

OPTICAL GRID NETWORK DIMENSIONING,
PROVISIONING, AND JOB SCHEDULING

ALI ASGHAR SHAIKH

A THESIS

IN

THE DEPARTMENT

OF

COMPUTER SCIENCE AND SOFTWARE ENGINEERING

PRESENTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY
CONCORDIA UNIVERSITY
MONTRÉAL, QUÉBEC, CANADA

APRIL 2014

© ALI ASGHAR SHAIKH, 2014

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis prepared

By: **Mr. Ali Asghar Shaikh**

Entitled: **Optical Grid Network Dimensioning,
Provisioning, and Job Scheduling**

and submitted in partial fulfillment of the requirements for the degree of
Doctor of Philosophy (Computer Science)

complies with the regulations of this University and meets the accepted standards
with respect to originality and quality.

Signed by the final examining committee:

_____ Chair

Dr. Abdallah Shami - ECE, UWO, (External Examiner)

Dr. Thomas Fevens - CSE, (Examiner)

Dr. Terry Fancott - CSE, (Examiner)

Dr. John Zhang - ECE, (Examiner)

Dr. Brigitte Jaumard - CSE, (Supervisor)

Approved _____

Chair of Department or Graduate Program Director

_____ 20 _____

Christopher W. Trueman, Dean

Faculty of Engineering and Computer Science

Abstract

Optical Grid Network Dimensioning, Provisioning, and Job Scheduling

Ali Asghar Shaikh, Ph.D.

Concordia University, 2014

An optical grid network reliably provides high speed communications. It consists of grid resources (e.g., computing and data servers) and huge-data paths that are connected to geographically dispersed resources and users. One of the important issues is dimensioning optical grid networks, i.e., to determine the link bandwidth utilization and amount of server resources, and finding the location of servers. Another issue is the provisioning of the job requests (maximization of services) on the capacitated networks, also referred to as Grade of Service (GoS). Additionally, job scheduling on the servers has also an important impact on the utilization of computing and network resources. Dimensioning optical grid network is based on Anycast Routing and Wavelength Assignment (ACRWA) with the objective of minimizing (min-ACRWA) the resources. The objective of GoS is maximizing the number of job requests (max-ACRWA) under the limited resources. Given that users of such optical grid networks in general do not care about the exact physical locations of the server resources, a degree of freedom arises in choosing for each of their requests the most appropriate

server location. We will exploit this anycast routing principle – i.e., the source of the traffic is given, but the destination can be chosen rather freely. To provide resilience, traffic may be relocated to alternate destinations in case of network/server failures.

This thesis investigates dimensioning optical grids networks and task scheduling. In the first part, we present the link capacity dimensioning through scalable exact Integer Linear Programming (ILP) optimization models (min-ACRWA) with survivability. These models take step by step transition from the classical RWA (fixed destination) to anycast routing principle including shared path protection scheme. In the second part, we present scalable optimization models for maximizing the IT services (max-ACRWA) subject to survivability mechanism under limited link transport capacities. We also propose the link capacity formulations based on the distance from the servers and the traffic data set. In the third part, we jointly investigate the link dimensioning and the location of servers in an optical grid, where the anycast routing principle is applied for resiliency under different levels of protection schemes. We propose three different decomposition schemes for joint optimization of link dimensioning and finding the location of servers. In the last part of this research, we propose the exact task scheduling ILP formulations for optical grids (data centers). These formulations can also be used in advance reservation systems to allocate the grid resources. The purpose of this study is to design efficient tools for planning and management of the optical grid networks.

Contents

List of Figures	ix
List of Tables	xii
Acronyms	xiv
1 Introduction	1
1.1 Problem Statement	4
1.2 Thesis Contribution	6
1.3 Plan of Thesis	7
2 Background	8
2.1 WDM Optical Networks	8
2.1.1 Routing and Wavelength Assignment (RWA)	9
2.1.2 WDM Mesh Optical Network Survivability	12
2.1.3 Traffic Models	15
2.2 Optical Grid Networks	16
2.2.1 Advance Reservation	19

2.2.2	Anycast Routing and Wavelength Assignment (AC-RWA)	20
2.2.3	Fault Tolerance in Optical Grid Network Survivability	21
2.3	Scheduling Algorithms and Strategies	22
2.4	Task Scheduling	23
3	Literature Review	25
3.1	min-ACRWA	26
3.2	max-ACRWA	28
3.3	loc-ACRWA	29
3.4	sch-ACRWA	32
4	Optimization Models for min-ACRWA	36
4.1	Notations	38
4.2	Optimization Models for Single Link Failure	39
4.2.1	Master Problem	39
4.2.2	Pricing Problem	41
4.2.3	Solution of the CG-ILP formulation	46
4.3	Performance Evaluation of min-ACRWA	50
4.3.1	Quality of the Solutions	51
4.3.2	Influence of the Number of Server Sites and the Topology	58
4.4	Data Sets	61
4.5	Shared Risk Link Group (SRLG)	62
5	Optimization Models for max-ACRWA	66

5.1	Optimization Models for Different Levels of Protections	67
5.1.1	Master Problem	69
5.1.2	Pricing Problem	71
5.1.3	Solution of the CG-ILP Formulation	75
5.2	Performance Evaluation of max-ACRWA	76
5.2.1	Network and Traffic Instances	77
5.2.2	Protection Schemes, GoS and Bandwidth Requirements	81
5.2.3	Impact of the Number of Servers	82
5.2.4	Comparison of Link Capacity Methods	82
5.3	Data Sets	84
6	Optimization Models for loc-ACRWA	90
6.1	CG-ILP Model I: Path Pair Based Configurations	92
6.1.1	Master Problem	94
6.1.2	Pricing Problem	98
6.2	CG-ILP Model II: Source Based Configurations	103
6.2.1	Master Problem	104
6.2.2	Pricing Problem	105
6.3	CG-ILP Model III: Server Based Configurations	108
6.3.1	Master Problem	111
6.3.2	Pricing Problem	114
6.4	Solution Process	115

6.5	Numerical Results for loc-ACRWA	120
6.5.1	Data Instances	120
6.5.2	Comparison of Solution Schemes	121
6.5.3	Comparison of the Accuracy of the Solutions of the Three Models	123
6.5.4	Comparison of the Server Selection by the three models	128
6.5.5	Comparison of the Bandwidth Utilization for Model I	129
6.5.6	Single Versus Multiple Backup Paths (5 Servers)	133
7	Scheduling: Joint Optimization in Optical Grid Networks	138
7.1	Optimization Models for sch-ACRWA	139
7.1.1	Classical ILP Formulation	140
7.1.2	Column Generation (CG) Formulation	143
7.2	Numerical Results	149
7.2.1	Experimental Setup	149
7.2.2	Results	151
8	Conclusion and Future Work	155
8.1	Summary	155
8.2	Conclusion	157
8.3	Future Work	158
8.4	Publications	158
	Bibliography	159

List of Figures

2.1	Lightpaths in WDM optical networks [72].	10
2.2	Routing schemes [72].	11
2.3	Protection schemes.	14
2.4	Dedicated/shared path protection.	15
2.5	Homogeneous optical grid network.	18
2.6	Heterogeneous optical grid network.	19
2.7	An example of DAG.	24
4.1	European network topologies.	52
4.2	CSP: Total number of wavelengths.	53
4.3	SPR: Total number of wavelengths.	54
4.4	CSP & SPR: Total number of wavelengths for large instances.	55
4.5	Performances of H1 and H2 compared to CG-ILP under CSP-A.	57
4.6	Performances of H1 and H2 compared to CG-ILP under SPR-A.	58
4.7	Running time for SPR-A protection scheme.	59

4.8	Comparison of the running times for different numbers of server nodes on the EU-base topology (CG-ILP algorithm) under CSP-A scheme.	60
4.9	Comparison of the running times for different numbers of server nodes on the EU-base topology (CG-ILP algorithm under SPR-A).	61
4.10	Impact of the topology connectivity (CG-ILP algorithm): Running times for the SPR-A protection scheme.	62
4.11	SPR-A vs. CSP-A protection schemes depending on the # of server nodes on the EU-base topology with respect to the number of bandwidth units – (CSP-A - SPR-A) / CSP-A.	63
4.12	SPR-A vs. CSP-A protection schemes depending on the density of the network topology with 5 server sites with respect to the number of bandwidth units – (CSP-A - SPR-A) / CSP-A.	64
5.1	An example of computing the shared backup path.	71
5.2	Column Generation algorithm for max-ACRWA.	76
5.3	Germany network topology [51].	85
5.4	Illustration of DIST-CAP transport capacity computation.	86
5.5	Grade of Services under different protection schemes.	86
5.6	Grade of Services under two different selections of node servers, with respect to single node failures.	87
5.7	Grade of Services under uniform traffic instances.	87
5.8	Variation of the bandwidth requirements with the transport capacity calculation under uniform traffic.	88

5.9	Variation of the bandwidth requirements with the transport capacity calculation under non-uniform traffic.	88
5.10	Grade of Services under non-uniform traffic instances.	89
6.1	Model I: Configurations for single link or node or server node or server failure Scenario 4.	92
6.2	Several configuration examples for the single link failure scenario. . .	93
6.3	Model I: Configurations for single link or node failure scenario.	97
6.4	Model II: Configurations Example.	104
6.5	Model III: Configurations Example.	109
6.6	Flowchart of the solution process.	117
6.7	LP results on different stopping criteria on M-III with 100 jobs. . . .	118
6.8	Uniform traffic: Model I vs. Model II and Classical vs. MILP solution approach.	121
6.9	Non Uniform traffic: Model I vs. Model II and Classical vs. MILP solution approach.	122
6.10	Comparison of all three models with respect to their bandwidth requirements	127
6.11	Selection of server nodes under link protection scheme.	129
6.12	Comparison of the bandwidth requirements of the three models. . . .	130
6.13	ILP solution for working and backup bandwidth units under Model I (5 Servers).	131
6.14	Different protection schemes - Model I - 5 server nodes.	132

6.15	Different protection schemes - Model I - 3 server nodes.	133
7.1	Column Generation each server span of of different size and dependencies.	153

List of Tables

5.1	Execution time (sec.)	84
6.1	Summary of the three models and their solution variants.	119
6.2	Optimal (Models I and II) and optimized (Model III) LP values (5 servers).	124
6.3	Best ILP values (5 servers).	125
6.4	Optimal MILP values (with fixed server locations).	128
6.5	Some ILP gaps.	134
6.6	Model I - Working paths - Link or node or server node or server protection scheme (5 servers).	136
6.7	Model I - Backup paths - Link or node or server node or server protection scheme (5 servers).	137
7.1	Classical ILP and Column Generation Results (CPU is in seconds).	154

Acronyms

AR	Adaptive Routing
AR	Advance Reservation
AC-RWA	Anycast Routing and Wavelength Assignment
CG	Column Generation
CSP	Classical Shared Path Protection
CSP-A	Classical Shared Path Protection with Anycast
DAG	Directed Acyclic Graph
DLE	Dynamic Lightpath Establishment
EON	Elastic Optical Networking
FAR	Fixed-Alternate Routing
FR	Fixed Routing
GoS	Grade of Service
IaaS	Infrastructure-as-a-Service
ILP	Integer Linear Programing
MILP	Mixed Integer Linear Programing
MP	Master Problem

OADM	Optical Add-Drop Multiplexers/demultiplexers
OEO	Optical-Electrical-Optical
OXC s	Optical Cross-Connects
PP	Pricing Problem
RMP	Restricted Master Problem
RWA	Routing and Wavelength Assignment
SLE	Static Lightpath Establishment
SPR	Shared Path Protection with Relocation
SPR-A	Shared Path Protection with Relocation and Anycast
SPR	Shortest Path Routing
SRLG	Shared Risk Link Group
STM	Scheduled Traffic Model
WDM	Wavelength Division Multiplexing
ZT	Zettabytes

Chapter 1

Introduction

As the complexity of wavelength-routed WDM networks — where each fiber contains multiple wavelengths — continues to increase, more efficient and scalable tools are needed to address and solve the dimensioning, routing, and resource allocation problems in optical grids. Dimensioning of optical grid networks differs from the dimensioning of the classical optical networks due to: (i) anycast routing, where destinations are not pre-determined, (ii) the identification of the best server locations, (iii) different destinations in the survivability mechanisms in order to offer protection against single server node failures, and (iv) advance reservation for long lived jobs. These differences cause the need for new network management tools for optical grids, including their dimensioning.

In a grid computing environment, users submit their jobs to a grid controller/ manager, which allocates the resources. Indeed, the grid controller assigns the jobs to geographically distributed computing facilities. It corresponds to the so-called *Advance*

Reservation (AR) mechanism: it provides guaranteed services [74] by allocating the required resources before launching the jobs. The support of grid computing through optical networks enables data-intensive applications, such as distributed computing, parallel programming, information sharing, or eScience. These applications are not new paradigms, but as the computing speed increases significantly, higher volume of data can be transferred efficiently. Optical grid networks, also referred to as Lambda Grids [17], were designed in response to the need for higher data transfer [24]. Some eScience Grid examples can be found in [10]: (i) Large Hadron Collider Computing Grid Project, (ii) Biomedical Informatics Research Network, and (iii) George E. Brown Network for Earthquake Engineering and Simulation.

Another area of commercial oriented grid computing is cloud computing. Their major applications are served in data centers for storing the data, hosting the web based applications as well as computing software. These services are referred to as “Infrastructure-as-a-service (IaaS)” paradigm. In this paradigm, physical resources (servers) are typically virtualized. This virtualization concept refers to the partitioning of a single physical resource to multiple virtual resources (1:N) or an aggregation of multiple physical resources in a single virtual resource (N:1) [29].

A major concern in deploying optical grids is resiliency. Resiliency ensures that service continuity under failure conditions is of utmost importance. To deal with potential network failures, various network resilience strategies for WDM networks have been devised (for an extensive overview, see [16] [54]). For instance, end-to-end (or path) protection schemes have been developed protecting against single link

failures. Here, a primary path is protected by a link-disjoint backup path that is used in case of link failure (this link disjointness guarantees that the primary and backup path will never fail at the same time for any single link failure). This corresponds to the framework of Classical Shared Path Protection (CSP).

These protection strategies can however be optimized for the optical grid scenario, by exploiting anycast routing principle in the optical grid scenarios. Here, a user submitting a job only cares about timely and correct processing of his/her job, but is indifferent about the location of its execution. So, instead of reserving a backup path to the resource indicated by the Grid scheduler under failure-free conditions, it could be better to relocate the job to another resource if this implies network resource savings. This corresponds to the so-called Shared Path protection with Relocation (SPR). Another approach is to freely choose any server node, but both the working and the backup servers should be identical. This scheme corresponds to the so-called Classical Shared Path protection with Anycast (CSP-A). We can extend CSP-A to allow different working and backup servers. This scheme corresponds to the so-called Shared Path protection with Relocation and Anycast (SPR-A). These protection schemes: CSP, CSP-A, SPR, and SPR-A will be discussed in more details in Chapter 4.

The maximization of the Grade of Service (GoS) or the provisioning of the job requests are referred to as the routing and wavelength assignment (RWA) problem. In this problem, the solution methods have to allocate the maximum number of job requests under the constraints of a limited transport capacity. The capacity of each

link depends on the traffic data set (job requests). Finding the location of the servers is one important issue when dimensioning optical grid networks. For this reason, it is better to optimize the location of servers while finding the paths (working and backup) in order to further optimize the bandwidth utilization.

Some optical grid applications, including parallel/distributed computing and concurrent programming, in which each job is divided into multiple tasks and some tasks are dependent on other tasks. Based on the requirements, that is, bandwidth, number of CPUs, deadline of each task, the grid manager assigns the appropriate resources to each task. A grid manager can also aggregate all the tasks and assign each task on the server and on the link in a specific time. In order to achieve the best utilization of the network and the computing resources, joint optimization of network and computing resources is the best choice [38]. This strategy is often called RWA and Directed Acyclic Graph (DAG) scheduling or Task Scheduling. It is an important aspect of an efficient utilization of grid resources, which in this thesis referred to as sch-ACRWA.

1.1 Problem Statement

One of the important problems in the optical grid network is dimensioning the network; given a set of traffic, and finding the location of servers, provisioning of communication paths, determining the capacity of servers and their job scheduling. In order to solve a dimensioning problem, a routing and wavelength assignment procedure is commonly used. Moreover, some ILP formulations have been proposed, but they are

not scalable. To solve large-size instances, ILP formulations based heuristics have been also proposed (discussed in Chapter 3). The proposed algorithms are capable of solving large-size instances, but the solutions are not optimal. To get optimal solution to practical-size problems in optical grid networks, exact solutions of ILP models are required.

In an optical grid environment, only some nodes contain computing resources (servers), and the users connect to different nodes to use the resources for executing their jobs. Joint optimization of finding the locations of these computing resources and of determining the required number of wavelengths corresponds to the optical grid dimensioning (loc-ACRWA) problem. It amounts to efficiently determine the resources for the planning and management of optical grid networks. The objective of the loc-RWA is to minimize network resources while allocating the network resources for all the requested connections. To solve the loc-ACRWA joint optimization problem, different techniques involving ILP formulation and heuristic algorithms have been used. The ILP formulations presented in the literature are exact but not scalable while heuristics are scalable but not optimal. Thus, scalable exact formulations are required for the joint optimization of the RWA and the location of servers (loc-ACRWA).

Another important issue is the task scheduling problem. This problem exists when expensive resources are shared by multiple users in optical grid networks locally and remotely. In this environment, computing or storage resources are placed in few locations and are accessible from any location in the network. For an efficient use of

these expensive and limited resources an optimization tool for optical grid networks is needed.

1.2 Thesis Contribution

The objective of my research is to develop off-line exact optimization models for the planning and management of optical grid networks. These models include routing and wavelength assignment with survivability, finding the best location of servers, and job scheduling for destination (server) nodes. A Column Generation (CG) technique that involves a decomposition of the original problem into two sub-problems has been considered. This technique leads to a scalable solution with optimal or near-to-optimal results.

- The first part of my research involves the development of scalable exact ILP formulations with survivability, which we referred to as min-ACRWA. The objective of the min-ACRWA is to minimize the total bandwidth units while accepting all the requested connections.
- In the second part, we developed the capacitated network optimization models for dimensioning optical grid networks and provisioning the traffic data set, which we referred to as max-ACRWA. The provisioning of traffic data sets evaluates the grade of service on the limited transport capacity. We have also proposed a formula for initializing the transport capacity on each link. This formula is based on the traffic data sets and the locations of the servers in the

network topology.

- In the third part, we developed three different ways of decomposition of the original problem in order to solve joint optimization for dimensioning. We call this problem loc-ACRWA, where the locations of the servers and network bandwidth are optimized together.
- The last part of this thesis concerns the task scheduling problem, which we referred to as sch-ACRWA. We developed task scheduling optimization models, which can be used in the field of parallel programming and distributed computing for optical grid network environments.

1.3 Plan of Thesis

The remainder of the thesis proceeds as follows. Chapter 2 describes the general background on optical grid networks for the dimensioning including survivability. Literature review related to the thesis research is discussed in Chapter 3. In Chapter 4, optimization models for dimensioning optical grid networks is presented. In this work, we consider the unlimited transport capacity on each link. Chapter 5 presents the limited transport capacity optimization models for dimensioning problem, and Grade of Service (GoS) is evaluated. Joint optimization models for finding the paths and the locations of servers are presented in Chapter 6. Chapter 7 proposes joint optimization models for task scheduling on the servers. Finally, the conclusion of this thesis and the future work are discussed in Chapter 8.

Chapter 2

Background

This chapter presents an overview of Wavelength Division Multiplexing (WDM) optical networks. It is followed by a discussion on optical grid networks, which are emerging networks originating from WDM optical networks and grid computing.

2.1 WDM Optical Networks

Since the late 1990s, daily use of applications, such as mobile phones, Internet, video conferences, high resolution videos, and online-games, has increased together with the bandwidth demand of these multimedia applications. WDM optical networks offer large-bandwidth communication channels. An optical network consists of optical fibers as links and optical devices, such as Optical Cross-Connects (OXC), Optical Add-Drop multiplexers/demultiplexers (OADM) and other devices at nodes.

A WDM optical network supports a very large number of end systems connectivity in efficient manner. It offers high aggregate throughput with a high bandwidth capacity, and as well as high bit rate [62]. Two commonly used topologies in public WDM optical networks are the mesh topology and the ring topology. A WDM mesh topology is used in back-haul networks while WDM ring topology is used in metropolitan or access networks. In this research, we only consider the WDM mesh network topology.

2.1.1 Routing and Wavelength Assignment (RWA)

Routing and wavelength assignment refers to selecting a suitable route (lightpath) and allocating an available wavelength, for each connection respectively. A lightpath is a path between two nodes in WDM optical networks, where no buffer or Optical-Electrical-Optical (OEO) conversion is required at intermediate nodes. The lightpath is created on the same wavelength throughout the path, referred to as wavelength-continuity constraint [13], as shown in Figure 2.1. The routing and wavelength assignment problem is a NP-complete problem. To solve it, it is often decoupled into two subproblems [72]: the routing problem and the wavelength assignment problem. Some common routing and wavelength assignment techniques are discussed next.

Routing Schemes

This section describes the three general approaches used in establishing lightpaths in WDM mesh optical network [72].

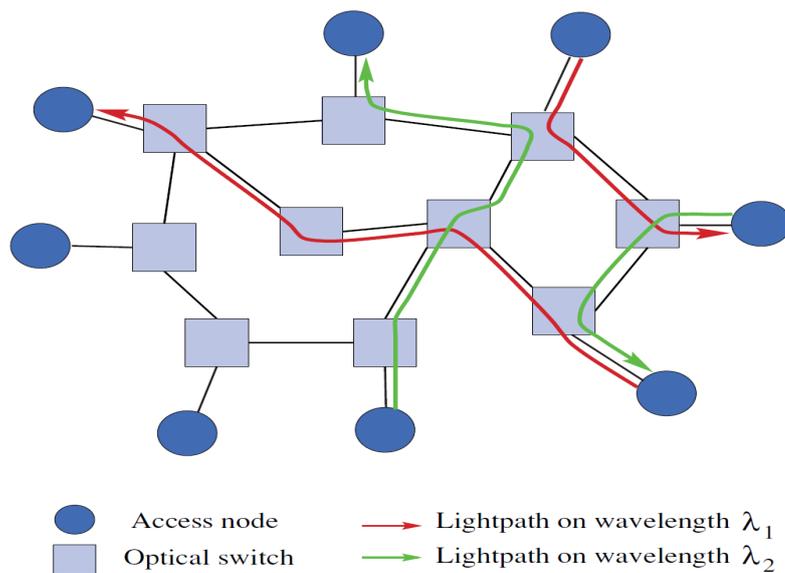


Figure 2.1: Lightpaths in WDM optical networks [72].

Fixed Routing (FR). A single fixed route is predetermined for each source-destination pair. One common example is fixed shortest-path routing (by using Dijkstra's or Bellman-Ford algorithms), where shortest paths are used, shown in Figure 2.2(a).

Fixed-Alternate Routing (FAR). Multiple fixed routes are precomputed for each source-destination pair. Each node maintains multiple routes for all other nodes available in the network topology. Figure 2.2(b) shows two routes from node 0 to node 2.

Adaptive Routing (AR). It finds the route based on network link state (load) information, to reduce a chance of connection blocking for incoming demands. A

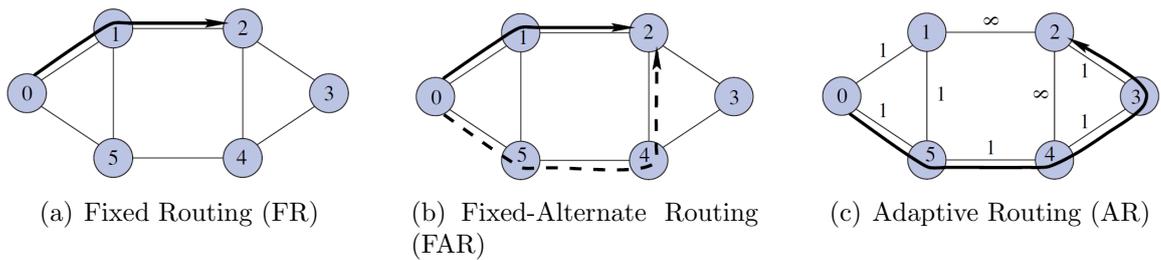


Figure 2.2: Routing schemes [72].

common approach to finding the route is the shortest-cost-path routing, where more busy links have higher cost than the less busy links. Figure 2.2(c) shows a shortest-cost-path from node 0 to node 2.

FR and FAR are much simpler than AR, but may suffer from higher *connection blocking*.

Wavelength Assignment

For the case in which lightpaths are established one at a time (either incremental or dynamic traffic), wavelength assignment heuristics are used. Some commonly proposed heuristics are [72]:

Random. This scheme first searches all available wavelengths, then assigns one randomly among them, usually with uniform probability.

First-Fit: In this scheme, all wavelengths are numbered ($\lambda_1, \lambda_2, \dots, \lambda_n$). When searching for a free available wavelength, lowest-numbered wavelengths are considered before highest-numbered wavelengths.

Least-Used (LU). This scheme selects the least used wavelength in the network which maintains load balancing on the network.

Most-Used (MU). This scheme is the opposite of LU, that is, to select the most-used wavelength in the network.

Min-Product (MP). This heuristic scheme is used in multi-fiber networks: each link contains multiple fibers. In single-fiber networks, MP becomes FF. The goal of MP is to pack wavelengths into fibers so that comparatively use a smaller number of fibers.

In addition to these wavelength assignment heuristics, one step solutions based on ILP formulations are also proposed. These formulations are used in static traffic, for further detail of ILP formulations refer to the survey papers [30], [31], and [33].

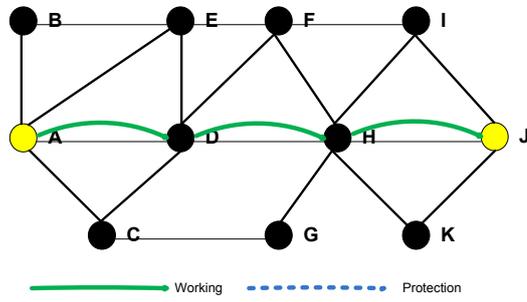
2.1.2 WDM Mesh Optical Network Survivability

Network survivability is also an important issue in WDM optical networks. Two protection techniques are commonly used in WDM optical networks: *proactive* and *reactive*. The former computes working and alternate-backup paths, and reserve resources for them before establishing a lightpath. This technique is referred to as pre-planned protection. The latter finds a backup path when a failure occurs — this technique is called online-provisioning. The proactive technique has a guaranteed service, and requires less restoration time, although it is less efficient in terms of resource utilization. On the other hand, the reactive technique is more efficient but

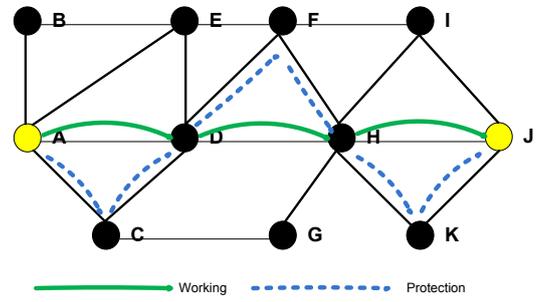
may fail to find a backup path due to a lack of resources. Thus, the reactive technique does not guarantee a successful restoration [48], [49].

Schemes for the protection of network resources include, *link protection* (Figure 2.3(b)), *segment protection* (Figure 2.3(c)), and *path protection* (Figures 2.3(e)). In link protection, each network link (edge) of a path is protected separately but the nodes are not protected. While in segment protection, a set of consecutive links and some (Figure 2.3(c)) or all (Figure 2.3(d)) intermediate nodes are protected. Path protection comes in two flavors; link protection and node protection, shown in Figure 2.3(e) and Figure 2.3(f). Node protection entails link protection.

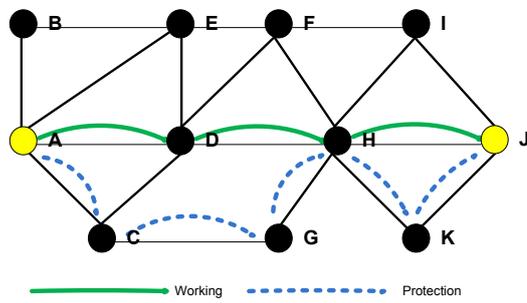
Link protection recovers the failure of working channels on a single fiber link through local re-routing, while path protection recovers the failure of working path through end-to-end (source-destination) re-routing. Finally, segment protection recovers the failure of a segment through two end nodes. Path protection offers better capacity utilization while link protection offers faster restoration. Segment protection offers capacity utilization and network restoration in between those offered by path protection and link protection [53]. In this particular example (Figure 2.3(c)), working path A-D-H-J comprises two segments. First segment is the A-D-H subpath, and its protection is provided by A-C-G-H subpath. Second segment is the H-J subpath, and its protection is provided by H-K-J subpath, this scheme is known as Basic Segment Protection [27]. Another segment protection scheme is Segment Protection with Overlap [27], shown in Figure 2.3(d). In this example, the working path comprises two segments, i.e., A-D-H and H-J, but the protection segments are A-C-G-H



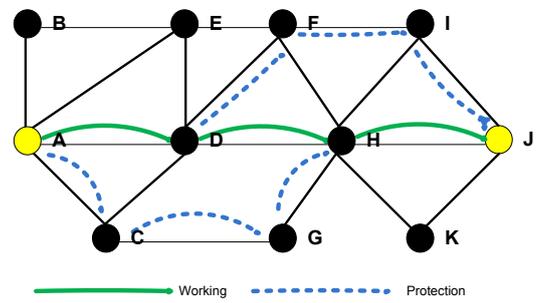
(a) Topology with one connection



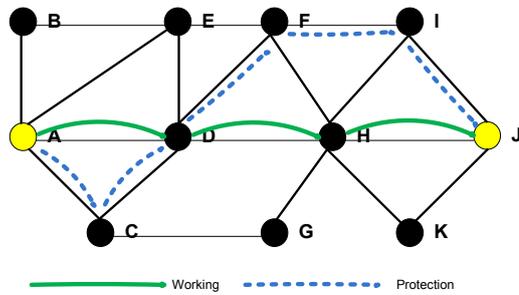
(b) Link protection for a link failure



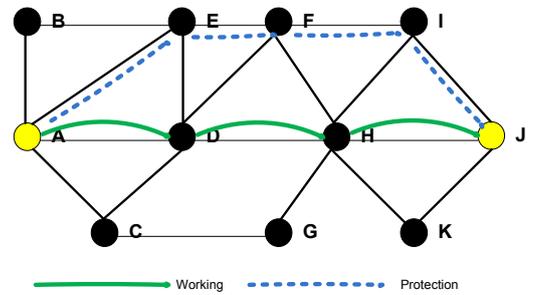
(c) Segment protection for a link failure



(d) Segment protection for a node failure



(e) Path protection for a link failure



(f) Path protection for a node failure

Figure 2.3: Protection schemes.

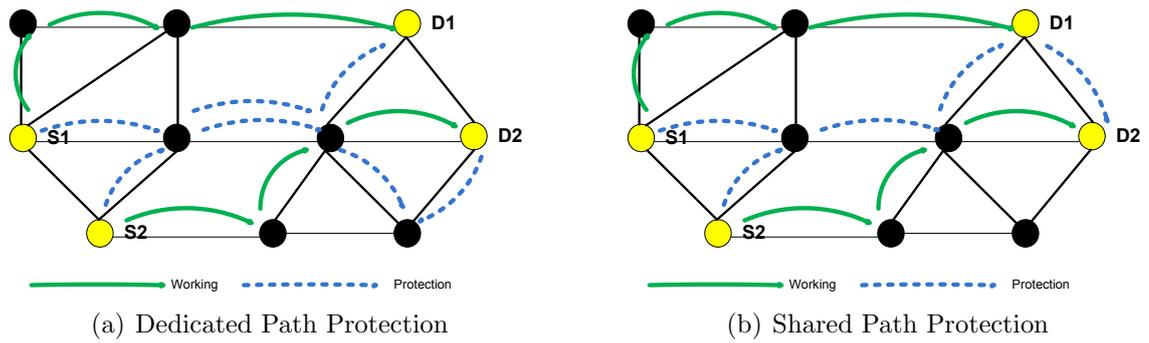


Figure 2.4: Dedicated/shared path protection.

and D-F-I-J. In this sense, it also covers node H which is a connection node of two segments; this scheme protects all intermediate nodes.

These protection schemes may be used for dedicated or shared protections. For dedicated protection, each working link or segment or path has its own protection. However, for shared protection, a protection link or segment or path may be shared by multiple working paths. Figure 2.4(a) and Figure 2.4(b) show respectively dedicated and shared path protection of two connections, that is, $S1$ to $D1$ and $S2$ to $D2$.

2.1.3 Traffic Models

Zang *et al.* [72] describe three commonly used traffic models. This section highlights the key aspects of these models.

Static Traffic. In this model, the set of requested connections is known in advance, and the connections remain in the network for a long time. Here, the RWA problem is also known as Static Lightpath Establishment (SLE).

Incremental Traffic. In this case, a lightpath is established for each incoming connection. Like static traffic, connections also remain for a long time, but connection requests arrive sequentially.

Dynamic Traffic: Like incremental traffic, a lightpath is established for each requested connection. However, connections remain for some finite time on the network.

Because in both incremental and dynamic traffic models, lightpaths are established dynamically, they are referred as Dynamic Lightpath Establishment (DLE). Of course, this mode of establishing lightpaths helps to minimize the resource and also minimizes blocking for incoming connections.

Kuri *et al.* [39] present an extension of static traffic model called Scheduled Traffic Model (STM). Apart from source and destination, STM also includes start and end times of each connection. The model can be used with *fixed windows* or *flexible windows*. In fixed windows, start and end times cannot be altered while in flexible windows, time can slide within a larger window.

2.2 Optical Grid Networks

This section presents an overview of optical grid networks, and their dimensioning and scheduling problems. An optical grid network is an emerging network originated from WDM optical networks and grid computing. In grid networks, distributed resources (computing or storage elements as well as scientific instruments) are interconnected to support compute-intensive and data-intensive applications [61]. Nowadays, most

critical scientific applications, multimedia applications, and business grids need to exchange huge amounts of data between the distributed sites. Optical networks are employed to provide high-bandwidth optical fibers and lightpaths for data transfer between interconnected grid resources. The grid is upgraded to the so-called Optical Grid [70].

The term “grid” arises from electrical “power grid”, the idea is that accessing the computing power and storage of computers connected through some types of networks is similar, as accessing to electrical power from an electrical grid [2]. The consumers of electricity do not care which electric grid station provides electricity. Similarly, the users of an optical grid network do not need to worry about where a given job will be executed. Hundreds of computer grids are available around the world; they are used in different areas of research, such as biological science, earth science, high energy physics, engineering, among others. Currently, there are few service providers who commercially offer grid resources on-demand, such as Amazon’s cloud computing “Elastic Compute Cloud” [68].

Recall that an optical grid network corresponds to geographically spread resources in different locations, connected through an optical transport network, and consisting of core and access networks. The core network is connected through Optical Cross Connects (OXC) and optical fibers, and in access networks, each site is connected to the OXC through optical fibers or any other media. A site comprises users and the computing resources. Each optical fiber contains a limited technology-dependent number of wavelengths, and each wavelength has also technology-dependent data rate

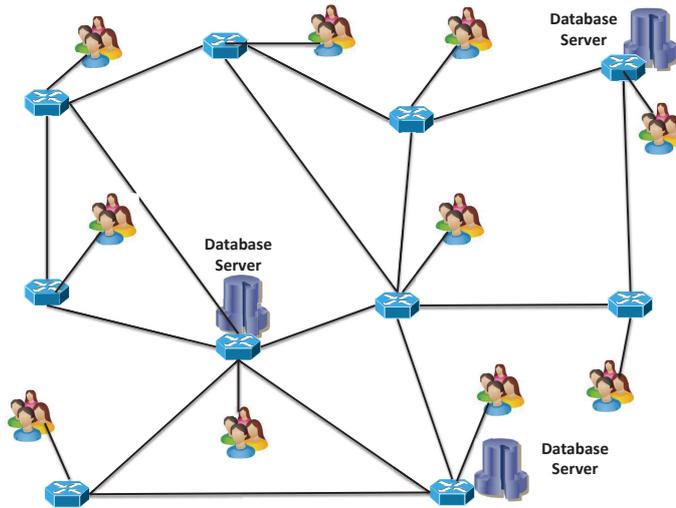


Figure 2.5: Homogeneous optical grid network.

