Universitat Politècnica de Catalunya

Optical Communications Group

Off-Line and In-Operation Optical Core Networks Planning

Alberto Castro

A thesis presented in partial fulfillment of the requirements for the degree of

Philosophy Doctor

Advisor:

Luis Velasco Esteban

September, 2014

Optical Communications Group (GCO)

Universitat Politècnica de Catalunya (UPC)

C/ Jordi Girona, 1-3

Campus Nord, D6-213

08034 Barcelona, Spain

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Escola de Doctorat

Acta de calificación de tesis doctoral

Curso académico: 2013/2014

Nombre y apellidos
Alberto Castro Casales

Programa de doctorado
Computer Architecture

Unidad estructural responsable del programa
Computer Architecture Department

Resolución del Tribunal

Reunido el Tribunal designado a tal efecto, el doctorando / la doctoranda expone el tema de la su tesis doctoral titulada _____

_____.

Acabada la lectura y después de dar respuesta a las cuestiones formuladas por los miembros titulares del tribunal, éste otorga la calificación:

☐ NO APTO ☐ APROBADO ☐ NOTABLE ☐ SOBRESALIENTE

(Nombre, apellidos y firma) Presidente/a	(Nombre, apellidos y firma) Secretario/a	
(Nombre, apellidos y firma) Vocal	(Nombre, apellidos y firma) Vocal	(Nombre, apellidos y firma) Vocal

_____, _____ de _____ de _____

El resultado del escrutinio de los votos emitidos por los miembros titulares del tribunal, efectuado por la Escuela de Doctorado, a instancia de la Comisión de Doctorado de la UPC, otorga la MENCIÓN CUM LAUDE:

☐ SÍ ☐ NO

(Nombre, apellidos y firma) Presidente de la Comisión Permanente de la Escuela de Doctorado	(Nombre, apellidos y firma) Secretaria de la Comisión Permanente de la Escuela de Doctorado

Barcelona a _____ de _____ de _____

Abstract

The ever increasing IP traffic volume has finally brought to light the high inefficiency of current wavelength-routed over rigid-grid optical networks in matching the client layer requirements. Such an issue results in the deployment of large-size, expensive and power-consuming Multiprotocol Label Switching (MPLS) layers to perform the required grooming/aggregation functionality. To deal with this problem, the emerging flexgrid technology, allowing for reduced size frequency grids, is being standardized.

Flexgrid optical networks divide the spectrum into frequency slots providing finer granularity than rigid networks based on Dense Wavelength Division Multiplexing (DWDM). To find a feasible allocation, new Routing and Spectrum Allocation (RSA) algorithms for flexgrid optical networks need to be designed and evaluated. Furthermore, due to the flexibility of flexible optical networks, the aggregation functions and statistical multiplexing can be partially located in the optical layer. In addition, given the special characteristics of flexible optical networks, the traditional mechanisms for protection and recovery must be reformulated.

Optical transport platforms are designed to facilitate the setting up and tearing down of optical connections (lightpaths). Combining remotely configurable optical cross-connects (OXCs) with a control plane provides the capability of automated lightpath set-up for regular provisioning, and real-time reaction to the failures, being thus able to reduce Operational Expenditures (OPEX). However, to exploit existing capacity, increase dynamicity, and provide automation in future networks, current management architectures, utilizing legacy Network Management Systems (NMS) need to be radically transformed.

This thesis is devoted to design optical networks and to devise algorithms to operate them.

Network design objective consists of:

i. Analyzing the cost implications that a set of frequency slot widths have on the Capital Expenditures (CAPEX) investments required to deploy MPLS-over-flexgrid networks;

ii. Studying recovery schemes, where a new recovery scheme specifically designed for flexgrid-based optical networks is proposed.

As for network operation, we focus on:

i. Studying provisioning, where two provisioning algorithms are proposed: the first one targets at solving the RSA problem in flexgrid networks, whereas the second one studies provisioning considering optical impairments in translucent DWDM networks;

ii. Getting back to the recovery problem, we focus on algorithms to cope with restoration in dynamic scenarios. Several algorithms are proposed for both single layer and multilayer networks to be deployed in the centralized Path Computation Element (PCE);

iii. One of the main problems in flexgrid networks is spectrum defragmentation. In view of that, we propose an algorithm to reallocate already established optical connections so as to make room for incoming requests. This algorithm is extended with elasticity to deal with time-varying traffic.

The above algorithms are firstly implemented and validated by using simulation, and finally experimentally assessed in real test-beds. In view of PCE architectures do not facilitate network reconfiguration, we propose a control and management architecture to allow the network to be dynamically operated; network resources can be made available by reconfiguring and/or re-optimizing the network on demand and in real-time. We call that as *in-operation network planning*.

It shall be mentioned that part of the work reported in this thesis has been done within the framework of several European and National projects, namely STRONGEST (FP7-247674), IDEALIST (FP7-ICT-2011-8), and GEANT (FP7-238875) funded by the European Commission, and ENGINE (TEC2008-02634) and ELASTIC (TEC2011-27310) funded by the Spanish Science Ministry.

Resumen

El volumen creciente del tráfico IP, ha puesto de manifiesto, finalmente, la alta ineficiencia de las redes ópticas actuales de grid rígido basadas en la tecnología de multiplexación por división de longitud de onda (WDM por su sigla en inglés) en la adecuación a los requisitos de capa de cliente. Dicho problema genera que se deba desplegar una red con capas de conmutación multi-protocolo mediante etiquetas (MPLS por su sigla en inglés) de gran tamaño, costosa y de alto consumo energético para poder realizar la funcionalidad de agregación requerida. Para hacer frente a este problema, la tecnología flexgrid emergente, que permite grids con frecuencias de menor tamaño, está siendo estandarizada.

Las redes ópticas flexgrid dividen el espectro en slots de frecuencia, lo que proporciona una granularidad más fina en comparación a las redes rígidas basadas en WDM. Para encontrar una asignación de espectro factible, nuevos algoritmos de enrutamiento y asignación de espectro (RSA por su sigla en inglés) para redes ópticas flexgrid deben ser diseñados y evaluados. Además, debido a la flexibilidad de las redes ópticas flexibles, las funciones de agregación y de multiplexación estadística pueden ser parcialmente situadas en la capa óptica. Asimismo, dadas las características especiales de las redes ópticas flexibles, los mecanismos tradicionales de protección y recuperación deben reformularse.

Las plataformas de transporte ópticas están diseñadas para facilitar la creación y destrucción de conexiones ópticas. La combinación de enrutadores ópticos configurables remotamente (OXCs por su sigla en inglés) con un plano de control, proporciona la capacidad de crear conexiones automáticamente para el aprovisionamiento habitual, y la reacción en tiempo real a los fallos, para así poder reducir los gastos operativos (OPEX por su sigla en inglés). Sin embargo, para aprovechar la capacidad existente, aumentar la dinamicidad y proporcionar automatización a las redes del futuro, las arquitecturas actuales de gestión, que utilizan sistemas legados de gestión de red (NMS por su sigla en inglés), necesitan ser transformadas de manera radical.

Esta tesis está dedicada al diseño de redes ópticas y a la creación de algoritmos para operarlas.

El objetivo de diseño de red se compone de:

i. El análisis de las implicancias en el costo que tiene un conjunto de slots de frecuencia en la inversión de capital (CAPEX por su sigla en inglés) necesaria para implementar redes MPLS-over-flexgrid;

ii. El estudio de esquemas de recuperación, donde se propone un nuevo esquema de recuperación diseñado específicamente para las redes ópticas basadas en flexgrid.

En cuanto a la operación de la red, nos centramos en:

i. El estudio de aprovisionamiento, donde se proponen dos algoritmos de aprovisionamiento: el primero de ellos tiene como objetivo solucionar el problema de RSA en redes flexgrid, mientras que el segundo estudia aprovisionamiento considerando la degradación óptica en redes WDM translúcidas;

ii. Volviendo al problema de la recuperación, nos centramos en algoritmos de restauración para escenarios dinámicos. Se proponen varios algoritmos, tanto para redes mono-capa como redes multi-capa, que serán desplegados en un PCE centralizado;

iii. Uno de los principales problemas en las redes flexgrid es la desfragmentación del espectro. En vista de ello, se propone un algoritmo para reasignar las conexiones ópticas ya establecidas con el fin de hacer espacio a las nuevas conexiones entrantes. Este algoritmo se extiende con elasticidad para ser utilizado en escenarios con tráfico variable en el tiempo.

Los algoritmos anteriores son en primer lugar implementados y validados utilizando simulación, y finalmente son evaluados experimentalmente en bancos de pruebas reales. En vista de que las arquitecturas de PCE no facilitan la reconfiguración de la red, proponemos una arquitectura de control y gestión para permitir que la red pueda ser operada de forma dinámica; hacer que los recursos de la red estén disponibles mediante reconfiguración y/o re-optimización de la red bajo demanda y en tiempo real. A eso lo llamamos *planificación en operación de la red*.

Es importante destacar que, parte del trabajo presentado en esta tesis se ha realizado en el marco de varios proyectos europeos y nacionales, a saber, STRONGEST (FP7-247674), IDEALIST (FP7-ICT-2011-8), y GEANT (FP7-238875) financiados por la Comisión Europea, y ENGINE (TEC2008-02634) y ELASTIC (TEC2011-27310), financiados por el Ministerio Español de Ciencia.

