. 21, no. 1, pp. 5–12, 2001.

[100] M. Xu, C. Liu, and S. Subramaniam, "PODCA: A passive optical data center architecture," in *Proceedings of the IEEE International Conference on Communications (ICC)*, 2016, pp. 1–6.

[101] A. Singla, A. Singh, K. Ramachandran, L. Xu, and Y. Zhang, "Proteus: a topology malleable data center network," in *Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks*, 2010, p. 8.

[102] J. Wu, M. Xu, S. Subramaniam, and H. Hasegawa, "Evaluation and Performance Modeling of Two OXC Architectures (Invited Paper)," in *The 37th IEEE Sarnoff Symposium*, 2016.

[103] M. Xu, C. Liu, and S. Subramaniam, "Podca: A passive optical data center network architecture," *Journal of Optical Communications and Networking*, vol. 10, no. 4, pp. 409–420, 2018.

[104] F. Lezama, G. Castanon, A. M. Sarmiento, and I. Martins, "Routing and spectrum allocation in flexgrid optical networks using differential evolution optimization," *ICTON*, 2014.

[105] J. Wu, M. Xu, S. Subramaniam, and H. Hasegawa, "Joint banding-node placement and resource allocation for multigranular elastic optical networks," in *Proceedings of the Global Communications Conference (GLOBECOM)*, 2017, pp. 1–6.

[106] J. Wu, S. Subramaniam, and H. Hasegawa, "Optimal nonuniform wavebanding in wdm mesh networks," *Photonic Network Communications*, vol. 31, no. 3, pp. 376–385, 2016.

[107] J. Wu, M. Xu, S. Subramaniam, and H. Hasegawa, "Joint banding-node placement and resource allocation for multigranular elastic optical networks," *Journal of Optical Communications and Networking*, vol. 10, no. 8, pp. C27–C38, 2018.

[108] K. Christodoulopoulos, I. Tomkos, and E. A. Varvarigos, "Routing and spectrum allocation in ofdm-based optical networks with elastic bandwidth allocation," in *Proceedings of the Global Communications Conference (GLOBECOM)*, 2010, pp. 1–6.

[109] S. Shirazipourazad, C. Zhou, Z. Derakhshandeh, and A. Sen, "On routing and spectrum allocation in spectrum-sliced optical networks," in *INFOCOM, 2013 Proceedings IEEE*, 2013, pp. 385–389.

[110] M. Jinno, B. Kozicki, H. Takara, A. Watanabe, Y. Sone, T. Tanaka, and A. Hirano, "Distance-adaptive spectrum resource allocation in spectrum-sliced elastic optical path network [topics in optical communications]," *IEEE Communications Magazine*, vol. 48, no. 8, 2010.

[111] K. Christodoulopoulos, I. Tomkos, and E. Varvarigos, "Elastic bandwidth allocation in flexible ofdm-based optical networks," *Journal of Lightwave Technology*, vol. 29, no. 9, pp. 1354–1366, 2011.

[112] M. Zhang, C. You, H. Jiang, and Z. Zhu, "Dynamic and adaptive bandwidth defragmentation in spectrum-sliced elastic optical networks with time-varying traffic," *Journal of Lightwave Technology*, vol. 32, no. 5, pp. 1014–1023, 2014.

[113] Y. Sone, A. Hirano, A. Kadohata, M. Jinno, and O. Ishida, "Routing and spectrum assignment algorithm maximizes spectrum utilization in optical networks," in *Optical Communication (ECOC), 2011 37th European Conference and Exhibition on*. IEEE, 2011, pp. 1–3.

[114] Z. Zhu, W. Lu, L. Zhang, and N. Ansari, "Dynamic service provisioning in elastic optical networks with hybrid single-/multi-path routing," *Journal of Lightwave Technology*, vol. 31, no. 1, pp. 15–22, 2013.

[115] X. Chen, J. Li, P. Zhu, R. Tang, Z. Chen, and Y. He, "Fragmentation-aware routing and spectrum allocation scheme based on distribution of traffic bandwidth in

elastic optical networks," *Journal of Optical Communications and Networking*, vol. 7, no. 11, pp. 1064–1074, 2015.

[116] C.-F. Hsu, Y.-C. Chang, and S.-C. Sie, "Graph-Model-Based Dynamic Routing and Spectrum Assignment in Elastic Optical Networks," in *J. OPT. COMMUN. NETW.*, vol. 8, no. 7, July, 2016.

[117] A. N. Patel, P. N. Ji, J. P. Jue, and T. Wang, "Routing, wavelength assignment, and spectrum allocation algorithms in transparent flexible optical wdm networks," *Optical Switching and Networking*, vol. 9, no. 3, pp. 191–204, 2012.

[118] M. Zhang, W. Shi, L. Gong, W. Lu, and Z. Zhu, "Bandwidth defragmentation in dynamic elastic optical networks with minimum traffic disruptions," in *Proceedings of the IEEE International Conference on Communications (ICC)*, 2013, pp. 3894–3898.

[119] R. Wang and B. Mukherjee, "Spectrum management in heterogeneous bandwidth networks," in *Proceedings of the Global Communications Conference (GLOBECOM)*, 2012, pp. 2907–2911.

[120] ——, "Spectrum management in heterogeneous bandwidth optical networks," *Optical Switching and Networking*, vol. 11, pp. 83–91, 2014.

[121] W. Fadini and E. Oki, "A subcarrier-slot partition scheme for wavelength assignment in elastic optical networks," in *High Performance Switching and Routing (HPSR), 2014 IEEE 15th International Conference on*. IEEE, 2014, pp. 7–12.

[122] S. Subramaniam and R. A. Barry, "Wavelength assignment in fixed routing wdm networks," in *Proceedings of the IEEE International Conference on Communications (ICC)*, vol. 1, 1997, pp. 406–410.

[123] X. Zhang and C. Qiao, "Wavelength assignment for dynamic traffic in multi-fiber wdm networks," in *Computer Communications and Networks, 1998. Proceedings. 7th International Conference on*, 1998, pp. 479–485.

[124] S. Xu, L. Li, S. Wang, and C. Chen, "Wavelength assignment for dynamic traffic in wdm networks," in *Networks, 2000.(ICON 2000). Proceedings. IEEE International Conference on*, 2000, pp. 375–379.

[125] S. Xu, L. Li, and S. Wang, "Dynamic routing and assignment of wavelength algorithms in multifiber wavelength division multiplexing networks," *IEEE Journal on selected areas in communications*, vol. 18, no. 10, pp. 2130–2137, 2000.

[126] G. Li and R. Simha, "On the wavelength assignment problem in multifiber wdm star and ring networks," in *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 3, 2000, pp. 1771–1780.

[127] K. Liu, "Routing and wavelength assignment algorithm in multi-fiber wdm optical networks," in *Photonics and Optoelectronics, 2009. SOPO 2009. Symposium on*, 2009, pp. 1–4.

[128] A. Coiro, M. Listanti, A. Valenti, and F. Matera, "Power-aware routing and wavelength assignment in multi-fiber optical networks," *IEEE/OSA Journal of optical communications and networking*, vol. 3, no. 11, pp. 816–829, 2011.

[129] J. Wu, S. Subramaniam, and H. Hasegawa, "Dynamic routing and spectrum assignment for multi-fiber elastic optical networks," in *Photonic Networks and Devices*. Optical Society of America, 2018, pp. NeTu4F–1.