(bandwidth) [65]. An optical grid network may consist of homogeneous or heterogeneous resources. Homogeneous resources refer to all the server nodes with the same functionality, i.e., each server node offers a similar type of services. For example, Figure 2.5 shows server nodes that offer data-intensive services. However, heterogeneous resources grid network offer different types of services, as shown in Figure 2.6. In this particular case, one node offers video services only, another node offers information services. There is yet another node that offers two services: application (computing) and data-intensive services.

In terms of traffic volume, it is expected that by 2016, global data center traffic could reach 6.6 zettabytes ($1 \text{ ZB} = 10^{21}$ bytes), and nearly two thirds thereof will be cloud traffic [1]. This growing demand of traffic requires a reliable and high-bandwidth communication medium, i.e., optical fibers. These fibers can be efficiently used by applying Dense Wavelength Division Multiplexing (DWDM) technology, i.e., running

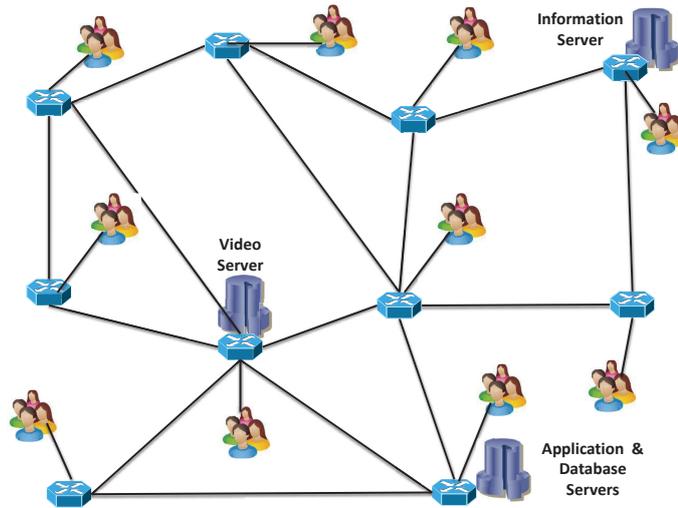


Figure 2.6: Heterogeneous optical grid network.

multiple wavelength carriers simultaneously over the same physical fiber to provide large bandwidth and thus as cost-effective solution to the network providers. Given the continually rising bandwidth demands, today's solutions can run 100Gb/s per wavelength (40-80 wavelengths on each fiber pair using DWDM). Currently, flexible grid networks are being considered: the flexible grid refers to the adaptive transceivers and intelligent nodes, allowing service providers to increase the bandwidth without overhauling it [25]. This new paradigm is called Elastic Optical Networking (EON).

2.2.1 Advance Reservation

Advance reservation (AR) system is also important in some fields of optical grid networks including data-intensive and video conferences for surgery. For example, if a surgeon is assisting a colleague to perform a surgery at a remote site, AR ensures availability of required bandwidth on the specified time [23].

2.2.2 Anycast Routing and Wavelength Assignment (AC-RWA)

The optical network is a prominent candidate for high data rate communications, reliable and economical as compared to others. In traditional optical networks, users have fixed destinations to execute their jobs, while, in an optical grid network, a user does not care about where the job is to be executed; this is known as *anycast routing*, also referred to as *location transparency* [50]. This major difference of optical grid networks require the architecture of a flexible optical layer, routing, wavelength assignment, dimensioning, and task scheduling strategies [20].

Given the amount of traffic, the determination of required resources (number of servers and link capacity) in optical grids is referred to as the dimensioning problem. A dimensioning problem in optical grid networks is different than in classical optical networks in two ways [21]. First one, needs to find suitable destination; optical grids work on anycast routing, where only the source is known and the destination can be selected to be any best node that can execute the requested job/task. Secondly, the task can also be lost because of lack of executing resources.

A key problem in optical grid networks is how to efficiently manage the available infrastructure in order to satisfy user requirements and maximize resource utilization. This is in large part influenced by the routing and scheduling of tasks [63], which leads to develop efficient routing and scheduling strategies.

2.2.3 Fault Tolerance in Optical Grid Network Survivability

In an optical grid network, WDM mesh optical network survivability techniques can be used. Faults can also occur in optical grids as in traditional optical networks, and these faults may occur because of the failure of a link, a node, or server resources. In a grid environment, users do not care about the faults due to anycast principle. In anycast routing, destinations are not fixed, so if there is any resource failure on the primary server, a submitted job should be diverted to the backup server. Different schemes are used for the backup server, but in optical grid networks, resources are pre-computed for backup [47]. In addition to those hardware faults, there is also the possibility of software faults occurring in applications, operating systems, protocols, among others. Common software faults include unhandled exceptions (run time errors), division by zero, and memory leaks.

Two recovery strategies exist for providing fault tolerance in an optical grid network: *Job check-pointing* and *replication*. Job check-pointing periodically stores the image of a job, which can be restored in case of failure. In replication, a job is sent to the primary as well as to the replication (secondary) server. If there is a failure on the primary server, the replication server will continue taking the execution of the job [14]. For a recent survey on strategies for fault tolerance in optical grid networks see [12].

2.3 Scheduling Algorithms and Strategies

A schedule of tasks is the assignment of tasks to specific times on the resources. Two common types of scheduling algorithms under research are *static* and *dynamic*. In static, also referred to as *advance*, a set of tasks is known in advance and has to be mapped onto the resources before the execution starts. In contrast, in dynamic, also referred to as *immediate*, each arriving task has to be mapped onto the network [64]. Some applications may have a deadline time to execute their tasks.

In an optical grid network, allocation of both resources (network and CPU) is called *co-allocation*. Access to the resources depends on the policy imposed by the resource owners. This policy is developed based on the type of executing jobs on the networks. An optical grid network has an important role in the area of computing that needs large/complex computations, expensive licensed software, or large data storage. Some applications also need large flow of data between a user and executing servers, e.g., data storage, and complex flow for climate-research, high-energy physics. For these large and complex flows, an optical grid is the best candidate due to its large bandwidth, low latency at economical cost [71].

Mostly combinatorial scheduling problems are NP-Complete [67]; many scheduling heuristics have been proposed in the literature. The objective of these algorithms is to minimize the execution time of a given set of tasks. The best known heuristic is the listing algorithm, e.g., Algorithm 2.1 [3].

Algorithm 2.1 List scheduling

Sort the list of tasks according to priority schemes.
for each task **do**
 Find computing resource (r) that allow earliest finish time
 Schedule task on r .
end for

2.4 Task Scheduling

In parallel programming or distributed computing, each job is decomposed into multiple tasks. To execute such tasks, the systems need to allocate the computing resources and communication paths to them. The allocation of computing resources among multiple tasks is known as *task scheduling*. In task scheduling, tasks are represented in a Directed Acyclic Graph (DAG), also known as a *task graph*. Some tasks can be executed in parallel if they are independent, if they are dependent on other tasks then they are executed on an incremental (one-by-one) basis.

A directed acyclic graph $G = (V, E, w, c)$ represents a list of tasks. The set of nodes (V) represents tasks with expected execution time, and a set of edges (E) represents communication paths and precedence between nodes. The computation cost and communication cost are represented by $w(n)$ and $c(e_{ij})$ respectively [60]. All instructions/operations are executed in a sequential order and there is no parallelism within a task. The nodes are strict with respect to their input and output; it means that a node can not start execution before receiving the input and it can not produce output until the execution has finished. Each node may be assigned multiple tasks, all direct predecessor tasks of node n_i are represented by $pred(n_i)$ and all direct successor tasks of the node n_i are represented by $succ(n_i)$. If node $n_i \in V$ does not contain

predecessor, it is represented as $pred(n_i) = \emptyset$, and is known as source node. Similarly, $succ(n_i) = \emptyset$, is known as sink node.

An example of DAG is shown in Figure 2.7. A job is divided into six tasks T_1, T_2, \dots, T_6 , and it needs 3 different types of resources R_1, R_2, R_3 . In DAG, each node contains three types of information: task id, resource id, and estimated execution time. The label on each directed edge represents an amount of data needed to be transmitted from one node to another. This DAG indicates that task T_1 has been executed and produce the output before starting the execution of the tasks T_2, T_3, T_4 . However, tasks T_2, T_3, T_4 can be executed in parallel. Similarly, tasks T_5 and T_6 can not start execution before the completion of tasks T_2, T_3 and T_3, T_4 respectively.

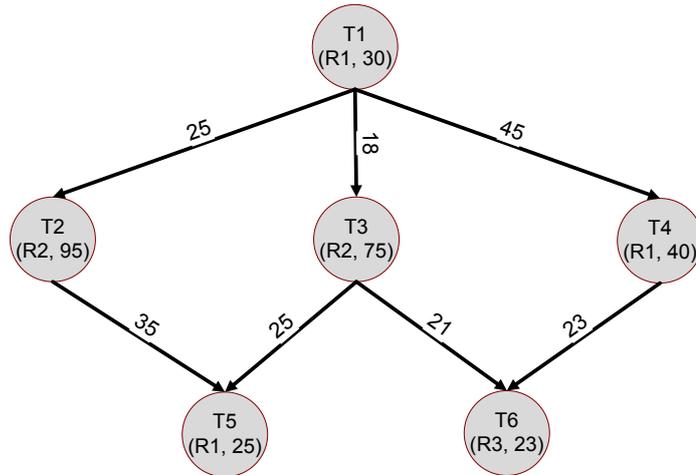


Figure 2.7: An example of DAG.

This example corresponds to a heterogeneous optical grid network. While in case of homogeneous, all the tasks require the same types of resources so that the type of resources (e.g., R_1, R_2 , and so on) can be omitted from task nodes.

Chapter 3

Literature Review

Optical networks are employed to facilitate reliable and faster communications for data transfer between interconnected grid resources, and the grid computing is updated to the so-called Optical Grid. In the recent years, the improvement of communication systems in distributed computing and storage-systems has received some attention. An optical grid network is a promising candidate for reliable and cost effective communication systems. This chapter presents the literature review regarding dimensioning (ACRWA) and task scheduling problems in optical grid networks.

The vast research literature devoted to RWA focuses on finding a suitable routing path and wavelength assignment, assuming both source and destination of connection requests are given (i.e., the *unicast routing* case). The most studied objectives are the minimum number of wavelengths (min-RWA) and the maximum grade of services, i.e., number of granted requests (max-RWA). For an extensive overview of such classical RWA literature, we refer to [22] and [72] and more specifically to the Integer Linear

Programming (ILP) reviewed models in [30], [31], and [33].

As highlighted in the introduction, we address the *anycast routing* case, where the problem is complicated by the fact that the destination is not known a priori, but can be freely chosen (among a given set of possible destinations, i.e., server sites). Note that proposed algorithms in the literature of dimensioning optical grid networks assume that wavelength converters are available at each optical node, and we also make the same assumption in our proposed models. Next, we introduce the studied research problems, followed by their related literature review.

3.1 min-ACRWA

In the first part of our study, we deal with link dimensioning in optical grid networks. We consider the objective of minimizing the number of wavelengths (we refer to as min-ACRWA) summed over all network links, i.e., the number of bandwidth units, see Chapter 4. We will assume that the locations of the servers are given. In order to select the best locations, one can either use [20], or the new models we developed in Chapter 6. Several Integer Linear Programming (ILP) formulations and heuristics have been proposed for traditional RWA problem: For recent review of ILP formulations and comparison see [30], [31] and [33] and for heuristics [72].

Zang *et al.* [73] present path protection routing and wavelength assignment for the traditional WDM mesh optical networks, where destination of each request is given. They proposed an ILP formulation for obtaining optimal solution. For scalability,

the original problem has been decomposed in two ILP sub-problems; one for local optimization and another for a global optimization. The protection constraint is a path protection duct-layer. A duct is a group of cables buried together.

Based on the work of Zang *et al.* [73], Buysse *et al.* [6] proposed path protection routing and wavelength assignment ILP formulation for optical grid networks. The objective is to minimize the number of bandwidth units for a given (fixed) destination for the working path and anycast principle is used for the backup path. This ILP formulation successfully solves only instances with up to size of 20 requested connections on Geant2 network topology (17 nodes, 45 directional links), due to scalability problems. The path protection under single-link-failure survivability has been considered. An experimental result shows that 20% of the number of wavelengths have been saved as compared to the case, where destination is given (fixed) in both working and backup paths [73].

Similarly, Buysse *et al.* [6] modify their ILP formulation for anycast routing principle in [7] for both working and backup paths. In addition to the ILP formulation, they also proposed a heuristic based on ILP formulation for large instances. The heuristic is able to solve large (up to 150) size instances, with an optimality gap of more than 6%. Aforementioned work assumes all the nodes are equipped with wavelength conversion.

In order to overcome the scalability and optimality gap issues, we present large scale optimization models based on column generation [15]. Therein, the original problem is decomposed according to a set of configurations, where a configuration is

added only if it contributes to the improvement of the current value of the objective, presented and discussed in Chapter 4. Indeed, CG models have already been successfully used for solving several design/management problems in optical and wireless networks, e.g., p -cycle based problems [57] and resource allocation in WiMax networks [8].

3.2 max-ACRWA

In the second step, we deal with the provisioning of job requests under limited number of bandwidth units on each link. We consider the objective of maximizing the grade of service (we refer to as max-ACRWA) assuming limited transport capacity on each link, see Chapter 5. We consider here an off-line network design problem, aiming to decide on the network and server resource dimensions. Again a set of server sites is given.

Determining the proper dimensioning of the links is one of the important issues in an optical grid network. Most researchers evaluate the minimum bandwidth requirements under unlimited capacity constraints. Usually, they do not take into account that the transport capacity values can only take a limited number of discrete values [6], [7], [72]. However, for instance, in SDH/SONET (Synchronous Digital Hierarchy/Synchronous Optical NETworks) networks, the capacities are determined by the interface speeds which only take discrete values (i.e., 155.52; 622.08; 2,488.32; 9,953.28 and 39,813.12 Mbps), which always multiply by exactly four. Furthermore,

in Coarse Wavelength Division Multiplexing (CWDM) or Dense Wavelength Division Multiplexing (DWDM) systems, the number of wavelengths to be used varies in steps of 8, 16, 24, 32, 40, 48, 64, 80, 96, 120 etc. wavelengths per fiber [37].

For these reasons, determining the link transport capacities is a critical issue in optical grid networks. No paper has yet addressed the issue of discrete capacity values in such a context. While both min-RWA and max-RWA problems have been well investigated for classical optical networks, all the aforementioned models for optical grid networks only consider the min-ACRWA problem with unlimited transport capacities. In this study, we therefore to study the max-ACRWA problem with different protection schemes and transport capacity calculation methods. The proposed CG-ILP model is based on a flow formulation, presented and discussed in Chapter 5.

3.3 loc-ACRWA

We extend min-ACRWA problem by finding the locations of servers while provisioning the working and backup paths (we refer to as loc-ACRWA), see Chapter 6. This means, the number of server sites is given but their locations are optimized while finding the working and backup paths.

The optimization of the servers locations in an optical grid environment has some resemblance with some classical facility location problems, namely the p -median and p -center problems, which have been widely investigated in the literature (see e.g., [52] and [55]). Both problems deal with the locations of p facilities. The p -median problem

searches the facility locations such that the sum of the shortest demand weighted distance between "customers" and p "facilities" is minimized. On the other hand, the p -center problem identifies the facility locations in order to minimize the maximal distance for all demand points. While the optical grid network dimensioning problem shares some features of the p -median problem, the former problem is more complex due to: (i) the distance function (expressed in terms of, e.g., optical hops) and (ii) the additional requirements of backup paths in order to ensure network survivability. These last two features make the joint optimization of finding server locations and dimensioning both working and backup paths much more complex than the facility locations in a p -center or p -median context.

Buyse *et al.* [6] solve the dimensioning problem in two steps for finding the paths and locations of the servers. The first step finds the locations of servers and assigns servers based on a given source and job arrival rate of each connection. The second step optimizes the routing and wavelength assignment for working and protection paths based on the given source and destination and locations of servers (determined in the first step).

Another four step solution for solving the grid dimensioning problem was proposed by Develder *et al.* [20]. In their study, they investigate the locations and capacity of servers and also determine link capacity. The first step is similar to that used by Buyse *et al.* [6]. The second step calculates the server capacities. The third step determines the access link (inter-site) bandwidth, and prefers to execute local jobs at their own server if server resource is available. Otherwise jobs will be executed on

remote servers. Shortest Path Routing (SPR) algorithm is used for routing from user to the server. In the last step, link bandwidth are calculated based on traffic matrix (source-destination) solved in the first three steps.

Leenheer *et al.* [42] also investigate capacity of each server, and the link bandwidth to install while meeting the given maximum job loss-rate criterion. The authors proposed an iterative approach as in [20]. They assume Poisson job arrival without any buffer at the server node; if no free server is found at the job arrival time, the job is lost.

A joint optimization for network bandwidth units (link dimensioning) and amounts of server resources is studied in [19]. This work is based on CG-ILP formulation, and solve large size instances for different types of backup path relocation (protection levels). An extension of [19] for relocation server site failures is presented in [18].

Larumbe and Sansò [40] investigated the optimal locations of data centers, with the objective of minimizing the average network delay (convex objective), without taking into account any reliability concern or link dimensioning. Chakareski [9] also studied the locations of data centers, with the aim of minimizing the overall operating cost of the network (again a convex objective), in a context of multi-service networks. They take the link and resource capacity into account, but do not address survivability.

Survivability is also a major issue in the area of data communications, where in case of link failure on the working path, the traffic is routed on the backup path. In the environment of optical grids, where users are transparent to the service centers

(data center), survivability is one of the most important issues. For this reason, we have also considered finding the backup paths while optimizing the working paths and the locations of servers. This additional feature complicates the dimensioning of optical grid/cloud networks.

To the best of our knowledge, we propose for the first time a joint optimization ILP formulation for determining the link capacity (including survivability) and the locations of servers. In this regard, we propose three different mathematical models with three different decomposition schemes in order to address the optimality gap issues, discussed in Chapter 6.

3.4 sch-ACRWA

In the last part of this study, we have developed and proposed task scheduling optimization models for servers (data centers) for dependent tasks in an optical grid network environment (we refer to as sch-ACRWA), see Chapter 7. As discussed in Chapter 2, Section 2.4, in a parallel programming or distributed computing environment, each job is divided into multiple tasks, and their dependency are represented in a Directed Acyclic Graph (DAG). In order to achieve the best utilization of network and computing resources, joint optimization of network and computing resources is the best choice [38]; this is called RWA and DAG scheduling, also referred to as task scheduling. Some literature of task scheduling does not consider the RWA. Task scheduling is an important aspect of efficient utilization of grid resources.

Joint optimization of computing and network resources, based on DAG, has been investigated in many projects [26], [36], [38], [43], [44], [69], and [76]. Traditional ILP formulation and heuristic algorithms (discussed next in more detail) have been proposed.

The task scheduling problem is NP-complete in its general form [67]. Two step listing algorithms for joint DAG scheduling of computing resource and network resource allocation are presented in [36], [38], and [69]. The first step sorts the task-list in descending order based on bottom-levels. A bottom-level is a technique commonly used in list scheduling algorithms, where a task that needs a longer execution-time is given a higher priority. The second step sequentially schedules a task on a server where it can finish early. The objective is to minimize the schedule length under the constraint of executing all the jobs. In [38] and [69], network utilization efficiency is also considered while minimizing the demand completion times. For efficient network utilization, Yan *et al.* [69] used hop-bytes techniques whereas Kannasoot *et al.* [38] used starting and ending times of data transfer. Hop-bytes is the metric proposed in [4], and is calculated based on the required communication bytes and distance. Results show that adaptive routing (AR) algorithm is more effective for reducing the schedule length than other two routing methods (fixed routing and fixed-alternate routing) [36], [69]. AR has a drawback of link utilization because it uses the links which has less load to avoid the blocking contention. For simplicity, Kanasoot *et al.* [38] use the fixed routing scheme.

Others ([43] and [44]) have also proposed ILP formulation and two heuristic

listing algorithms for a joint optimization of RWA and DAG (task) scheduling. In their work, the authors include task deadline constraints. Experiments on NSFNET show that network utilization can be improved by up to 28% without sacrificing on the schedule completion time. Guo *et al.* [26] present another ILP formulation and heuristic of joint optimization, network resources and computing resources. Their aim is to determine the actual finish time as compared to the expected time.

Zhu *et al.* [76] proposed a fault-tolerant scheduling heuristic algorithm (for both computing and network resources) based on DAG, called Grid Resource Protection (GRP) scheme for DAG scheduling. In this scheme, each task is assigned to two different servers: the primary and backup. This scheduling algorithm jointly allocates server and network resources. GRP scheme is more reliable but is less efficient in resource utilization. For a recent survey on fault tolerance in optical grid networks see [12].

In addition to joint DAG scheduling optimization, many algorithms for DAG scheduling have been proposed in the last few decades for only computing resources in the area of cloud/grid computing, distributed systems. These studies do not consider the communication paths for dispersed resources, some recent algorithms are discussed in [45] and [75].

All works discussed before are based on single DAG scheduling, recently a heuristic scheduling algorithm for multiple DAGs is presented in [56]. This algorithm considers the main objective as minimizing DAGs completion time and also try fairness among the DAGs, and the lowest transportation time.

In this study, we consider task scheduling with RWA based on DAG and propose an exact ILP formulation using column generation techniques. This formulation jointly optimizes the computer resources and the communication paths with fixed routing, and it can work with single or multiple DAGs. This means, there is no task overlap on the servers, and the data transfer time is also considered for dependent tasks, see Chapter 7.

Chapter 4

Optimization Models for min-ACRWA

Different approaches have been proposed to solve the classical RWA problem in WDM optical networks, with/without addressing the question of the protection. Different objective functions have been considered with the most studied ones being minimizing the cost of the network resources (min-RWA problem), or maximizing the grade of services (max-RWA problem), or minimizing the blocking probability. In optical grid networks, the RWA problem evolves toward the so-called AnyCast Routing and Wavelength Assignment (ACRWA) in order to handle the anycast requests with non pre-determined destinations.

In this chapter, we propose scalable optimization models for ACRWA with the objective of minimizing the total network capacity (min-ACRWA) for optical grid

networks including survivability. In the next chapter we will discuss about the max-ACRWA.

We started with the design of an optimization model for Shared Path Protection with Relocation (SPR) and compared it to Classical Shared Path Protection (CSP). In the SPR case, working path destination is fixed (given) and protection path can be any server node. In the CSP case, working and protection paths have the same fixed destination. This case study has been published in [28]. Note that, this study only considers heterogeneous optical grid networks, where all the server nodes (data centers) have the same types of resources.

In the next step, this first optimization model is extended to the anycast routing principle, i.e., Shared Path Protection with Relocation and Anycast (SPR-A), and Classical Shared Path Protection with Anycast (CSP-A). Here, only the source of each connection is given; destination can be any server node. In case of SPR-A, the destinations for both paths can be any two server nodes (not necessary the same), while in case of CSP-A, the destination can be any server node and must be the same for both paths. Column Generation Integer Linear Programming (CG-ILP) formulation for all aforementioned cases is presented in Section 4.2 under single link failure scenario. This case study has been published in the “Journal of Optical Communication Networks (JOCN)” [58].

In the last section of this chapter, we also discussed the Shared Risk Link Group (SRLG), where multiple links are buried together within in single duct. Thus, these multiple links have the same risk of failure. In this context, we present a SRLG

formulation. At the end, we also present single node failure scenario formulation.

4.1 Notations

A network topology and traffic input is modeled as follows:

$G = (V, L, V^{\text{SVR}})$, directed graph representing an optical grid, where V is the node set, L is the set of (directed) links and $V^{\text{SVR}} \subset V$ is the server node set.

V Node set, indexed by $v \in V$, representing the OXCs and possibly collocated server sites (computational and/or storage servers).

L Directional link set, indexed by ℓ . Each pair of connected nodes is usually connected by two links, one in each direction.

$V^{\text{SVR}} \subset V$. Server node set, indexed by v or v^{SVR} , comprising the server sites (capable of processing grid jobs), i.e., potential candidate destinations.

V^{S} Set of job requests originating at source node $v^{\text{S}} \in V \setminus V^{\text{SVR}}$, or set of job originating at source node $v^{\text{S}} \in V \setminus V^{\text{SVR}}$ for the destination server $v^{\text{SVR}} \in V^{\text{SVR}}$

$D_v = |v^{\text{S}}|$, i.e., number of job unit demands from source node v^{S} , or $= |v^{\text{S}}|$, i.e., number of job unit demands between source v^{S} and destination v^{SVR} .

4.2 Optimization Models for Single Link Failure

Optimization models are developed through the Column Generation (CG) technique. A CG model is a decomposition of the original problem into two sub problems: Master Problem (MP) and Pricing Problem (PP). Each generic PP has been solved for each source node v^s , and we denote it by PP_{v^s} . The MP and the PP are solved in an integrated way, and their solving technique depends on the decomposition of the original problem. For further details of CG technique, readers are referred to [5], [32], [46] and for linear programming concepts [15] and [41].

An objective of MP is same as the original problem, i.e., minimization of network resources while satisfying all the requested connections. The Master problem takes care of satisfying all the demands with shared path protection. On the other hand, the PP finds link-disjoint working and protection paths for each request-set ($v^s \in V^s$) based on maximizing the shared path protection. This CG-ILP formulation is scalable in terms of network topology and the number of requested connections in the traffic model.

4.2.1 Master Problem

The master problem of the column generation ILP model uses two sets of variables: $z_c \in \mathbb{Z}_{(c \in C)}^+$ and $b_\ell^B \in \mathbb{Z}_{(\ell \in L)}^+$. The value of each variable z_c is equal to the number of selected copies of configuration c . Variable b_ℓ^B is equal to the maximum required bandwidth units on link ℓ . The PP_{v^s} generates potential configuration related with

source node v^s . To complete the characterization of the configurations, we need the following parameters:

$p_{c\ell}^W = 1$ if link ℓ is used by the working path of configuration c , 0 otherwise.

$p_{c\ell}^B = 1$ if link ℓ is used by the backup path of configuration c , 0 otherwise.

The objective function which minimizes the total network capacity, can be written as follows:

$$\min \text{COST}(z, b^B)$$

where

$$\text{COST}(z, b^B) = \sum_{\ell \in L} \left(b_{\ell}^B + \sum_{c \in C} p_{c\ell}^W z_c \right). \quad (4.1)$$

In order to satisfy all the demands, we have defined following constraints:

$$\sum_{c \in C_v} z_c \geq D_v \quad v \in V^s. \quad (4.2)$$

Note that the demand of requests originating at $v = v^s$ is not necessarily satisfied by a single configuration.

The next set of constraints expresses the capacity requirement for link ℓ' in a backup path. Indeed, if ℓ' protects link ℓ , with ℓ belonging to several working paths (modeled here throughout the various configurations associated with working paths containing ℓ), we must ensure that ℓ' has a large enough transport capacity:

$$\sum_{c \in C} p_{c\ell}^W p_{c\ell'}^B z_c \leq b_{\ell'}^B \quad \ell, \ell' \in L : \ell \neq \ell'. \quad (4.3)$$

4.2.2 Pricing Problem

Each Pricing Problem (PP_{v^s}) corresponds to the design of a potential configuration, i.e., a potential working and backup provisioning for the job requests originating from a given source node $v^s \in V^s$. Per definition of the pricing problem, the objective function corresponds to the reduced cost of the configuration variable of the master problem, i.e., of variable z_c for $c \in C_v$, assuming we search for configurations in C_v .

In addition, the interest of the pricing problem lies in the identification of improving configurations, i.e., configurations c such that, if their corresponding variable z_c is added in the restricted master problem, it will contribute to improve (here, to minimize further) the current value of the objective of the restricted master problem. Such configurations are the ones with a negative reduced cost. In other words, assuming we minimize the reduced cost of the current pricing problem associated with source node v^s , either the minimum reduced cost is negative, and then we have obtained an improving configuration that we add to the current restricted master problem, or the minimum reduced cost is positive. In the latter case, we conclude that, at this stage, no more improving configuration associated with v^s can be found, unless the values of the dual variables change following the addition of another configuration associated with another source node.

The PP_{v^s} contains 5 sets of variables, described as follows:

$$p_\ell^W = 1 \text{ if link } \ell \text{ is used by the working path, } 0 \text{ otherwise.}$$

$$p_\ell^B = 1 \text{ if link } \ell \text{ is used by the protection path, } 0 \text{ otherwise.}$$

$p_{\ell\ell'}^{\text{WB}}$ = 1 if backup path link ℓ' protects working path link ℓ , 0 otherwise.

d_v^{W} = 1 if server node $v \in V^{\text{SVR}}$ is a destination node of the working path,
0 otherwise.

d_v^{B} = 1 if server node $v \in V^{\text{SVR}}$ is a destination node of the backup path,
0 otherwise.

Let us express the objective function of the pricing problem associated with source node v^s , or PP_{v^s} for short, i.e., the reduced cost of variable $z_c, c \in C_v$. For doing so, we need the dual values of the constraints involving variable z_c :

$u^{(4.2)} \geq 0$, value of the dual variable associated with constraint (4.2)- v^s ,

$u^{(4.3)} \leq 0$, values of the dual vector associated with constraints (4.3).

The reduced cost, $\overline{\text{COST}}$, of PP_{v^s} , to be minimized, can then be written as follows:

$$\overline{\text{COST}} = \sum_{\ell \in L} p_{\ell}^{\text{W}} - u_{v^s}^{(4.2)} - \sum_{\ell \in L} \sum_{\ell' \in L: \ell \neq \ell'} u_{\ell\ell'}^{(4.3)} p_{\ell}^{\text{W}} p_{\ell'}^{\text{B}}. \quad (4.4)$$

First two sets of constraints are related to the working and backup provisioning of the job requests originating from v^s , which take care of the working and backup path definitions.

$$\sum_{\ell \in \omega^+(v)} p_\ell^W - \sum_{\ell \in \omega^-(v)} p_\ell^W = \begin{cases} +1 & \text{if } v = v^S \\ -1 & \text{if } v = v^{SVR} \\ 0 & \text{otherwise} \end{cases} \quad v \in V, \quad (4.5)$$

$$\sum_{\ell \in \omega^+(v)} p_\ell^B - \sum_{\ell \in \omega^-(v)} p_\ell^B = \begin{cases} +1 & \text{if } v = v^S \\ -d_v^B & \text{if } v \in V^{SVR} \\ 0 & \text{otherwise} \end{cases} \quad v \in V. \quad (4.6)$$

The next two sets of constraints deal with the overlap and the sharing of links pertaining to the working and backup paths: link disjoint.

$$p_\ell^W + p_\ell^B \leq 1 \quad \ell \in L \quad (4.7)$$

$$p_\ell^W + p_{\ell'}^B \leq 1 \quad \ell, \ell' \in L : \quad (\ell \text{ and } \ell' \text{ are opposite to each other}). \quad (4.8)$$

If we consider CSP, we need the following constraints:

$$d_v^B = \begin{cases} 1 & v \text{ is the primary server (i.e., given destination)} \\ 0 & \text{else} \end{cases} \quad (4.9)$$

when we consider SPR, we replace constraints (4.9) with the following constraints:

$$\sum_{v \in V^{\text{svr}}} d_v^{\text{B}} = 1. \quad (4.10)$$

As can be observed, the expression of the reduced cost (4.4) is nonlinear. In order to linearize it, we introduce the variables $p_{\ell\ell'}^{\text{WB}} \in \{0, 1\}$ such that:

$$p_{\ell\ell'}^{\text{WB}} = p_{\ell}^{\text{W}} p_{\ell'}^{\text{B}}, \quad p_{\ell}^{\text{W}}, p_{\ell'}^{\text{B}} \in \{0, 1\}, \quad \ell, \ell' \in L : \ell \neq \ell',$$

and add the following three sets of constraints.

$$p_{\ell\ell'}^{\text{WB}} \geq p_{\ell}^{\text{W}} + p_{\ell'}^{\text{B}} - 1 \quad \ell', \ell \in L : \ell \neq \ell' \quad (4.11)$$

$$p_{\ell\ell'}^{\text{WB}} \leq p_{\ell}^{\text{W}} \quad \ell', \ell \in L : \ell \neq \ell' \quad (4.12)$$

$$p_{\ell\ell'}^{\text{WB}} \leq p_{\ell'}^{\text{B}} \quad \ell', \ell \in L : \ell \neq \ell'. \quad (4.13)$$

After adding the linearize constraints, the expression of the objective (i.e., reduced cost) of the pricing problem PP_{v^s} becomes:

$$\overline{\text{COST}} = \sum_{\ell \in L} p_{\ell}^{\text{W}} - u_{v^s}^{(4.2)} - \sum_{\ell \in L} \sum_{\ell' \in L : \ell \neq \ell'} u_{\ell\ell'}^{(4.3)} p_{\ell\ell'}^{\text{WB}}. \quad (4.14)$$

All aforementioned constraints define the CSP and the SPR case, now let us modify above model for the CSP-A and the SPR-A. The change will occur only in the pricing problem. The flow constraints (4.5) will replace with (4.15), and remove

the constraints (4.9). We have to also add the single working path destination by adding the constraints (4.16). In case of CSP-A, the working and backup paths destination must be same, and can be enforced by constraints (4.17).

$$\sum_{\ell \in \omega^+(v)} p_\ell^W - \sum_{\ell \in \omega^-(v)} p_\ell^W = \begin{cases} +1 & \text{if } v = v^S \\ -d_v^W & \text{if } v \in V^{\text{SVR}} \\ 0 & \text{otherwise} \end{cases} \quad v \in V \quad (4.15)$$

$$\sum_{v \in V^{\text{SVR}}} d_v^W = 1, \quad (4.16)$$

$$d_v^B = d_v^W \quad v \in V^{\text{SVR}}. \quad (4.17)$$

In addition to CG-ILP, we have also developed heuristic algorithm, H2, in an attempt to design a more scalable heuristic algorithm than heuristic proposed in [28] (we refer here as H1). As we will see in Section 4.3.1, we were quite successful in that attempt for the scalability aspect, less for the accuracy part. A key difference between H1 heuristic and H2 heuristic is that in H2, we combine all the requests originating from the same source node, as in the master problem of CG-ILP, while in H1, requests are considered on an individual basis, which increases the complexity of H1.

Shortest paths are computed using different weights for primary and backup

path calculation. Backup weights account for sharing of wavelengths, while working weights account for the length of the path only:

$\text{WEIGHT}_\ell^{\text{W}}$: Primary weights are all taken equal to one, meaning that when computing shortest paths with those weights, we indeed consider the length of the working paths in terms of the number of links they contain.

$\text{WEIGHT}_\ell^{\text{B}}$: Backup weights are initialized to one, and will contain the complement of the protection bandwidth requirements with respect to the maximum link bandwidth requirement, see line 8. The reason is as follows. When computing shortest paths, we can either minimize or maximize their overall bandwidth requirements. When maximizing, instead of changing the shortest path algorithm in a longest path algorithm, one can also complement the protection weights with respect to the largest weight in order to go on using a shortest path algorithm (this is what is done on line 8 of the heuristic).

The underlying idea of the definition of the weights for the search of the backup path is that there are more opportunities for sharing with the links already contributing to bandwidth protection, or, in other words, the more protection bandwidth a link has, the more protection bandwidth sharing the link offers.