Table of Contents

Page

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

The ever increasing IP traffic volume has finally brought to light the high inefficiency of current wavelength-routed over rigid-grid optical networks in matching the client layer requirements. Such an issue results in the deployment of large-size, expensive and power-consuming Multiprotocol Label Switching (MPLS) layers to perform the required grooming/aggregation functionality. To deal with this problem, the emerging flexgrid technology, allowing for reduced size frequency grids, is being standardized.

Flexgrid optical networks divide the spectrum into frequency slots providing finer granularity than rigid networks based on Dense Wavelength Division Multiplexing (DWDM). If a connection requires a greater capacity than one slot, a number of adjacent slots must be assigned to it. Consequently, a specific amount of slots are allocated to fulfill the bitrate requested; hence the 10 Gb/s demands will occupy fewer spectrum than those of 400 Gb/s, and thus obtaining an efficient use of network resources. In order to find a feasible allocation, new Routing and Spectrum Allocation (RSA) algorithms for flexgrid optical networks need to be designed and evaluated.

Furthermore, due to the flexibility of flexible optical networks, the aggregation functions and statistical multiplexing can be partially located in the optical layer. Therefore, studies should be carried out comparing the performance of the different alternatives. In addition, given the special characteristics of flexible optical networks, the traditional mechanisms for protection and recovery must be reformulated.

Regarding network capacity planning, it requires the placement of network resources to satisfy expected traffic demands and network failure scenarios. Today, the network capacity planning process is typically an offline process, and is based on very long planning cycles (e.g., yearly, quarterly). Generally, this is due to the static and inflexible nature of the current networks. This is true for both the transport layer (optical) and for the IP/MPLS layer. However, the latter should be inherently more dynamic compared to the underlying transport infrastructure. Notwithstanding both layers might use automated Traffic Engineering (TE) techniques to place traffic where the network resources are.

Increasing transport capacity to meet predicted traffic changes and to react against failures needs for network flexibility. However, due to the fixed and rigid nature of provisioning in the transport layer, network planning and TE still require significant human intervention, which entails high Operational Expenditures (OPEX). In addition, to ensure that the network can support the forecast traffic and all the failure scenarios that need to be protected against, operators add spare capacity (over-provision) in different parts of the network to address likely future scenarios. This entails inefficient use of network resources and significantly increases network Capital Expenditures (CAPEX).

Notwithstanding, optical transport platforms are designed to facilitate the setting up and tearing down of optical connections (lightpaths) within minutes or even seconds to create Wavelength Switched Optical Networks (WSON) [G.8080]. Combining remotely configurable optical cross-connects (OXCs) with a control plane provides the capability of automated lightpath set-up for regular provisioning, and real-time reaction to the failures, being thus able to reduce OPEX. However, to exploit existing capacity, increase dynamicity, and provide automation in future networks, current management architectures, utilizing legacy Network Management Systems (NMS) need to be radically transformed.

In a scenario where lightpath provisioning can be automated, network resources can be made available by reconfiguring and/or re-optimizing the network on demand and in real-time. We call that as *in-operation network planning*.

The work here presented is being developed within the Optical Communications Group. This thesis continues, in part, the work developed during the PhD theses of Marc Ruiz and Oscar Pedrola.

1.2 Goals of the thesis

The main objective of this thesis is designing optical networks and devising algorithms to operate them. Two specific goals are defined:

G.1 – Off-line planning in optical networks

This work consists in two sub-goals:

- **G.1.1**.: CAPEX study for flexgrid network.
- **G.1.2**.: Devise new recovery schemes specifically designed for flexgrid-based optical networks.

G.2 – Dynamic Provisioning and In-Operation Planning in WSON and Flexgrid networks

This work is divided into four sub-goals:

- **G.2.1**: Propose algorithms for on-line provisioning in WSON and Flexgrid optical networks.
- **G.2.2**: Devise algorithms for on-line restoration in WSON and Flexgrid optical networks.
- **G.2.3**: Design spectrum-reallocation algorithms to increase resource utilization in Flexgrid optical networks.
- **G.2.4**: Conceive a control and management architecture to dynamically optimize the network.

A summary of the goals of the thesis is presented in Table 1-1.

Table 1-1 Thesis goals

<table>
<tr><th colspan="2"></th><th>Goals</th></tr>
<tr><td rowspan="2">Off-line Planning</td><td>CAPEX</td><td>G1.1
CAPEX study for flexgrid optical networks.</td></tr>
<tr><td>Resilience architectures</td><td>G1.2
Recovery mechanisms for flexgrid networks.</td></tr>
<tr><td colspan="2" rowspan="4">Dynamic Provisioning and In-Operation Planning</td><td>G 2.1
Dynamic provisioning.</td></tr>
<tr><td>G 2.2
Dynamic restoration.</td></tr>
<tr><td>G 2.3
Re-optimization.</td></tr>
<tr><td>G 2.4
In-operation planning.</td></tr>
</table>

1.3 Thesis outline

The remainder of this thesis is as follows.

Chapter 2 briefly introduces the main background of DWDM and flexgrid optical networks, including recovery mechanisms. Moreover, general concepts of metaheuristics frameworks, used along the thesis, are described.

Chapter 3 explores the state-of-the-art of off-line planning, dynamic operation and re-optimization for optical core networks.

Chapter 4 presents our contribution to off-line optical network planning (goal G 1.1). This chapter is based on already published material [Pe12.1.] and [Pe12.2].

Chapter 5 devises new recovery mechanisms specifically devised for flexgrid optical networks (goal 1.2). This chapter is based on already published material [Ca12.2], [Ca12.4], and [Ca14.2].

Chapter 6 presents our contribution about dynamic provisioning in both, translucent WSON and flexgrid optical networks (goal 2.1). This chapter is based on already published material [Ca12.1], [Ca12.3], and [Ve12.1].

Chapters 7 and 8 are devoted to dynamic restoration (goal 2.2). These chapters are based on already published material [Ca13.1], [Ca13.2], [Ca14.1], [Ma13], [Pa14.2], [Ve13], and [Ve14.2].

Chapter 9 is focused on spectrum reallocation (goal 2.3). This chapter is based on already published material [Ca12.1], [Ca13.2], [Ca13.3], and [Pa14.1].

Chapter 10 introduces our contribution for in-operation planning (goal 2.4). This chapter is based on already published material [Ca13.3], [Ca14.1], [Pa14.1], [Ve14.1], and [Ve14.2].

Finally, Chapter 11 presents the main conclusions of the thesis.

All the network topologies and traffic profiles used in the studies included in this thesis are collected in Appendix A.

Chapter 2

Background

As a starting point for this thesis, in this chapter we introduce the concepts and technologies that will be used in the remaining chapters.

Firstly, some concepts related to translucent DWDM and flexgrid optical networks are introduced. We describe the routing and optical resources allocation algorithms for both optical transmission technologies. Secondly, existing path recovery mechanisms are presented. Finally, a brief introduction to optimization is presented and the metaheuristic frameworks used in this thesis are presented.

2.1 Optical networks

2.1.1 Opaque, translucent and transparent networks

Optical networks are evolving from traditional *opaque* infrastructures towards *transparent* infrastructures [Ra99]. An opaque network performs reamplifying, reshaping, and retiming (3R) regeneration for every wavelength channel at each node traversed by the connection. This allows regenerating the optical signal and improves the transmission quality. On the other hand, transparency implies that signal remains within the optical domain from the source to the destination node. Thereby, transparency eliminates the optical-electronic-optical (OEO) conversions or 3R regenerators at the intermediate nodes. This yields important benefits such as freedom on the bit rates and data formats of the transported signals, elimination of the well-known electronic bottleneck, and reduction of both network cost and power consumption [Po08].

The key enablers to deploy transparent all-optical networks are, on the one hand, the advances on relevant optical signal functions (e.g., amplification, filtering, and

dispersion compensation) and, on the other hand, the availability of network nodes capable of routing and switching in the optical domain (e.g., *Reconfigurable Optical Add-Drop Mulitplexers* or ROADMs and *Optical Cross-Connects* or OXCs). Such an all-optical switching network is referred to as a Wavelength-Switched Optical Networks (WSON).

Nevertheless, due to the lack of electronic regeneration, two constraints impose fundamental limitations to transparent WSON: the wavelength continuity constraint (WCC) and the optical physical impairments [Ya05.1]. The WCC implies that an optical connection (i.e., *lightpath*) is subject to use the same wavelength on each link along the route. This restriction raises important challenges at the time of computing and establishing lightpaths, specifically to achieve an acceptable network performance in terms of connection blocking probability. On the other hand, physical impairments such as amplification noise, cross-talk, dispersion (polarization mode dispersion (PMD), chromatic dispersion (CD)) and fiber nonlinearity may accumulate while the signal travels from the source node to the destination node. In consequence, the maximum optical transmission reach is limited, since the received signal quality may render the bit error rate (BER) unacceptable and the lightpath unusable [Po08], [Hu05]. Although BER is the main performance parameter to measure the signal quality, in transparent WSON it can only be measured at the edges. Therefore, efficient substitutes to estimate the BER such as the Optical Signal Noise Ratio (OSNR) [Hu05] and the Q factor [Po08] are needed. In this thesis, we rely on the OSNR as signal quality performance indicator.

In the evolution from opaque networks towards transparent WSON, *translucent* networks are considered as an intermediate step [Ya05.1]. This infrastructure uses a set of sparsely but strategically 3R regenerators placed throughout the network to do signal regeneration [Sh07]. Therefore, a translucent network is a cost-efficient infrastructure that aims at attaining an adequate trade-off between network construction cost, mainly due to the high cost of 3R regenerators, and the service provisioning performance, adequate end-to-end quality of transmission (QoT).

In transparent/translucent WSON, the longer the path an optical signal is transported on, the more amplifiers are needed to compensate the power loss due to the fiber and node attenuations. The increased number of amplifiers makes the *Amplifier Spontaneous Noise* (ASE) a significant impairment factor to the signal quality. The OSNR is used to capture the impact of the ASE noise [Hu05]. Fig. 2-1 illustrates the computation of the estimated OSNR at the receiver of a LSP. The LSP traverses $N-1$ links each allocating a different number of amplifier spans.

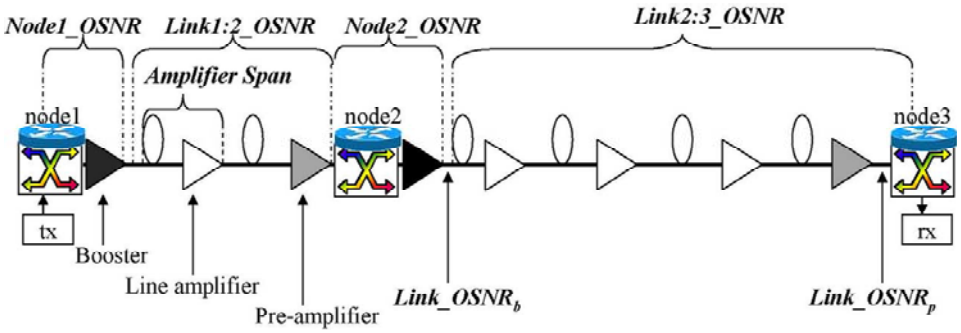

Fig. 2-1 Contributors for the path OSNR computation (reproduced from [Ma10]).

To estimate the OSNR level at node *N*, the following assumptions are made:

- The pre-amplifier (at the input port of a node) and the line amplifier have the same characteristics, but the booster amplifier (at the output port of the node) has, in general, different characteristics such as the noise figure.
- For the computation of the OSNR in a lightpath involving nodes (including source and destination nodes) only the first $N-1$ nodes contribute to the OSNR estimation. The destination node is not considered since it is not entirely crossed by the lightpath.
- The maximum OSNR level (i.e., $OSNR_{\max}$) of a LSP is the OSNR level at the optical transmitter, that is, at either the source node or after using a 3R regenerator at an intermediate node.

As described in [Ts08], there are two components contributing to the OSNR level estimation: the *Link_OSNR* that considers the total ASE noise induced by all the (pre- and line) amplifier spans of a link (e.g., *Link1:2_OSNR* in Fig. 2-1) and the *Node_OSNR* which considers the ASE noise caused by the node booster amplifier. Thus, the estimated OSNR (*Total_OSNR*) for a transparent segment traversing links and nodes is given by:

$$Total\_OSNR_N = \frac{1}{\left(\sum_{j=1}^{N-1}\frac{1}{Link\_OSNR_j}+\sum_{j=1}^{N-1}\frac{1}{Node\_OSNR_j}\right)} \quad (2.1)$$

where the $Link\_OSNR_j$ and the $Node\_OSNR_j$ are the OSNR for the j^{th} link and node, respectively. It is worth noting that the expression (2.1) is simplified and holds if the amplifier's gain is automatically controlled in order to basically maintain a constant optical power per wavelength channel at the output of the amplifiers.

In addition to the ASE noise, other accumulated impairments such as the PMD, CD and the nonlinearities may impact on the signal quality of the transparent segments. These impairments are integrated into the constraint model as OSNR penalties [St01], [Pa08], [Le09.1], [Cu08], [Ts08], [Sa09]. To this end, the

degradation caused by these physical impairments needs to be kept within defined acceptable ranges or constraints.

The TE Link OSNR consists of two OSNR values (in dB) that allow computing the *Link_OSNR*, namely, the *Link_OSNR*$_b$ and the *Link_OSNR*$_p$ (see Fig. 2-1). Thus, the *Link_OSNR* can be computed according to (2.2) using the OSNR levels at both ends of the link, i.e., *Link_OSNR*$_b$ (after the booster amplifier of the link) and the *Link_OSNR*$_p$ (after the pre-amplifier of the same link) in Fig. 2-1.

$$Link\_OSNR = \frac{1}{\frac{1}{Link\_OSNR_p} - \frac{1}{Link\_OSNR_b}} \tag{2.2}$$

2.1.2 Flexgrid

The flexgrid technology [Ji09], [Ji10], [Ge12], provides higher spectrum efficiency and flexibility in comparison to traditional WSON. By leveraging key advances in optical multi-level modulation techniques and the design of both bandwidth-variable transponders (BV-Ts) and bandwidth-variable wavelength selective switches (BV-WSSs), the main component enabling the design of bandwidth-variable wavelength cross-connects (BV-OXCs), flexgrid optical networks are able to provide both sub- and super-wavelength traffic accommodation. While BV-Ts may work under both single- and multi-carrier advanced modulation formats such as Quadrature Phase Shift Keying (QPSK), Quadrature Amplitude Modulation (QAM), and Opitcal Orthogonal Frequency-Division Multiplexing (O-OFDM) [Ji10], BV-OXCs can be assembled using existing devices like the *WaveShaper* programmable optical processor [Finisar]. Thanks to this flexible technology, a flexgrid optical network can adjust to varying traffic conditions over time, space and bandwidth, thereby creating a network scenario where wavelength channels are both switched and dimensioned (bitrate/reach/signal bandwidth) according to temporary traffic requirements.

To this end, flexgrid optical networks divide the available optical spectrum into a set of frequency slots (FSs) of a fixed finer spectral width in comparison to the current ITU-T DWDM rigid frequency grid (50GHz) [G.694.1]. Current proposals for the slot size are 25GHz, 12.5GHz and 6.25GHz, the latter two being mentioned in the industry as potential minimum bandwidth granularities. Therefore, traffic demands are assigned a given number of FSs according to their requested bit-rate, the selected modulation technique and the considered frequency grid (i.e., the slot width) [Ji10].

Let introduce some notation. Let graph G (*N*, *E*) represent the topology of a flexgrid optical network, where *N* is the set of optical nodes and *E* the set of optical links connecting two nodes, and let *S* be the set of FSs available in each link $e \epsilon E$.

Then, given a connection request d requesting for $B(d)$ Gb/s, the spectral resources that need to be allocated are a function of the spectral efficiency (B_{mod} in b/s/Hz) of the chosen modulation format and the slot width, denoted as $F(S)$ in GHz. In addition, QPSK and 16-QAM modulation formats are considered in this thesis. On the one hand, 16-QAM ($B_{16\text{-}QAM}$=4 b/s/Hz) needs half spectrum width than QPSK (B_{QPSK}=2 b/s/Hz) to transmit the same bit rate. On the other hand, 16-QAM has worse receiver sensitivity than QPSK so both the length in hops, which entails cascade filters, and the distance of optical connections using the 16-QAM modulation format is limited.

Without loss of generality, given B_{mod} and $F(S)$, we compute the amount of contiguous slots required by connection request d, denoted as $S(d)$, as in [Ji10]:

$$S(d) = \left\lceil \frac{B(d)}{B_{\text{mod}} \cdot F(S)} \right\rceil \tag{2.3}$$

To illustrate eq. (2.3), Table 2-1 shows the number of contiguous slots that each connection request needs under different slot widths and using the considered modulation formats.

It must be mentioned that Eq. (2.3) tends to under-estimate the number of FSs required, as it assumes that *B(d)* consists only of payload data. However, in general, this is not the case, as different overhead data (e.g., around 10% extra) may be required. Such overhead may vary according to the modulation format selected. For instance, in OFDM-based systems, overhead symbols are required to avoid inter-symbol interference. Additionally, the selection of the modulation format may depend on each particular demand bit-rate. These issues, however, are left out of the scope of this thesis.

Table 2-1 Contiguous slots required per connection request.

	Requested bit rate ($B(d)$)							
Slot width ($F(S)$)	**10 Gb/s**		**40 Gb/s**		**100 Gb/s**		**400 Gb/s**	
	QPSK	16-QAM	QPSK	16-QAM	QPSK	16-QAM	QPSK	16-QAM
50 GHz	1	1	1	1	1	1	4	2
25 GHz	1	1	1	1	2	1	8	4
12.5 GHz	1	1	2	1	4	2	16	8
6.25 GHz	1	1	4	2	8	4	32	16

Routing and Spectrum Allocation

In WSON, a RWA algorithm provides optical connections finding a physical route and assigning a wavelength to the connection request. Note that when no

wavelength converters are available in the network, the WCC must be ensured assigning a wavelength that must be currently unused in every link in the route.

In flexgrid optical networks, the similar problem is called the Routing and Spectrum Allocation (RSA) since it finds a physical route and allocates a set of slots to connection requests. However, besides the spectrum continuity constraint similar to WCC in RWA, the RSA problem adds the spectrum contiguity constraint, i.e. the frequency slots allocated for a connection request must be contiguous in the spectrum along the links in its route. RSA algorithms are used either in the network planning phase when the set of connection requests is known in advance (offline RSA algorithms) [Ch11], [Wa11.2], [Kl11], [Ve12.3], or to dynamically provision connections at arrivals of requests (dynamic RSA algorithms) [Wa11.3].