4.2.3 Solution of the CG-ILP formulation

Column Generation (CG) techniques offer highly efficient solution methods for linear programs with a very large number of variables, where the constraints can be

Algorithm 4.1 Heuristic H2 - SPR-A Protection Scheme

- 1: **Step 1:** *Initialization*
- 2: For all $\ell \in L$: $b_\ell^B \leftarrow 0$; $\text{WEIGHT}_\ell^W \leftarrow 1$,
- 3: **Step 2:** *Primary and backup paths*
- 4: **for all** $v^S \in V \setminus V^{\text{SVR}}$ **do**
- 5: Concatenate all the requests originating at v^S into a single aggregated request, denoted by $k(v^S)$, with a bandwidth requirement such that: $b_{k(v^S)} = \sum_{k \in K_s} b_k$.
- 6: **Step 2a:** *Selection of the grid server location*
- 7: **for all** $\ell \in L$ **do**
- 8: $\text{WEIGHT}_\ell^B \leftarrow \left(\max_{\ell \in L} b_\ell^B \right) - b_\ell^B + 1$
- 9: **end for**
- 10: **for all** $v^{\text{SVR}} \in V^{\text{SVR}}$ **do**
- 11: Compute the shortest path $p_{v^S v^{\text{SVR}}}$ from v^S to v^{SVR} with weights WEIGHT^W
- 12: **end for**
- 13: $p_s^W \leftarrow \arg \min_{v^{\text{SVR}} \in V^{\text{SVR}}} \{ \text{LENGTH}(p_{v^S v^{\text{SVR}}}) \}$ where $\text{LENGTH}(p_{v^S v^{\text{SVR}}})$ is computed according to WEIGHT^W
- 14: **Step 2b:** *Tentative selection of the primary path*
- 15: Temporarily remove from G the links of p_s^W
- 16: **Step 2c:** *Selection of the backup path and confirmation/new computation of the primary path*
- 17: **if** there exists a path from v to a server site **then**
- 18: For all $v^{\text{SVR}} \in V^{\text{SVR}}$: Compute the shortest path $p_{v^S v^{\text{SVR}}}$ from v^S to v^{SVR} with weights WEIGHT^B
- 19: $p_s^B \leftarrow \arg \min_{v^{\text{SVR}} \in V^{\text{SVR}}} \{ \text{LENGTH}(p_{v^S v^{\text{SVR}}}) \}$ where $\text{LENGTH}(p_{v^S v^{\text{SVR}}})$ is computed according to WEIGHT^B
- 20: Restore graph G (put back all links)
- 21: **else**
- 22: Restore initial graph G (put back all links)
- 23: Compute the shortest pair of link disjoint paths between v^S and v^{SVR} with weights WEIGHT^W and WEIGHT^B , for all $v^{\text{SVR}} \in V^{\text{SVR}}$.
- 24: Let p' and p'' be the two resulting routes. Let

$$p_s^W = \arg \min \{ \text{LENGTH}(p'), \text{LENGTH}(p'') \};$$

$$p_s^B = \arg \max \{ \text{LENGTH}(p'), \text{LENGTH}(p'') \}.$$
- 25: **end if**
- 26: Update the bandwidth requirements (b_ℓ^W and b_ℓ^B) on the links of the primary and backup paths. For b_ℓ^B , the updating formula is as follows:

$$b_\ell^B = \max_{\ell' \in L} \left(\sum_{k \in K: \ell' \in p_k^W, \ell \in p_k^B} b_k \right),$$

where p_k^W (resp. p_k^B) is the aggregated working (resp. backup) path of request k .

- 27: **end for**
-

expressed implicitly. In order to satisfy the demand constraint in the master problem, we have to generate few as promising as possible configurations at the outset, referred to as Restricted Master problem (RMP). This was achieved by solving PP_{v^s} for $v^s \in V^s$, after modifying its objective as follows:

$$\min \sum_{\ell \in L} (p_\ell^W + p_\ell^B). \quad (4.18)$$

The set of constraints is made of constraints (4.5)-(4.17) except the linearization constraints (4.11) to (4.13), depending on the protection scheme and their associated constraints. The detail of the CG-LP and ILP solution process is described in Algorithm 4.2.

Obtaining Integer Solutions Once the linear relaxation of the restricted master problem has been solved (upto Step 2 of Algorithm 4.2), one needs to derive an integer solution (Step 3 of Algorithm 4.2). In order to get an exact integer solution, one would need to use a branch-and-price method [5], which usually turns out to be a non scalable solution process.

We therefore propose to solve the ILP made only of the columns generated in order to reach the optimal solution of the linear relaxation of the restricted master problem, using the ILP solver of CPLEX. Let z_{LP}^* (resp. \tilde{z}_{ILP}) be the optimal value of the linear relaxation of the restricted master problem (resp. the value of the integer

Algorithm 4.2 Solution of the CG-ILP model

Step 1. *Initialization*

Build a set of initial configurations in order to set an initial Restricted Master Problem (RMP).

Step 2. *Solution of the linear relaxation of the master problem*

Solve the LP relaxation of the current RMP

OPT \leftarrow .FALSE.

while OPT = FALSE **do**

 OPT \leftarrow .TRUE.

for each source node v^s **do**

 Solve PP_{v^s}

if $\overline{\text{COST}}(PP_{v^s}) < 0$ **then**

 OPT \leftarrow .FALSE.

 Add the improving configuration associated with PP_{v^s} to the current RMP

 Re-optimize the LP relaxation of the enlarged RMP

end if

end for

end while

Step 3. *Deriving an optimal or a near optimal integer solution*

Solve the ILP model made of the current set of columns (variables) of the RMP, using either a branch-and-bound technique or a rounding off technique.

solution obtained using the above described process). Then, the optimality gap

$$\text{GAP} = \frac{\tilde{z}_{\text{ILP}} - z_{\text{LP}}^*}{\tilde{z}_{\text{ILP}}}$$

measures the accuracy of the integer solution. In practice, for the reported results, we observed less than 1% gap, i.e., the integer solution is indeed an optimal one or near to optimal.

4.3 Performance Evaluation of min-ACRWA

A performance evaluation of min-ACRWA model has been conducted on different European network topologies, comprising of 28 nodes and different numbers of bidirectional links (EU-base, EU-dense, and EU-sparse topologies) , as depicted in Figure 4.1.

We have considered the following four protection schemes under single link failures scenario:

CSP Classical Shared Path protection, i.e., the working and backup path has same fixed (given) destination.

SPR Shared Path protection with Relocation, i.e, the working path destination is fixed (given) and backup path can be any server node ($v^{\text{SVR}} \in V^{\text{SVR}}$).

CSP-A Classical Shared Path protection with Anycast, i.e., the destination can be any server node but the same for both working and backup paths.

SPR-A Shared Path protection with Relocation and Anycast, i.e., the destination for both paths can be two server nodes (not necessarily the same).

Traffic instances were generated as follows: for a given number, say $|K|$, of job requests, we randomly select $|K|$ pair of (source & destination) nodes $v^s \in V \setminus V^{\text{SVR}}$ and $v^{\text{SVR}} \in V^{\text{SVR}}$ for CSP and SPR case (for CSP-A and SPR-A only the source node is selected). The number of times a source node is selected gives the number of job requests originating from that node. Nodes which are hosting server nodes are

excluded.

We consider different sets of fixed server nodes: $V^{\text{svr}^3} = \{\text{London, Vienna, Berlin}\}$, $V^{\text{svr}^5} = V^{\text{svr}^3} \cup \{\text{Lyon, Zurich}\}$, $V^{\text{svr}^7} = V^{\text{svr}^5} \cup \{\text{Munich, Zagreb}\}$.

We use the IBM ILOG CPLEX solver (release 11) to solve the ILP models under a C++ implementation. All programs have been run on a cluster server node with 1 CPU of 2.2 GHz AMD Opteron 64-bit processor, 8GB RAM. In the forthcoming figures, each data point corresponds to average results over 10 random traffic instances.

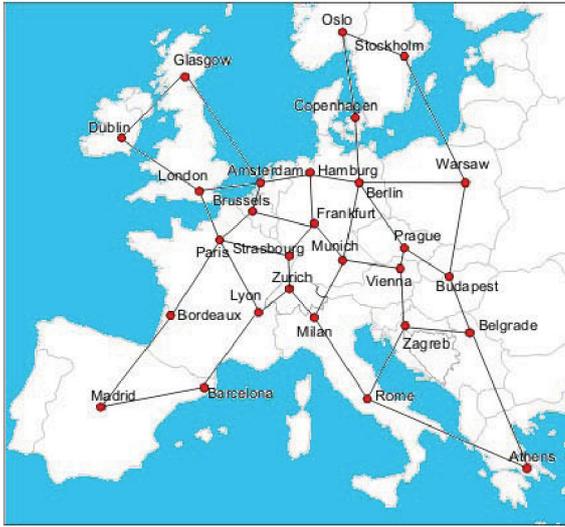
4.3.1 Quality of the Solutions

First, we discuss the case where source and destination is given in the CSP and SPR protection schemes, and then CSP-A and SPR-A where only source node is given.

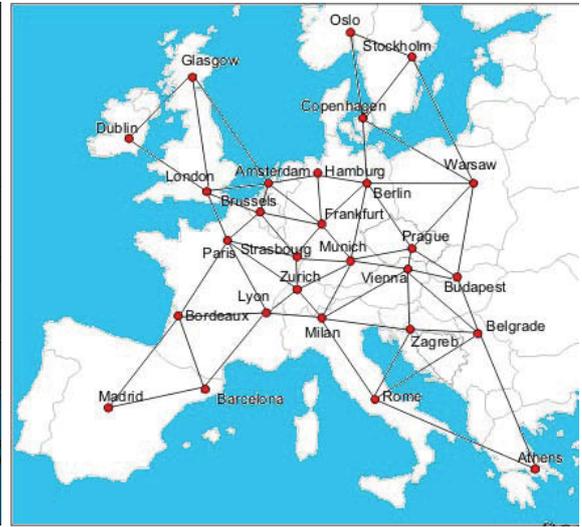
CSP and SPR

We have compared our CG-ILP formulation with classical ILP [6], and heuristic [7] on European network topology considering different protection schemes under single link failure. Before we go into the details of the comparative performances of the different methods on large demand instances, we first evaluate the results on small instances. In Figure 4.2 and Figure 4.3, we plotted the total number of wavelengths, which the different methods output for the optimized capacity value, with a number of requests varying from 5 to 20. In all plots, ILP refers to the ILP model of [6], [28], and CG to the CG-ILP model of Section 4.2, and Heuristic to the heuristic of [28].

For demand sets with more than 11 demands, difficulties start to appear when it



(a) EU-base Network Topology



(b) EU-dense Network Topology



(c) EU-sparse Network Topology

Figure 4.1: European network topologies.

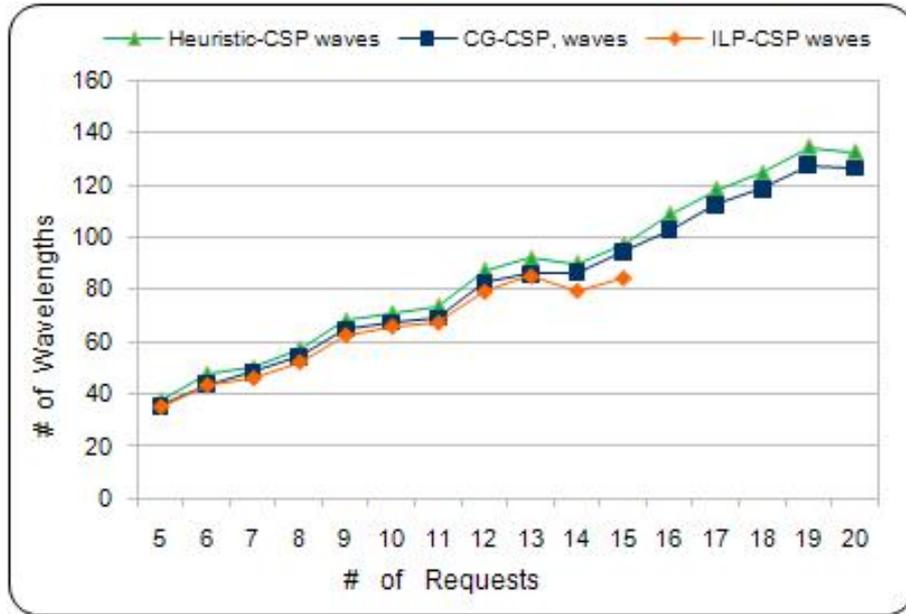


Figure 4.2: CSP: Total number of wavelengths.

comes to solving the classical ILP model for the CSP scheme, in a reasonable time frame. Indeed, out of the 10 solved instances, there were always one or two instances which could not be solved within the 72 hours time limit we set ourselves. This is why for demand set with more than 15 requested connections, we did not use the classical ILP model anymore.

The heuristic performs quite well, but with comparable computing times, and solutions of slightly inferior quality than CG, when the number of requests remain small. We calculated the average gap for the request size range 5 to 13 (since the ILP average does not include all 10 instances for demands beyond 13 requests). With respect to the comparison between the two protection schemes, there is only a gap of 2.46% in the CSP and 1.11% in the SPR case if we compare the values of the ILP with the CG solutions. The heuristic generates inferior results compared to ILP:

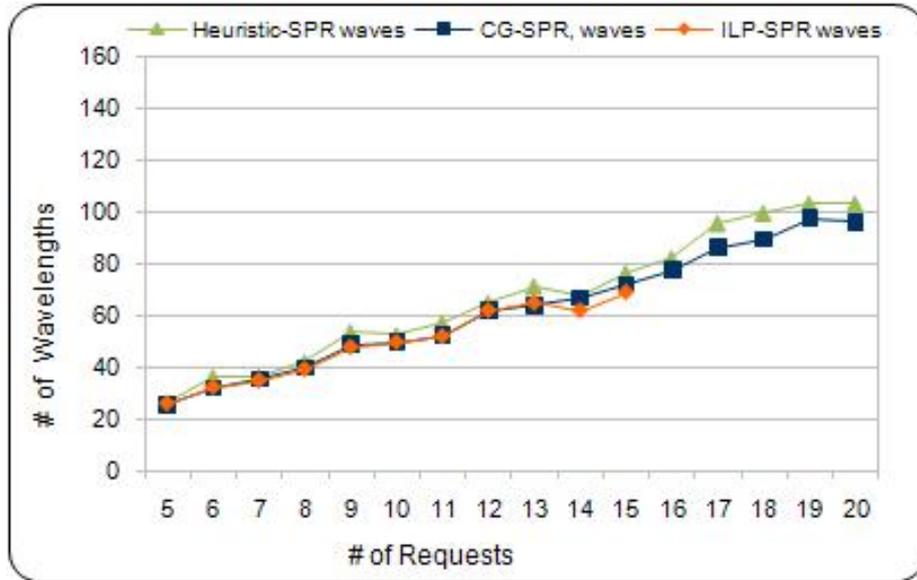


Figure 4.3: SPR: Total number of wavelengths.

On average, a difference of 8.63% for CSP, 8.15% for SPR. Comparing the results generated by the CG method and the heuristic, we come to the conclusion that the gap between their optimized solutions remains fairly constant: For CSP, there is a difference of 5.48% and for SPR, it is 5.71%. This leads us to the conclusion that the CG has an output which estimates the optimal output very well and the heuristic has suboptimal solutions, which are of satisfactory quality.

For large size instances, the trend is fairly similar as shown in Figure 4.4 where we plotted the total number of wavelengths for the demand sets with 50 to 300 requested connections. We ascertain that the difference between the total number of wavelengths for the heuristic and CG averages to 4.99% for the CSP case and 6.92% for SPR.

As a last observation, note that the conclusions made in [6] and [7] are confirmed

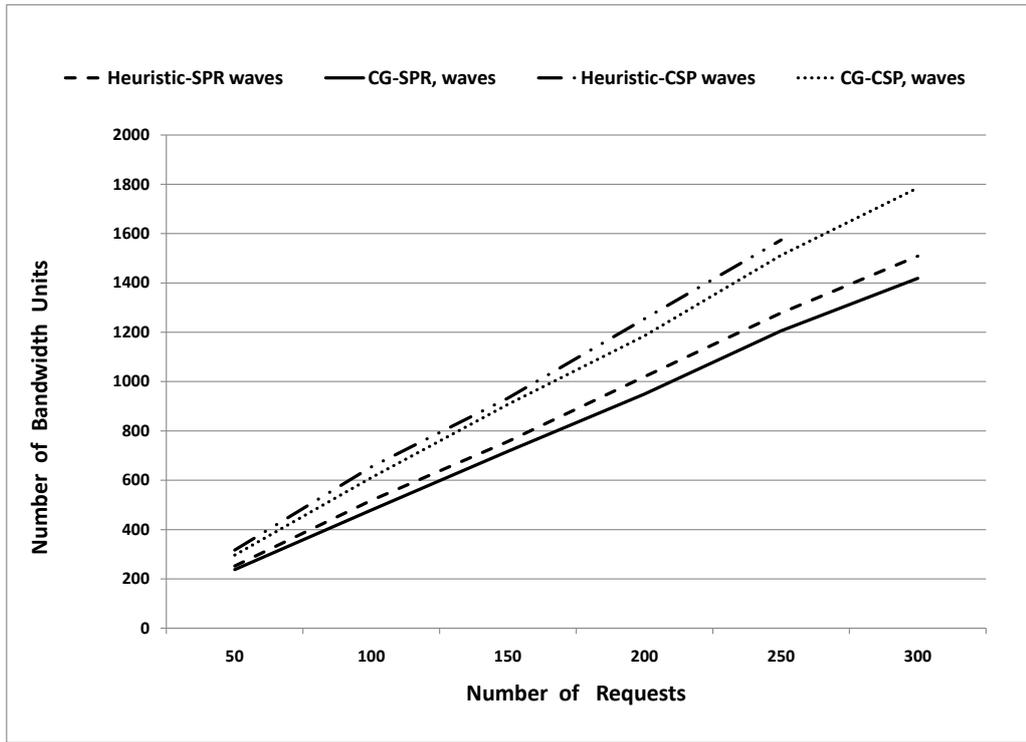


Figure 4.4: CSP & SPR: Total number of wavelengths for large instances.

for large traffic demand instances: relocation impacts the network dimension by introducing a network load reduction (NLR). Here, it amounts to $\pm 22\%$, independently of the requested number of connections.

Although we did not develop a branch-and-price algorithm for solving exactly the CG-ILP model, we get fairly small optimality gaps (difference between the value of the incumbent solution¹ and the lower bound provided by the LP solution of the last generated RMP), i.e., smaller than 1% on average, for the solutions output by the CG-ILP model.

¹Optimal ILP solution of the last RMP of the column generation algorithm.

CSP-A and SPR-A

Next, we are comparing the CSP-A with SPR-A protection schemes, and again on European EU-base (Figure 4.1(a)) topology with 5 server resource locations ($V^{\text{svr}5}$). We will also compare the different number of resources such as 3, 5, and 7 to analyze the impact of various number of resources. Finally, we will compare the three different network topologies (Figure 4.1) to see the impact of different number of links.

If we now look at the CSP-A and SPR-A protection schemes, where the server location is not given at the outset. We have compared here CG-ILP with heuristic H1 [7], and we have also developed heuristic H2 (see Algorithm 4.1), the results are described in Figures 4.5 and 4.6. We have noted that CG-ILP has an optimality gap $< 0.5\%$ which means we get optimal solutions from a practical point of view. In both figures, we provide the relative performances of the two heuristics, H1 and H2, with respect to CG-ILP. The relative optimality gaps are computed as follows:

$$\frac{\text{COST}_{\text{H1}}^* - \text{COST}_{\text{CG-ILP}}^*}{\text{COST}_{\text{CG-ILP}}^*} \quad \text{and} \quad \frac{\text{COST}_{\text{H2}}^* - \text{COST}_{\text{CG-ILP}}^*}{\text{COST}_{\text{CG-ILP}}^*},$$

where COST_{\square} denotes the cost value found by the \square model/algorithm. Comparisons are made in Figure 4.5 for the CSP-A protection scheme, and in Figure 4.6 for the SPR-A protection scheme. The key observations are that the H1 heuristic provides better solutions than the H2 heuristic, but at the expense of longer computing times, as discussed below. Indeed, for both protection schemes, the H1 heuristic provides solutions with an average of 5% accuracy, compared to the CG-ILP solutions, while

the relative accuracy varies between 10% and 20% for the H2 heuristic.

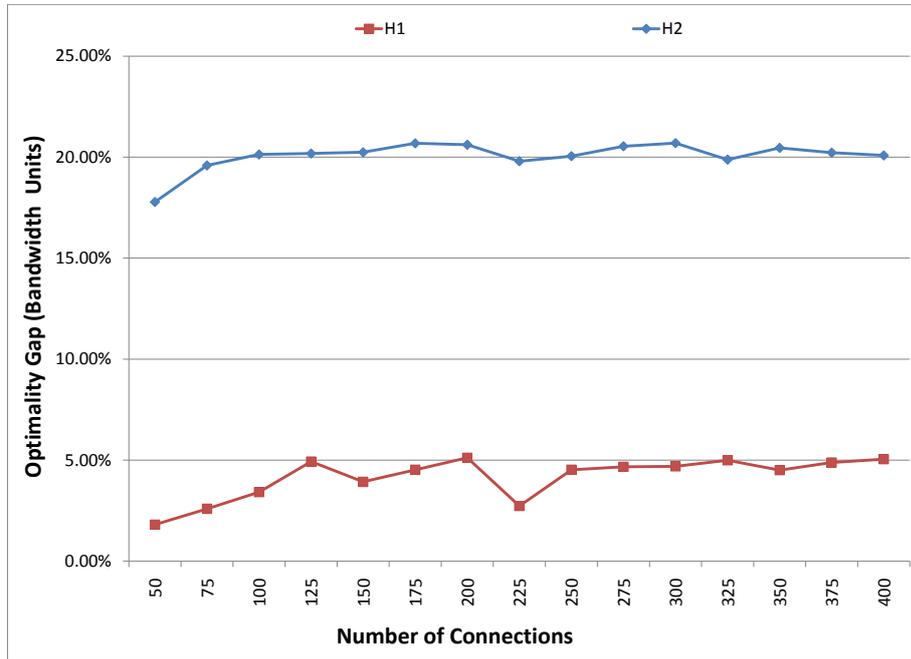


Figure 4.5: Performances of H1 and H2 compared to CG-ILP under CSP-A.

We observe that both CG-ILP and H2 algorithms are not sensitive to the number of requests, with H2 being much faster than CG-ILP, as shown in Figure 4.7. On the other hand, H1 is increasing with the number of requests, and when the number of requests exceeds 500, H1 has higher computing times than CG-ILP. As shown by the results depicted in Figure 4.7, H1 provides better solutions than H2. However, when accuracy is not a major concern, but routes need to be found very fast, H2 is an interesting alternate choice and scales to very large demand sets.

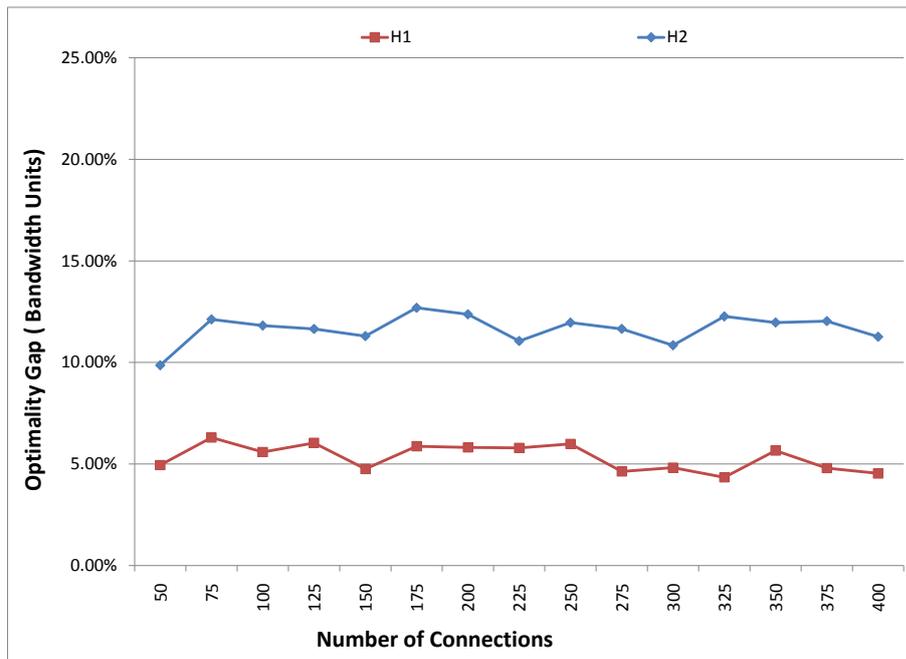


Figure 4.6: Performances of H1 and H2 compared to CG-ILP under SPR-A.

4.3.2 Influence of the Number of Server Sites and the Topology

Number of servers We compare here the performances of the CG-ILP algorithm with different numbers of resources (server nodes): 3, 5, and 7. Running time results are shown in Figure 4.8 (resp. 4.9) for the CSP-A (resp. SPR-A) protection scheme.

We observe, that for the CSP-A scheme, computing times are higher for 5 server locations than for 3, while computing times for 3 are higher than those for 7 server locations. For the SPR-A scheme, the running times with 3 server nodes are higher than with 5, and running times with 5 server nodes are higher than those with 7 server locations. We made experiments with another data set, where the Berlin server was relocated in Copenhagen. Again, the results (not shown here) gave similar trend of

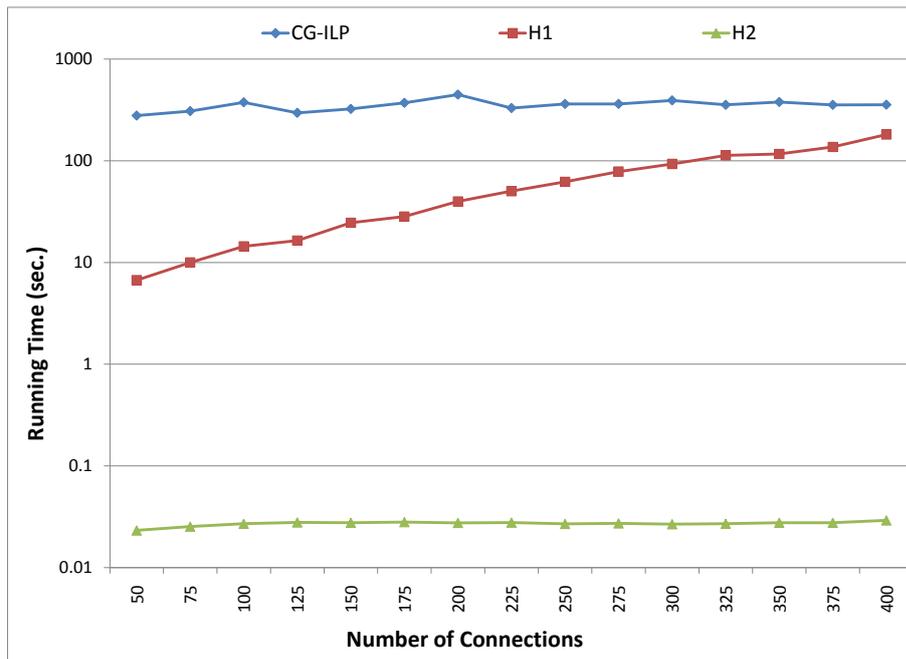


Figure 4.7: Running time for SPR-A protection scheme.

running times. Therefore, from the two case studies, no clear trend can be observed in run-time dependency on the number of server sites.

Impact of the topology connectivity We next analyze the effect of the topology. For doing so, we considered the European networks comprising the same number of nodes, but with different number of links (i.e., connectivity). We again considered the case for 5 server sites. Consequently, we investigate the performance of algorithm CG-ILP on the 3 topologies of the European network (see Figure 4.1) described at the beginning of Section 4.3: EU-base, EU-dense, EU-sparse with an average node degree of 2.93, 4.21, and 2.5 respectively. Contrarily to the number of server sites, the topology seems a lot more influential, where a highly meshed network severely penalizes the execution time for CG-ILP, as observed in Figure 4.10. This was to be

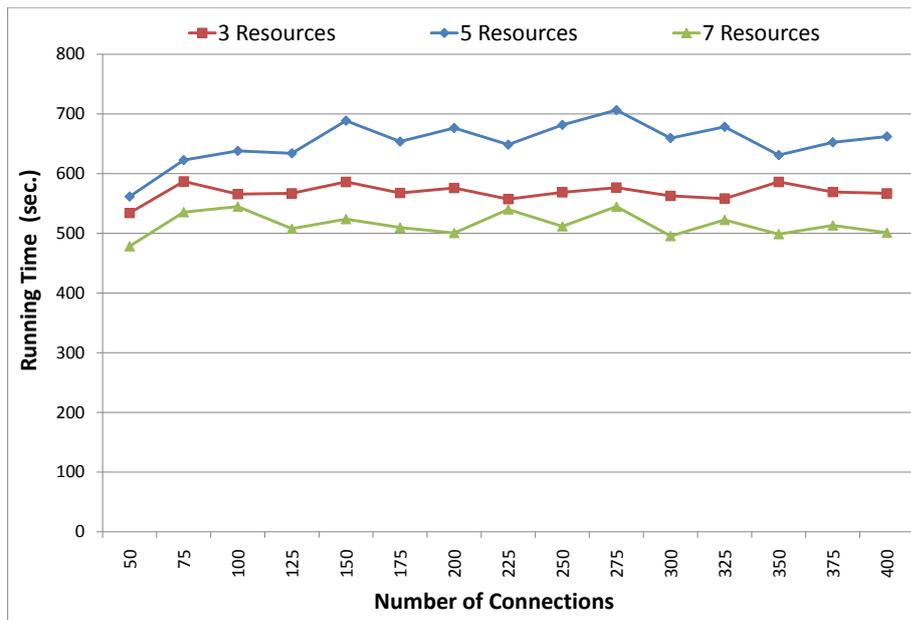


Figure 4.8: Comparison of the running times for different numbers of server nodes on the EU-base topology (CG-ILP algorithm) under CSP-A scheme.

expected, since the number of possible paths increases.

Bandwidth savings by exploiting relocation Lastly, we compared the bandwidth requirements of CSP-A and SPR-A, depending on the number of server nodes and the network topology. In Figure 4.11, we plotted the bandwidth savings that result from using the SPR-A scheme rather than the CSP-A scheme, using the ratio $(\text{bandwidth (CSP-A)} - \text{bandwidth (SPR-A)}) / \text{bandwidth (SPR-A)}$. In all cases, there are meaningful bandwidth savings, which is rather stable with the number of job requests (experiments have been conducted for 50 up to 400 requests). On average, it is around 13% for 3 and 5 servers, and increases to around 21% for 7 servers. Indeed, the more servers, the more flexibility for an anycast scheme. With respect to the impact of the topology, the trend is as expecting, more bandwidth savings as the

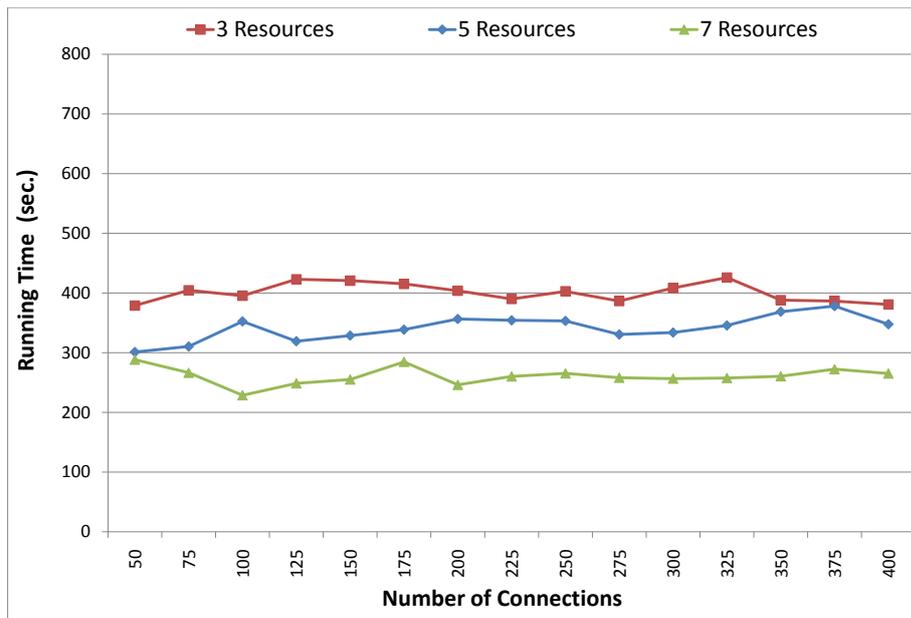


Figure 4.9: Comparison of the running times for different numbers of server nodes on the EU-base topology (CG-ILP algorithm under SPR-A).

density is decreasing (see Figure 4.12) , i.e., bandwidth savings go from an average of 7% on a EU-dense topology, to an average of 13% for the EU-base topology, and then to above 21% for the EU-sparse topology.

4.4 Data Sets

In this study, we experimented four protection schemes (CSP, SPR, CSP-A, SPR-A) with different topologies (Eu-base, Eu-sparse, Eu-dense), and three sets of server nodes (data centers), i.e., V^3 , V^5 , and V^7 . We concluded that our CG-ILP formulations saves upto 20% bandwidth units from classical unicast routing to anycast routing. We have also compared CG-ILP formulations with heuristic and it improves upto 10% bandwidth units.

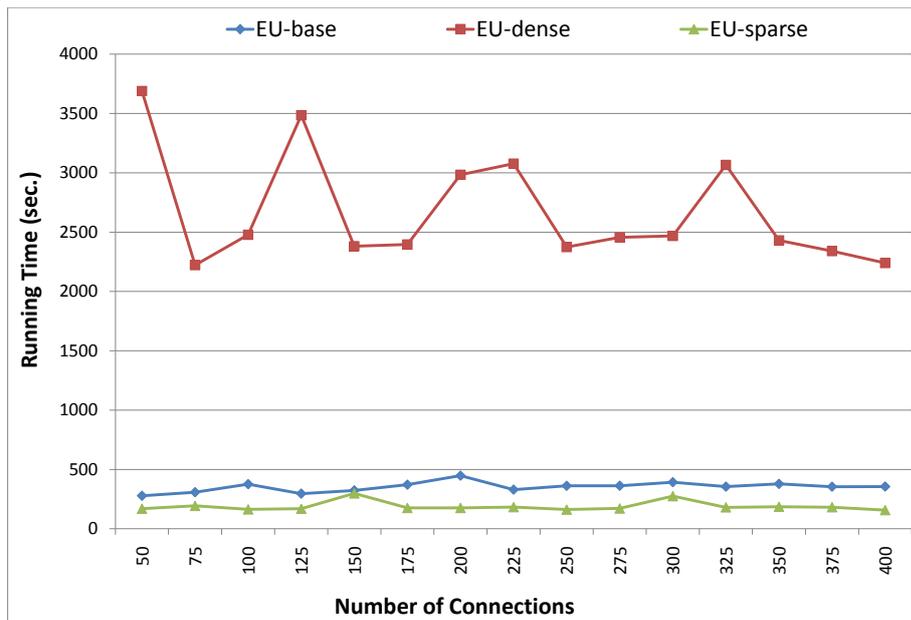


Figure 4.10: Impact of the topology connectivity (CG-ILP algorithm): Running times for the SPR-A protection scheme.

4.5 Shared Risk Link Group (SRLG)

We have studied the survivable optical grid network under single link failure. However, there may exist a bundle of fibers (links) which are buried together or in the same duct or pass through the same bridge [73]. Thus, these links may fail at the same time.

A subset of links which has the same risk of failures is known as Shared Risk Link Group (SRLG). SRLG can also be applied for double link failures or any other subset of links / nodes sharing a common risk, called general SRLG [66].