The difficulty of the RSA problem lays in the fact that, apart from the spectrum continuity constraint in each link along the routing path, the spectrum contiguity constraint must be also ensured. To solve the RWA+WCC problem, some works in the literature propose to use a dedicated graph for each of the wavelengths available in the DWDM grid. That algorithm using parallel graphs can be adapted to solve the RSA problem by defining channels, i.e. sets of spectrum contiguous frequency slots; a dedicated graph must be used for each channel. Channels were introduced in [Ve12.3] as a concept to simplify the offline RSA problem. The use of connection request-tailored channels allows removing the spectrum contiguity problem from mathematical formulations. Channels can be grouped as a function of the number of slots, e.g., the set of channels C_2 = {{1,1,0,0,0,0,0,0}, {0,1,1,0,0,0,0,0}, {0,0,1,1,0,0,0,0}, … {0,0,0,0,0,0,1,1}} includes every channel using 2 contiguous slots, where each position is 1 if a given channel uses that slot. The size of the complete set of channels C that need to be defined is bounded by $|S| \cdot n$, where S represents the set of FSs and n is the number of different amounts of contiguous slots that connections can request, e.g. if connections can request for either 1, or 2, or 4, or 16 contiguous slots, n=4. Then, the number of parallel graphs that need to be maintained and the number of times the shortest path algorithm is run can be as large as 3200 in a 5 THz spectrum using 6.25 GHz slots.

Multi-flow Transponders (MF-TPs)

The flexible grid described above supports bandwidth variable transponders (BV-Ts) that can tune both bitrate and spectrum width to the values required to support a specific connection. Therefore, one single BV-T, e.g. 400Gb/s, can be used for the entire connection, which can be specifically configured to cope with the actual needs. However, when BV-Ts need to transmit at low bit rates (e.g. 100Gb/s), part of their capacity is wasted. Therefore, the multi-flow transponder (MF-TP), also known as sliceable bandwidth variable transponder (SBVT), has been proposed in [Ji12] and [Lo14]. MF-TPs can provide even higher levels of elasticity and efficiency to the network by enabling transmitting from one point to multiple destinations, changing the bitrate to each destination and the number of destinations on demand.

2.2 Off-line planning

Planning, i.e. designing and dimensioning the network, generally consists in determining the nodes and links that need to be installed and which is the equipment to be purchased to serve the foreseen traffic while minimizing network CAPEX. Because those tasks are done before the network enters into operation it is commonly known as off-line planning or static planning. Indeed, the classical network life-cycle typically consists of several steps that are performed sequentially (Fig. 2-2). Starting with inputs from the service layer and from the state of the resources in the already deployed network, a planning phase needs to be carried out to produce recommendations that the next phase uses to design the network for a given period. That period is not fixed and actual time length usually depends on many factors, which are operator and traffic type specific. Once the planning phase produces recommendations, the next step is to design, verify and manually implement the network changes. While in operation, the network capacity is continuously monitored and that data is used as input for the next planning cycle. In case of unexpected increases in demand or network changes, nonetheless, the planning process may be restarted.

While the network is in operation, its capacity is continuously monitored and that data is used as input for the next planning cycle. In case of unexpected increases in demand or network changes, nonetheless, the planning process may be restarted. That period is not fixed and actual time length usually depends on many factors, which are operator and traffic type specific.

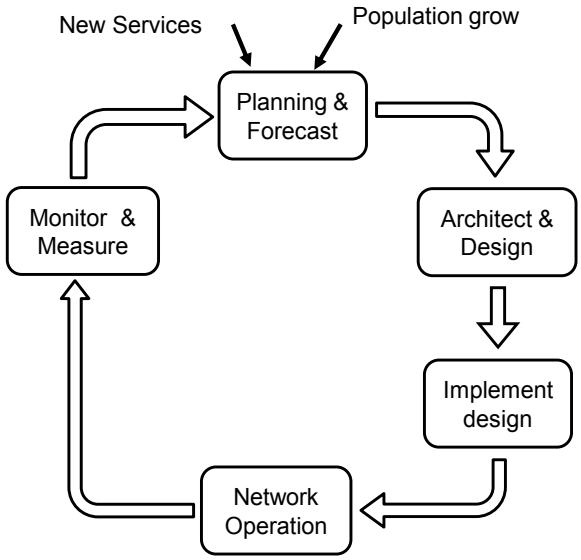

Fig. 2-2 Classical network life-cycle.

2.3 Static vs. dynamic network operation

Operation of the currently deployed carriers' transport networks is very complex; multiple manual configuration actions are needed for provisioning purposes (e.g., hundreds of thousands of node configurations per year in a mid-size network). In fact, transport networks are currently configured with big static *fat pipes* based on capacity over-provisioning, since they are needed for guaranteeing traffic demand and Quality of Service (QoS). Furthermore, network solutions from different vendors typically include a centralized service provisioning platform, using vendor-specific NMS implementations along with an operator-tailored umbrella provisioning system, which may include a technology specific Operations Support System (OSS). Such complicated architectures (Fig. 2-3) generate complex and long workflows for network provisioning: up to two weeks for customer service provisioning and more than six weeks for core routers connectivity services over the optical core.

Fig. 2-4 illustrates the fact that such static networks are designed to cope with the requirements of several failure scenarios, and predicted short-term increases in bandwidth usage, thus requiring capacity over-provisioning and significantly increasing CAPEX. It shows a simple network consisting in three routers connected to a central one through a set of lightpaths established on an optical network. Two different scenarios are considered, although the same amount of IP traffic is conveyed in each of them. In the scenario A, router R3 needs three lightpaths to be established to transport its IP traffic towards R4, whereas R1 and R2 need only one lightpath each. In contrast in the scenario B, R1 and R2 need two lightpaths whilst R3 needs only one lightpath. In static networks, where lightpaths in the optical network are statically established, each pair of routers has to be equipped with the

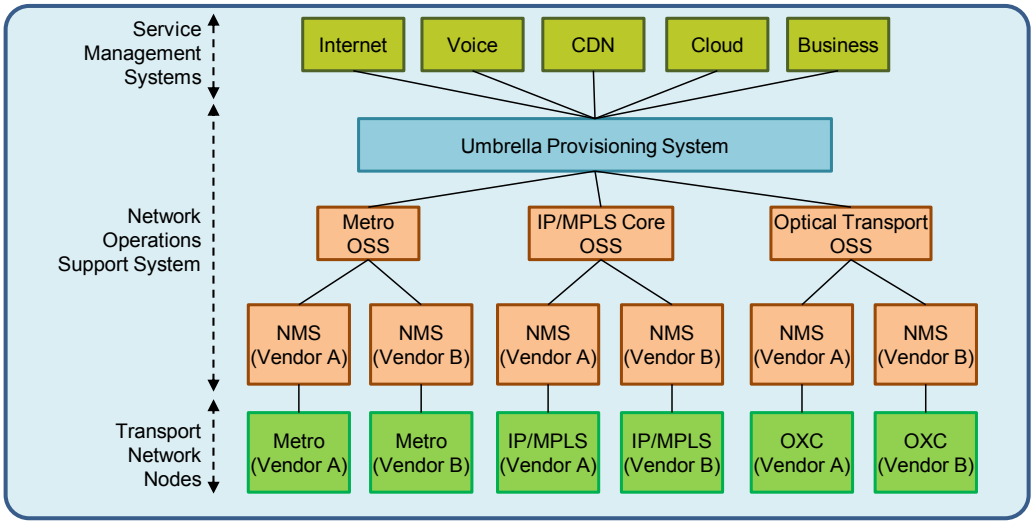

Fig. 2-3 Current static architecture.

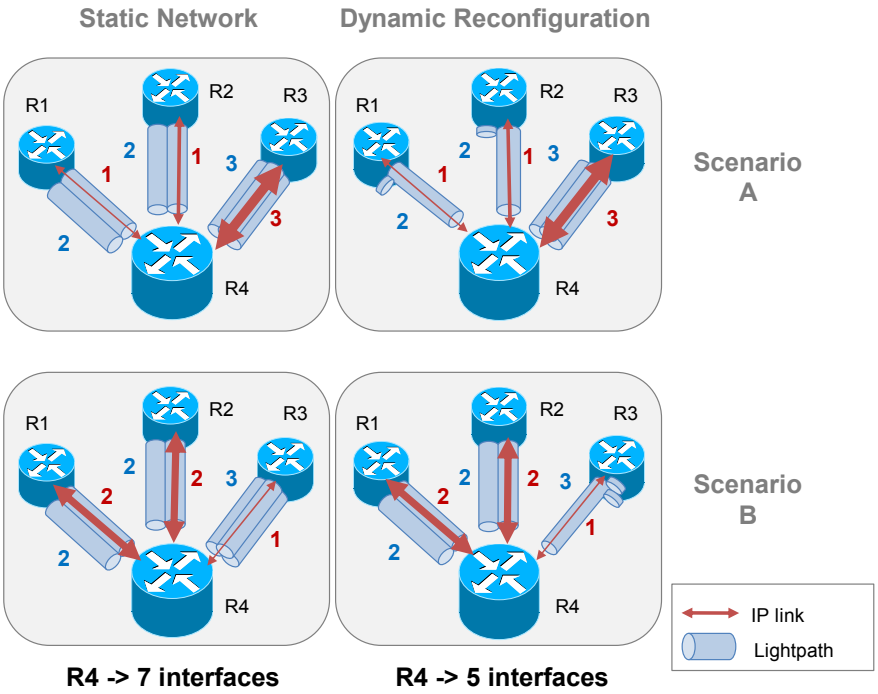

Fig. 2-4 An example of dynamic planning and reconfiguration.

number of interfaces for the worst case, resulting in R4 being equipped with 7 interfaces. However, if the optical network can be dynamically reconfigured setting up and tearing down lightpaths on demand, each router can be dimensioned separately for the worst case, regardless of the peering routers. As a result, R4 would need to be equipped with only 5 interfaces, thus saving 28.5% of interfaces.

2.4 Control plane

As technologies are developed to allow the network to become more agile, it may be possible to provide response to traffic changes by reconfiguring the network near real-time. In fact, some operators have deployed Generalized Multi-Protocol Label Switching (GMPLS) control plane, mainly for service set-up automation and recovery purposes; the GMPLS framework includes LSP signaling and Traffic Engineering Database (TED) updating protocols. However, GMPLS-based control plane governs only parts of the network and does not support holistic network reconfiguration. This functionality will require an *in-operation planning tool* that interacts directly with the data and control planes and operator polices via OSS platforms, including the NMS.

The main component in the control plane related to this thesis is the Path Computation Element (PCE) [Fa06], [Pa13], which is defined as an entity to serve paths computation requests. The PCE Communication Protocol (PCEP) [Va09] is

Table 2-2 Comparison of PCE architectures.

Stateless PCE	Stateful PCE	Active Stateful PCE
Performs path computation using topology and the state of the network resources in the TED.	Maintains the LSP-DB with the LSPs that are active in the network.	Capable of responding to changes in network resource availability and predicted demands and reroute existing LSPs for increased network resource efficiency [Cas13].

used to carry paths computation requests and PCE responses. Requests can be processed independently of each other or in groups, utilizing a view of the network topology stored in the TED (stateless PCE) or considering as well information regarding Label Switched Paths (LSPs) that have been set-up in the network, stored on the LSP Database (LSP-DB) (stateful PCE) [Cr14]. Finally, a PCE is said to be Active [Cr13] if it can modify established LSPs based on traffic trends. Table 2-2 compares the features of the different PCE architectures.

In addition, significant standardization effort has assisted in defining control plane architectures and protocols. Starting from the distributed paradigm lead by the GMPLS framework, control plane have lately moved towards a centralized one led by the development of the Software-Defined Network (SDN) concept with the introduction of OpenFlow [OpenFlow]. IETF is also moving in that direction with the definition of the Application-Based Network Operations (ABNO) architecture [Ki14]. ABNO includes a number of standard components and interfaces which, when combined together, provide a method for controlling and operating the network. A simplified view of the ABNO architecture is represented in Fig. 2-5. In addition to the PCE, it includes:

- The ABNO controller as the entrance point to the network for NMS/OSS and the service layer for provisioning and advanced network coordination. It acts as a system orchestrator invoking its inner components accordingly to a specific workflow.
- The Virtual Network Topology Manager (VNTM) [Ok09.1] coordinates Virtual Network Topology (VNT) configuration by setting up or tearing down lower-layer LSPs, and advertising the changes to higher-layer network entities.
- The Provisioning Manager is responsible for the establishment of LSPs. This can be done by interfacing the control plane or by directly programming the data path on individual network nodes or acting as an OpenFlow controller.

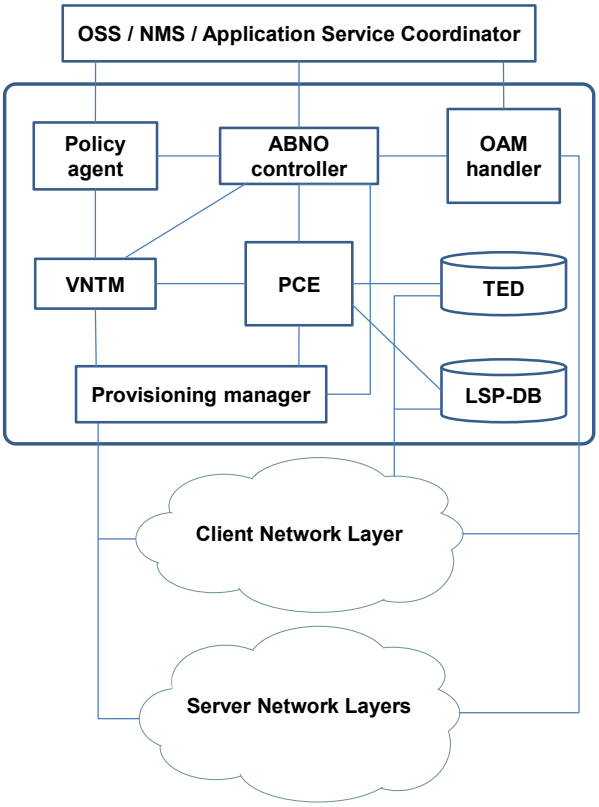

Fig. 2-5 Simplified ABNO architecture.

- The Operations, Administration, and Maintenance (OAM) handler is responsible for detecting faults and taking actions to react to problems in the network. It interacts with the nodes to initiate OAM actions such as monitoring and testing new links and services.

2.5 Recovery

2.5.1 Protection and restoration in optical networks

Owing to the huge bitrate associated to each established path, *recovery* schemes need to be used to guarantee that the associated client connectivity demand continue being served even in case of failures [Gr03]. As in WSON, recovery can be provided by either *protection*, where the failed working path is substituted by a pre-assigned backup one, or *restoration*, which is based on rerouting the working path. Backup paths use resources, i.e. each of the frequency slots in a fiber link, that are *dedicated* to protect a single working path, or they can be *shared* to provide protection to multiple working paths. As a consequence, the former scheme is called dedicated path protection (DPP) and the latter shared path protection (SPP) (see Fig. 2-6).

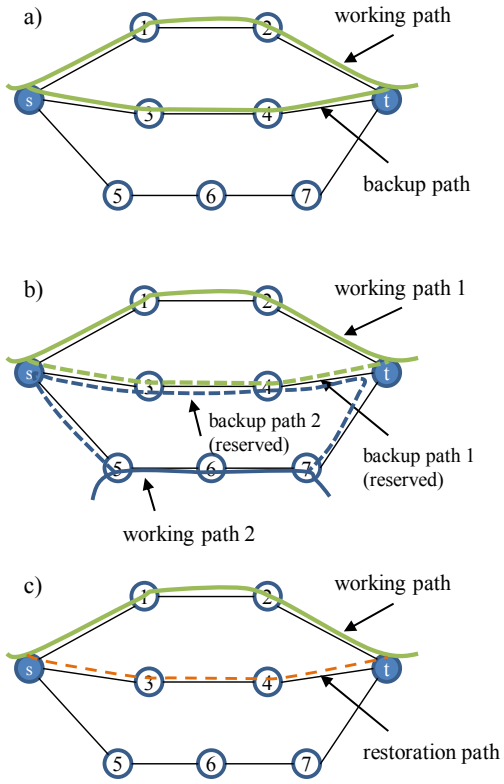

Fig. 2-6 DPP (a), SPP (b), and path restoration (c).

Although protection schemes reserve resources to guarantee that all protected paths are recovered in case of any single failure, SPP provides better resource utilization than DPP due to spare resources are shared among several working paths. On the one hand, restoration is the most efficient scheme since resources are only allocated after a failure impacts a working path. On the other hand, recoverytimes are usually much shorter when protection is used. In the case of DPP, since spare resources are already in use, recovery times are really short, being slightly longer for SPP since spare resources are reserved beforehand and activated in case of failure. Protection (DPP and SPP) and restoration schemes have been traditionally used in WSON.

2.5.2 Recovery schemes for flexgrid networks

In flexgrid optical networks, the bitrate requested by client demands is converted to a number of frequency slots to be allocated, using eq. (2.3). Therefore, Service Level Agreements (SLA) between network operators and clients can now include specific terms so that all or *only part* of the requested bitrate is restored in case of failure [So11]. Let q^d be percentage of bitrate to be assured to demand d in case of failure, hence, the number of slots, $n^d_{recovery}$, that must be used for recovery demand d can be computed so to:

$$n^{d}_{recovery} \cdot B_{\text{mod}} \cdot F(s) > q^{d} \cdot B(d) \tag{2.4}$$

For illustrative purposes, Fig. 2-7 shows how path restoration works. In Fig. 2-7a one client demand is requesting some amount of bitrate that translates into 4 slots. The working path serves all the requested bitrate. After a failure has impacted the working path, a restoration path is computed and established in Fig. 2-7b. On the other hand, Fig. 2-8 depicts how path protection works. In Fig. 2-8a one client demand is requesting some amount of bitrate that also translates into 4 slots. The working path serves all the requested bitrate and there are some slots reserved along the backup route. After a failure has impacted the working path, the backup path is activated in Fig. 2-8b. Note that both recovery paths guarantee that, at least, 75% of the requested bitrate is served in case of failure.

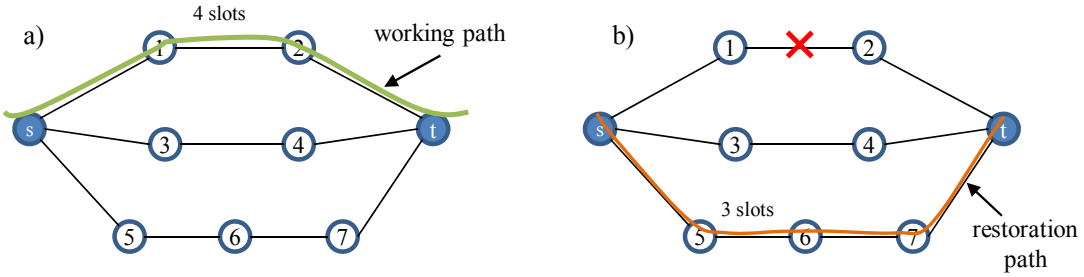

Fig. 2-7 Path restoration in action.