In order to provide the survivability under SRLG failures, the pricing problem of min-ACRWA model can be modified as follows. Let L^{SRLG} be a SRLG set of link, and \mathcal{L} denotes the set of SRLG sets. Now, if we want to add the survivability under SRLG

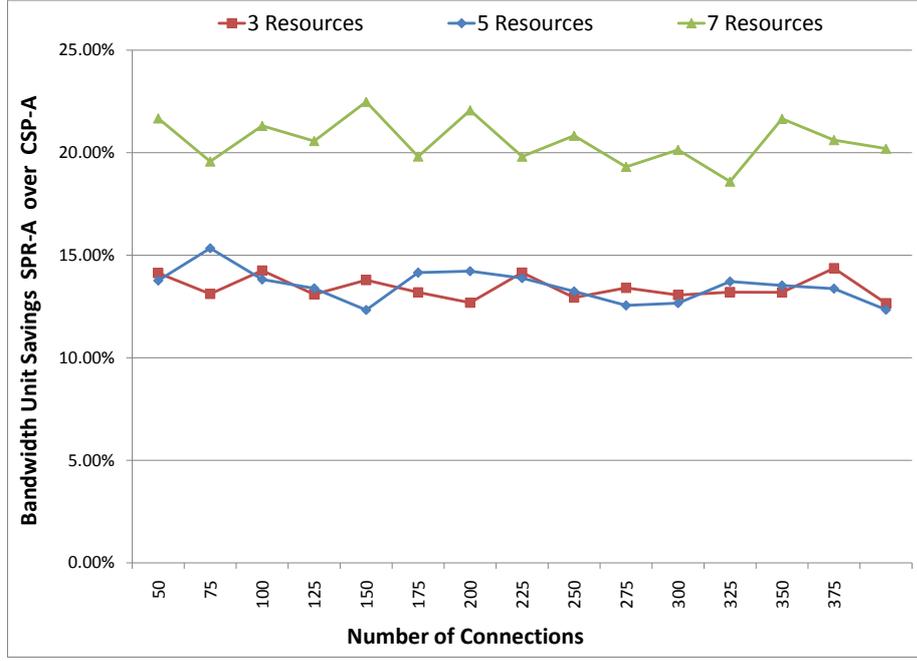


Figure 4.11: SPR-A vs. CSP-A protection schemes depending on the # of server nodes on the EU-base topology with respect to the number of bandwidth units – $(\text{CSP-A} - \text{SPR-A}) / \text{CSP-A}$.

failure then constraints (4.7) and (4.8) will be replaced by the constraints (4.19) and (4.20).

$$\sum_{\ell \in L^{\text{SRLG}}} p_{\ell}^{\text{W}} + \sum_{\ell \in L^{\text{SRLG}}} p_{\ell}^{\text{B}} \leq 1 \quad L^{\text{SRLG}} \in \mathcal{L} \quad (4.19)$$

$$\sum_{\ell \in L^{\text{SRLG}}} p_{\ell}^{\text{W}} + \sum_{\ell' \in L^{\text{SRLG}}} p_{\ell'}^{\text{B}} \leq 1 \quad L^{\text{SRLG}} \in \mathcal{L} : (\ell \text{ and } \ell' \text{ are opposite to each other}). \quad (4.20)$$

We have developed the min-ACRWA formulation with single link failure (in Section 4.2), Now we are discussing the single node failure. There are two ways to make survivable optical grids under node failure: the first one is to provide protection on

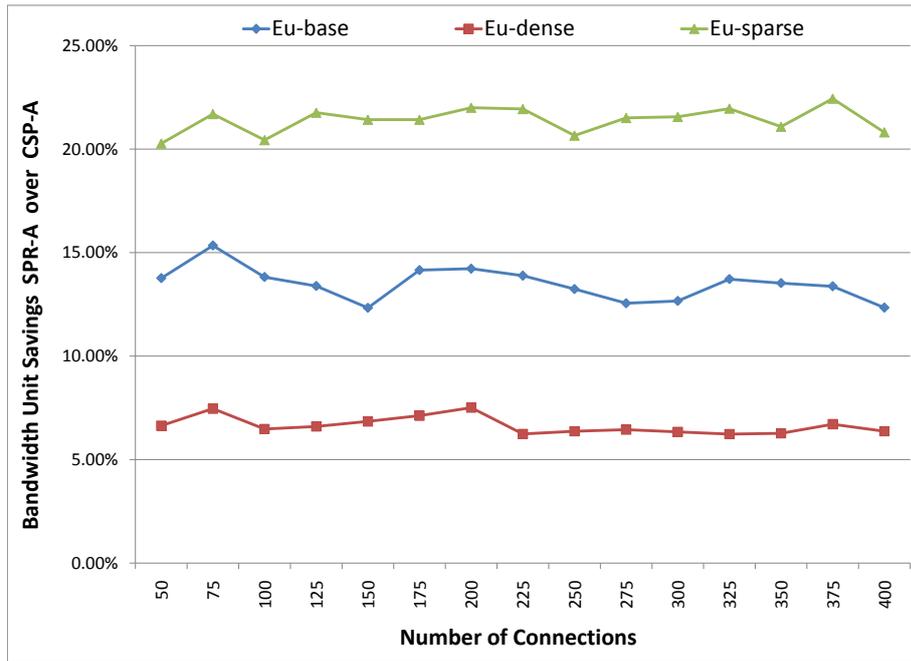


Figure 4.12: SPR-A vs. CSP-A protection schemes depending on the density of the network topology with 5 server sites with respect to the number of bandwidth units – $(\text{CSP-A} - \text{SPR-A}) / \text{CSP-A}$.

intermediate nodes only, and the alternative is to provide protection on intermediate nodes as well as on the destination node (i.e., server node). This alternative method leads to the protection of network resources and computing resources. Note that node failure also covers the link failure.

In order to provide intermediate node protection in the previous model (Section 4.2), the master problem will remain the same, and the link disjoint constraints (4.7) and (4.8) have to be replaced by the following three sets of constraints in the pricing

problem.

$$\sum_{\ell \in \omega^+(v)} p_\ell^W + \sum_{\ell \in \omega^+(v)} p_\ell^B \leq 1 \quad v \neq v^S, v \notin V^{\text{SVR}} \quad (4.21)$$

$$\sum_{\ell \in \omega^-(v)} p_\ell^W + \sum_{\ell \in \omega^-(v)} p_\ell^B \leq 1 \quad v \neq v^S, v \notin V^{\text{SVR}} \quad (4.22)$$

$$p_\ell^W + p_{\ell'}^W + p_\ell^B + p_{\ell'}^B \leq 1 \quad \ell \in \omega^{(v)}, v = v^S \quad (4.23)$$

(ℓ and ℓ' are opposite to each other).

If we want to include the server node failure, then we have to add constraints (4.24) in the case of SPR-A protection scheme.

$$d_v^W + d_v^B \leq 1 \quad v \in V^{\text{SVR}}. \quad (4.24)$$

Chapter 5

Optimization Models for max-ACRWA

An optical grid network provides high speed communications for large scale applications and services may require an ultra-high bit rate network services at the order of the transmission capacity of the network infrastructures. In the context of resilient optical grids, we investigate how to maximize the grade of services for given transport capacities, while maximizing the protection level.

This chapter presents scalable optimization models, solved with the help of Column Generation (CG) technique, for maximizing IT services under limited link transport capacities. We assume the use of the anycast routing principle to identify the server nodes for executing the jobs, and a shared path protection mechanism in order to offer protection against single link/node failures. We also investigate different calculation methods of the link transport capacities in order to maximize the grade

of services, while taking into account the bandwidth requirements.

Computational results are presented on different traffic distributions. They show that the proposed link dimensioning can save more than 35% bandwidth in optical grid networks, in comparison with the classical link dimensioning strategies. We also investigate the different protection schemes against single link failures, single node failures, single server node failures, single node and server node failures. Further, we compare their bandwidth requirements, as well as their impact on the grade of services (GoS).

Results show that there is no significant increase of the bandwidth requirements and no meaningful impact on the GoS when moving from a single link protection scheme to a single node (including server nodes) protection scheme. This work is published in “Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), 2011” [34] and an extension of that work has been published in the journal of “Telecommunication Systems” [59].

5.1 Optimization Models for Different Levels of Protections

Again, the original optimization model is decomposed around two problems, solved alternatively and in sequence in a column generation framework, One of them is the so-called restricted master problem which selects the best configurations, among the set of already generated configurations, in order to maximize the objective, i.e., the grade

of services in this study. We distinguish the master problem from the restricted master problem. The master problem contains all potential configurations: it is a theoretical problem in the sense that it cannot be solved assuming all configurations are made explicit. The restricted master is a "subversion" of the master problem, where only a very small subset of configurations are explicitly embedded. Of course, the objective is to find the optimal solution of the master problem, using an implicit enumeration of all potential configurations, thanks to the column generation techniques.

The second problem corresponds to a series of so-called pricing problems, each associated with a single source node, which generates "augmenting" configurations, i.e., configurations such that, if added to the current restricted master problem, improves the value of its objective value. The optimization model relies on so-called configurations, defined in 4.2.2.

The network topology and traffic modeled is similar as discussed in 4.1. Let the optical grid be defined by its set of nodes, V , indexed by v , and its set of directed links, L , indexed by ℓ . We assume that each link has a transport capacity of W_ℓ wavelengths, and that there are some nodes hosting a server $v^{\text{SVR}} \in V^{\text{SVR}}$ (or a data center). The definition of the link transport capacities is described in Section 5.2.1. Let V^{S} be the job requests vector, indexed by v^{S} , and D_v is equal to the number of requested jobs originating from $v^{\text{S}} \in V \setminus V^{\text{SVR}}$. Note that it is not useful to consider demands originating from a node where a server is located, at least with respect to the link transport capacities. We assume that the granting of those requests only depend on the availability of resources on the server located at the node from which

they originate.

We first describe the details of the Restricted Master Problem (RMP), followed by the pricing problem and their solution method. Note that, the restricted problem and the pricing problems are solved alternatively, and feed on each other until the optimality conditions are met.

5.1.1 Master Problem

The master problem uses two sets of variables, $z_c \in \mathbb{Z}^+$, such that component vector z_c is equal to the number of copies of configurations c , and $b_\ell^B \in \mathbb{Z}^+$, counting the backup sharing bandwidth units on their associated link. $p_{c\ell}^W$ and $p_{c\ell}^B$ correspond to the parameters in the RMP but to variables in the pricing problems. Note that they are same as we have discussed in Chapter 4, Section 4.2.1.

The objective function aims at maximizing the number of granted requests, and can be written as follows:

$$\max \sum_{c \in C} z_c. \tag{5.1}$$

There are three sets of constraints in the master problem. The first two sets of constraints are the same as min-ACRWA model in Chapter 4, Section 4.2.1, and the third set of constraints ensures the capacity of each link (W_ℓ).

The first set of constraints ensures the selected set of configurations for each source

node $v^s \in V^s$, must not exceeds than the requested size (D_v).

$$\sum_{c \in C_v} z_c \leq D_v \quad v^s \in V^s. \quad (5.2)$$

The next set of constraints computes the required spare capacity requirement for each link.

$$\sum_{c \in C} p_{cl}^w p_{cl'}^b z_c \leq b_{\ell'}^b \quad \ell', \ell \in L : \ell' \neq \ell. \quad (5.3)$$

Note that, this constraint compute the backup capacity in the sharing mode, i.e., if two or more working paths are link disjoint, then they can share the same backup capacity, as shown in Figure 5.1. Therein, solid lines represent working paths and dashed lines represent backup paths. The bandwidth link requirements of the backup path computation for the illustrated working paths (v_1, v_2, v_4 with 3 units and v_3, v_4 with 5 units) are: $\ell_2=3$, $\ell_5=5$, and $\ell_6=5$.

The third set of constraints checks that the overall required capacity (primary and backup) does not exceed the available transport capacity for each link. The value of the W_ℓ parameters can be calculated using the proposed transport capacity calculation methods, which are described in Subsection 5.2.1.

$$\sum_{c \in C} p_{cl}^w z_c + b_\ell^b \leq W_\ell \quad \ell \in L. \quad (5.4)$$

Recall that while the master problem (MP) includes all possible configurations, the Restricted Master Problem (RMP) is made of a very small subset of promising

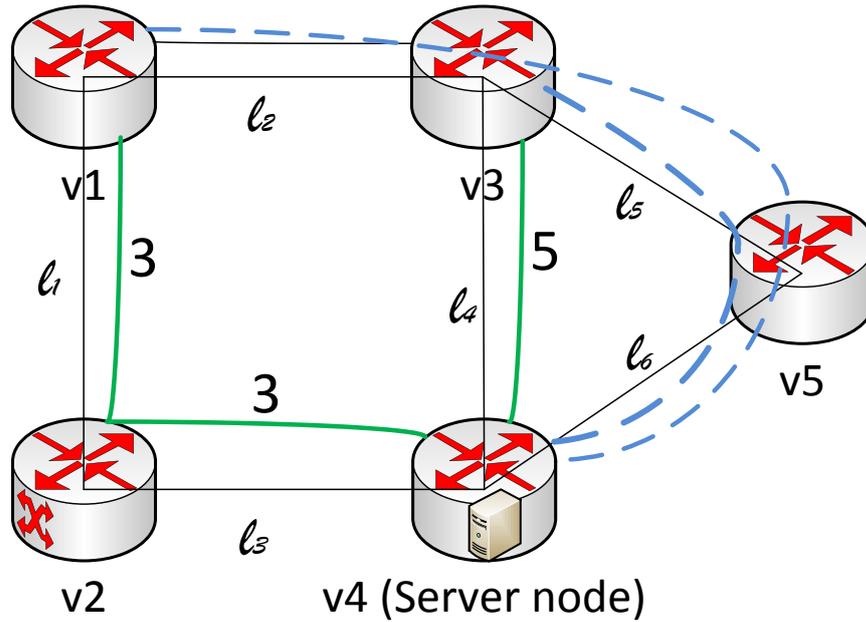


Figure 5.1: An example of computing the shared backup path.

configurations. Note that the job requests with the same origin are not necessarily processed on the same server nodes, and consequently, there might be multiple primary paths for each source node. There even might be several primary paths between the same pair of source and destination nodes, due to bandwidth requirements. We assume each job (or aggregation of jobs) transfer requires a whole wavelength per link, referred to as bandwidth unit.

5.1.2 Pricing Problem

The second element in the decomposition induced by the column generation model corresponds to the so-called Pricing Problems (PP). Here, there are as many different ones as the number of source nodes. In addition, pricing problems differ from one iteration to the next, as the values of the dual variables differ from one iteration to

the next. Note that each pricing problem takes care of the requests originating from one particular source node.

We denote by PP_{v^s} the pricing problem associated with source node v^s . When solving the pricing problem, we either find a new configuration which, if added to the RMP, may improve the value of the current cost (objective function) of the RMP (Restricted Master Problem), or we move to the solution of the next pricing problem. If, after solving all pricing problems, we have been unable to generate a single augmenting configuration, we can then conclude that we have reached the optimal value of the LP relaxation of the master problem.

We now state the expression of the pricing problem PP_{v^s} associated with source node v^s , starting with its objective function. It corresponds to the so-called reduced cost [15], whose expression depends on the values of the dual variables of the current restricted master problem. Let $u_{v^s}^{(5.2)}$ be the value of the dual variable associated with constraint (5.2- v^s), $u_{\ell\ell'}^{(5.3)}$ be the value of the dual variable associated with constraint (5.3- $\ell\ell'$), and $u_{\ell}^{(5.4)}$ be the value of the dual variable associated with constraint (5.4- ℓ).

This PP_{v^s} also contains the same set of variables described in Chapter 4, Section 4.2.2, i.e., p_{ℓ}^W for the working path, p_{ℓ}^B for the protection path, d_v^W for the working server, d_v^B for the backup server, and $p_{\ell\ell'}^{WB}$ for the linearization of the objective term.

The expression of the reduced cost of PP_{v^s} , to be maximized, can be written as follows:

$$\overline{\text{COST}} = 1 - u_{v^s}^{(5.2)} - \sum_{\ell \in L} \sum_{\ell' \in L: \ell \neq \ell'} u_{\ell\ell'}^{(5.3)} p_{\ell}^W p_{\ell'}^B - \sum_{\ell \in L} u_{\ell}^{(5.4)} p_{\ell}^W. \quad (5.5)$$

The first two sets of constraints determine the primary and backup paths using a flow formulation:

$$\sum_{\ell \in \omega^+(v)} p_\ell^W - \sum_{\ell \in \omega^-(v)} p_\ell^W = \begin{cases} +1 & \text{if } v = v^s \\ -d_v^W & \text{if } v \in V^{\text{SVR}} \\ 0 & \text{otherwise} \end{cases} \quad v \in V \quad (5.6)$$

$$\sum_{\ell \in \omega^+(v)} p_\ell^B - \sum_{\ell \in \omega^-(v)} p_\ell^B = \begin{cases} +1 & \text{if } v = v^s \\ -d_v^B & \text{if } v \in V^{\text{SVR}} \\ 0 & \text{otherwise} \end{cases} \quad v \in V \quad (5.7)$$

where $\omega^+(v)$ (resp. $\omega^-(v)$) denotes the set of outgoing (resp. incoming) links at node v . Each constraint considers three cases: (i) v is a source node, (ii) v is a destination (candidate server) node, and (iii) v is an intermediate node different from a source/destination node.

In the previous set of flow constraints, we consider potential routing to all designated server nodes, and the next two sets of constraints ensure the selection of exactly one server (destination) node for the primary and the backup paths:

$$\sum_{v \in V^{\text{SVR}}} d_v^W = 1 \quad (5.8)$$

$$\sum_{v \in V^{\text{SVR}}} d_v^B = 1. \quad (5.9)$$

As can be observed in expression (5.5) of the objective function of PP_{vs} , there is a nonlinear term $p_\ell^W p_{\ell'}^B$, which we need to linearize using variable $p_{\ell\ell'}^{WB} = p_\ell^W p_{\ell'}^B$.

Linearization constraints are written as follows:

$$p_{\ell\ell'}^{WB} \geq p_\ell^W + p_{\ell'}^B - 1 \quad \ell', \ell \in L : \ell \neq \ell' \quad (5.10)$$

$$p_{\ell\ell'}^{WB} \leq p_\ell^W \quad \ell', \ell \in L : \ell \neq \ell' \quad (5.11)$$

$$p_{\ell\ell'}^{WB} \leq p_{\ell'}^B \quad \ell', \ell \in L : \ell \neq \ell'. \quad (5.12)$$

Note that, due to the maximization of the reduced cost, constraints (5.11) and (5.12) are not necessary (i.e., are redundant).

After adding constraints (5.10), (5.11), and (5.12), the non-linear objective expression (5.5) can be rewritten as follows (linearize expression):

$$\overline{\text{COST}} = 1 - u_{vs}^{(5.2)} - \sum_{\ell \in L} \sum_{\ell' \in L: \ell \neq \ell'} u_{\ell\ell'}^{(5.3)} p_{\ell\ell'}^{WB} - \sum_{\ell \in L} u_\ell^{(5.4)} p_\ell^W. \quad (5.13)$$

Protection against single link failures: The basic protection offers protection against any single link failure and is ensured with the following set of constraints:

$$p_\ell^W + p_\ell^B \leq 1 \quad \ell \in L \quad (5.14)$$

$$p_\ell^W + p_{\ell'}^B \leq 1 \quad \ell, \ell' \in L : \quad (\ell \text{ and } \ell' \text{ are opposite to each other}) \quad (5.15)$$

which guarantee that primary and backup paths are link disjoint.

Protection against single link or intermediate node failure: In order to ensure such a protection, primary and backup paths need to be node disjoint. It is ensured by the following constraints:

$$\sum_{\ell \in \omega^+(v)} p_\ell^W + \sum_{\ell \in \omega^+(v)} p_\ell^B \leq 1 \quad v \notin V^{\text{SVR}} \cup \{v^S\}, v \in V \quad (5.16)$$

$$\sum_{\ell \in \omega^-(v)} p_\ell^W + \sum_{\ell \in \omega^-(v)} p_\ell^B \leq 1 \quad v \notin V^{\text{SVR}} \cup \{v^S\}, v \in V. \quad (5.17)$$

Note that, in addition, the above constraints ensure that paths are loop-less. Observe also that the case where there is no intermediate node in a path, i.e., if the source node is directly connected to a server node, is included in the proposed modeling.

Protection against single link or node or server node failure: In order to include server node protection, we must force the server node selection to be different for the primary and backup paths. This comes in addition to the previous constraints, and is expressed as follows:

$$d_v^W + d_v^B \leq 1 \quad v \in V^{\text{SVR}}. \quad (5.18)$$

5.1.3 Solution of the CG-ILP Formulation

A column generation solution of max-ACRWA is similar as for min-ACRWA (discussed in Chapter 4, Section 4.2.3), except the difference of reduced cost test. In case of minimization, we check if reduced cost ($\overline{\text{COST}}$) is less than 0 (negative) than we add

the configuration (column) in the restricted master problem. As a contrast, in case of maximization, if $\overline{\text{COST}}$ is greater than 0 (positive), than we add the configuration in the restricted master problem. We have also use the same technique for obtaining integer solution, and we observe that, there was no gap between the LP and ILP results. A diagram of alternatively solution of the master and the pricing problem is shown in Figure 5.2.

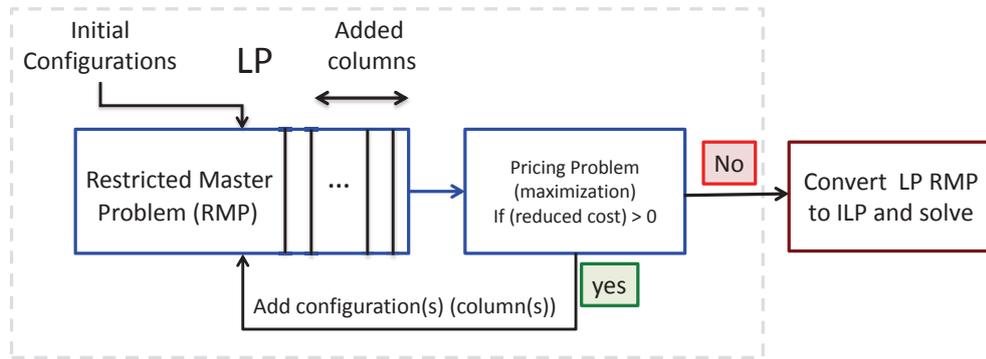


Figure 5.2: Column Generation algorithm for max-ACRWA.

5.2 Performance Evaluation of max-ACRWA

This section is subdivided into four subsections. In Subsection 5.2.1, we discuss the network and job request instances, as well as the methods we use in order to set the link transport capacities. In Subsection 5.2.2, we compare the Grade of Services (GoS) and the bandwidth requirements under different protection schemes, i.e., single link vs. single intermediate node vs. single node failures. Note that single intermediate node and single node (intermediate and server node) failures include single link failures. The impact of the number of servers is investigated in Subsection 5.2.3. In

Subsection 5.2.4, we investigate the impact of the different link transport capacity calculation methods, on the grade of services and the bandwidth requirements.

5.2.1 Network and Traffic Instances

Experiments were conducted in order to validate the model proposed in Section 5.1, and then to evaluate the performance of a grid network under various traffic loads (uniform and non-uniform), in terms of grade of services and of bandwidth requirements. Comparisons are also made in order to evaluate the impact of link vs. node protection on the bandwidth requirements.

Implementation of the model was done in C++ with the help of ILOG CPLEX for solving the (integer) linear programs. Programs were run on a single node with Intel Xeon E5462 quad-core processors 3 GHz, each with 8 GB RAM.

We used the European network topology (28 nodes and 41 bidirectional links) with two different sets of resource centers (i.e., server nodes):

$$V_{\text{EU}}^3 = \{\text{London, Vienna, Berlin}\}, \text{ or}$$

$$V_{\text{EU}}^5 = V_{\text{EU}}^3 \cup \{\text{Lyon, Zurich}\}.$$

We also use the Germany network topology (50 nodes and 88 bidirectional links), shown in Figure 5.3. We assume server resources are installed as follows:

$$V_{\text{GER}}^3 = \{\text{Braunschweig, Frankfurt and Muenchen}\}, \text{ or}$$

$$V_{\text{GER}}^5 = V_{\text{GER}}^3 \cup \{\text{Dortmund, Erfurt}\}.$$

We generated various randomly generated traffic instances with a variable number (100 up to 1,000) job/service requests. Instances are incrementally generated: the set

of 100 requests is a subset of the 200 request set, and so forth. We do not consider requests such that their origin correspond to a server nodes $v \in V^{\text{SVR}}$ as such requests can be straightforwardly granted (assuming the capacity of the server nodes is not a bottleneck issue).

We considered two types of traffic distribution, uniform and non-uniform. The non-uniform distribution takes care of the number of users/jobs associated with each node, and the number of demand requests originating from node v is calculated as follows:

$$D_v = \frac{\text{population}_{v^s}}{\sum_{v \in V \setminus V^s} \text{population}_v} \times \text{Overall_Demand}, \quad (5.19)$$

where population_{v^s} denotes the number of users around node v^s and where Overall_Demand represents the overall number of job requests. Recall that (see beginning of Section 5.1.1 for the explanations) that $D_v = 0$ if $v \in V^{\text{SVR}}$. Therefore, the ratio of server nodes demands is randomly distributed to other nodes ($v \notin V^{\text{SVR}}$).

We next discuss how to set the link transport capacities. We consider two main ways for setting the link transport capacities, **DIST-CAP** and **PWR2-CAP**, which are next described. Both ways aim at identifying power of 2 values of the transport capacities [37].

DIST-CAP: We set the link transport capacities, considering that the links which are closer to the server nodes require higher transport capacities. This way, transport capacity W_ℓ depends on the overall number of job/service requests and its

hop-distance toward the closest server node(s) (ties are arbitrarily broken):

$$W_{\ell} = \text{Constant} \times \frac{\text{Overall Number of Job Requests}}{100 \times (2^{\text{hop-distance towards closest node server})}}.$$

An example of the link transport capacity calculation on a network topology sample is illustrated in Figure 5.4, where Constant = 4. Let us assume that there are 200 job requests over the grid. Both nodes v_{d_1} and v_{d_2} host a server. For links ℓ_1 and ℓ_6 (see Figure 5.4), we get:

$$W_{\ell_1} = 4 \times \frac{200}{100 \times 2^0} = 8,$$

$$W_{\ell_6} = 4 \times \frac{200}{100 \times 2^1} = 4,$$

and so on.

Similarly, we have used Constant = 16 for European and Constant = 12 for Germany network topologies in the subsequent experiments.

Constant value was set based on various experiments in order to estimate its best value so as to get a reasonable grade of services. In contrast to assigning the same transport capacity on each link, such a transport capacity calculation reduces by more than 35% the bandwidth requirement, without a meaningful effect on the GoS.

PWR2-CAP: We first compute the required number of wavelengths on each link, assuming a shortest path routing (SP-PWR2-CAP), or an optimal routing (OP-PWR2-CAP) for all requests. Let REQ-CAP[ℓ] be the resulting capacity value. Then,

we consider two different ways of computing W_ℓ , which are next described.

SP-ROUND-PWR2-CAP or **OP-ROUND-PWR2-CAP**: W_ℓ is set to the rounded value of $\text{REQ-CAP}[\ell]$ to the closest power of 2, so that 3 and 5 are rounded to $4 = 2^2$, values 6 to 11 are rounded to $8 = 2^3$, and so on.

SP-ALEA-PWR2-CAP or **OP-ALEA-PWR2-CAP**: W_ℓ is set to

$$\text{ALEA} \{ \text{ROUND_DOWN_PWR2}(\text{REQ-CAP}_\ell), \text{ROUND_UP_PWR2}(\text{REQ-CAP}_\ell) \}$$

where ROUND_DOWN_PWR2 (resp. ROUND_UP_PWR2) corresponds to rounding down (resp. up) to the closest power of 2 value, and where $\text{ALEA} \{a, b\}$ is a function which randomly selects either a or b .

It consequently leads to four possible transport capacity computations:

OP-ROUND-PWR2-CAP: **OP-PWR2-CAP**, then round to the closest power of 2 value.

SP-ROUND-PWR2-CAP: **SP-PWR2-CAP**, then round to the closest power of 2 value.

OP-ALEA-PWR2-CAP: **OP-PWR2-CAP**, then randomly select either the rounded down value (the closest lower value which is a power of 2) or the rounded up value (the closest upper value which is a power of 2).

SP-ALEA-PWR2-CAP: **SP-PWR2-CAP**, then randomly select either the rounded down value or the rounded up value of power2.

5.2.2 Protection Schemes, GoS and Bandwidth Requirements

Using the uniform traffic distributions, we compare the Grade of Services (GoS) according to the selected protection scheme on the European and the Germany network topologies. Link transport capacities are set using the DIST-CAP method.

We compared the following three protection schemes: single link failure, single node (intermediates only) failure, and single node including server node failure. Note that the second and third protection schemes also include the protection against single link failures. Experiments were conducted on the European and the Germany networks, for different numbers of job/service requests.

Results are summarized in Figure 5.5, where the height of each vertical bar corresponds to the average, over 10 traffic instances, of the number of granted job requests, in percentage, for a given protection scheme, see the legend in Figure 5.5. Results show that there are no significant differences for the grade of services values under the three different protection scenarios for both network topologies, independently of the number of jobs. These can be explained by the values of the capacity constraints and would definitely changed if transport capacities had smaller values.

Additionally, we had a look to the bandwidth requirements. We noted that protection against single node failures use 3% more bandwidth capacity in the European (and 2% in the Germany) network topology than protection against single link failures. In brief, average hop count (i.e., number of links) of the working and backup paths for the job requests is 4.58, 4.77, and 4.87 for the European (resp. 6.85, 7.01, and 7.05 for the Germany) network topology for single link, intermediate node, and

intermediate and server node failures, respectively.

Execution times are smaller than 2 minutes for the European network while they can reach up to 103 minutes for the Germany network.

5.2.3 Impact of the Number of Servers

Again we use the uniform traffic, to analyze the GoS with different number of servers (V^3 and V^5) on European and Germany network topologies. Link transport capacities are set using the DIST-CAP method described in Section 5.2.1.

We first investigate the impact of the number of servers, i.e., 3 versus 5 server nodes, on GoS. In Figure 5.6, we provide the GoS (in percentage) for various numbers of job requests, up to 1,000 requests for the European network, up to 400 for the Germany network. Again, each bar height corresponds to an average value over a set of 10 traffic instances. On average, we observe that, with the server location set V_{EU}^5 , 2% more job requests are granted than with V_{EU}^3 , while with V_{GER}^5 , 5% more requests are granted than with V_{GER}^3 , see Subsection 5.2.1 for the definition of the server location sets. Furthermore, the average required number of hops (i.e., links) with V_{EU}^3 (resp. V_{GER}^3) is 4.98 (resp. 7.22) for the European (resp. Germany) network.

5.2.4 Comparison of Link Capacity Methods

We next compare the grade of services depending on the method for setting the link transport capacities. Experiments were conducted using both uniform and non-uniform traffic demands on the European network topology.

GoS results are first shown for uniform traffic in Figure 5.7, under a single node failure protection scheme. We observe, see Fig. 5.7, that the rounding to the closest PWR2-CAP (OP-ROUND-PWR2-CAP and SP-ROUND-PWR2-CAP) has a higher GoS (92.06% and 92.56%) than the DIST-CAP and than the randomized PWR2-CAP (OP-ALEA-PWR2-CAP and SP-ALEA-PWR2-CAP) calculation methods. On the other hand, DIST-CAP leads to a better GoS (88.12%) than the random PWR2-CAP (81.25% and 84.09%). This leads to the conclusion that PWR2-CAP with the rounding methods to the closest power of 2 values is a good choice for the optical grid dimensioning in the case of uniform traffic.

Figures 5.8(a) and 5.8(b) show the assigned (transport capacity) and effectively used capacity, respectively. While DIST-CAP leads to higher reserved (spare) and used capacity, i.e., average of 5.56 and 4.75 bandwidth units per connection, the OP-XXX-PWR2-CAP calculations leads to the lowest reserved (spare) and used capacity (reserved 3.22, 3.45; used 3.20, 3.45 bandwidth units, on average). The conclusion is then that OP-XXX-PWR2-CAP with the rounding to the closest power of 2 value is a promising method with respect to: GoS, reserved and used capacity.

In Table 5.1 we report the minimum, maximum and average execution times (in seconds). These results indicate that, when the transport capacity decreases, the optimization process takes more time.

All the aforementioned results have been obtained with uniform traffic instances. We next discuss some results for non-uniform traffic instances. Figure 5.10 shows the grade of services of the different PWR2-CAP methods. The OP-XXX-PWR2-CAP

Unit	DIST-CAP	OP-ROUND-PWR2-CAP	SP-ROUND-PWR2-CAP	OP-ALEA-PWR2-CAP	SP-ALEA-PWR2-CAP
Minimum	105	250	112	251	112
Maximum	182	1,358	292	1,010	270
Average	145	539	175	478	172

Table 5.1: Execution time (sec.).

calculations leads to the highest GoS (92%). They require less reserved/spare or used bandwidth, as can be seen in Figure 5.9. The execution times follow the same trends as for the uniform traffic.

Finally, we can conclude that the OP-XXX-PWR2-CAP calculations provide the highest GoS while requiring less bandwidth for both types of traffic instances, i.e., whether uniform or non-uniform.

5.3 Data Sets

We experimented two topologies, i.e, European and Germany network topologies, upto 1000 job requests and our CG-ILP formulations solved optimally within reasonable time. These experiments were conducted with uniform and non-uniform traffic data sets under different level of protection schemes. Additionally, we have also proposed different link capacity formulations and it led upto 35% bandwidth units reduction.

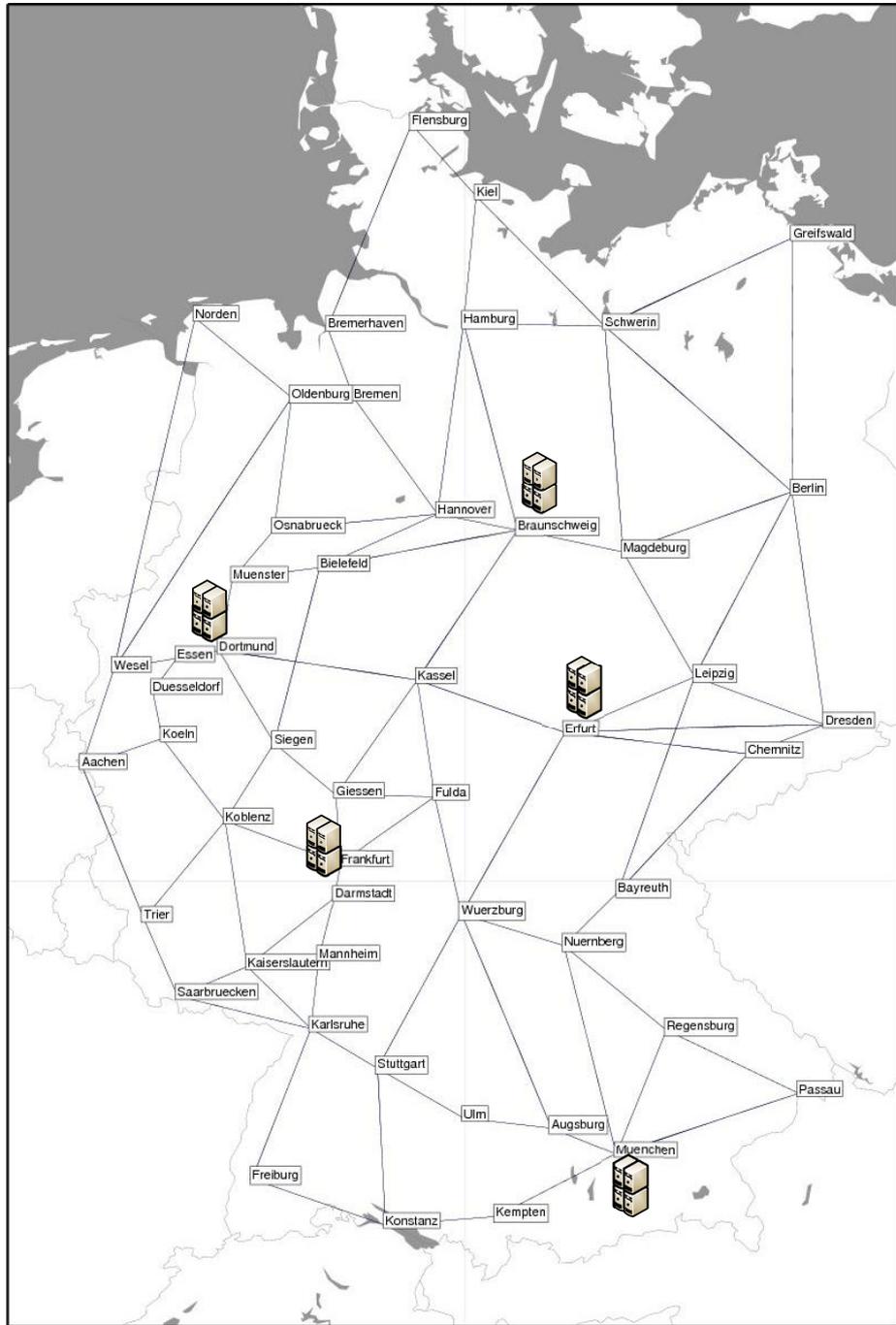


Figure 5.3: Germany network topology [51].