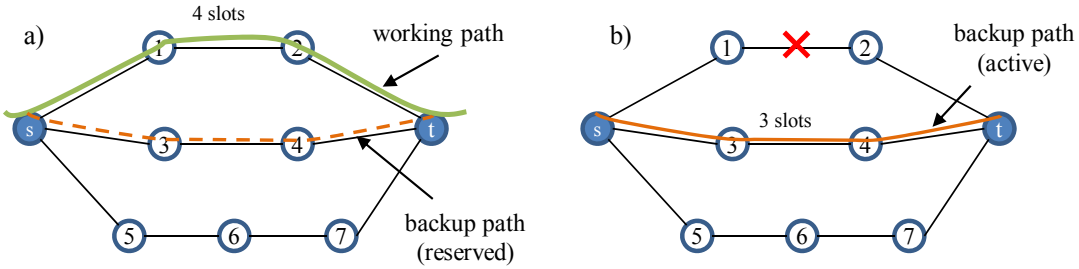

Fig. 2-8 SPP in action.

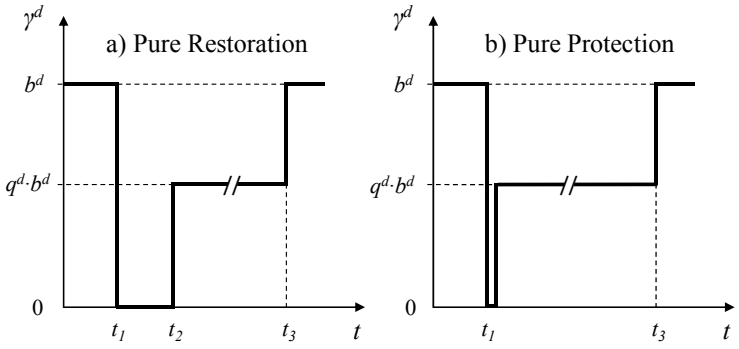

Fig. 2-9 Partial path restoration (a) and protection (b).

Fig. 2-9 presents the evolution of the bitrate actually served (y^d) for a demand d as a function of time when restoration (Fig. 2-9a), and protection (Fig. 2-9b) is used for recovery. In the period $t<t_1$ all the requested bitrate is served. In $t=t_1$ a failure impacts the working path of demand d and in $t=t_3$ the failure is repaired. When restoration is used for recovery in Fig. 2-9a, no bitrate is served until the restoration path is established for the demand in $t=t_2$. On the opposite, when protection is used for recovery, the minimum bitrate is served just after the failure is detected in $t=t_1$, as shown in Fig. 2-9b. In $t=t_3$ the failure is eventually repaired and all the requested bitrate is served again in both schemes.

2.6 Metaheuristics

A heuristic is an algorithm to obtain problem solutions, where the guarantee of finding optimal solutions is sacrificed for the sake of getting good solutions in a significantly reduced amount of time. Among heuristics, constructive methods and local search methods can be distinguished. Constructive algorithms generate solutions from scratch by adding (to an initially empty partial solution) components, until a solution is complete. Local search algorithms start from some initial solution and iteratively try to replace the current solution by a better solution in an appropriately defined neighborhood of the current solution.

A metaheuristic can be defined as an iterative master process that guides and modifies the operations of subordinate heuristics to efficiently produce high-quality solutions. The subordinate heuristics may be high (or low) level procedures, a simple local search, or just a constructive method. Metaheuristics are generic algorithm frameworks which can be applied to different optimization problems with relatively few modifications. Examples of metaheuristics are Tabu Search, Variable Neighborhood Search, Ant Colony, and Simulated Annealing [Ge10].

In the next subsections we present the metahueristics use in this thesis.

2.6.1 Greedy Randomize Search Procedure

The Greedy Randomize Search Procedure (GRASP) is an iterative two phase meta-heuristic method based on a multi-start randomized search technique with a proven effectiveness in solving hard combinatorial optimization problems. It was first presented in [Fe89], [Fe95], and later formalized and given its acronym in [Fe94]. Since then, it has been used to solve a wide range of problems (see e.g., [Pa10], [Vi11], [Pe10]) with many and varied applications in the real life such as the design of communication networks, collection and delivery operations and computational biology. For recent and comprehensive surveys of GRASP we refer the reader to [Re03], [Re08], [Re10.1], [Re10.2], [Fe09.1], [Fe09.2].

In the first phase of the multi-start GRASP procedure, a greedy randomized feasible solution of the problem is built by means of a construction procedure. Then, in the second phase, a local search technique to explore an appropriately defined neighborhood is applied in an attempt to improve the current solution. These two phases are repeated until a stopping criterion is met, and once the procedure finishes the best solution found over all GRASP iterations is returned. Note that with the basic GRASP methodology, iterations are independent from each other as previous solutions of the algorithm do not have any influence on the current iteration. One approach to include memory in the GRASP procedure is with Path Relinking (PR), a method which was first introduced in [Gl96], as an strategy to integrate both intensification and diversification in the context of tabu search [Gl97]. This approach generates new solutions by exploring the trajectories connecting high-quality solutions. The path evaluated starts at a so-called *initiating* solution and moves towards a so-called *guiding* solution which is usually taken from an stored set of good quality solutions called the *elite set*.

PR was first applied in the context of GRASP in [La99], and widely applied ever since. A wide variety of examples and applications of GRASP+PR are presented in [Re05]. After a solution is output from the multi-start phase (i.e., construction plus local search), PR is applied between the current solution and a selected solution from the elite set. Then, the best solution found in this iteration is candidate for inclusion in the elite set and it is only added if a certain quality and diversity criteria is met.

2.6.2 Biased Random Key Genetic Algorithm

The Biased Random Key Genetic Algorithm (BRKGA), a class of Genetic Algorithms (GA), has been recently proposed to effectively solve optimization problems, in particular, network related problems such as routing in IP networks and RWA in optical networks [Go10], [No10], [Re11], [Go11]. Compared with other meta-heuristics, BRKGA has provided better solutions in shorter running times. As in GAs, each individual solution is represented by an array of n genes (chromosome), and each gene can take any value in the real interval [0, 1]. Each chromosome encodes a solution of the problem and a fitness value, i.e., the value of the objective function. A set of p individuals, called a population, evolves over a number of generations. At each generation, individuals of the current generation are selected to mate and produce offspring, making up the next generation. In BRKGA, individuals of the population are classified into two sets: the elite set p_e, those individuals with the best fitness values, and the non-elite set. Elite individuals are copied unchanged from one generation to the next, thus keeping track of good solutions. The majority of new individuals are generated by crossover combining two elements, one elite and another non-elite, selected at random. An inheritance probability (ρ_e) is defined as the probability that an offspring inherits the gene of its elite parent. Finally, to escape from local optima a small number of

mutant individuals (randomly generated) to complete a population are introduced at each generation. A deterministic algorithm, named *decoder*, transforms any input chromosome into a feasible solution of the optimization problem and computes its fitness value.

In the BRKGA framework, the only problem-dependent parts are the chromosome internal structure and the decoder, and, thus, one only needs to define them to completely specify a BRKGA heuristic.

2.7 Conclusions

In this chapter some background in optical networks and optimization has been introduced. The basics of translucent DWDM and flexgrid optical networks, the technologies in which our work will be based, were presented. Next, some background related to recovery was introduced. Finally, optimization tools, mainly focused on heuristics, were described.

Next chapter reviews the state-of-the-art regarding the objectives of this thesis.

Chapter 3

Review of the State-of-the-Art

Aiming at confirming that our objectives are not yet covered, in this chapter we review the previous work related to this thesis that can be found in the literature. The first section is devoted to review off-line network planning; next, we survey the current works regarding dynamic network operation, including provisioning and restoration. Finally, we review works related to network re-optimization.

3.1 Static Planning

3.1.1 CAPEX studies

The network optimization problem has been tackled in the literature with a variety of objectives, perhaps the most recurring if the one focused on multilayer networks (see e.g., [Zh02], [Pi04], [Ho06], [Ko06], [Ch10], [Ru11]).

In [Zh02], authors investigated the traffic-grooming problem in a DWDM mesh network. An ILP formulation for traffic grooming in such a DWDM mesh network was presented. Moreover, the performance of the single-hop grooming approach and multi-hop grooming approach were compared. The results showed that the end-to-end aggregate traffic between the same node pair tends to be groomed on to the same lightpath channel, which directly joins the end points, if the optimization objective is to maximize the network throughput. The authors extended the optimization problem to a network-revenue model and found a different grooming scheme, which can be used to design an efficient heuristic algorithm on network-revenue model.

In fact, each time a novel optical transport technology emerges such a problem has to be redefined. For example, in [Ho06], authors proposed two greedy heuristics for

the multi-layer synchronous digital hierarchy (SDH) over DWDM network planning problem, one pure random start approach and a GRASP-like heuristic. The planning process does include explicit equipment specifications and costs and can be adjusted to several different kinds of hardware. However, the authors clarified that the GRASP approach could be improve and therefore the current solution provides just a promising basis.

Moreover, in [Ko06], authors studied the LSP routing problem for three different network models, namely, overlay, augmented, and peer model, in IP/MPLS over DWDM networks. In the overlay model, both topologies (IP/MPLS and DWDM) are completely decoupled; therefore, algorithms in each layer have information regarding only its own topology. In the peer model, all the topology information is shared between the two layers, so algorithms in each layer has a complete view of the multilayer topology. Note that, in this model the amount of information to be shared is very high. Finally, in the augmented model, some information is shared between layers. In this thesis, we will design multilayer networks assuming the overlay model.

In [Ch10], authors proposed to decompose the network into clusters and select a hub node in each cluster to groom traffic originating and terminating locally. At the second level of the hierarchy, the hub nodes form a virtual cluster for the purpose of grooming intra-cluster traffic. In this thesis, we assume that the metro traffic is first aggregated at the metro routers, named edge routers in this thesis, to create the first level of the core hierarchy. In fact, we add a second level within the core hierarchy, named as transit routers in this thesis.

More recently, in [Ru11], authors deal with the survivable multilayer IP/MPLS-over-WSON optimization problem. We take as reference this work and apply similar principles for IP/MPLS over flexgrid multilayer networks.

3.1.2 Resilience Schemes

Both, protection and restoration schemes have been widely studied in the literature [Ta07], [Ei11], [Ch09], [Pa12], [Hu11], [Da09], [Va11], [Va12], [So11], [Cl05]. Because of the trade-off between efficiency and recovery time, some authors have focused on reducing recovery times of restoration pre-computing K shortest disjoint paths [Ta07] or even preconfigured cycles [Ei11]. Other works have studied multi-path routing, e.g. authors in [Ch09] propose a mechanism for multiple lightpaths provisioning which satisfies extremely high bandwidth requirements whilst minimizing the amount of traffic affected by single link failures upper-bounding also differential delay. Authors in [Pa12] propose using multi-path provisioning for flexgrid optical networks as a mechanism to increase network performance. The application of multi-path to recovery was introduced by authors in [Hu11] where K disjoint paths are both provisioned and protected using SPP scheme.

The concept of partial protection by provisioning multiple paths was studied by authors in [Da09] and extended to mixed-line-rate scenario in [Va11], [Va12]. A slightly different partial recovery scheme was introduced in [So11] for flexgrid-based networks. It consists in provisioning only one path for the full requested bitrate and using restoration in case of failure to find a single backup path to recover part of the requested bitrate (*bandwidth squeezing*); this entails reducing the number of frequency slots that are actually recovered from a failure.

Authors in [Cl05] propose combining protection and restoration aiming at adding extra availability in case of double failures; protection is used to recover from the first failure and restoration to recover from the second one. Note however that in this scheme only one path is active at any given time.

In this thesis, we take advantage from protection and restoration schemes to simultaneously recover, partially or totally, affected connections. Note that this is only possible in flexgrid networks as a result of the flexible channel allocation.

3.2 Dynamic Provisioning

3.2.1 Translucent optical networks

This section surveys a set of relevant works in the field of translucent WSON from two standpoints, namely, the problem of placing 3R regenerators and the IA-RWA algorithm for on-line provisioning.

In translucent WSON, the problem of the regenerator placement aims to plan the most cost-efficient network when placing 3R regenerators is solved during the design network phase. In [Ya05.2], authors proposed four different regeneration placement algorithms to find the number of regenerators and the network nodes for which the number of established LSPs is maximized. The first algorithm, called Nodal Degree First (NDF) places the pool of 3R regenerators at the nodes with the highest nodal degree. The second algorithm, named Centered Node First (CNF), computes the nodes more “centered” to place the regenerator resources. A node is more “centered” than another node when the former is traversed by a larger number of shortest hop/distance paths for all the network node-pairs. In the third algorithm Traffic Load Prediction (TLP), it is assumed that some certain information about future network traffic demands is available. Then, a traffic-prediction regeneration placement algorithm is executed. By means of a predefined wavelength-routed algorithm, the aim is to route all the static connection demands and identify the nodes where regeneration is mostly used. Similarly, the fourth algorithm, called Signal Quality Prediction (SQP) places the regenerators at nodes for which the maximum distance of a transparent path segment is not exceeded at the time of computing a lightpath. Such a distance is derived as estimation, under the worst case, of the accumulated physical impairments. This may lead to over-

dimensioning the regenerator resources placed throughout the network [Le09.1]. More efficient approaches rely on using a signal quality metric (e.g., Q factor) when placing the regenerators [Pa08], [Le09.1].

Once the regenerator resources are placed in the network, the objective of the IA-RWA algorithm is to compute paths for the lightpath requests fulfilling the requirement of adequate optical signal quality and achieving optimal network resource utilization. Several IA-RWA algorithms have been proposed in the literature being roughly sorted according to the constraint used to estimate/evaluate the optical signal quality. The first category considers IA-RWA algorithms, which aim at not exceeding a maximum transparent distance [Sh02]. The second category includes the IA-RWA algorithms which estimate or analytically compute the optical signal quality using a single or a set of physical performance indicators such as the OSNR, estimated BER, the PMD, etc. [Ya05.1], [Ya05.2].

In this thesis, we rely on the OSNR computation to evaluate the quality of the signal conveyed by optical connections. We use OSNR to decide whether regenerators need to be used.

3.2.2 Flexgrid optical networks

Many works in the literature have addressed the RWA problem subject to the WCC constraint. Authors in [Le96] propose to use a dedicated graph for each of the wavelengths available in the spectrum. This algorithm has been used afterwards to provide link disjoint path pair for path protection in [Yu05], [Ve09]. In that RWA algorithm, a shortest path algorithm, e.g. the Dijkstra's algorithm [Bh99], is used to compute the shortest path in each parallel graph. Wavelength assignment is performed by choosing the wavelength that provides the shortest of all these paths; a number of heuristics have been developed for wavelength assignment including first fit (FF) [Zh00].

Due to the high number of parallel graphs required to represent the spectrum slots, such solutions become impractical in flexgrid optical network, the authors in [Ji10] propose a heuristic algorithm to be used in rings consisting in a routing and wavelength assignment algorithm based on FF under the spectrum-continuity constraint. Authors in [Wa11.3] proposed three different RSA algorithms for mesh networks. However, they explicitly omit different modulation formats, physical layer constraints like distance as a function of the requested bit rate, slot width, and guard bands between adjacent connections to avoid spectral clipping caused by the cascaded optical filters in the nodes.

Authors in [Wa11.1] studied the influence of the network traffic consisting of requests of mixed line bit rate on the optimal width of frequency slots. In fact, the optimal frequency slot width needs to be found since a larger slot width wastes spectrum and reduces network capacity, whereas a smaller slot width may improve

the spectrum utilization but imposes higher hardware requirements. They conclude that the optimal slot width can be computed as the greatest common factor of the spectrum width requested by the optical connections. In a similar work, the authors in [Sa11.1] studied the network performance as a function of the slot width. They showed that 12.5GHz is the best trade-off between grid granularity and blocking probability when all connections request for 400Gb/s. The results we present in this thesis show that such statements should be somehow clarified, weighting better the amount of connections per bit rate.

In [Sa11.2] the authors studied provisioning in mixed-rate WSONs to guarantee enough quality of transmission of the connections. Note that this is a complex task, as a result of the coexistence of optical connections operating at different bit-rates and modulation formats. To minimize setup times in dynamic scenarios, the authors proposed to use guard bands, i.e., leaving unused wavelength channels between lightpaths in order to reduce the effect of physical layer impairments. Authors in [Ri11] compared the cost-efficiency of 10/40/100Gb/s mixed-rate and 25/50/100Gb/s flexgrid architectures for translucent transport networks. They proved that despite the high price of individual transponders, flexgrid solutions can be more cost-efficient than mixed-rate solutions because of the better compatibility between different data rates, increasing the reach of channels and simplifying the spectrum allocation. In line with the latter study, in this thesis we assume that flexgrid transponders are available and thus phase modulated formats are used for all data rates. In particular, two modulation formats are considered: the quadrature phase shift keying (QPSK) and the 16-ary quadrature-amplitude modulation (16-QAM).

3.3 Dynamic Restoration

3.3.1 Restoration in MPLS-over-WSON

Some previous works have studied survivability in multi-layer networks (MLN). From the planning viewpoint, authors in [Ru11] studied the survivable MLN problem as a CAPEX minimization problem.

CAPEX can be notably reduced by using dynamic provisioning, as authors demonstrated in [Kr95]. The feasibility of GMPLS-based distributed restoration, where each node computes the restoration route for the sourced LSPs after a failure has been detected, was experimentally proved in [Ve10] for single layer and in [Gh08] for MLN. Notwithstanding, low restorability can be achieved mainly because of resource contention during the signaling phase as a consequence of outdated TED [Lu05]. To alleviate to some extent contention-caused blocking, the authors in [Sa07] proposed a contention detection scheme for GMPLS-based signaling.

However, the root cause of the problem is in the path computation and there it is where the use of a centralized PCE can help. Authors in [Al11] studied a mechanism to perform temporal reservations in the PCE's TED after a path has been computed waiting for arrival of the corresponding TED update messages.

Another mechanism to avoid contention, especially during restoration, is to create a TED copy and perform sequential restoration, as proposed in [Gi10.1] for single layer networks. Also in [Gi10.1], authors proposed to wait until all the restoration path computation requests arrive at the PCE to perform bulk path computation; nodes send GMPLS signaling Notify messages to the PCE upon a failure has been detected. That way the PCE can obtain the total number of affected lightpaths and wait until all path computation requests arrive.