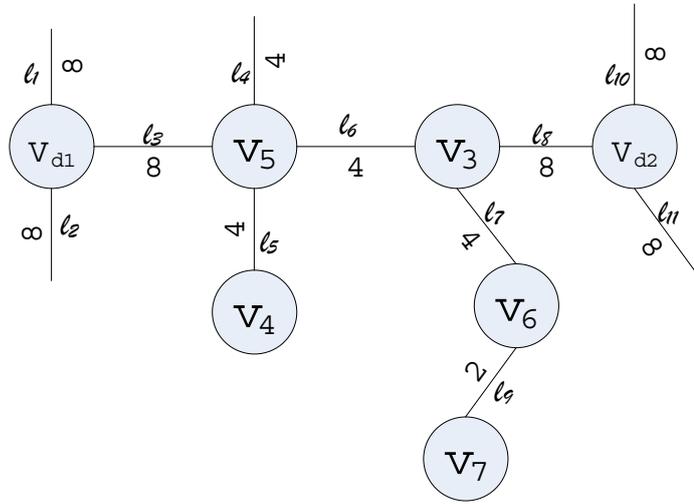
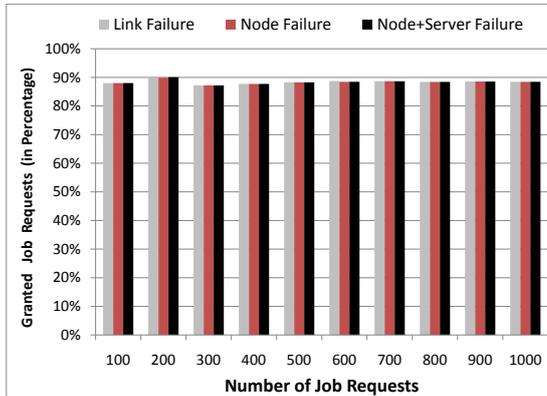
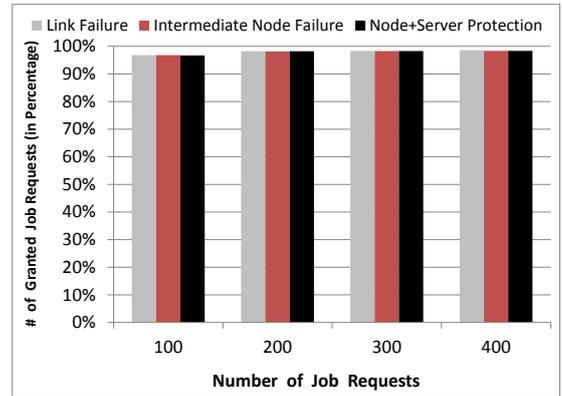


Figure 5.4: Illustration of DIST-CAP transport capacity computation.

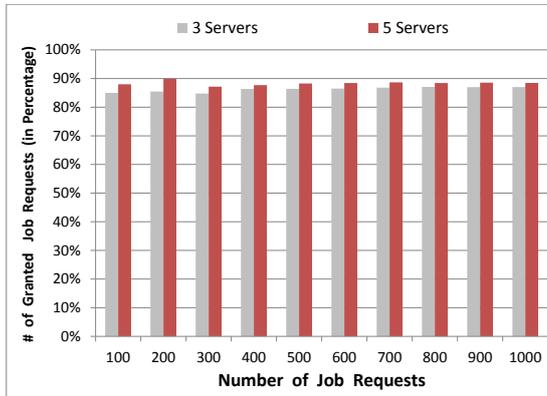


(a) European Network (V_{EU}^5)

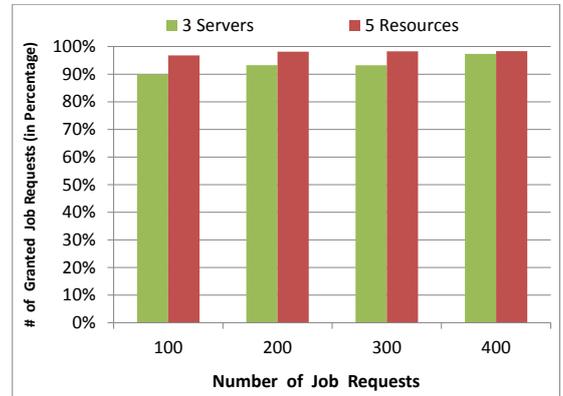


(b) Germany Network (V_{GER}^5)

Figure 5.5: Grade of Services under different protection schemes.



(a) European Network



(b) Germany Network

Figure 5.6: Grade of Services under two different selections of node servers, with respect to single node failures.

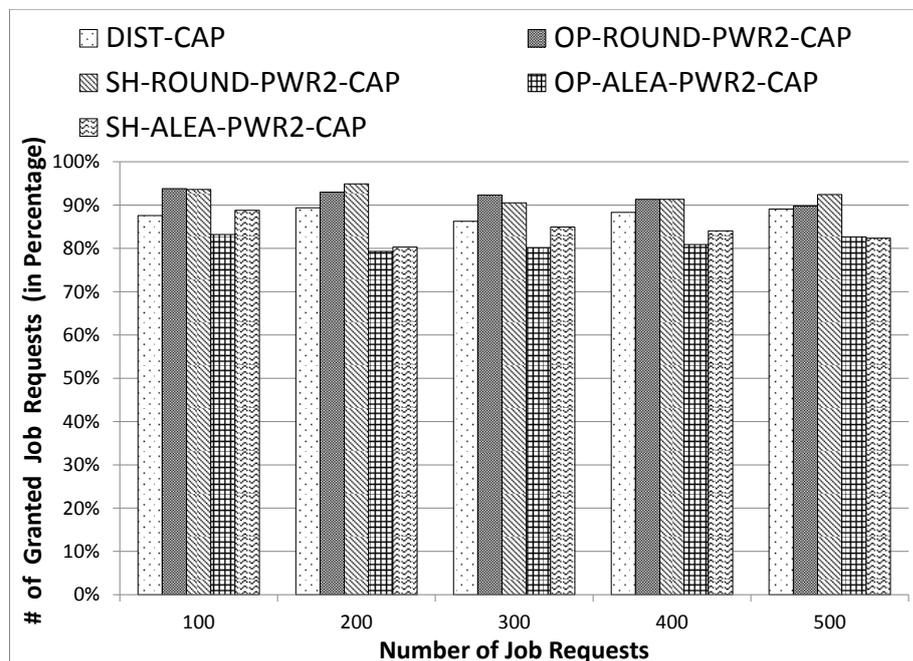
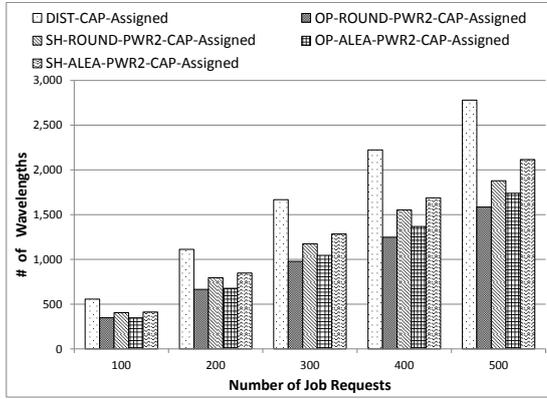
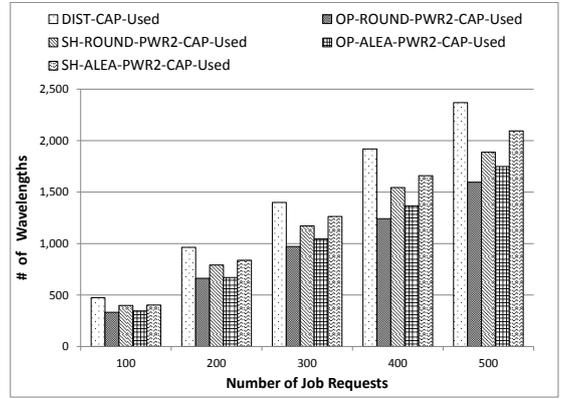


Figure 5.7: Grade of Services under uniform traffic instances.

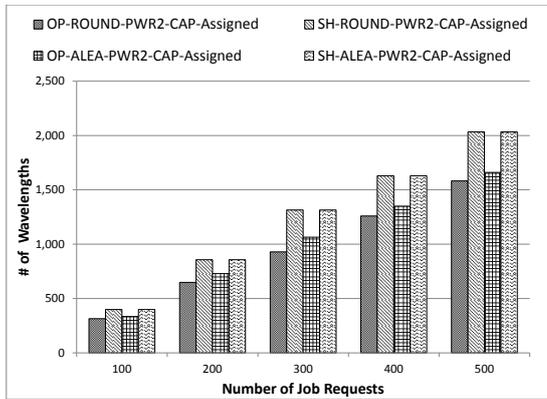


(a) Assigned Capacity

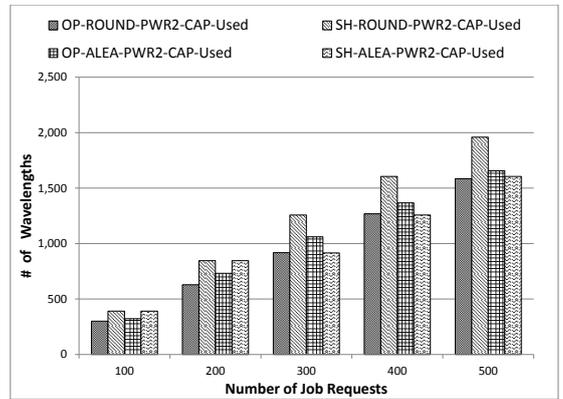


(b) Used Capacity

Figure 5.8: Variation of the bandwidth requirements with the transport capacity calculation under uniform traffic.



(a) Assigned Capacity



(b) Used Capacity

Figure 5.9: Variation of the bandwidth requirements with the transport capacity calculation under non-uniform traffic.

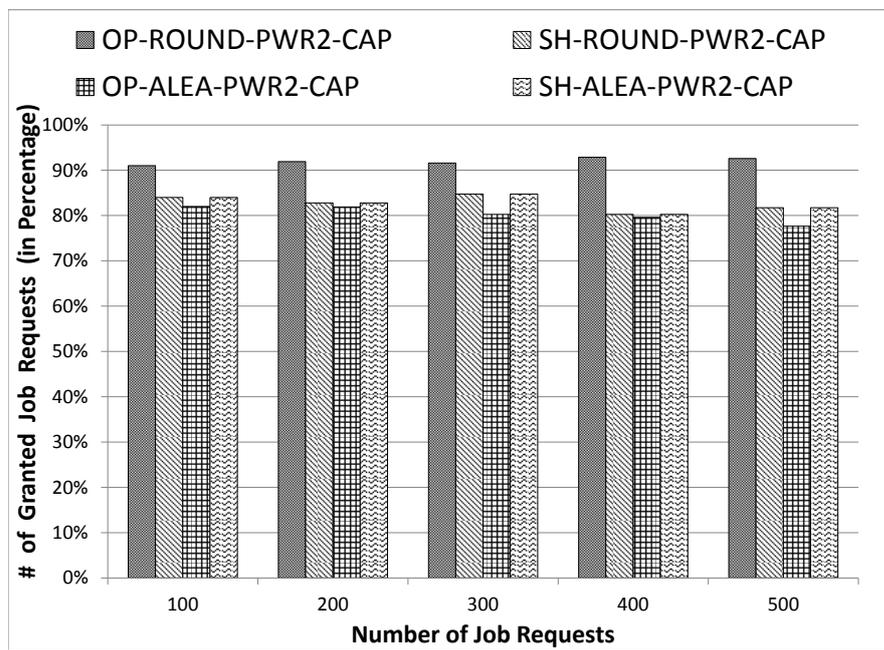


Figure 5.10: Grade of Services under non-uniform traffic instances.

Chapter 6

Optimization Models for loc-ACRWA

We have already proposed scalable optimization models in Chapters 4 and 5 for the capacitated and the uncapacitated networks, respectively. The objective in the uncapacitated network is to minimize the total required bandwidth units for a given data set. In contrast, the objective in the capacitated network is to maximize the grade of service (GoS) under limited link transport capacities. Note that, in both cases, the locations of the server nodes are given.

In this chapter, we again address a dimensioning problem by adding the server node location to the link dimensioning for optical grid/cloud networks: given a set of traffic requests, determine (i) the link transport capacities, and (ii) the location of server nodes (data centers). Therein, we propose joint optimization models for the locations of servers while finding the paths (working and backup). The objective is

to minimize the total required bandwidth units under the constraints of satisfying all the job requests.

Notations for topology and traffic input for the modeling are the same as discussed in Chapter 4, Section 4.1. Note that, the number of server nodes (n_s) is given, and their locations will be an output of the optimization process.

We are interested in resilient optical grids, and therefore, in this study, we will investigate four failure scenarios, described below.

- *Single Link Failure Scenario 1*: Protection against any single link failure.
- *Single Link or Server Node Failure Scenario 2*: Scenario 1, with the additional protection against any single server node failure.
- *Single Link or Node or Server Node Failure Scenario 3*: Scenario 2, with the additional protection against any intermediate node failure.
- *Single Link or Node or Server Node or Server Failure Scenario 4*: Scenario 3, with the additional protection against any single server failure.

Observe that Scenarios 1 and 3 are the same as we have discussed in Chapter 5, Scenario 2 adds server node failure in Scenario 1, and the last scenario adds the server (resource) failure in Scenario 3. A Scenario 4 example is shown in Figure 6.1. There, if the server located in v_1 fails, we need to reroute the requests originating from

v_1 , but also the requests which have been directed to the server located at v_1 , i.e., the requests originating from v_2 , so that the protection bandwidth requirement on (v_5, v_6) amounts to 4 units. Similarly, if the server located in v_6 fails, the protection bandwidth requirement on link (v_2, v_1) is 7 units.

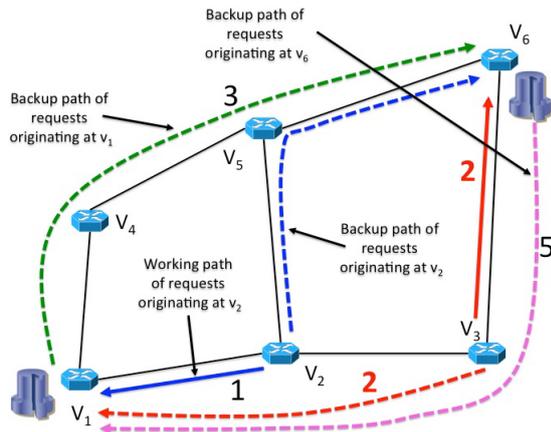


Figure 6.1: Model I: Configurations for single link or node or server node or server failure Scenario 4.

We have developed three different decomposition models, presented each in a separate section, followed by their solution methods and numerical results.

6.1 CG-ILP Model I: Path Pair Based Configurations

The first proposed model relies on a decomposition model, which is an extension of the model proposed in Chapter 4, with the addition of the selection of the best locations for the servers. The decomposition relies on a set of configurations, where each configuration is associated with a source node v^s , and made of a pair of one

working and one backup path both originated at v^s , except for the nodes which are hosting a server, where we assume that the traffic will be served locally and thus require no path. An example is sketch in Figure 6.2, where demands originating at source node v_{10} can be accommodated by configurations c_1 and c_5 (either only one of them, or distributed over the two of them). Similarly demands originating at node v_7 (resp. v_4) can be accommodated by configurations c_3 or c_4 (resp. c_2), and so on

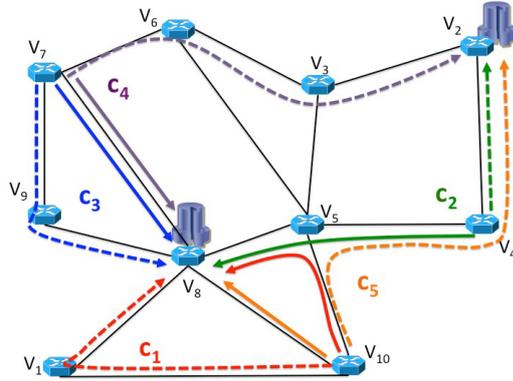


Figure 6.2: Several configuration examples for the single link failure scenario.

We define the configurations and the given parameters: C set of all configurations with $C = \bigcup_{v \in V^s} C_v$, indexed by c , C_v set of configurations associated with source node v . Parameters p_{cl}^w refers to the working path link and p_{cl}^b to the backup path link. Note that these parameters are the same as we have defined in Chapter 4, Section 4.2.1, other parameters are defined as follows:

$p_{cv}^w = 1$ if the working path goes through node v in configuration c , 0 otherwise.

$a_{cv} = 1$ if node v is selected as a server location either by the working or the backup path in configuration c , 0 otherwise.

$a_{cv}^w = 1$ if v is the server location of the working path in configuration c .

Note that we build configurations for provisioning job requests originating from any v^s , which in a particular configuration can be served at any of the possible server locations. Yet, we will retain and select only the ones associated with the selected server locations.

6.1.1 Master Problem

The master problem uses three sets of variables, $z_c \in \mathbb{Z}^+$, such that component vector z_c is equal to the number of copies of configurations c , and $b_\ell^b \in \mathbb{Z}^+$, counting the backup sharing bandwidth units on their associated link. Note that they are same as we have discussed in the Chapter 4, Section 4.2.1. Third set of variables is $y_v \in \{0, 1\}$ equals 1 if we set a server at node v , 0 otherwise. We first present the master problem for the first failure scenario and then explain how to modify it for the second, third, and fourth failure scenarios.

Single Link Failure Scenario 1

The objective function which minimizes the total network capacity is as follows:

$$\min \sum_{\ell \in L} \left(b_{\ell}^{\text{B}} + \sum_{c \in C} p_{c\ell}^{\text{W}} z_c \right) \quad (6.1)$$

subject to:

$$D_v y_v + \sum_{c \in C_v} z_c \geq D_v \quad v \in V^{\text{s}} \quad (6.2)$$

$$\sum_{c \in C} p_{c\ell}^{\text{W}} p_{c\ell'}^{\text{B}} z_c \leq b_{\ell'}^{\text{B}} \quad \ell, \ell' \in L : \ell' \notin \{\ell, \text{OPP}(\ell)\} \quad (6.3)$$

$$\sum_{c \in C_v} a_{cv} z_c \leq M y_v \quad v \in V \quad (6.4)$$

$$\sum_{v \in V} y_v \leq n_{\text{s}} \quad (6.5)$$

$$z_c \in \mathbb{Z}^+ \quad c \in C \quad (6.6)$$

$$b_{\ell}^{\text{B}} \in \mathbb{Z}^+ \quad \ell \in L \quad (6.7)$$

$$y_v \in \{0, 1\} \quad v \in V. \quad (6.8)$$

Constraints (6.2) are the demand constraints, which ensure that all requests are assigned to and granted on a server. If node v hosts a server, we assume that all the job requests originating from v are readily served by v , unless the server is offline due to some failures, a case which is not considered in a single link failure scenario. Constraints (6.3) are used to compute the bandwidth requirement for link ℓ' in a backup path. It is similar as constraints (4.3) in Chapter 4, Section 4.2.1. Constraints (6.3) are valid under the assumption that each configuration contains a single pair of working and protection paths, which are link or node disjoint depending on the failure scenario under consideration. Constraints (6.4) prevent from selecting a configuration

in which node v has been selected as a server location when $y_v = 0$. The maximum number of data center locations is controlled in (6.5). Constraints (6.6) to (6.8) define the domains of the variables.

Single Link or Server Node Failure Scenario 2

In order to provide failure Scenario 2, there is no change in the master problem.

Single Link or Node or Server Node Failure Scenario 3

The single link failure scenario model needs to be slightly modified in order to handle the failure Scenario 3, which adds the single node or server node failures to the first scenario. The objective function remains unchanged, but we need to add the following set of constraints in order to take into account the bandwidth requirements for the backup paths associated with working paths that are not node disjoint. An example of such a case is illustrated in Figure 6.3 with node v_5 belonging to two different working paths, while the two backup paths share two links. Thus, the bandwidth requirement on link $\ell = (v_8, v_9)$ must account for both affected paths under possible failure of node v_5 .

$$\sum_{c \in C} p_{cv}^W p_{c\ell}^B z_c \leq b_\ell^B \quad \ell \in L, v \in V. \quad (6.9)$$

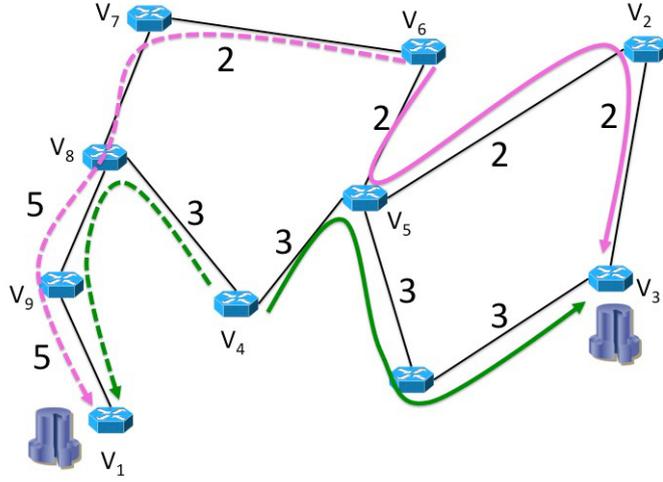


Figure 6.3: Model I: Configurations for single link or node failure scenario.

Single Link or Node or Server Node or Server Failure Scenario 4

A backup path is now needed for the requests originating at a node where a server is hosted. First, we need to modify the demand constraints (6.2), as we now need to allow the selection of configurations with a backup path for the requests originating from a node hosting a server location:

$$\sum_{c \in C_v} z_c \geq D_v \quad v \in V^S \quad (6.10)$$

Next, in the computation of the amount of protection bandwidth that is needed, we need to include the amount associated with the backup paths associated with requests originating or ending at a node hosting a server:

$$\sum_{c \in C} a_{cv}^W p_{cl}^B z_c \leq b_\ell^B \quad \ell \in L, v = v^{SVR}. \quad (6.11)$$

6.1.2 Pricing Problem

In this section, we will describe the pricing problems associated with the formulations corresponding to each failure scenario. Indeed, we cannot solve the complete master problems as described as they contain too many variables. One needs to recall that, in practice, the key idea of column generation techniques is to work only with a very small but meaningful subset of variables (or columns) of the master problem, forming the so-called Restricted Master Problem (RMP), discussed in Chapter 4, Section 4.2.

Whatever the failure scenario, each pricing problem (PP for short) corresponds to one configuration for one source node, denoted by PP_{vs} , and outputs one backup paths, and one working path if the source node does not host a server. Hence, in order to alleviate the notations, we will omit the c index, in the sections describing the pricing problems.

Single Link Failure Scenario 1

While p_ℓ^W and p_ℓ^B designated parameters in the master problem, they now denote variables in the pricing problem, and are used to find a pair of working and backup paths. Similarly, a_v^W and a_v^B , and $a_v = \max\{a_v^W, a_v^B\}$ denote variables such that $a_v^W = 1$ (resp. a_v^B) if v is the location of the destination server of the working (resp. backup) path under construction in the configuration associated with the pricing problem. Similarly, $a_v = 1$ if a server (either working or backup) is located at node v , 0 otherwise.

The objective function of the PP_{vs} , i.e., the reduced cost of variable z_c , for the

search of a configuration c associated with a source node v^s , can be written as follows:

$$\overline{\text{COST}}_1^I = \sum_{\ell \in L} p_\ell^W - u_{v^s}^{(6.2)} - \sum_{\ell \in L} \sum_{\ell' \in L: \ell' \notin \{\ell, \text{OPP}(\ell)\}} u_{\ell\ell'}^{(6.3)} p_{c\ell}^W p_{c\ell'}^B - \sum_{v \in V} u_v^{(6.4)} a_v \quad (6.12)$$

where $u_{v^s}^{(6.2)} \geq 0$, $u_{\ell\ell'}^{(6.3)} \leq 0$ and $u_v^{(6.4)} \leq 0$ are the values of the dual variables associated with constraint (6.2), (6.3), and (6.4) respectively. Note that the dual values are indexed with the constraint numbers they are associated with.

The set of constraints can be written as follows:

$$\sum_{\ell \in \omega^+(v)} p_\ell^W - \sum_{\ell \in \omega^-(v)} p_\ell^W = \begin{cases} 1 - a_v^W & \text{if } v = v^s \\ -a_v^W & \text{otherwise} \end{cases} \quad v \in V \quad (6.13)$$

$$\sum_{\ell \in \omega^+(v)} p_\ell^B - \sum_{\ell \in \omega^-(v)} p_\ell^B = \begin{cases} 1 - a_v^B & \text{if } v = v^s \\ -a_v^B & \text{otherwise} \end{cases} \quad v \in V \quad (6.14)$$

$$p_\ell^W + p_{\ell'}^W + p_\ell^B + p_{\ell'}^B \leq 1 \quad \ell, \ell' \in L, \ell \neq \ell' (\ell \text{ \& } \ell' \text{ are opposite to each other}) \quad (6.15)$$

$$a_v \geq a_v^W \quad v \in V \quad (6.16)$$

$$a_v \geq a_v^B \quad v \in V \quad (6.17)$$

$$\sum_{v \in V} a_v^W \leq 1 \quad (6.18)$$

$$\sum_{v \in V} a_v^B \leq 1 \quad (6.19)$$

$$p_\ell^W, p_\ell^B \in \{0, 1\} \quad \ell \in L \quad (6.20)$$

$$a_v, a_v^W, a_v^B \in \{0, 1\} \quad v \in V. \quad (6.21)$$

Constraints (6.13) and (6.14) are flow conservation constraints in order to establish working and backup paths. When we solve a given pricing problem for a given source node v^s , we do not know yet whether v^s will host a server. In case v^s hosts a server, i.e., $a_{v^s}^W = 1$, then we do not need to generate a working path, hence the term $1 - a_{v^s}^W$ in the first part of the case in constraints (6.13). Similarly for the backup path, if v^s hosts a server, there is no need of generating a backup path in a single link failure scenario, hence the term $1 - a_{v^s}^B$ in constraints (6.14). Note that, if $a_{v^s}^B = 0$, a better solution can be found by setting $a_{v^s}^B = 1$, hence a contraction, therefore $a_{v^s}^B = 1$ if $a_{v^s}^W = 1$, and due to constraint (6.14), there is no backup path either, which is fine in the context of single link failure.

The pair of link disjoint working and backup paths is ensured by constraints (6.15). The relationship $a_v = \max\{a_v^W, a_v^B\}$ is guaranteed by the combination of constraints (6.16) and (6.17) and the minimization of the third term of objective expression, see (6.12). Constraints (6.18) and (6.19) ensure that each path, working or backup, has exactly one destination server. Again, the minimization of the first and third term in the objective (6.12) guarantees that those constraints are satisfied as equality constraints in the optimal solution. Constraints (6.20) and (6.21) define the domains of the variables.

In order to linearize the quadratic term of the reduced cost objective, i.e., $p_\ell^W p_\ell^B$, we

introduce the set of variables $p_{\ell\ell'}^{\text{WB}} = p_{\ell}^{\text{W}} p_{\ell'}^{\text{B}}$ and the following three sets of constraints:

$$p_{\ell\ell'}^{\text{WB}} \geq p_{\ell}^{\text{W}} + p_{\ell'}^{\text{B}} - 1 \quad \ell, \ell' \in L : \ell \neq \ell' \quad (6.22)$$

$$p_{\ell}^{\text{W}} \geq p_{\ell\ell'}^{\text{WB}} \quad \ell, \ell' \in L : \ell \neq \ell' \quad (6.23)$$

$$p_{\ell'}^{\text{B}} \geq p_{\ell\ell'}^{\text{WB}} \quad \ell, \ell' \in L : \ell \neq \ell'. \quad (6.24)$$

Note that the last two sets of constraints (6.23) and (6.24) are indeed redundant, taking into account the objective (i.e., the reduced cost) of the pricing problem.

Single Link or Server Node Failure Scenario 2

Recall that, there is not any change from failure Scenario 1 to 2 in the master problem, so that the expression of the reduced cost is the same. In order to add the server node protection, following set of constraints must be added to ensure that different node locations are selected for the working and the backup servers:

$$a_v^{\text{W}} + a_v^{\text{B}} \leq 1 \quad v \neq v^{\text{S}}, v \in V. \quad (6.25)$$

Single Link or Node or Server Node Failure Scenario 3

The expression of the reduced cost is modified due to the addition of the set of constraints (6.9):

$$\overline{\text{COST}}_2^{\text{I}} = \overline{\text{COST}}_1^{\text{I}} - \sum_{v \in V} \sum_{\ell \in L} u_{v\ell}^{(6.9)} p_v^{\text{W}} p_{\ell}^{\text{B}} \quad (6.26)$$

where $u^{(6.9)} = (u_{v\ell}^{(6.9)} \leq 0)$ is the dual value vector associated with constraints (6.9).

We can easily linearize the quadratic terms $p_v^W p_{\ell'}^B$ in (6.26), in the same way we linearized the quadratic terms $p_{\ell}^W p_{\ell'}^B$ in (6.12).

Aforementioned constraints are for a single link protection scheme. For a single link or node or server node, constraints (6.15) need to be replaced by constraints (6.27) and also add the constraints (6.25).

$$\sum_{\ell \in \omega^-(v)} p_{\ell}^W + \sum_{\ell \in \omega^-(v)} p_{\ell}^B \leq 1 \quad v \neq v^S, v \in V. \quad (6.27)$$

Constraints (6.27) ensure that, for each node, there is at most one incoming link. Note that, except for the source, constraints (6.27) also ensure protection against a single server failure, except if there is a server located at the source node, this last case will be dealt with in Scenario 4.

Single Link or Node or Server Node or Server Failure Scenario 4

The expression of the reduced cost is modified due to the addition of the set of constraints (6.11):

$$\overline{\text{COST}}_3^I = \overline{\text{COST}}_2^I - \sum_{v \in V^{\text{SVR}}} \sum_{\ell \in L} u_{v\ell}^{(6.11)} a_v^W p_{\ell}^B \quad (6.28)$$

where $u^{(6.11)} = (u_{v\ell}^{(6.11)})$, with $u_{v\ell}^{(6.11)} \leq 0$ is the dual value vector associated with constraints (6.11).

In addition to constraints (6.27) and (6.25), constraints (6.14) needs to be modified

as follows:

$$\sum_{\ell \in \omega^+(v)} p_\ell^B - \sum_{\ell \in \omega^-(v)} p_\ell^B = \begin{cases} 1 & \text{if } v = v^s \\ -a_v^B & \text{otherwise} \end{cases} \quad v \in V. \quad (6.29)$$

Indeed, if the case of a server failure, a backup path needs to be provided for the requests originating at its location, and it is ensured by (6.29).

6.2 CG-ILP Model II: Source Based Configurations

In Model II, each configuration is again related with a single source node, with the difference that each source node demand is provisioned selecting a single configuration. Indeed, each configuration may select one or more working paths and a single backup path. The idea is to reduce the number of configurations to be generated, and to define a decomposition where the difficulties of the master and pricing problems are more balanced. Examples of configurations for Model II are depicted in Figure 6.4.

The variables (z_c, b_ℓ^B) and parameters (a_{cv}, p_ℓ^B) have the same definition as in the Model I in Section 6.1. The new variables and parameters are:

$b_{v\ell\ell'} \in \mathbb{Z}^+$ counts the number of backup units associated with source node, $v = v^s$, and link ℓ to be protected by ℓ' .

$\varphi_{c\ell}^W \in \mathbb{Z}^+$ is equal to the number of primary working units on link ℓ .

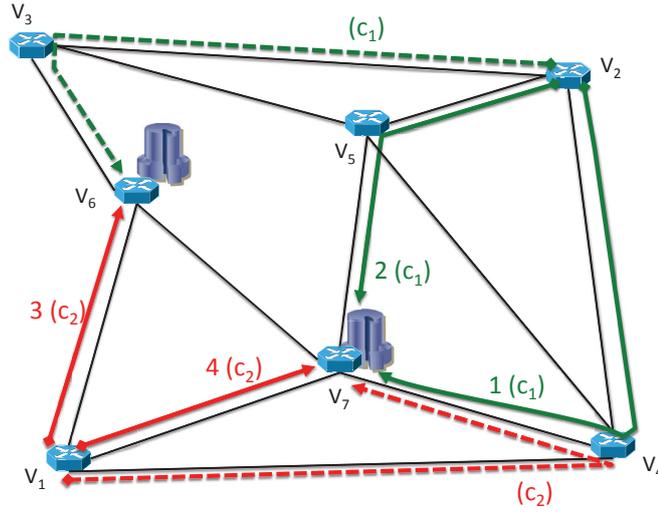


Figure 6.4: Model II: Configurations Example.

We first present the master problem, then pricing problem.

6.2.1 Master Problem

The objective is the same as in Model I, but its mathematical expression differs. It is written as follows:

$$\min \sum_{\ell \in L} \left(b_{\ell}^B + \sum_{c \in C} \varphi_{c\ell}^W z_c \right) \quad (6.30)$$

subject to:

$$y_v + \sum_{c \in C_v} z_c \geq 1 \quad v \in V^S \quad (6.31)$$

$$\sum_{c \in C_v} \varphi_{c\ell}^W p_{c\ell}^B z_c \leq b_{v\ell\ell'} \quad \ell, \ell' \in L : \ell \notin \{\ell', \text{OPP}(\ell')\}, v \in V^S \quad (6.32)$$

$$\sum_{v \in V^s} b_{v\ell\ell'} \leq b_{\ell'}^B \quad \ell, \ell' \in L : \ell \notin \{\ell', \text{OPP}(\ell')\} \quad (6.33)$$

$$\sum_{c \in C} a_{cv} z_c \leq M y_v \quad v \in V \quad (6.34)$$

$$\sum_{v \in V} y_v \leq n_s \quad (6.35)$$

$$z_c \in \{0, 1\} \quad c \in C \quad (6.36)$$

$$b_{\ell}^B \in \mathbb{Z}^+ \quad \ell \in L \quad (6.37)$$

$$b_{v\ell\ell'} \in \mathbb{Z}^+ \quad \ell, \ell' \in L : \ell \notin \{\ell', \text{OPP}(\ell')\}, v \in V^s \quad (6.38)$$

$$y_v \in \{0, 1\} \quad v \in V. \quad (6.39)$$

Constraints (6.31) ensure that at least single configuration is selected for each source node. Note that if a source node is selected as one of the server nodes, then there is no need to accept any configuration regarding the source node: this is ensured by adding first term, i.e., y_v . The backup sharing is controlled in constraints (6.32) and (6.33). The last two sets of constraints, (6.34) and (6.35), define the selection of the server nodes and of their limits respectively. They are identical to constraints (6.4) and (6.5). Constraints (6.36) to (6.39) define the domains of the variables.

6.2.2 Pricing Problem

Recall that each pricing problem corresponds to a single source, and there is also possibility of multiple working paths and single backup path.

The reduced cost expression is similar to the previous model, except the change

of φ_ℓ^W in the second term, and written as follows:

$$\overline{\text{COST}} = \sum_{\ell \in L} \varphi_\ell^W - u_{v^s}^{(6.31)} - \sum_{\ell \in L} \sum_{\ell' \in L: \ell \neq \ell'} u_{\ell \ell'}^{(6.32)} \varphi_\ell^W p_{\ell'}^B - \sum_{v \in V} u_v^{(6.34)} a_v \quad (6.40)$$

where $u^{(6.31)} \geq 0$, $u_{\ell \ell'}^{(6.32)} \leq 0$ and $u_v^{(6.34)} \leq 0$ are the values of the dual variables associated with constraints (6.31), (6.32) and (6.34), respectively.

We next describe the sets of constraints:

$$\sum_{\ell \in \omega^+(v)} \varphi_\ell^W - \sum_{\ell \in \omega^-(v)} \varphi_\ell^W = \begin{cases} D_v & \text{if } v = v^s \\ -d_v & \text{otherwise} \end{cases} \quad v \in V \quad (6.41)$$

$$\sum_{\ell \in \omega^+(v)} p_\ell^B - \sum_{\ell \in \omega^-(v)} p_\ell^B = \begin{cases} 1 & \text{if } v = v^s \\ -a_v^B & \text{otherwise} \end{cases} \quad v \in V \quad (6.42)$$

$$\varphi_\ell^W \leq M p_\ell^W \quad \ell \in L \quad (6.43)$$

$$p_\ell^W + p_{\ell'}^W + p_\ell^B + p_{\ell'}^B \leq 1 \quad \ell, \ell' \in L : \ell = \text{OPP}(\ell') \quad (6.44)$$

$$a_v^B \leq a_v \quad v \in V \quad (6.45)$$

$$d_v \leq M a_v \quad v \in V \quad (6.46)$$

$$\sum_{v \in V} a_v^B \leq 1 \quad (6.47)$$

$$\sum_{v \in V} d_v = D_v \quad (6.48)$$

$$\varphi_\ell^w \in \mathbb{Z}^+ \quad \ell \in L \quad (6.49)$$

$$p_\ell^w, p_\ell^b \in \{0, 1\} \quad \ell \in L \quad (6.50)$$

$$d_v \in \mathbb{Z}^+ \quad v \in V \quad (6.51)$$

$$a_v^b, a_v \in \{0, 1\} \quad v \in V. \quad (6.52)$$

The required demands D_v are satisfied by flow constraints (6.41) for working path(s). These paths may be selected on different server nodes. Multiple paths are indirectly enforced by dual values, if it increases the backup sharing. The variable d_v indicates the number of working path units on the server v . Next, constraints (6.42) select single backup path. Note that the same comments apply for those flow constraints as for the flow constraints of the pricing problems of Model I. Constraints (6.43) set the value of p_ℓ^w , where 1 indicates that working path is provisioned on link ℓ , and used for link disjoint paths and linearization in other constraints. Constraints (6.44), (6.45), and (6.46) are the same or the equivalent of (6.15), (6.16) and (6.17) in the pricing problem of Model I presented in Section 6.1. Required demands D_v are ensured by constraints (6.48). The last four sets of constraints, (6.49) to (6.52) set domains of the variables.

In order to linearize the third term of objective, we introduce the variable $\delta_{\ell v} \in \mathbb{Z}^+$, where $\delta_{\ell v} = \varphi_\ell^w p_\ell^b$, and add the following three sets of constraints:

$$\delta_{\ell\ell'} \leq D_v p_{\ell'}^B \quad \ell, \ell' \in L : \ell' \notin \{\ell, \text{OPP}(\ell)\} \quad (6.53)$$

$$\delta_{\ell\ell'} \leq \varphi_{\ell}^W \quad \ell, \ell' \in L : \ell' \notin \{\ell, \text{OPP}(\ell)\} \quad (6.54)$$

$$\delta_{\ell\ell'} \geq \varphi_{\ell}^W - D_v (1 - p_{\ell'}^B) \quad \ell, \ell' \in L : \ell' \notin \{\ell, \text{OPP}(\ell)\}. \quad (6.55)$$

After linearization, the reduced cost expression can be written as follows:

$$\min \sum_{\ell \in L} \varphi_{\ell}^W - u_{v,s}^{(6.31)} + \sum_{\ell \in L} \sum_{\ell' \in L: \ell \neq \ell'} u_{\ell\ell'}^{(6.32)} \delta_{\ell\ell'} + \sum_{v \in V} u_v^{(6.34)} a_v. \quad (6.56)$$

Aforementioned constraints provide single link failures Scenario 1, for adding the server node failures (Scenario 2) the following set of constraints must be added:

$$a_v^W + a_v^B \leq 1 \quad v \in V \quad (6.57)$$

where, constraints (6.57) ensure to distinct selection of working and backup servers.

6.3 CG-ILP Model III: Server Based Configurations

In this model, configurations are centered around potential locations of servers. For a given potential server location at node v , a configuration c is characterized by the set of primary and backup demands satisfied by the server located at v , see Figure 6.5 for

an illustration. Therein, there are 2 configurations, one centered at v_6 , the other one at v_7 . Primary paths are represented by plain lines, while backup paths are represented by dotted lines. The first configuration comprises one backup path (for the demand originating at v_2), and one primary path (for part of the demand originating from v_1). The second configuration comprises one backup path (for the demand from v_1), and three primary paths (one for part of the demand from v_1 , and two for the demand from v_2). Let C_v be the set of all configurations centered at location $v^{\text{SVR}} \in V^{\text{SVR}}$, and C be the overall set of configurations. We have: $C = \bigcup_{v \in V^{\text{SVR}}} C_v$.

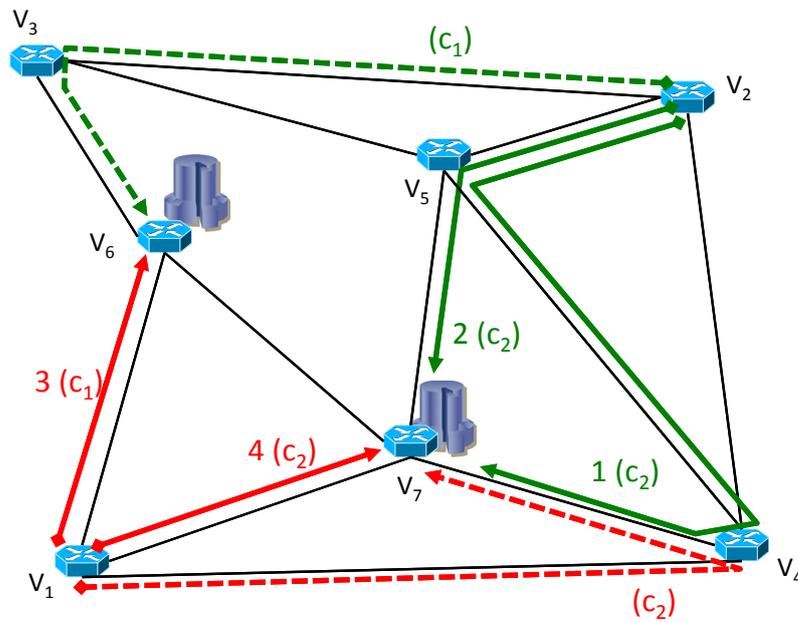


Figure 6.5: Model III: Configurations Example.

There is a unique backup path for the demands originating at a given source node v , while the demand can be served using several primary paths. By splitting the demand on several primary paths, it allows saving some bandwidth for ensuring

protection. For instance, in the example of Figure 6.5, the splitting of the demand originating at v_1 on two node disjoint paths (except for the source and destination nodes) allows a backup of bandwidth requirement restricting to 4 units, while the overall primary demand is 7 units.

In addition of $(z_c)_{c \in C}$ and $(b_\ell^B)_{\ell \in L}$ defined in previous two models, there are two other vectors, $(y_v)_{v \in V}$ and $(x_{v\ell\ell'})_{\ell, \ell' \in L, v \in V}$ such that:

$y_v = 1$ if there is a server located at node $v \in V$, 0 otherwise.

$x_{v\ell\ell'} \in \mathbb{Z}^+$, equal to the number of backup bandwidth units required on link ℓ' in order to protect link ℓ with respect to the primary paths originating from v^S .

Next, parameters of the configuration c are defined as follows:

$d_{cv}^W \in \mathbb{Z}^+$ equals the number of requests originating from v , which are accommodated (primary paths) by the server node v associated with c .

$b_{cv\ell}^W \in \mathbb{Z}^+$ equals the number of required working bandwidth units on link ℓ for provisioning primary paths from source node v^S to the server location v^{SVR} of configuration c .

$p_{cv\ell}^B \in \{0, 1\}$, where $p_{cv\ell}^B = 1$ if link ℓ is used by the unique backup path from v^S to the server location v^{SVR} of configuration c , 0 otherwise.

$p_{cv}^B \in \{0, 1\}$, with $p_{cv}^B = 1$ if there exists one backup path (there cannot be more than one) from v^S to the server location v^{SVR} of configuration c , 0 otherwise.

6.3.1 Master Problem

Recall that the original problem is decomposed into the master problem and several pricing problems, one for each potential server location. We next describe the master problem followed by the pricing problem for the single link failure scenario.

The objective function of the master problem is as the original problem, i.e., minimize the total network bandwidth units used for both working and backup paths, and can be written as follows.

$$\min \sum_{\ell \in L} \left(b_{\ell}^B + \sum_{c \in C} \sum_{v \in V^s} b_{cv\ell}^W z_c \right) \quad (6.58)$$

Subject to:

$$\sum_{v \in V} y_v \leq n_s \quad (6.59)$$

$$\sum_{c \in C_v} z_c \leq y_v \quad v \in V \quad (6.60)$$

$$D_v y_v + \sum_{c \in C \setminus C_v} d_{cv}^W z_c \geq D_v \quad v \in V^s \quad (6.61)$$

$$\sum_{c \in C} p_{cv}^B z_c = 1 \quad v \in V^s \quad (6.62)$$

$$\sum_{c \in C} b_{cv\ell}^W z_c \leq M \left(1 - \sum_{c \in C} p_{cv\ell}^B z_c \right) \quad \ell \in L, v \in V^s \quad (6.63)$$

$$\sum_{c \in C} b_{cv\ell}^W z_c \leq x_{v\ell\ell'} + M \left(1 - \sum_{c \in C} p_{cv\ell'}^B z_c \right) \quad \ell, \ell' \in L : \ell' \notin \{\ell, \text{OPP}(\ell)\}, v \in V^s \quad (6.64)$$

$$\sum_{v \in V^s} x_{v\ell\ell'} \leq b_{\ell'}^B \quad \ell, \ell' \in L : \ell' \notin \{\ell, \text{OPP}(\ell)\} \quad (6.65)$$

$$z_c \in \{0, 1\} \quad c \in C \quad (6.66)$$

$$y_v \in \{0, 1\} \quad v \in V \quad (6.67)$$

$$b_\ell^B \in \mathbb{Z}^+ \quad \ell \in L \quad (6.68)$$

$$x_{v\ell\ell'} \in \mathbb{Z}^+ \quad \ell \in L, \ell' \in L : \ell' \notin \{\ell, \text{OPP}(\ell)\}, v \in V^S. \quad (6.69)$$

Constraint (6.59) is equivalent to:

$$\sum_{c \in C} z_c \leq n_s,$$

and expresses that we select at most n_s server locations.

Constraint (6.60) imposes that, if v is selected as a server location, then at most one configuration must be selected for that server location. On the other hand, if v is not selected as a server location, then no configuration centered at v is selected. Constraints (6.61) are the demand constraints for the working path. Note that the summation is over $C \setminus C_v$ and not C_v as in Model I, due to the different definition of the configurations in the two models. Note that if v is the data center location of configuration c , then demand of v is trivially satisfied (as v hosts a server). Otherwise, primary paths need to be established in order to grant the job requests of source node v . Constraints (6.62) enforce that, for a given source node, there is a unique backup path (but possibly several working paths). Constraints (6.63) ensure that, for a given source (demand), primary and backup paths are link disjoint. However, there can be

several primary paths going through the same given link. In other words,

$$\sum_{c \in C} p_{cv\ell}^B z_c \in \{0, 1\} \quad \text{in the optimal solution,}$$

while

$$\sum_{c \in C} b_{cv\ell}^W z_c \in \mathbb{Z}^+ \quad \text{in the optimal solution.}$$

Note that the M constant value can be selected equal to the demand of the source node D_v in constraints (6.63) and (6.64).

In order to compute the protection bandwidth requirements, we need to proceed in two steps. First, with constraints (6.64), we compute the protection bandwidth requirements related to a given source node. Second, since we only allow one backup path, any link belonging to the backup path should account for the protection of any link belonging to a working path, as expressed in (6.65).

Constraints (6.65) can be equivalently written:

$$\max_{\ell \in L} \sum_{v \in V^s} x_{v\ell\ell'} \leq b_{\ell'}^B \quad \ell' \in L. \quad (6.70)$$

They are identical to the constraints (6.33) of Model 2. Their purpose is to compute the backup bandwidth requirements on ℓ' taking into account the bandwidth sharing.

6.3.2 Pricing Problem

Pricing problem will output one configuration c , i.e., for a given server node v^{SVR} selected to be a server location, it will identify the source nodes V^{S} served or not for the primary paths (fully or partially) and backup paths by v^{SVR} .

The objective function of the reduced cost expression can be written as follows:

$$\begin{aligned} \min \sum_{\ell \in L} \sum_{v \in V^{\text{S}}} b_{v\ell}^{\text{W}} - u_{v^{\text{SVR}}}^{(6.60)} - D_{v^{\text{SVR}}} u_{v^{\text{SVR}}}^{(6.61)} - \sum_{v \in V^{\text{S}} \setminus \{v^{\text{SVR}}\}} u_v^{(6.61)} d_v^{\text{W}} - \sum_{v \in V^{\text{S}}} u_v^{(6.62)} p_v^{\text{B}} \\ - \sum_{\ell \in L} \sum_{v \in V^{\text{S}}} u_{v\ell}^{(6.63)} (b_{v\ell}^{\text{W}} + M p_{v\ell}^{\text{B}}) - \sum_{\ell \in L} \sum_{\ell': \ell' \notin \{\ell, \text{OPP}(\ell)\}} \sum_{v \in V^{\text{S}}} u_{v\ell\ell'}^{(6.64)} (b_{v\ell}^{\text{W}} + M p_{v\ell'}^{\text{B}}) \quad (6.71) \end{aligned}$$

where $u_v^{(6.60)} \leq 0$, $u_v^{(6.61)} \geq 0$, $u_v^{(6.62)} \leq 0$, $u_{\ell v}^{(6.63)} \leq 0$, $u_{\ell\ell' v}^{(6.64)} \leq 0$ be the (real) values of the dual variables associated with constraints (6.60), (6.61), (6.62), (6.63), (6.64), respectively.

Constraints: instead of flows originating from a single source node, we have multi-flows, with possibly one flow originating from each source node. The first two sets of constraints may find the working and the backup paths to the given server node v^{SVR} for each source node v^{S} .

$$\sum_{\ell \in \omega^+(v')} b_{v\ell}^{\text{W}} - \sum_{\ell \in \omega^-(v')} b_{v\ell}^{\text{W}} = \begin{cases} d_v^{\text{W}} & \text{if } v' = v \neq v^{\text{SVR}} \\ -d_v^{\text{W}} & \text{if } v' = v^{\text{SVR}} \neq v \\ 0 & \text{otherwise} \end{cases} \quad v \in V^{\text{S}}, v' \in V \quad (6.72)$$

$$(6.73)$$

$$\sum_{\ell \in \omega^+(v')} p_{v\ell}^B - \sum_{\ell \in \omega^-(v')} p_{v\ell}^B = \begin{cases} p_v^B & \text{if } v' = v \neq v^{\text{SVR}} \\ -p_v^B & \text{if } v' = v^{\text{SVR}} \neq v \\ 0 & \text{otherwise} \end{cases} \quad v \in V^S, v' \in V \quad (6.74)$$

$$d_v^W \leq D_v \quad v \in V^S \quad (6.75)$$

$$b_{v\ell}^W \in \mathbb{Z}^+ \quad \ell \in L, v \in V^S \quad (6.76)$$

$$p_{v\ell}^B \in \{0, 1\} \quad \ell \in L, v \in V^S \quad (6.77)$$

$$d_v^W \in \mathbb{Z}^+ \quad v \in V^S \quad (6.78)$$

$$p_v^B \in \{0, 1\} \quad v \in V^S. \quad (6.79)$$

There is no need to explicitly forbid primary and backup paths to use different links, as such a requirement is indirectly reinforced in the constraints (6.63) and (6.64) of the master problem, where we compute the requirements for the backup bandwidth. In addition, it is likely that the working path and the backup path of a given source will not belong to the same configuration. Constraints (6.75) ensure requested number of jobs for each source node v^S . Constraints (6.76) to (6.79) define the domains of the variables.

6.4 Solution Process

In order to obtain the integer solution from the RMP, we have used two ILP heuristic solution schemes. The first one, referred below as the classical one, is very similar

to the one in i.e., corresponds to the use of an ILP solver on the constraint matrix associated with the set of generated columns in order to reach the optimal solution of the linear relaxation, discussed in Chapter 4, Section 4.2.3. The second, referred below as Mixed Integer Linear Programming (MILP), For all three models, and described in the flowchart of Figure 6.6.

We solve the continuous relaxation of the models using the column generation method until either the optimality condition was satisfied, or until an ε -optimality condition is satisfied when the convergence is too slow. Indeed, for Model III, we observed a very slow convergence in practice, which is not due to degeneracy. We then define a stopping condition in order to allow reaching an optimized LP value within reasonable computing times: if the LP objective value is not improved by more than 0.05% (Model III(a)) or 0.01% (Model III(b)) during the last 50 iterations, we stop the LP solution process. While this implies we can no longer guarantee to reach the optimal LP solution, it allows reaching optimized LP solutions within a reasonable amount of time. The choice of the numerical values used in the stopping criteria are justified by additional experiments presented in Figure 6.7. In this Figure, we did additional intensive testing for finding out the best compromise between the parameters of the stopping condition and the accuracy of the last output LP solution. Therein, we see that if we only require 30 instead of 50 iterations, the accuracy of the LP bound is definitively deteriorating.

Rather than developing a costly branch-and-cut algorithm, we next solve the incumbent integer restricted master problem (RMP) (i.e., the restricted master problem

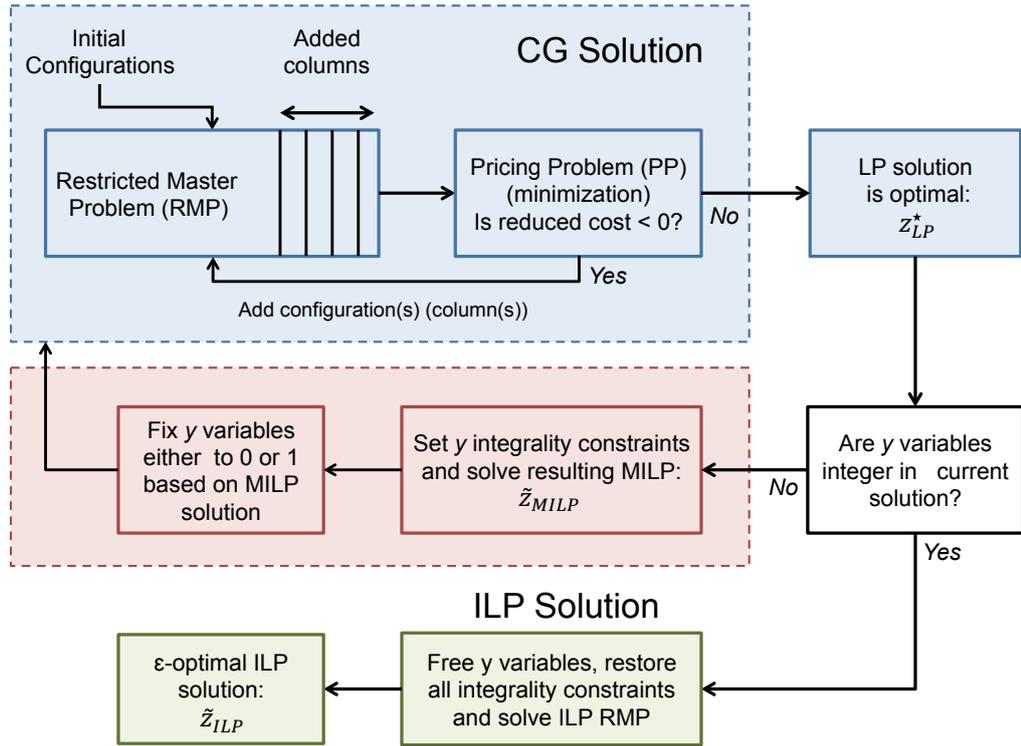


Figure 6.6: Flowchart of the solution process.

made of the columns generated until the linear relaxation was (ε) -optimally solved) using the MILP and then the ILP solver of IBM ILOG Cplex engine. Rather than solving directly the integer RMP, we first only restore the integrality conditions for the y_v variables, solve the corresponding MILP RMP, fix the y_v variables to their value in the MILP optimal solution (denoted by \tilde{z}_{MILP} since, while it is the optimal value of the incumbent MILP RMP, it is not necessarily the optimal MILP solution of the original model), and then again generate improving columns using the remaining continuous variables (i.e., the x_{\square} and the z_c variables). Once the (ε) -optimality condition is again satisfied for the LP relaxation of the RMP with binary valued y_v variables, we release the y_v variables and solve the resulting incumbent ILP RMP



Figure 6.7: LP results on different stopping criteria on M-III with 100 jobs.

using the ILP Cplex solver. Let \tilde{z}_{ILP} denote the value of the resulting ILP solution.

For all models, we generated an initial solution, i.e., an initial set of configurations, as follows: We use the pricing problems, without changing the set of constraints, but with replacing the reduced cost objective by the minimization of the overall working and backup bandwidth requirements instead. This way of proceeding is not necessarily the fastest way to generate initial configurations, but a very practical one, as it does not require developing new models or algorithms.

The characteristics of our three models and their solutions have been summarized in Table 6.1.

Model	Configuration	Variables	Solution
I	Associated with 1 source: 1 working, 1 backup path ≥ 1 config. used per source	$z_c \in \mathbb{Z}^+$ $x_\ell^B \in \mathbb{Z}^+; x_{\ell\ell'}^v \in \mathbb{Z}^+$ $y_v \in \{0, 1\}$	See flowchart in Figure 6.6 LP solved optimally $\rightsquigarrow z_{LP}^{*,I}$ ILP solved through CPLEX engine
	Associated with 1 source: ≥ 1 working, 1 backup paths exactly 1 config. per source only 1 backup path per source	$z_c \in \mathbb{Z}^+$ $x_\ell^B \in \mathbb{Z}^+; x_{\ell\ell'}^v \in \mathbb{Z}^+$ $y_v \in \{0, 1\}$ 2 nd and 3 rd scenarios: $x_{v\ell'}^{v_s} \in \mathbb{Z}^+$	See flowchart in Figure 6.6 LP solved optimally $\rightsquigarrow z_{LP}^{*,II}$ ILP solved through CPLEX engine
III	Associated with 1 server: ≥ 1 working, = 1 backup paths	$z_c \in \mathbb{Z}^+$ $x_\ell^B \in \mathbb{Z}^+$	III(a) Stop CG if during last 50 iterations, LP cost reduction was $< 0.05\%$ III(b) Stop CG if last during 50 iterations, LP cost reduction was $< 0.01\%$
	exactly 1 config. per server	$y_v \in \{0, 1\}$	III(c) Start from ILP solution of Model II, and stop CG if during 50 iterations LP cost reduction was $< 0.5\%$
	only 1 backup path per source	2 nd and 3 rd scenarios: $x_{v\ell'}^{v_s} \in \mathbb{Z}^+$	LP solved heuristically $\rightsquigarrow z_{LP}^{*,III} \geq z_{LP}^{*,III}$ ILP solved through CPLEX engine

Table 6.1: Summary of the three models and their solution variants.

6.5 Numerical Results for loc-ACRWA

This section investigates numerical results of three optimization models discussed in the previous sections. These results are evaluated on four different failure scenarios, discussed in the starting of this chapter. Initial results have been published in [35], and other results have been submitted in European Journal of Operational Research (EJORS).

6.5.1 Data Instances

We considered the network topology of the European backbone network shown in Figure 4.1(b). We conducted experiments with 3 and 5 server nodes.

Different demand sets have been randomly generated as sets of unit requests, assuming them to be uniformly distributed among the source nodes. A unit request calls for a full wavelength to be provisioned between the source and a destination to be chosen among the server nodes (using anycast). Experiments have been conducted with 50 up to 1,000 unit requests, where for each demand size we have repeated the experiment for 5 or 10 instances. Also, note that when increasing the request set for a particular random seed from D_1 to D_2 (thus with $|D_2| > |D_1|$ unit requests), we have $D_2 \supset D_1$.

6.5.2 Comparison of Solution Schemes

We have started simulation on Model-I and Model-II, and compared two solution schemes: classical and MILP (discussed in Section 6.4). Both models were tested on the European network topology (28 nodes and 41 bidirectional links) with five server (cluster) nodes, whose locations has to be optimized. We generate uniform and non-uniform incremental traffic instances from 50 to 500 (discrete) job requests. In the non-uniform traffic instances, number of jobs is proportional to the population of the node (city).

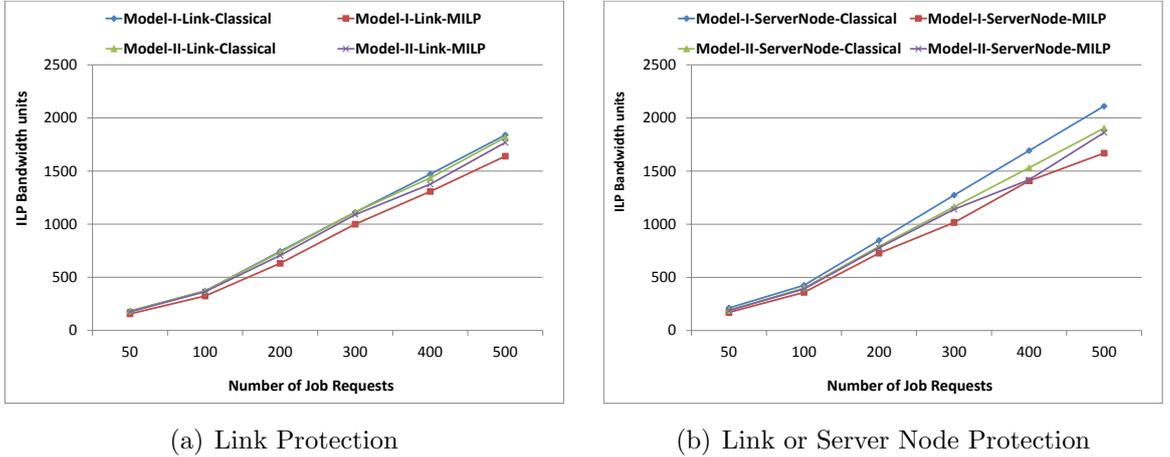
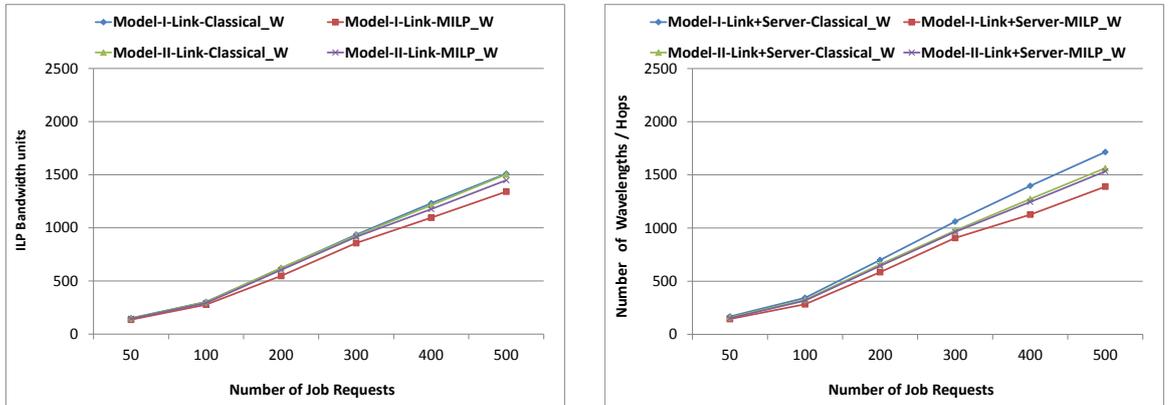


Figure 6.8: Uniform traffic: Model I vs. Model II and Classical vs. MILP solution approach.

Evolution of bandwidth requirements with an increasing number of job requests is illustrated in Figure 6.8 for uniform traffic. In Figure 6.8(a) (resp. 6.8(b)), results are provided for protection against single link failures (resp. single link or single server node failures). Each point represents the average number of wavelengths (hops/links), averaged over 5 traffic instances, for a given number of job requests. We observe that Model I requires less bandwidth than Model II under both protection schemes.

Moreover, MILP approach takes less bandwidth than the classical, i.e., 12% in Model I and 5% in Model II. This concludes that MILP approach is more efficient than the classical one for both models, in terms of computing times and solution accuracy.

The same experiments were conducted with the non-uniform traffic instances, and the results are shown in Figure 6.9. Conclusions are very similar than for the uniform traffic instances, except that the differences between the different models are solution approaches are bigger. This means, the non-uniform traffic instances consumes around 20% less bandwidth units than the uniform one.



(a) Link Protection

(b) Link or Server Node Protection

Figure 6.9: Non Uniform traffic: Model I vs. Model II and Classical vs. MILP solution approach.

We conclude that the MILP approach produces better solutions than the classical one with uniform and non-uniform traffic instances. We have also observed the both type of traffic instances have same trends. For this reason, in the remaining sections we would considered only MILP approach with uniform traffic instances.

6.5.3 Comparison of the Accuracy of the Solutions of the Three Models

In this section, we compare the quality of the solutions as output by the three CG models, as described in Sections 6.1 to 6.3, in a scenario with 5 servers within a single link failure protection scheme.

We looked at the optimal (Models I and II) or optimized (Model III) values of the LP relaxations (see Table 6.2), the optimal MILP values (see Table 6.4), and the best ILP values (see Table 6.3), together with the computing times (for reaching the optimal/optimized LP values in Table 6.2 and the overall computing times for reaching the final ILP values in Table 6.3) and the number of generated/selected configurations. All numbers correspond to averages over 10 randomly generated instances.

The comparisons of the various lower (optimal values of the LP relaxations) and upper (ILP values) bounds lead to the following observations. Note that the LP values of Model III are valid lower bounds due to the stopping condition described in Section 6.4. However, all ILP values are upper bounds on the optimal ILP values.

The lowest LP and ILP values are obtained by Model I, which is the only Model in which multiple backup paths are allowed. In Section 6.5.6, we will further investigate the protection schemes generated by Model I. Under the restricted assumption of a single backup path, Model III provides, by far, the best lower bounds, except for 800 and 900 jobs, but then the upper bounds (ILP values) of Model II and Model III(b) or (c) are very close under the assumptions of a single backup path.

# jobs	z_{LP}^*			\tilde{z}_{LP}			CPU _{LP}						Number of Generated Configurations				
	I	II	III	III(a)	III(b)	III(c)	I	II	III(a)	III(b)	III(c)	I	II	III(a)	III(b)	III(c)	
100	127	181	160	152	152	152	332	8,878	5,194	28,223	4,153	134	986	552	1349	589	
200	265	330	341	326	348	348	322	8,477	3,361	14,253	2,665	133	1011	448	1064	463	
300	405	483	491	469	482	482	364	5,855	6,348	23,828	3,774	136	754	585	1266	545	
400	549	648	654	631	707	707	310	6,135	5,559	17,679	4,107	132	742	555	1175	543	
500	695	798	831	803	784	784	303	8,537	7,049	22,874	4,688	133	1012	633	1287	591	
600	842	970	981	947	962	962	332	6,463	7,651	21,237	4,638	133	781	624	1,129	568	
700	995	1,127	1,153	1,152	1,139	1,139	298	5,794	8,628	13,350	5,092	130	761	619	1,081	508	
800	1,120	1,280	1,320	1,295	1,293	1,293	328	5,905	5,920	17,498	4,756	132	767	565	1,220	543	
900	1,274	1,434	1,476	1,439	1,508	1,508	333	6,505	5,890	16,940	4,028	131	790	563	1,159	473	
1,000	1,428	1,628	1,663	1,590	1,594	1,594	286	5,883	7,656	20,281	3,807	130	700	569	1,292	544	

Table 6.2: Optimal (Models I and II) and optimized (Model III) LP values (5 servers).

# jobs	\hat{z}_{ILP}						CPU _{OVERALL} (includes all solution phases)						Number of Generated Configurations					# of selected configs.
	I	II	III(a)	III(b)	III(c)	I	II	III(a)	III(b)	III(c)	I	II	III(a)	III(b)	III(c)			
100	281	306	309	309	306	496	9,361	23,144	88,682	21,120	217	1,019	697	1,512	657	35		
200	557	606	604	604	606	491	8,888	14,967	44,118	12,542	218	1,041	579	1,237	526	38		
300	822	902	896	896	902	509	6,508	21,610	62,974	16,893	214	798	678	1,339	641	37		
400	1,099	1,199	1,192	1,192	1,199	465	6,741	23,912	56,626	19,127	216	784	679	1,370	654	43		
500	1,384	1,480	1,490	1,490	1,480	463	9,378	26,622	64,373	20,768	209	1,069	798	1,434	723	41		
600	1,637	1,762	1,780	1,787	1,762	499	7,843	28,821	64,216	22,264	221	865	711	1,329	682	43		
700	1,993	2,037	2,079	2,079	2,037	408	7,420	31,583	47,127	19,269	178	862	710	1,261	598	35		
800	2,309	2,309	2,354	2,354	2,309	465	8,078	23,723	54,991	18,023	178	904	680	1,333	693	30		
900	2,450	2,591	2,658	2,658	2,591	503	9,003	25,496	53,568	14,341	219	950	640	1,328	527	40		
1,000	2,730	2,892	2,922	2,922	2,892	448	9,152	31,251	68,404	21,260	218	900	719	1,493	783	40		

Table 6.3: Best ILP values (5 servers).

While Model III(b) gives the best lower bounds among the three variants of Model III, i.e., when the stopping criteria is no improvement larger than 0.01% during the last 50 iterations (Model III(b)), the computing times of Model III(b) are much larger than Model III(a), without providing significantly better ILP solutions. Observe that while the LP values of Model III(c) are not as good as those of Model II, due to the stopping condition for Model III(c) that does not guarantee reaching the optimal LP value.

We now look at the empirical lower bounds as given by the \tilde{z}_{MILP} values. In Figure 6.10, we provide the curves of the number of unit requests vs. \tilde{z}_{MILP} , together with the confidence intervals associated with the values \tilde{z}_{MILP} over a set of 5 randomly generated problems. We then observe that the standard deviation values are rather small, and that all three models generate similar results, except for Model I, especially when the number of unit requests increases. Indeed, since Model I allows several backup paths, it achieves a smaller overall bandwidth requirement.

As a conclusion on the comparison of the lower and upper bounds for all three models, we observe that, while those gap values are not very small, they are, to some extent, very similar to the gap values observed for the p -center problem [11], which is a priori simpler as there is no backup path to compute.

The computing times (CPU) (up to the LP) for Model I are always the lowest ones, while under the assumption of a single backup path, Model III(c) has the lowest ones, thanks to the “warm” start and the relaxed optimality condition. However, if we take into account the computing times of generating an initial solution, Model III(a) is the

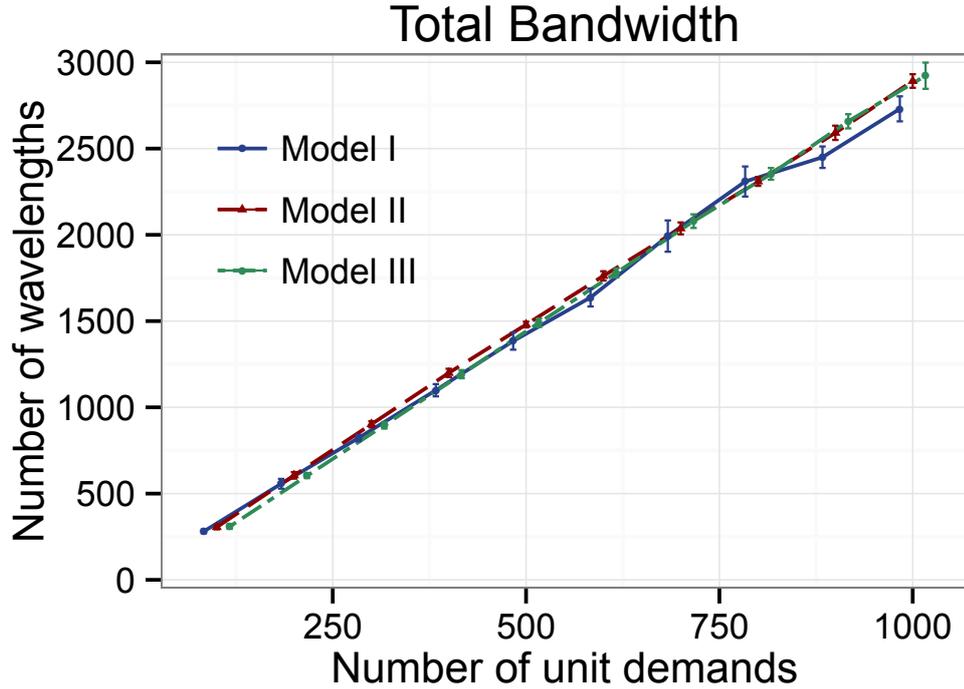


Figure 6.10: Comparison of all three models with respect to their bandwidth requirements

most economical one. We conclude that it would be worth designing a fast heuristic in order to provide an initial solution, as the complete solution of Model II, even if it provides a good initial solution, is quite computationally expensive. However, while providing a “warm” start helps to speed up the LP solution of Model III(c), it does not help to speed it up for the ILP solution: the overall computing times are smaller than Models III(a) and III(b), but much higher than those of Model II (or even Model I).

We have also analyzed the number of generated configurations before reaching the (ε) -optimal LP and ILP solutions. Model I generates by far the lowest number of configurations, while, under the assumption of a single backup path, it is Model

III(c), whether it is for reaching the (ε -)optimal LP values or the best ILP values.

# jobs	z_{MILP}^*				
	I	II	III(a)	III(b)	III(c)
100	268	303	250	273	266
200	500	604	540	456	541
300	784	896	623	707	780
400	1,037	1,193	911	861	1,030
500	1,241	1,475	1,184	1,140	1,271
600	1,552	1,743	1,499	1,574	1,460
700	1,487	2,018	1,904	1,574	1,783
800	1,527	2,276	1,951	1,793	1,904
900	2,320	2,562	2,210	1,996	2,308
1,000	2,589	2,857	2,343	2,378	2,058

Table 6.4: Optimal MILP values (with fixed server locations).

6.5.4 Comparison of the Server Selection by the three models

In order to further investigate the different ILP values of the three models, we examined the selection of server nodes made by all three models under the single link failure protection scheme. Results are depicted in Figure 6.11. Along the vertical axis, we put the average number of times (expressed as a percentage) a given server location was selected by a given model. We observe that the three models mostly select the same set of server nodes, i.e., London, Lyon, Zurich, Berlin, Vienna. Only Model I selects Madrid as a server node in some instances.

Although the models do select the same server locations, they do not necessarily select the same paths, and the same number of backup paths (one or several). Figure 6.12 plots the amount of working (Figure 6.12(a)) and backup capacity (Figure

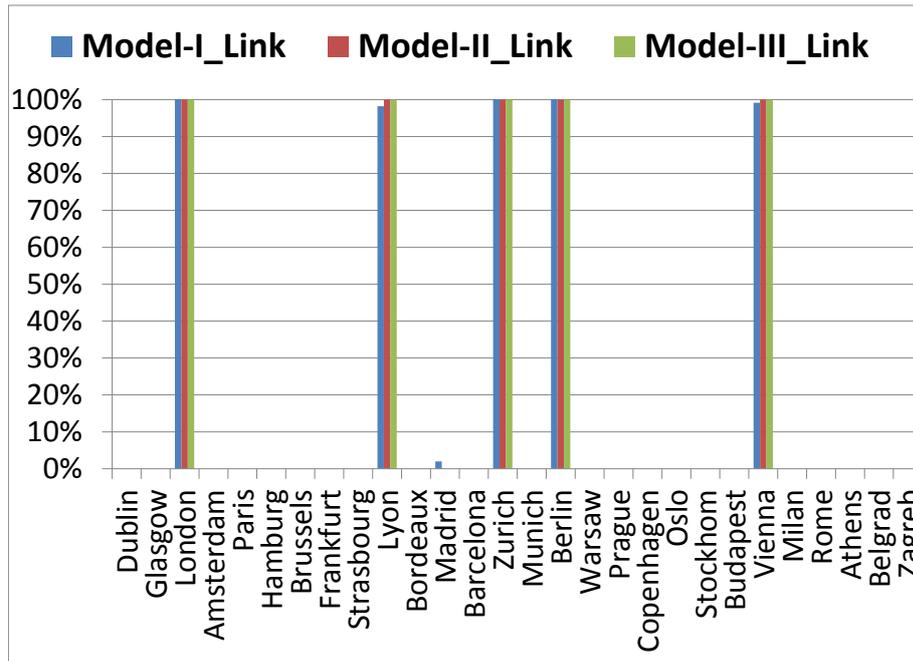
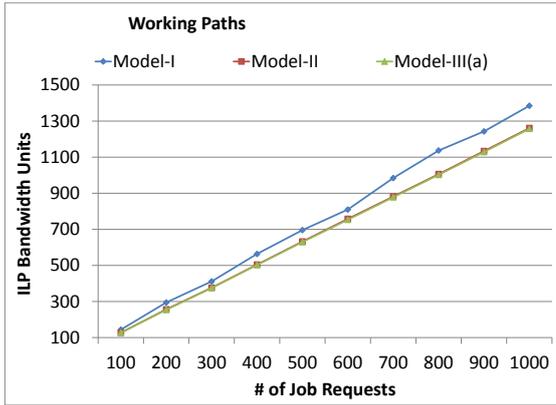


Figure 6.11: Selection of server nodes under link protection scheme.

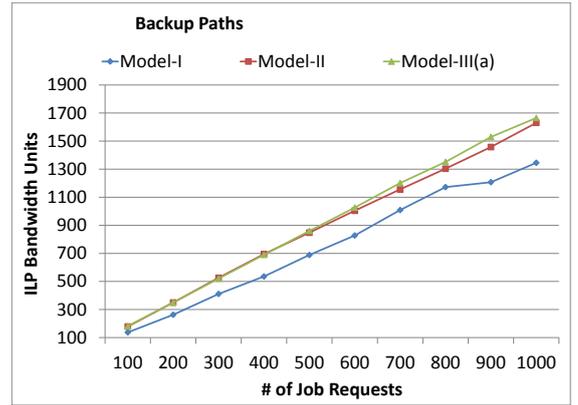
6.12(b)) for each for the three models. While, we observed (as in Table 6.3) that the bandwidth requirements are very similar for Models II and III(a), the working bandwidth requirements are larger for Model I, while its backup requirements are less than for Models II and III, while the sum of those bandwidth requirements are similar to those of Models II and III. This is mainly explained by the path structure: only one backup path for Models II and III, while there could be several of them for Model I.

6.5.5 Comparison of the Bandwidth Utilization for Model I

Comparative bandwidth requirements for the three failure scenarios are depicted in Figure 6.13. We only explicitly present results for Model I, since the qualitative trends



(a) Working bandwidth requirements



(b) Backup bandwidth requirements

Figure 6.12: Comparison of the bandwidth requirements of the three models.

are the same for all three models, even if the numerical values are not exactly the same. For the bandwidth requirements, we distinguish the working and the backup bandwidth requirements, as a function of the number of unit requests.

We observe that, for Model I, for the first two failure scenarios, the total backup bandwidth is comparable to the working. Indeed, as the backup paths are shared among disjoint working paths, they do not require more bandwidth, even if they may be longer. This observation is independent of the number of requests. However, for the third failure scenario, i.e., where we protect against link+node+server failures, backup path capacity always exceeds the working capacity, and the more so as the number of requests increases. This can be explained by the fact that when a data center collocated with network node v_s fails, also the requests originating at that node need to be rerouted, whereas in the case of only network node failure, we only need to reroute traffic that was sent to the data center (server node) at v_s coming from other source nodes $v \neq v_s$.

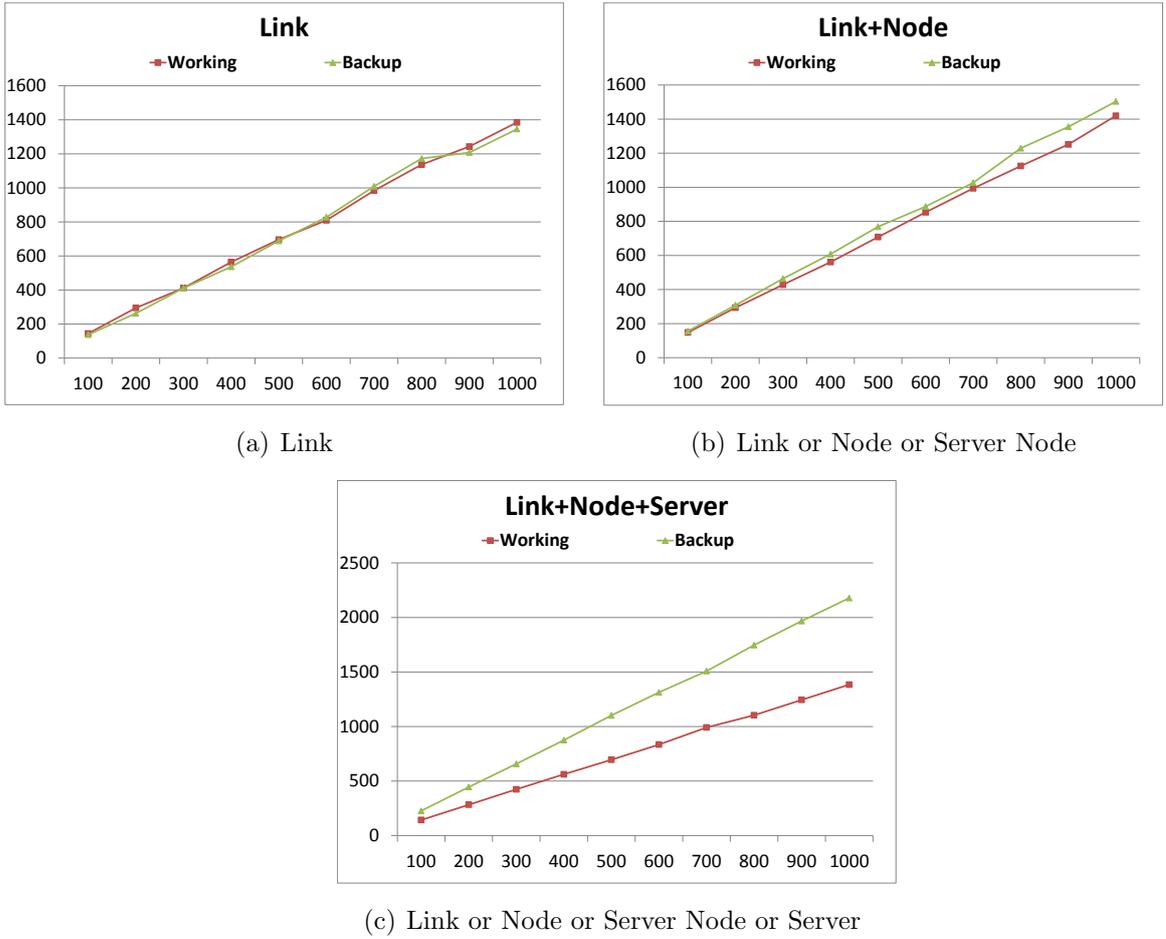


Figure 6.13: ILP solution for working and backup bandwidth units under Model I (5 Servers).

Figure 6.14 plots the overall bandwidth requirements for each failure scenario, where heights of each bar are averages over 10 randomly generated instances with the same number of requests. As expected from the analysis of the results in Figure 6.13, we observe that the single link or node failure scenario only use slightly more bandwidth than the single link failure scenario: on average, there is a 6% difference. However, relative differences, i.e., $(\tilde{z}_{ILP}^{III}(\text{link+node+server}) - \tilde{z}_{ILP}^{III}(\text{link})) / \tilde{z}_{ILP}^{III}(\text{link})$, increases to around 30% for the comparison with the single link/node/server failure scenario.

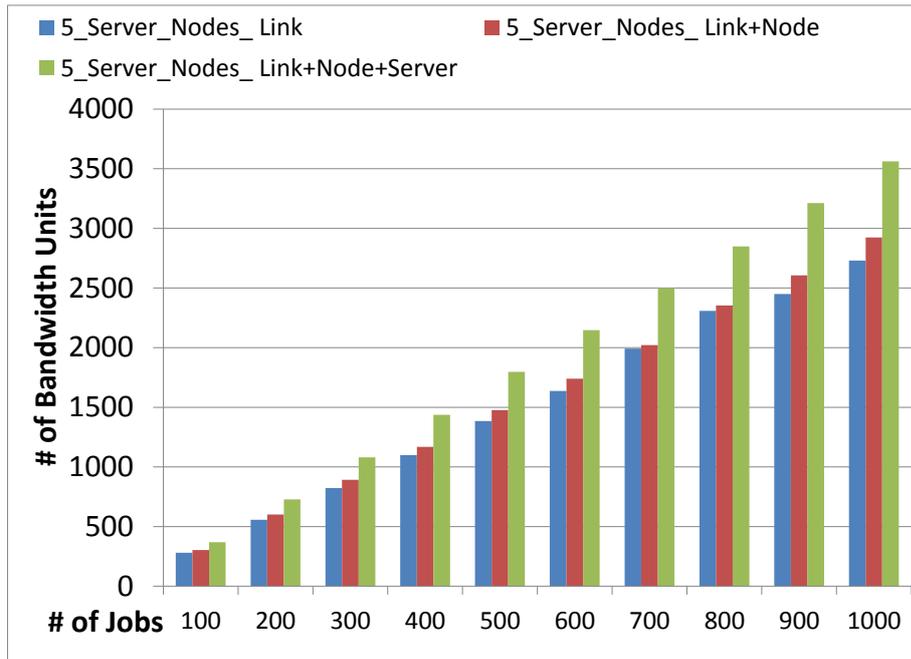


Figure 6.14: Different protection schemes - Model I - 5 server nodes.

We next present some results for 3 server nodes, again only for Model I, in Figure 6.15. Therein, the single link or node or server node, and the single link or node or server node or server failure scenarios require an average of 7% and 29%, respectively, extra bandwidth units than the single link failure scenario.

Comparison between 3 and 5 server locations shows that the case with 3 server locations uses an extra number of bandwidth units around 22%, 23%, and 22% over the requirements for 5 server locations in the single link, single link or node or server node, single link or node or server node or server failure scenario, respectively.

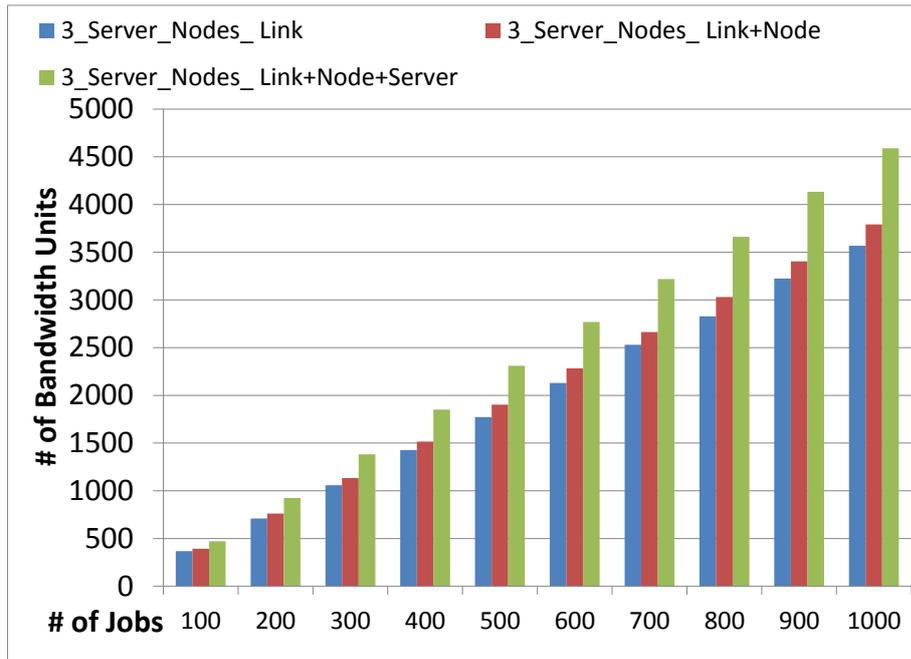


Figure 6.15: Different protection schemes - Model I - 3 server nodes.

6.5.6 Single Versus Multiple Backup Paths (5 Servers)

All Models

In order to further assess the accuracy of the best solutions using the lower and upper bounds obtained by the different models, we computed the following gaps, where we use Model III(a) for the comparison with Model III:

$$\text{GAP}_{\text{II vs. I}} = \frac{\text{ILP}_{\text{II}} - \text{ILP}_{\text{I}}}{\text{ILP}_{\text{I}}} \quad \text{GAP}_{\text{III vs. I}} = \frac{\text{ILP}_{\text{III}} - \text{ILP}_{\text{I}}}{\text{ILP}_{\text{I}}}.$$

Numerical values are described in Table 6.5, from which we deduce that the average accuracy varies from 0.02% to 9.85% depending on the number of traffic requests.

request size	GAP II vs. I	GAP III vs. I
100	8.75%	9.85%
200	8.74%	8.35%
300	9.70%	8.96%
400	9.11%	8.47%
500	6.91%	7.63%
600	7.64%	8.72%
700	2.24%	4.34%
800	0.02%	1.93%
900	5.77%	8.50%
1,000	5.91%	7.03%

Table 6.5: Some ILP gaps.

Model I

Recall that Model I allows multiple working and multiple backup paths for each source node. So we analyze the impact of allowing multiple paths to the same or different destination nodes under different protection schemes.

Results corresponding to averages over 10 randomly generated instances for each number of requests, are shown in Table 6.6 and Table 6.7. Each sub-table (a,b,c) corresponds to a specific failure scenario. Each column is associated with a number of distinct destinations for the working (or the backup) paths and the number of distinct working or backup paths (see the heading of the columns). For instance 2 destinations and 4 paths in Table 6.6 means that there are some source nodes with 4 different working paths (i.e., they pairwise differ by at least two links) and there are two distinct destinations (i.e., server nodes) for those four paths. In each column, i.e., a path pattern for source nodes, we indicate the percentage in terms of source

nodes served by this path pattern. The last column is a check that no path pattern has been forgotten, i.e., the sum of all percentages should be equal to 100%. The last row of each sub-table contains the average of each path pattern over the various sets of traffic instances (i.e., 100 instances).

As we can see in the different tables, the largest percentages correspond to: (i) the number of working paths with two different paths toward an identical destination and (ii) the number of backup paths with two different paths toward different (two) destinations, if we exclude the basic path pattern with a single path, both for the working and the backup path patterns.

It is also observed (as can be seen in the last column of each sub-table) that the single link and the single link/node failure scenarios do not differ much with respect to the number of bandwidth units (6%) as well as with respect to the ratio of multiple paths (3%). On the other hand, the single link/node/server failure scenario requires more bandwidth units (30%) and has more multiple paths (6% more for working and 10% for backup).

(a) Working Paths under Single Link Failure

# of dest.	1	1	1	1	2	2	3	2	3	4	Σ
# of paths	1	0	2	3	2	3	3	4	4	4	
Size	%										
100	58.3	17.9	11.5	1.2	6.3	2.8	0.0	2.0	0.0	0.0	100
200	51.2	17.9	12.3	2.0	10.3	6.0	0.0	0.4	0.0	0.0	100
300	53.2	17.9	15.0	1.8	9.6	2.1	0.0	0.4	0.0	0.0	100
400	42.5	17.9	16.3	5.2	8.7	6.3	0.4	2.0	0.4	0.4	100
500	42.1	17.9	19.4	4.4	9.1	4.0	0.8	0.8	0.8	0.8	100
600	45.7	17.9	16.8	4.3	7.9	5.0	0.0	1.8	0.4	0.4	100
700	50.4	17.9	15.6	0.9	9.4	3.6	0.4	1.8	0.0	0.0	100
800	62.5	17.9	9.5	0.0	5.4	2.4	0.0	1.2	0.6	0.6	100
900	49.3	17.9	15.7	1.4	9.6	3.6	0.4	1.4	0.4	0.4	100
1000	31.1	17.9	8.6	0.4	23.6	6.8	4.6	0.7	3.2	3.2	100
AVERAGE	48.6	17.9	14.1	2.1	10.0	4.3	0.7	1.2	0.6	0.6	100

(b) Working Paths under Single Link or Node Failure

# of dests.	1	1	1	1	2	2	3	2	3	4	Σ
# of paths	1	0	2	3	2	3	3	4	4	4	
Size	%										
100	54.3	17.9	7.9	2.1	10.4	6.1	0.4	1.1	0.0	0.0	100
200	47.9	17.9	11.4	2.9	11.1	6.8	0.4	1.8	0.0	0.0	100
300	45.4	17.9	15.4	1.8	9.3	6.1	0.7	3.6	0.0	0.0	100
400	45.4	17.9	12.1	1.8	12.5	7.1	0.7	1.8	0.4	0.4	100
500	42.9	17.9	16.1	3.9	11.8	6.1	0.0	0.7	0.4	0.4	100
600	42.5	17.9	12.1	2.9	15.4	7.5	0.4	0.7	0.4	0.4	100
700	40.4	17.9	14.3	5.0	13.6	5.0	1.1	2.9	0.0	0.0	100
800	41.3	17.9	16.7	2.4	12.7	7.1	0.8	0.4	0.4	0.4	100
900	43.2	17.9	12.9	2.1	13.2	6.8	0.4	2.1	0.7	0.7	100
1000	37.7	17.9	8.1	0.0	25.3	5.8	2.6	0.6	1.0	1.0	100
AVERAGE	44.1	17.9	12.7	2.5	13.5	6.4	0.7	1.6	0.3	0.3	100

(c) Working Paths under Single Link or Node or Server Failure

# of dests.	1	1	1	1	2	2	3	2	3	4	Σ
# of paths	1	0	2	3	2	3	3	4	4	4	
Size	%										
100	64.6	17.9	10.7	0.4	5.7	0.7	0.0	0.0	0.0	0.0	100
200	61.8	17.9	13.6	0.7	5.7	0.4	0.0	0.0	0.0	0.0	100
300	55.7	17.9	12.9	1.1	10.4	2.1	0.0	0.0	0.0	0.0	100
400	57.5	17.9	12.1	0.7	7.5	3.2	0.4	0.0	0.4	0.4	100
500	55.0	17.9	13.6	0.7	10.7	1.4	0.4	0.4	0.0	0.0	100
600	58.2	17.9	13.9	0.7	8.6	0.0	0.4	0.4	0.0	0.0	100
700	56.8	17.9	16.8	0.0	7.1	1.4	0.0	0.0	0.0	0.0	100
800	59.1	17.9	13.5	0.0	7.1	1.6	0.4	0.4	0.0	0.0	100
900	59.3	17.9	13.9	0.4	7.1	1.4	0.0	0.0	0.0	0.0	100
1000	49.0	17.9	5.5	0.3	24.0	0.6	1.9	0.0	0.3	0.3	100
AVERAGE	57.7	17.9	12.7	0.5	9.4	1.3	0.3	0.1	0.1	0.1	100

Table 6.6: Model I - Working paths - Link or node or server node or server protection scheme (5 servers).

(a) Backup Paths under Single Link Failure

# of dests.	1	1	1	1	2	2	3	2	3	4	Σ
# of paths	1	0	2	3	2	3	3	4	4	4	
Size	%										
100	49.3	17.9	10.0	0.4	3.6	3.6	1.8	1.1	1.4	1.4	100
200	42.1	17.9	10.7	0.0	17.1	5.6	3.2	1.2	1.2	1.2	100
300	42.9	17.9	10.0	0.7	19.6	6.8	1.8	0.4	0.0	0.0	100
400	27.4	17.9	9.1	1.6	19.4	8.7	5.6	0.8	4.8	4.8	100
500	32.5	17.9	7.9	0.0	23.4	7.1	3.6	2.8	2.4	2.4	100
600	33.9	17.9	9.3	0.4	19.3	7.1	3.9	1.8	3.2	3.2	100
700	46.0	17.9	6.7	0.8	19.0	4.4	1.6	2.0	0.8	0.8	100
800	59.9	17.9	7.5	1.2	9.1	0.8	1.2	0.8	0.8	0.8	100
900	35.0	17.9	12.1	1.1	23.9	3.2	2.5	0.7	1.8	1.8	100
1000	31.3	17.9	9.9	0.0	23.0	6.7	4.4	1.2	2.8	2.8	100
AVERAGE	40.0	17.9	9.3	0.6	17.8	5.4	2.9	1.3	1.9	1.9	100

(b) Backup Paths under Single Link or Node Failure

# of dests.	1	1	1	1	2	2	3	2	3	4	Σ
# of paths	1	0	2	3	2	3	3	4	4	4	
Size	%										
100	48.2	17.9	5.7	0.4	17.1	4.3	5.0	0.0	0.7	0.7	100
200	36.1	17.9	6.7	0.4	22.6	4.8	6.0	0.8	2.4	2.4	100
300	37.5	17.9	6.4	0.7	22.1	3.9	6.4	0.7	2.1	2.1	100
400	39.6	17.9	4.3	0.7	22.1	4.3	6.4	1.1	1.8	1.8	100
500	36.8	17.9	6.8	0.4	24.6	5.4	6.8	0.7	0.4	0.4	100
600	36.4	17.9	5.4	0.0	25.4	5.0	7.9	0.0	1.1	1.1	100
700	32.1	17.9	6.4	0.7	22.9	7.1	6.1	1.8	2.5	2.5	100
800	33.9	17.9	5.7	0.0	25.0	8.2	5.7	0.7	1.4	1.4	100
900	34.3	17.9	7.5	0.0	24.6	2.9	5.7	0.7	3.2	3.2	100
1000	37.9	17.9	7.5	0.0	24.6	6.4	2.9	0.7	1.1	1.1	100
AVERAGE	37.3	17.9	6.2	0.3	23.1	5.2	5.9	0.7	1.7	1.7	100

(c) Backup Paths under Single Link or Node or Server Failure

# of dests.	1	1	1	1	2	2	3	2	3	4	Σ
# of paths	1	0	2	3	2	3	3	4	4	4	
Size	%										
100	82.5	0.0	10.7	0.4	5.7	0.7	0.0	0.0	0.0	0.0	100
200	79.6	0.0	13.6	0.7	5.7	0.4	0.0	0.0	0.0	0.0	100
300	73.6	0.0	12.9	1.1	10.4	2.1	0.0	0.0	0.0	0.0	100
400	75.4	0.0	12.1	0.7	7.5	3.2	0.4	0.0	0.4	0.4	100
500	72.9	0.0	13.6	0.7	10.7	1.4	0.4	0.4	0.0	0.0	100
600	76.1	0.0	13.9	0.7	8.6	0.0	0.4	0.4	0.0	0.0	100
700	74.6	0.0	16.8	0.0	7.1	1.4	0.0	0.0	0.0	0.0	100
800	77.0	0.0	13.5	0.0	7.1	1.6	0.4	0.4	0.0	0.0	100
900	77.1	0.0	13.9	0.4	7.1	1.4	0.0	0.0	0.0	0.0	100
1000	66.9	0.0	5.5	0.3	24.0	0.6	1.9	0.0	0.3	0.3	100
AVERAGE	75.6	0.0	12.7	0.5	9.4	1.3	0.3	0.1	0.1	0.1	100

Table 6.7: Model I - Backup paths - Link or node or server node or server protection scheme (5 servers).

Chapter 7

Scheduling: Joint Optimization in Optical Grid Networks

In parallel programming or distributed computing environment, each job is decomposed into multiple tasks. To execute such tasks, the grid management systems need to allocate the computing resources and communication paths to them. The allocation of computing resources among multiple tasks is known as *task scheduling*. In task scheduling, task dependencies are represented in a Directed Acyclic Graph (DAG), also known as *task graph*. Some tasks can be executed in parallel if they are independent and if they are dependent, on the result of one another, they are executed on an incremental (one-by-one) basis.

Recall that, different approaches have been investigated for DAG scheduling based on heuristic algorithms (discussed in Chapter 3). In this study, we have developed exact Integer Linear Programming (ILP) formulations with classical ILP and Column

Generation (CG) ILP models, discussed next.

7.1 Optimization Models for sch-ACRWA

Now, we present classical ILP and column generation formulations followed by their numerical results. In these formulations, we assume there is no bandwidth bottleneck between the clients and the server nodes. We consider homogeneous optical grid networks, where all the server nodes (data centers) have the same type of resources. Additionally, we assume enough wavelengths on each fiber, so we only consider the data transfer time from one server node to another. A topology instance $G = (V, L, V^{\text{SVR}})$ is the same as discussed in Chapter 4, Section 4.1. Further, it is assumed that a single unique path between each pair of server nodes is predetermined. The resulting set of paths is represented by P between server nodes V^{SVR} , and each path indexed by $p^{v'v}$. Each link on the path is represented by $\ell \in p^{v'v}$. Job (tasks) instance notations used in the models are as follows:

Traffic instance input

$$G_{\text{DAG}_s} = (N, \text{PRED}, \text{EXEC}, \text{TRANSFER})$$

N set of tasks, indexed by n and its predecessor tasks set is denoted by PRED_n .

PRED matrix of links (precedence) between task nodes, if task n' is a predecessor of task n (denoted by $n' \in \text{PRED}_n$), then n' must be executed prior to task n .

EXEC vector of task's execution time indexed by EXEC_n that is estimated execution time of task n .

TRANSFER vector of data transfer time for each hop between the tasks. Each pair of task duration is indexed by $\text{TRANSFER}_{nn'}$, where task n' is a predecessor of task n .

7.1.1 Classical ILP Formulation

We have developed classical ILP formulation based on the following decision and auxiliary variables.

Decision Variables

$x \in \{0, 1\}$, where $x_{vn} = 1$ if task n is executed on server v , 0 otherwise.

$t^{\text{SPAN}} \in \mathbb{Z}^+$ is the finish time of the last task.

$t_{vn}^s \in \mathbb{Z}^+$ is start time of task n at server v .

$t_{vn}^f \in \mathbb{Z}^+$ is finish time of task n at server v .

Auxiliary Variables

$\delta \in \{0, 1\}$, where $\delta_{nn'}^{vv'} = 1$, if data transfer is required from server v' to server v when task n and its predecessor task $n' \in \text{PRED}_n$ are executed on v and v' (different servers), respectively, 0 otherwise.

$\alpha \in \{0, 1\}$, where $\alpha_{nn'}^v = 1$, if task n is finishing before task n' starts, or task n' is finishing before the starting of task n , at server v , otherwise $\alpha_{nn'}^v = 0$.

The objective is to minimize the schedule length, i.e., completion time of the last task.

$$\min t^{\text{SPAN}}. \quad (7.1)$$

The first set of constraints compute the schedule length, i.e., $t^{\text{SPAN}} = \max_{n \in N, v \in V} t_{vn}^{\text{F}}$.

$$t_{vn}^{\text{F}} \leq t^{\text{SPAN}} \quad n \in N, v \in V^{\text{SVR}}. \quad (7.2)$$

The second set of constraints computes the finish time for each task n and it is calculated based on the given estimated execution time EXEC_n .

$$t_{vn}^{\text{S}} + \text{EXEC}_n x_{vn} = t_{vn}^{\text{F}} \quad n \in N, v \in V^{\text{SVR}}. \quad (7.3)$$

The third set of constraints ensures that each task n must be served by at least one server ($v \in V^{\text{SVR}}$).

$$\sum_{v \in V^{\text{SVR}}} x_{vn} \geq 1 \quad n \in N. \quad (7.4)$$

The next five sets of constraints ensure the task precedencies:

If successor and predecessor tasks are executed on the same server, then successor task should be started after finishing time of all its predecessor tasks,

i.e., $t_{vn}^{\text{S}} \geq \max_{n' \in \text{PRED}_n} t_{vn'}^{\text{F}}$.

$$t_{vn}^{\text{S}} \geq t_{vn'}^{\text{F}} \quad n, n' \in N : n' \in \text{PRED}_n ; v \in V^{\text{SVR}}. \quad (7.5)$$

Otherwise, successor and predecessor tasks are executed on different servers. Constraints (7.6) to (7.8) ensure the communication path, if task n is executed on server v and its $n' \in \text{PRED}_n$ is executed on server v' .

$$x_{vn} + x_{v'n'} - 1 \leq \delta_{nn'}^{vv'} \quad n, n' \in N : n' \in \text{PRED}_n ; v, v' \in V^{\text{SVR}} : v \neq v' \quad (7.6)$$

$$\delta_{nn'}^{vv'} \leq x_{vn} \quad n, n' \in N : n' \in \text{PRED}_n ; v, v' \in V^{\text{SVR}} : v \neq v' \quad (7.7)$$

$$\delta_{nn'}^{vv'} \leq x_{v'n'} \quad n, n' \in N : n' \in \text{PRED}_n ; v, v' \in V^{\text{SVR}} : v \neq v'. \quad (7.8)$$

Note that, constraints (7.7) and (7.8) are redundant due to the minimization of $t_{v'n'}^{\text{F}}$ through constraints (7.9).

Consequently, constraints (7.9) ensure task n must be started after finishing time of $n' \in \text{PRED}_n$ plus data transfer time ($\text{TRANSFER}_{nn'}^{vv'}$).

$$t_{vn}^{\text{S}} \geq \text{TRANSFER}_{nn'}^{vv'} \delta_{nn'}^{vv'} + t_{v'n'}^{\text{F}} \quad n, n' \in N : n' \in \text{PRED}_n ; v, v' \in V^{\text{SVR}} : v \neq v', \quad (7.9)$$

where, $\text{TRANSFER}_{nn'}^{vv'}$ is the total transfer time from server v' (output of task n') to v (input for task n), i.e., $\text{TRANSFER}_{nn'}^{vv'} = |p^{v'v}| * \text{TRANSFER}_{nn'}$.

The last two sets of constraints ensure there is no overlap for the execution time between two tasks on the same server. We now consider a pair (n, k) of tasks with

no precedence relation, i.e., $n, k \in N : k \notin \text{PRED}_n$ and $n \notin \text{PRED}_k$.

$$t_{vn}^F < t_{vk}^S + M(1 - \alpha_{nk}^v) \quad n, k \in N : n \neq k, v \in V^{\text{SVR}} \quad (7.10)$$

$$t_{vk}^F < t_{vn}^S + M\alpha_{nk}^v. \quad n, k \in N : n \neq k, v \in V^{\text{SVR}}. \quad (7.11)$$

7.1.2 Column Generation (CG) Formulation

We have also investigated a column generation framework; original problem is decomposed into two subproblems, called Master Problem and Pricing Problem. These formulations are solved alternatively until the stopping optimality condition is satisfied, as discussed next.

Master Problem

Master Problem (MP) is a first component of original decomposed problem, and the objective is also the same as in the classical ILP. In addition to the topology and traffic instances notations, we use the following decision and auxiliary variables. A configuration is a potential allocation of tasks to a given node and is denoted by c . Let C be the overall set of configurations: $C = \bigcup_{v \in V} C_v$, where C_v is the set of configurations associated with $v \in V^{\text{SVR}}$. Moreover, we also define variables of MP and the configuration parameters which are feedback from the pricing problem and input to the MP.

Configuration Parameters

a $\in \{0, 1\}$, where $a_n^c = 1$, if task n is allocated in configuration $c \in C_v$,
0 otherwise.

t^S $\in \mathbb{Z}^+$, where $t_n^{S,c}$ is the starting time of task n in configuration
 $c \in C_v$ associated with server v .

t^F $\in \mathbb{Z}^+$, where $t_n^{F,c}$ is the finishing time of task n in configuration
 $c \in C_v$ associated with server v .

Decision Variables

z $\in \{0, 1\}$ with $z_c = 1$, if configuration $c \in C_v$ is selected for server
node $v \in V^{\text{SVR}}$, 0 otherwise.

t^{SPAN} $\in \mathbb{Z}^+$ is the completion time of the last task.

Auxiliary Variables

δ $\in \{0, 1\}$ with $\delta_{nn'}^{vv'} = 1$, if data transfer is required from server v'
to server v when task n and its predecessor task $n' \in \text{PRED}_n$ are
executed on v and v' , respectively, 0 otherwise.

The objective is to minimize the completion time of the last task:

$$\min t^{\text{SPAN}}. \tag{7.12}$$

The first set of constraints ensures that we do not select more (unit server) configurations than the number of servers ($V^{\text{SVR}} \subseteq V$):

$$\sum_{c \in C} z_c \leq |V^{\text{SVR}}|. \quad (7.13)$$

The second set of constraints ensures that at most one configuration can be selected for each server ($v \in V^{\text{SVR}}$):

$$\sum_{c \in C_v} z_c \leq 1 \quad v \in V^{\text{SVR}}. \quad (7.14)$$

The third set of constraints makes sure that each task has been assigned to at least one server:

$$\sum_{c \in C} a_n^c z_c \geq 1 \quad n \in N. \quad (7.15)$$

Note that, in practice, this inequality will be satisfied as an equality due to the objective function. We express as an inequality in order to ease the solution process.

The next set of constraints computes the schedule length (or span).

$$\sum_{c \in C} t_n^{\text{F},c} z_c \leq t^{\text{SPAN}} \quad n \in N. \quad (7.16)$$

Data transfer from one server to the next: if task n is run on server v and follows a prior task $n' \in \text{PRED}_n$, which was executed on server v' , it requires a communication

path from v' to v : For all $n, n' \in N : n' \in \text{PRED}_n, v \neq v', v, v' \in V^{\text{SVR}}$,

$$\sum_{c \in C_v} a_n^c z_c + \sum_{c \in C_{v'}} a_{n'}^c z_c - 1 \leq \delta_{nn'}^{vv'} \quad (7.17)$$

$$\delta_{nn'}^{vv'} \leq \sum_{c \in C_v} a_n^c z_c \quad (7.18)$$

$$\delta_{nn'}^{vv'} \leq \sum_{c \in C_{v'}} a_{n'}^c z_c. \quad (7.19)$$

Task precedence constraints: if there is a precedence constraint between two tasks, we need to ensure that the second task is executed after the first one has ended, with its data transferred if necessary:

$$\sum_{c \in C} t_n^{S,c} z_c \geq \sum_{c \in C} t_{n'}^{F,c} z_c + \sum_{v \in V^{\text{SVR}}} \sum_{v' \in V^{\text{SVR}}: v \neq v'} \text{TRANSFER}_{nn'}^{vv'} \delta_{nn'}^{vv'} \quad (7.20)$$

$n, n' \in N : n' \in \text{PRED}_n.$

Pricing Problem:

Each Pricing Problem (PP) corresponds to a single server. The PP generates potential and promising configuration for the master problem based on the dual values that are output by the solution of the last restricted master problem. In this configuration, each accepted task is assigned non-overlapped time (i.e., starting time and its duration) under the constraints of precedence. The configuration parameters a , t^S , and t^F defined in the MP are variables in the pricing problem. Further, $\alpha \in \{0, 1\}$, where $\alpha_{nn'} = 1$ if task n is executed before task n' under the assumption that both

tasks are executed on the same server, 0 otherwise.

The reduced cost objective is written as follows:

$$\begin{aligned}
\min \quad & u^{(7.13)} - u_{v^{\text{svr}}}^{(7.14)} - \sum_{n \in N} u_n^{(7.15)} a_n - \sum_{n \in N} u_n^{(7.16)} t_n^{\text{F}} \\
& - \sum_{v \in V^{\text{svr}}} \sum_{v' \in V^{\text{svr}}:v' \neq v} \sum_{n \in N} \sum_{n' \in N:n' \in \text{PRED}_n} \left(u_{vv'nn'}^{(7.17)} (a_n - a_{n'}) \right) - u_{vv'nn'}^{(7.18)} a_n - u_{vv'nn'}^{(7.19)} a_{n'} \\
& - \sum_{n \in N} \sum_{n' \in N:n' \in \text{PRED}_n} u_{nn'}^{(7.20)} (t_n^{\text{S}} - t_{n'}^{\text{F}}) \quad (7.21)
\end{aligned}$$

where, $u^{(7.13)} \leq 0$, $u_{v^{\text{svr}}}^{(7.14)} \leq 0$, $u_n^{(7.15)} \geq 0$, $u_n^{(7.16)} \leq 0$, $u_{vv'nn'}^{(7.17)} \leq 0$, $u_{vv'nn'}^{(7.18)} \geq 0$, $u_{vv'nn'}^{(7.19)} \geq 0$, $u_{nn'}^{(7.20)} \geq 0$, are dual values of the Restricted Master Problem (RMP).

Selection of the tasks to be executed on the server of the configuration If task n is not allocated in this configuration then its starting and finishing time must be zero.

$$t_n^{\text{S}} \leq M a_n \quad n \in N \quad (7.22)$$

$$t_n^{\text{F}} \leq M a_n \quad n \in N. \quad (7.23)$$

Precedence constraints If task $n' \in \text{PRED}_n$ is executed then it must be finished before the starting of task n .

$$t_{n'}^{\text{F}} \leq t_n^{\text{S}} \quad n, n' \in N : n' \in \text{PRED}_n. \quad (7.24)$$

Task ordering constraints We now consider a pair (n, k) of tasks with no precedence relation, i.e., $n, k \in N : k \notin \text{PRED}_n$ and $n \notin \text{PRED}_k$. This means, there should not overlap between the tasks that have no precedence/successor relationship.

$$t_n^F \leq t_k^S + M(1 - \alpha_{nk}) \quad n, k \in N : n \neq k \quad (7.25)$$

$$t_k^F \leq t_n^S + M\alpha_{nk} \quad n, k \in N : n \neq k. \quad (7.26)$$

Finish Time It can be calculated with starting time and given estimated execution time for each task n .

$$t_n^S + \text{EXEC}_n a_n = t_n^F \quad n \in N. \quad (7.27)$$

Solution of the CG-ILP Formulation

In order to solve the master and the pricing problem alternatively through the column generation technique, we must initialize the feasible configurations to the restricted master problem. To this end, we have developed a heuristic algorithm, which we referred to as Init-Heu, presented in Algorithm 7.1. We are given a set of tasks denoted by $n \in N$ and expected execution time EXEC. Each task may have some precedence tasks (PRED_n) with the assumption of that task n_i predecessors can be $n_{i-\ell}, n_{i-m}$, and so on. This greedy algorithm takes the task sequentially and assign to the sever based on the server availability and their predecessor tasks servers.

After the insertion of the initial set of configurations, the CG-ILP alternative

approach that is used is the one presented in Chapter 4, Algorithm 4.2.

7.2 Numerical Results

In this section, first we discuss network topology and traffic instances followed by experimental results.

7.2.1 Experimental Setup

Network Topology: European Network with five server locations (i.e., London, Lyon, Zurich, Berlin, Vienna).

Traffic Instances: We have created traffic instances through the algorithm 7.2, with the following parameters:

- Instance size (number of tasks) from 10 (*sizeIncr*) to 100 (*sizeTo*) with the interval of 10 (*sizeIncr*). An estimated execution time for each task is randomly generated between 10 (MIN_CPU) and 100 (MAX_CPU).
- Total number of dependencies 25% to 200% with the interval of 25%, e.g., instance size of 20 have dependencies of 5 (*TotalDepIncr*), 10, 15, ... 40 (*TotalDepTo*). A communication time (data transfer for each hop) from predecessor to successor task is randomly generated between 5 (MIN_BW) and 15 (MAX_BW). Each node has a limit of maximum 4 predecessor and 4 successors.

Algorithm 7.1 Init-Heu

1: **input parameters:** topology (G) and traffic (G_{DAG_s}) instance.
2: **comment:** *Initialization*
3: $assServer[] := -1$
4: $serverAvail[] := 0$
5: **comment:** *Assign non-overlapping time to each task n*
6: **for** each task n in G_{DAG} **do**
7: $t_n^S = \infty$
8: **comment:** *Find earliest available starting time on the server v under the precedence constraints*
9: **for** each server v in V^{SVR} **do**
10: Determine the predecessor task k with $\max_{n' \in \text{PRED}_n} t_{n'}^F$ and its server v'
11: **if** $v = v'$: same servers **then**
12: $drt := t_k^F$
13: **else**
14: $drt := t_k^F + \text{TRANSFER}_{nn'}^{vv'}$
15: **end if**
16: $t_n^S := \min\{t_n^S, \max(\text{serverAvail}[v], drt)\}$
17: update the $selServer$ for n .
18: **end for**
19: **comment:** *save earliest available starting time*
20: $t_n^F := t_n^S + \text{EXEC}_n$;
21: $serverAvail[\text{selServer}] = t_n^F$
22: $assServer[n] = \text{selServer}$
23: **end for**
24: **Output:** t_n^S and t_n^F and assigned server $assServer[n]$ for each task.

Algorithm 7.2 Traffic Generator

1: **input parameters:** $sizeIncr$, $sizeTo$, $TotalDepIncr$, $TotalDepTo$
2: **for** $size = sizeIncr \rightarrow sizeTo$ increment by $sizeIncr$ **do**
3: taskGraph G_{DAG_s} ;
4: $execList :=$ generate CPU time for $sizeIncr$ tasks between MIN_CPU and MAX_CPU
5: Add $execList$ to G_{DAG_s}
6: **for** $TotalDep := TotalDepIncr \rightarrow TotalDepTo$ increment by $TotalDepIncr$ **do**
7: $depList :=$ Generate $TotalIncrDep$ dependencies between MIN_BW and MAX_BW
8: Add $depList$ to G
9: **Output:** write G_{DAG_s} as an instance of $size$, $TotalDep$
10: **end for**
11: **end for**

7.2.2 Results

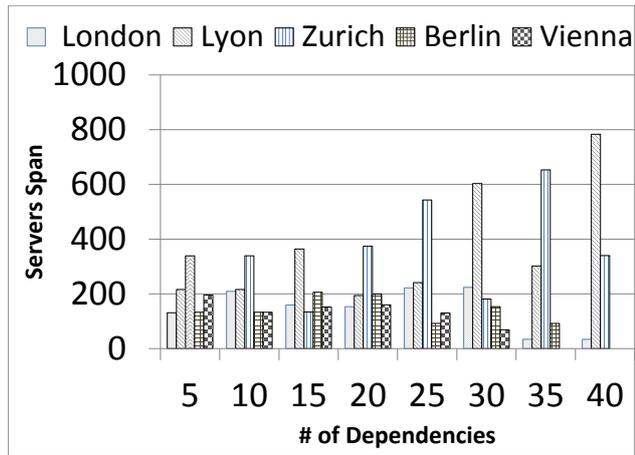
We have simulated different instances on the classical ILP and Column Generation (CG) formulations, results are shown in 7.1. The first column represents traffic instance including size and total number of dependencies. Columns 2 and 3 indicate CPU execution time and optimal LP solution, respectively, through classical LP solution. Columns 4 and 5 indicate CPU execution time and optimal ILP solution, respectively, through classical ILP solution. Column 6 to 10 are results of CG solution, where col. 6 indicates CPU time, 7 number of generated configurations, 8 LP optimal result, 9 an ILP solution using CPLEX engine for integer solution, and 10 gap between LP and ILP of CG solution, and it is calculated as follows.

$$Gap = \frac{\tilde{z}_{ILP} - z_{LP}^*}{z_{LP}^*}$$

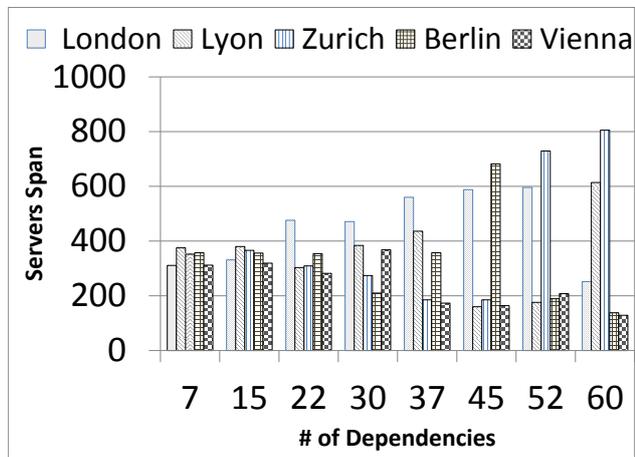
As can be observed from the Table 7.1, classical LP solution successfully solves all instances with very small CPU times, where 0 indicates less than a second. As a contrast, the classical ILP successfully solves all instances up to 20 tasks, while 30 tasks instances, it executes some instances but could not solve other instances due to limited memory. CG-ILP formulation solves upto size of 50 tasks with integrality gap issue mostly in the less number of dependencies, as shown in Table 7.1. The execution time (CPU) depends on the number of generated configurations, i.e, generating more configurations takes longer CPU times. If we compare Classical LP and CG-LP linear programming relaxation lower bounds, the CG-LP gives good lower bound because

classical LP is solved with all continuous (non-integer) variables while CG-LP consists of continuous variables in the restricted master problem and integer variables in the pricing problem.

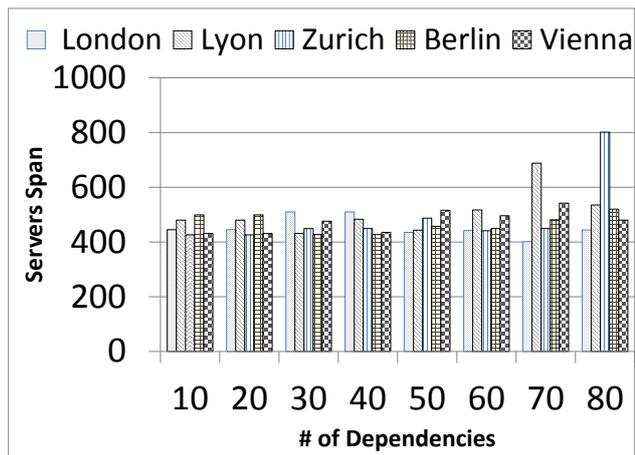
We have also determined the span of different size instances, such as 20, 30, and 40 shown in Figure 7.1. In this figure, x-axis represents the number of dependencies and y-axis represents servers length (span) corresponding to their number of tasks. It can also be observed that as the size increases their schedule length (highest span server) marginally increases specially in higher number of dependencies, but other servers length significantly increase, see e.g., size of 30 with 60 (Fig. 7.1(b)) and 40 with 80 (Fig. 7.1(c)) dependencies. These preliminary results have been submitted in Asia Communications and Photonics Conference (ACP-2013).



(a) Size of 20



(b) Size of 30



(c) Size of 40

Figure 7.1: Column Generation each server span of of different size and dependencies.

Instance	Classical LP		Classical ILP		Column Generation				
	CPU	LP	CPU	ILP	CPU	Conf.	LP	ILP	Gap
1	2	3	4	5	6	7	8	9	10
10-2	0	27.8	1	139	1	24	139	161	16%
10-5	0	30	1	150	0	11	150	150	0%
10-7	0	37.6	1	188	1	20	188	188	0%
10-10	0	40.8	1	204	2	24	204	210	3%
10-12	0	47.6	4	242	2	25	238	250	5%
10-15	0	69	1	345	2	30	345	346	0%
10-17	0	92.8	18	464	2	28	464	464	0%
10-20	0	92.8	9	464	2	28	464	464	0%
20-5	0	48	51	240	6	52	240	339	41%
20-10	0	67.8	14	339	4	36	339	339	0%
20-15	0	67.8	18	339	4	37	339	364	7%
20-20	0	74.8	21	374	6	41	374	374	0%
20-25	0	101.2	144	507	7	44	506	543	7%
20-30	0	119.6	27	598	9	48	598	603	1%
20-35	0	130.6	102	653	9	56	653	653	0%
20-40	0	156.6	5822	783	9	52	783	783	0%
30-7	0	43.4			27	89	375	375	0%
30-15	0	43.4			131	223	242	380	57%
30-22	0	53.8			679	344	287	476	66%
30-30	0	89.6	190	448	26	82	448	471	5%
30-37	0	107.4	325	537	26	80	537	560	4%
30-45	0	122			180	230	610	682	12%
30-52	0	135	666	675	32	84	675	729	8%
30-60	0	161	249	805	18	57	805	805	0%
40-10	0	37.2			3362	901	281	499	77%
40-20	0	41			12855	1815	289	499	73%
40-30	0	62			137	141	483	510	6%
40-40	0	62			6673	704	310	510	65%
40-50	1	62			3676	453	323	516	60%
40-60	1	66.4			5472	542	348	517	48%
40-70	1	130.4			58	97	652	688	6%
40-80	1	152.4			79	105	762	801	5%
50-12	0	55.4			19126	1162	343	625	82%
50-25	0	57.8			445	174	619	619	0%
50-37	0	122			83	103	610	800	31%
50-50	0	122			180	133	610	800	31%
50-62	0	122			181	130	610	800	31%
50-75	2	132.2			171	138	661	886	34%
50-87	1	169.2			589	284	846	1056	25%
50-100	4	169.2			2876	633	846	1066	26%

Table 7.1: Classical ILP and Column Generation Results (CPU is in seconds).

Chapter 8

Conclusion and Future Work

This chapter highlights the main contributions of this thesis, draw the conclusion of the thesis, and outline some possible future work.

8.1 Summary

The main goal of this research was to develop exact optimization models for dimensioning optical grid networks, provisioning job requests, and task scheduling. In this context, new scalable ILP models based on a column generation method have been developed in order to investigate different protection schemes. This work started with the CSP and went through step by step further to the SPR-A protection scheme in optical grids. The objective of these models is to minimize the network capacity while finding the working and backup paths. By splitting the traditional ILP formulation into a Restricted Master Problem and a Pricing Problem, column generation is able to

handle large network instances. Numerical results on the European Network topology have shown that CG leads to a very good approximation of the optimal result.

In the next step, we have investigated the provisioning and the dimensioning of optical grid networks under the maximization of the grade of services. We proposed an exact and scalable optimization model for off-line problems, and different link transport capacity calculation methods. Using the proposed large scale optimization model, we studied the impact of the number of server nodes on the European and the Germany network topologies. We compared the grade of services under different protection schemes, and observed that their values did not differ much. We also investigated the impact of different link capacity methods on the bandwidth requirements and the grade of services.

In the third step, we proposed and compared three mathematical models, with three different decomposition schemes in order to address optimization gap issues. These models jointly optimize the location of the servers while provisioning the working and backup paths. Unlike the previous models, the location of the server (data center) nodes are not given, and the optimization model finds the locations of the servers. We have also conducted experiments on different levels of protection schemes, including link, node, and server protections.

In the final step, we have developed task scheduling optimization models based on the DAG task scheduling, where some tasks are dependent on other tasks.

8.2 Conclusion

This thesis investigated different problems related to dimensioning optical grid networks, provisioning job requests, and task scheduling. It is concluded that the CC-ILP optimization models for link dimensioning with the objective of either minimizing the bandwidth units or maximization the grade of service, give better solutions as compared to the classical ILP and heuristics. These models are able to solve up to optimal or near to optimal integer solutions. Moreover, joint optimization of computing and networks resources with exact CG-ILP formulations solve the scalable practical-size problems. However, their integer solutions do not provide the optimal solutions. In the context of DAG scheduling for optical grids networks, the exact CG-ILP formulations are not able to solve practical-size problems.

At the end, we can conclude that the link dimensioning and efficient job provisioning (grade of service) proposed models, with the objective min-ACRWA and max-ACRWA, respectively, are efficient tools to solve practical size instances with an optimal integer (we assume less than 1% gap is an optimal) solution. The computing time is reasonable for the advance reservation systems, management, and planning of optical grid networks. On the other hand, link dimensioning including locations of servers models need some improvements with respect to the integer solutions, and job scheduling models also need some integer solution improvements in order to be scalable for practical sized problems.

8.3 Future Work

Recall that all the related, as well as our proposed work about dimensioning optical grid networks are done under the assumption of optical nodes with wavelength conversion capability. In the future, our proposed minimization and maximization models can be extended under the wavelength continuity assumption where the same wavelength will be allocated through all the hops of the paths. Joint optimization of dimensioning optical grid networks, where the location of servers and paths are optimized together, needs hybrid solutions of ILP and heuristics in order to increase the accuracy of the solutions. Similarly, in order to resolve the scalability issue in the task scheduling, hybrid solutions need to be developed as well. This task scheduling model can also be extended by the link dimensioning with time constraints among the server nodes for task dependency while allocating the non-overlapping times on the server nodes.

8.4 Publications

1. B. Jaumard, J. Buysse, A. Shaikh, M.D. Leenheer, and C. Develder. “Column generation for dimensioning resilient optical grid networks with relocation,” in IEEE Global Communications Conference (IEEE GLOBECOM), 2010 [28].
2. A. Shaikh, J. Buysse, B. Jaumard, and C. Develder. “Anycast routing for survivable optical grids: Scalable solution methods and the impact of relocation,” Journal of Optical Communications and Networking (JOCN), 2011 [58].

3. C. Develder, J. Buysse, A. Shaikh, B. Jaumard, M. D. Leenheer, and B. Dhoedt. “Survivable optical grid dimensioning: Anycast routing with server and network failure protection,” in IEEE International Conference on Communications (ICC), 2011 [19].
4. B. Jaumard and A. Shaikh, “Maximizing access to IT services on resilient optical grids,” in Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), 2011 [34].
5. A. Shaikh and B. Jaumard, “Optimized dimensioning of resilient optical grids with respect to Grade of IT Services,” in Telecommunication Systems, 2012, Accepted.
6. B. Jaumard, A. Shaikh, and C. Develder, “Selecting the best locations for data centers in resilient optical grid/cloud dimensioning”, in International Conference on Transparent Optical Networks (ICTON), 2012 [35].
7. B. Jaumard, A. Shaikh, and C. Develder, “ Best server location and dimensioning of optical grid networks” in European Journal of Operational Research (EJORS), submitted.
8. A. Shaikh and B. Jaumard, “Task assignment and scheduling in optical grid”, submitted in Asia Communications and Photonics Conference (ACP-2013).

Bibliography

- [1] Cisco global cloud index: Forecast and methodology, 2011 - 2016.
http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns1175/Cloud_Index_White_Paper.html/.
- [2] What is grid computing.
<http://www.gridcafe.org/what-is-the-grid.html/>.
- [3] T.L. Adam, K.M. Chandy, and J.R. Dickson. A comparison of list schedules for parallel processing systems. *Communication of the ACM*, 17(12):685–690, Dec. 1974.
- [4] T. Agarwal, A. Sharma, A. Laxmikant, and L.V. Kale. Topology-aware task mapping for reducing communication contention on large parallel machines. In *20th International Parallel and Distributed Processing Symposium (IPDPS)*, pages 1–10, Apr. 2006.
- [5] C. Barnhart, E.L. Johnson, G.L. Nemhauser, M.W.P. Savelsbergh, and P.H. Vance. Branch-and-price: Column generation for solving huge integer programs. *Operations Research*, 46(3):316–329, Jun. 1998.

- [6] J. Buysse, M.D. Leenheer, C. Develder, and B. Dhoedt. Exploiting relocation to reduce network dimensions of resilient optical grids. In *7th International Workshop Design of Reliable Communication Networks (DRCN)*, pages 100–106, Oct. 2009.
- [7] J. Buysse, M.D. Leenheer, B. Dhoedt, and C. Develder. On the impact of relocation on network dimensions in resilient optical grids. In *14th Conference on Optical Network Design and Modeling (ONDM)*, pages 1–6, Feb. 2010.
- [8] A. Capone and G. Carello. Scheduling optimization in wireless mesh networks with power control and rate adaptation. In *3rd Annual IEEE Communications Society on Sensor and Ad Hoc Communications and Networks (SECON)*, volume 1, pages 138–147, Sep. 2006.
- [9] J. Chakareski. Joint optimization of flow allocation and data center placement in multi-service networks. In *19th International on Packet Video Workshop (PV)*, pages 101–106, May 2012.
- [10] N. Charbonneau and V. Vokkarane. A survey of advance reservation routing and wavelength assignment in wavelength-routed WDM networks. *IEEE Communications Surveys & Tutorials*, pages 1–28, Nov. 2011.
- [11] D. Chen and R. Chen. New relaxation-based algorithms for the optimal solution of the continuous and discrete p -center problems. *Computers & Operations Research*, 36(5):1646–1655, 2009.

- [12] Y. Chen, A. Bari, and A. Jaekel. Strategies for fault tolerance in optical grid networks. In *Computing, Networking and Communications (ICNC)*, pages 628–632, Jan. 2012.
- [13] I. Chlamtac, A. Ganz, and G. Karmi. Lightpath communications: An approach to high bandwidth optical WAN's. *IEEE Transactions on Communications*, 40(7):1171–1182, Jul. 1992.
- [14] M. Chtepen, F.H.A. Claeys, B. Dhoedt, F.D. Turck, P. Demeester, and P.A. Vanrolleghem. Adaptive task checkpointing and replication: Toward efficient fault-tolerant grids. *IEEE Transactions on Parallel and Distributed Systems*, 20(2):180–190, Feb. 2009.
- [15] V. Chvatal. *Linear Programming*. W.H. Freeman and Company, New York/San Francisco, 1983.
- [16] A. Concaro, G. Maier, M. Martinelli, A. Pattavina, and M. Tornatore. QoS provision in optical networks by shared protection: An exact approach. In *Quality of Service in Multiservice IP Networks*, volume 2601, pages 419–432, 2003.
- [17] T. DeFanti, M. Brown, J. Leigh, O. Yu, E. He, J. Mambretti, D. Lillothun, and J. Weinberger. Optical switching middleware for the optiputer. *IEICE Trans. Fundamentals/Commun./Electon./Ing. & Syst.*, E86-B(8):2263–2272, Aug. 2003.

- [18] C. Develder, J. Buysse, B. Dhoedt, and B. Jaumard. Joint dimensioning of server and network infrastructure for resilient optical grids/clouds. *IEEE/ACM Transactions on Networking*, Submitted 2013.
- [19] C. Develder, J. Buysse, A. Shaikh, B. Jaumard, M. D. Leenheer, and B. Dhoedt. Survivable optical grid dimensioning: Anycast routing with server and network failure protection. In *IEEE International Conference on Communications (ICC)*, pages 1–5, Jun. 2011.
- [20] C. Develder, B. Dhoedt, B. Mukherjee, and P. Demester. On dimensioning optical grids and the impact of scheduling. *Photonic Network Communications*, 17(3):255–265, Jun. 2009.
- [21] C. Develder, M.D. Leenheer, T. Stevens, B. Dhoedt, F.D. Turck, and P. Demester. Scheduling in optical grids: A dimensioning point of view. In *International Conference on the Optical Internet - Australian Conference on Optical Fiber Technology (COIN-ACOFT)*, Jun. 2007.
- [22] R. Dutta and G.N. Rouskas. A survey of virtual topology design algorithms for wavelength routed optical networks. *Optical Networks Magazine*, 1(1):73–89, Jan. 2000.
- [23] S. Figueira, N. Kaushik, and S. Naiksatam. Advance reservation of lightpaths in optical-network based grids. In *IEEE First International Conference on Broadband Networks (BROADNETS)*, pages 985–992, May 2004.

- [24] I. Foster and C. Kesselman, editors. *The grid: blueprint for a new computing infrastructure*. Morgan Kaufmann, San Francisco, CA, USA, 2nd edition, 2004.
- [25] O. Gerstel, M. Jinno, A. Lord, and S.J.B. Yoo. Elastic optical networking: a new dawn for the optical layer? *IEEE Communications Magazine*, 50(2):s12 –s20, Feb. 2012.
- [26] W. Guo, Z. Wang, Z. Sun, W. Sun, Y.Jin, W. Hu, and C. Qiao. Task scheduling accuracy analysis in optical grid environments. *Photonic Network Communications*, 17(3):209–217, Jan. 2009.
- [27] B. Jaumard, N.N. Bhuiyan, S. Sebbah, F. Huc, and D. Coudert. A new framework for efficient shared segment protection scheme for WDM networks. In *High Performance Switching and Routing (HPSR)*, 2010.
- [28] B. Jaumard, J. Buysse, A. Shaikh, M.D. Leenheer, and C. Develder. Column generation for dimensioning resilient optical grid networks with relocation. In *IEEE Global Communications Conference (IEEE GLOBECOM)*, pages 1–6, Dec. 2010.
- [29] B. Jaumard and C. Develder. Dimensioning resilient optical cloud networks. *IEEE Communication Magazine*, Submitted 2013.
- [30] B. Jaumard, C. Meyer, and B. Thiongane. ILP formulations for the RWA problem for symmetrical systems. In P. Pardalos and M. Resende, editors, *Handbook*

- for *Optimization in Telecommunications*, chapter 23, pages 637–678. Kluwer, 2006.
- [31] B. Jaumard, C. Meyer, and B. Thiongane. Comparison of ILP formulations for the RWA problem. *Optical Switching and Networking*, 4(3):157–172, Nov. 2007.
- [32] B. Jaumard, C. Meyer, and B. Thiongane. On column generation formulations for the RWA problem. *Discrete Applied Mathematics*, 157(6):1291–1308, Mar. 2009.
- [33] B. Jaumard, C. Meyer, B. Thiongane, and X. Yu. ILP formulations and optimal solutions for the RWA problem. In *IEEE Global Communications Conference (IEEE GLOBECOM)*, pages 1918–1924, Dec. 2004.
- [34] B. Jaumard and A. Shaikh. Maximizing access to IT services on resilient optical grids. In *3rd International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, Oct. 2011.
- [35] B. Jaumard, A. Shaikh, and C. Develder. Selecting the best locations for data centers in resilient optical grid/cloud dimensioning. In *14th International Conference on Transparent Optical Networks (ICTON)*, pages 1–4, Jul. 2012.
- [36] Y. Jin, Y. Wang, W. Guo, W. Sun, and W. Hu. Joint scheduling of computation and network resource in optical grid. In *International Conference on Information, Communication and Signal Processing*, pages 1–5, Dec. 2007.

- [37] A. Juttner, T. Cinkler, and B. Dezsó. A randomized cost smoothing approach for optical network design. In *9th International Conference on Transparent Optical Networks (ICTON)*, pages 75–78, Jul. 2007.
- [38] N. Kannasoot and P.J. Jason. Resource-efficient task assignment and scheduling in optical grids. In *Optical Fiber Communication, Collocated National Fiber Optic Engineers Conference (OFC/NFOEC)*, pages 21–25, 2010.
- [39] J. Kuri, N. Puech, M.Gagnaire, E. Dotaro, and R. Douville. Routing and wavelength assignment of scheduled lightpath demands. *IEEE Journal on Selected Areas in Communications*, 21(8):1231–1240, Oct. 2003.
- [40] F. Larumbe and B. Sanso. Optimal location of data centers and software components in cloud computing network design. In *12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CC Grid)*, pages 841–844, May 2012.
- [41] L. Lasdon. *Optimization Theory for Large Systems*. New York: MacMillan, 1970.
- [42] M.D. Leenheer, C. Develder, T. Stevens, B. Dhoedt, M. Pickavet, and P. Demeester. Design and control of optical grid networks. In *Broadband Communications, Networks and Systems (BROADNETS)*, pages 107–115, Sep. 2007.
- [43] X. Liu, C. Qiao, W. Wei, X. Yu, T.Wang, W. Hu, W. Guo, and M.Y. Wu. Task scheduling and lightpath establishment in optical grids. *Journal of Lightwave Technology*, 27(12):1796–1805, Jun. 2009.

- [44] X. Liu, W. Wei, C. Qiao, T. Wang, W. Hu, W. Guo, and M.Y. Wu. Task scheduling and lightpath establishment in optical grids. In *IEEE International Conference on Computer Communications (INFOCOM)*, pages 1966–1974, Apr. 2008.
- [45] Z. Liu, T. Qin, W. Qu, and W. Liu. DAG cluster scheduling algorithm for grid computing. In *IEEE 14th International Conference on Computational Science and Engineering (CSE)*, pages 632–636, Aug. 2011.
- [46] M.E. Lubbecke and J. Desrosiers. Selected topics in column generation. *Operations Research*, 53(6):1007–1023, Dec. 2005.
- [47] F. Magoules, J. Pan, K. Tan, and A. Kumar. *Introduction to grid computing*. CRC Press, 2009.
- [48] G. Maier, A. Pattavina, S.D. Patre, and M. Martinelli. Optical network survivability: Protection techniques in the WDM layer. *Photonic Network Communications*, 4(3/4):251–269, Nov. 2002.
- [49] G. Mohan and C.S.R. Murthy. Lightpath restoration in WDM optical networks. *IEEE Network*, 14(6):24–32, 2000.
- [50] H. Nguyen, M. Gurusamy, and L. Zhou. Provisioning lightpaths and computing resources for scheduled grid demands with location transparency. In *Optical Fiber Communication Conference (OFC)*, pages 1–3, Feb. 2008.

- [51] S. Orłowski, M. Pióro, A. Tomaszewski, and R. Wessäly. SNDlib 1.0–Survivable Network Design Library. In *Proceedings of the 3rd International Network Optimization Conference (INOC), Spa, Belgium, Apr. 2007*.
- [52] S.H. Owen and M.S. Daskin. Strategic facility location: A review. *European Journal of Operational Research*, 111(3):423–447, Dec. 1998.
- [53] Q.V. Phung, H. Daryoush, H.N. Nguyen, and K.M. Lo. A segmentation method for shared protection in WDM mesh networks. In *14th IEEE International Conference on Networks (ICON)*, pages 1–6, Sep. 2006.
- [54] S. Ramamurthy and B. Mukherjee. Survivable WDM mesh networks. part i-protection. In *Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE INFOCOM)*, Mar. 1999.
- [55] J. Reese. Solution methods for the p-median problem: An annotated bibliography. *Networks*, 48(3):125–142, Aug. 2006.
- [56] F. L. Ren and J. Yu. Multiple DAGs scheduling based on lowest transportation and completion time algorithm on the cloud. In *Seventh ChinaGrid Annual Conference (ChinaGrid)*, pages 33–35, 2012.
- [57] A. Sebbah and B. Jaumard. Efficient and scalable design of protected working capacity envelope. In *Telecommunications Network Strategy and Planning Symposium*, pages 1–21, Oct. 2008.

- [58] A. Shaikh, J. Buysse, B. Jaumard, and C. Develder. Anycast routing for survivable optical grids: Scalable solution methods and the impact of relocation. *Journal of Optical Communications and Networking (JOCN)*, 9(3):767–779, Sep. 2011.
- [59] A. Shaikh and B. Jaumard. Optimized dimensioning of resilient optical grids with respect to grade of IT services. *Telecommunication Systems*, Accepted 2012.
- [60] O. Sinnen and L.A. Sousa. Communication contention in task scheduling. *IEEE Transactions on Parallel and Distributed Systems*, 16(6):503–515, Jun. 2005.
- [61] S. Soudan, B.B. Chen, and P.V. Primet. Flow scheduling and endpoint rate control in grid networks. *Future Generation Computer Systems*, 25(8):904–911, Sep. 2009.
- [62] T.E. Stern, G. Ellinas, and K. Bala. *Multiwavelength Optical Networks*. Cambridge, 2009.
- [63] T. Stevens, M.D. Leenheer, C. Develder, B. Dhoedt, K. Christodoulopoulos, and P. Kokkinos. Multi-cost job routing and scheduling in grid networks. *Future Generation Computer Systems*, 25(8):912–925, Sep. 2009.
- [64] S. Tanwir, L. Battestilli, H. Perros, and G. Karmous-Edwards. Dynamic scheduling of network resources with advance reservations in optical grids. *International Journal of Network Management*, 18(2):79–105, Mar. 2008.

- [65] P. Thysebaert, M.D. Leenheer, B. Volckaert, F.D. Turck, B. Dhoet, and P. Demeester. Scalable dimensioning of optical transport networks for grids excess load handling. *Photonic Network Communications*, 12(2):117–132, Sep. 2006.
- [66] A. Todimala and B. Ramamurthy. Survivable virtual topology routing under shared risk link groups in WDM networks. In *IEEE First International Conference on Broadband Networks (BROADNETS)*, pages 130–139, Oct. 2004.
- [67] J. Ullman. Np-complete scheduling problems*. *Journal of Computer and System Sciences*, 10(3):384–393, Jun. 1975.
- [68] J. Vykoukal, M. Wolf, and R. Beck. Services grids in industry – on-demand provisioning and allocation of grid-based business services. *Business & Information Systems Engineering*, 1(2):177–184, Apr. 2009.
- [69] Y. Wang, Y. Jin, W. Guo, W. Sun, and W. Hu. Joint scheduling for optical grid applications. *Journal Of Optical Networking*, 6(3):304–318, Mar. 2007.
- [70] Z. Wang, W. Guo, Z. Sun, Y. Jin, W. Sun, W. Hu, and C. Qiao. On accurate task scheduling in optical grid. In *First International Symposium on Advanced Networks and Telecommunication Systems*, pages 17–18, Dec. 2007.
- [71] R. Yahyapour and P. Wieder. Grid scheduling use cases. Technical Report 1.064, Grid Scheduling Architecture Research Group (GSA-RG), 2006.

- [72] H. Zang, J.P. Jue, and B. Mukherjee. A review of routing and wavelength assignment approaches for wavelength-routed optical WDM networks. *Optical Networks Magazine*, pages 47–60, Jan. 2000.
- [73] H. Zang, C. Ou, and B. Mukherjee. Path-protection routing and wavelength assignment (RWA) in WDM mesh networks under duct-layer constraints. *IEEE/ACM Transactions on Networking*, 11(2):248–258, Apr. 2003.
- [74] J. Zheng and H.T. Mouftah. Routing and wavelength assignment for advance reservation in wavelength-routed WDM optical networks. In *IEEE International Conference on Communications (ICC)*, pages 2722–2726, Aug. 2002.
- [75] W. Zheng and R. Sakellariou. A monte-carlo approach for full-ahead stochastic DAG scheduling. In *IEEE 26th International Parallel and Distributed Processing Symposium Workshops PhD Forum (IPDPSW)*, pages 99–112, May 2012.
- [76] M. Zhu, S. Xiao, W. Guo, A. Wei, Y. Jin, W. Hu, and B. Gellerd. Fault-tolerant scheduling using primary-backup approach for optical grid applications. In *Communications and Photonics Conference and Exhibition (ACP) Asia*, Nov. 2009.