Few works addressed the application of the GCO framework. As mention above, authors in [Ah12] presented a dynamic bulk provisioning framework with the objective of optimizing the use of network resources. PCE clients are allowed to bundle and simultaneously send multiple connection requests to the PCE where, in turn, several bundles can be concurrently processed together as a single bulk. To the best of our knowledge, however, the only works in the literature addressing bulk restoration are the ones presented in this thesis.

3.4 Re-optimization

In dynamic scenarios, wavelength channel fragmentation appears in WSON networks subject to the WCC constraint as a consequence of the unavailability of wavelength converters. Note that wavelength converters are rarely used in optical networks due to their high costs. Fragmentation increases the blocking probability of connection requests making worse the network Grade of Service (GoS).

The first work addressing the fragmentation problem in WSON was presented in [Le96], where a strategy named *move-to-vacant wavelength retuning* was proposed. It is based on re-allocating the already established optical connections to other wavelengths in order to make enough room for new connections that otherwise would be blocked. To reduce the disruption period, a parallel connection in the same route is first created in a free wavelength for each connection to be rerouted. This strategy provides the advantage that old and new routes share the same optical nodes so their computation and establishment are facilitated. Authors in [Mo99] presented an algorithm to improve that in [Le96], introducing rerouting in addition to the simple wavelength retuning. Several rerouting strategies are investigated in [Ch08], where the authors also proposed intentional rerouting, which runs rerouting periodically. The fragmentation problem appears also in grooming-capable optical networks and some works already addressed it. The work in [Ya08] presented two rerouting algorithms to alleviate resource inefficiency and improve the network throughput. Authors in [Ag09] presented a centralized flow

reallocation module to minimize the overall network cost. The standardized *make-before-break* rerouting technique (similar to the move-to-vacant one) is also experimentally tested establishing the new path for rerouting and transferring traffic from the current path to the new one before the old path is finally torn down. In this way, reallocation procedures do not cause any working traffic disruption.

Two main strategies have been proposed to reallocate (including rerouting and wavelength reassignment) already established paths in the context of WSON: periodic defragmentation and path-triggered defragmentation. The former strategy focuses on minimize fragmentation itself all over the network at given period of time, whereas the latter focuses on making enough room for a given connection request if it cannot be established with current resources allocation. Periodic defragmentation, requiring long computation times as a result of the amount of data to be processes, is essentially performed during low activity periods, e.g. during nights [Ag09]. Conversely, path-triggered defragmentation, involving only a limited set of already established connections, might provide solutions in shorter times and can be run in real time. It is worth noting that incoming connection requests or tear-downs arriving while network re-optimization or defragmentation operations are running must be kept apart until they terminate.

The problem of fragmentation is worse in flexgrid optical networks since, in addition to spectrum continuity, spectrum contiguity needs to be guaranteed. Some works recently addressed this issue [We11], [Am11], [Ta11]. Authors in [We11] proposed a spectrum defragmentation algorithm that takes advantage from BV-WSS that support wavelength conversion. Blocking probability can be reduced by 50% at the expense of increasing notably the cost of the photonic layer. Experimental wavelength conversion and its application to reduce spectrum fragmentation were presented in [Am11]. In [Ta11], the authors proposed an algorithm that performs spectrum defragmentation using rerouting instead of wavelength conversion. Since rerouting usually causes service disruption, they used the make-before-break technique to minimize disruption.

In this thesis, we apply re-optimization to flexgrid-based networks to solve the problem of spectrum fragmentation. Specifically, spectrum defragmentation is applied after a request, for a new connection or to expand the capacity of an existing one, cannot be served.

3.5 Conclusions

In this chapter, the state-of-the-art of the previous work related to this thesis has been reviewed.

Relevant planning problems for multilayer networks with WSON optical technology were presented. It is clear that being the Flexgrid technology so novel,

few studies are currently available, which clearly open opportunities to work in this topic.

Different resilience schemes for optical networks were surveyed. To best of our knowledge, multi-path partial recovery mechanisms combining protection and restoration to recover part of connections' bitrate in the event of link failures has not been faced so far.

Regarding dynamic provisioning, on the one hand, currently available IA-RWA algorithms for WSON translucent networks are very complex and the solutions not closed enough to optimality. On the other hand, algorithms to efficiently solve the RSA problem need to be devised.

As for dynamic recovery, current GMPLS control plane architectures for multilayer networks include algorithms for sequential path computation, which presents a poor resource sharing and high resource contention. This allows to devise new control plane architectures and algorithms that specifically focus on restoration scenarios, where several connection requests might arrive simultaneously to the PCE.

Solving the spectrum fragmentation problem is of paramount importance in flexgrid networks, as a result of the various amount of frequency slots allocated to optical connections. Although, spectrum defragmentation has been considered in WSON, no works facing this problem in the context of flexgrid are available in the literature.

Finally, Table 3-1 summarizes the above survey of the state-of-the-art. As a conclusion, the goals of this thesis have not been covered so far in the literature, mainly as a result of the novelty of flexgrid technology.

Table 3-1 SoA summary

Related goal		References
Static Planning	**CAPEX studies**	[Zh02], [Pi04], [Ho06], [Ko06], [Ch10], [Ru11]
	Resilience Schemes	[Ta07], [Ei11], [Ch09], [Pa12], [Hu11], [Da09], [Va11], [Va12], [So11], [Cl05]
Dynamic Provisioning	**Translucent networks**	[Ya05.1], [Ya05.2], [Le09.1], [Pa08]
	Flexgrid optical networks	[Le96], [Yu05], [Ve09], [Bh99], [Zh00], [Ji10], [Wa11.1], [Wa11.3], [Sa11.1], [Sa11.2], [Ri11]
Dynamic Restoration	**MPLS-over-WSON**	[Kr95], [Ve10], [Gh08], [Lu05], [Sa07], [Al11], [Gi10.1], [Ru11], [Ah12]
Re-optimization		[Le96], [Mo99], [Ch08], [Ya08], [Ag09] [Ag09], [We11], [Am11], [Ta11]

In the following chapters, we present the work conducted to accomplish the thesis goals.

Chapter 4

CAPEX study for flexgrid optical networks

In this chapter, we analyze the cost implications that a set of frequency slot widths has on the CAPEX investments required to deploy MPLS-over-flexgrid networks. We tackle the multilayer MPLS-over-flexgrid optimization (MIFO) problem. To this end, we model the MIFO problem by means of an ILP formulation. For the sake of a compelling analysis, exhaustive numerical experiments are carried out considering a set of realistic network topologies, network equipment costs and traffic instances. Results show that investments in optical equipment capable of operating under slot widths of 12.5 GHz, or even 25 GHZ, are more appropriated given the expected traffic evolution.

4.1 Motivation

Our goal in this chapter is to analyze for a number of candidate slot widths, the CAPEX needed to deploy a multi-layer MPLS-over-flexgrid architecture.

It is clear that finer grids will allow for more efficient spectrum utilization, and as a result, favor grooming data directly at the optical layer instead of requiring costly MPLS equipment for such functionality. Thus, given the fact that the network CAPEX, is a figure that network operators are always striving to reduce, the introduction of flexgrid technology is of paramount importance for future multi-layer networks [Ri11]. However, it must be noted that while reducing the need for grooming at the MPLS layer, this more advanced optical technology will also imply higher costs at the optical layer given the highly demanding (grid-dependent) filtering characteristics that BV-WSSs are required to have. In addition, due to the

increased spectrum fragmentation (particularly for the 12.5 GHz and 6.25 GHz grids), more complex network management, and therefore, more advanced control planes will be required, thereby leading to cost increases. Since exact costs for such components are still not available, in this chapter we consider a relative cost value to approximately quantify both these additional costs and, by this means, effectively determine which frequency grid will better address network operator's needs for cost-effective, spectrum-efficient network architectures.

4.2 Multilayer MPLS-over-Flexgrid Optimization (MIFO) problem

The aim of the MIFO problem is to effectively exploit network resources while, at the same time, minimizing CAPEX investments. This sets our target on the reduction of MPLS equipment. We consider QPSK as the modulation format for all traffic demands, which are assumed to be of 10, 40, 100 or 400 Gb/s each. This way, the focus is set on the evaluation of the CAPEX savings that can be achieved through the use of narrower slot widths

In flexgrid networks, the conversion from bitrate requested to spectrum width has impact on the spectral efficiency achieved, but also on the number and type of BV-Ts deployed. In this chapter, the grooming of demands into lightpaths aims at minimizing the spectrum usage. Hence, considering the type and BV-T costs provided in the results the results section (4.4), two demands of 10Gb/s, following the same path, would be groomed into a 40 Gb/s lightpath (requiring one 40Gb/s BV-T at each end) in both the 50GHz and 25GHz grids. On the other hand, in both the 12.5GHz and 6.25GHz grids, two 10Gb/s lightpaths (two 10Gb/s BV-Ts at each end) would be set up. Note that in the 12.5GHz grid the tie in the number of slots is broken by selecting the cheapest option, which in this case is two 10Gb/s BV-Ts. According to this discussion, it can be anticipated that networks using finer slot widths would feature a larger number of BV-Ts but with a considerably lower average bit-rate.

In order to tackle MIFO, we assume that a network topology representing a set of geographical locations as well as the interconnectivity among them (i.e. the fibers are already deployed) is given in advance. In these sites, network equipment can be installed if necessary. For illustrating proposes, in Fig 4-1(a) colored circles represent nodes that are source/destination of MPLS traffic demands, and white ones represent candidate locations where network equipment can be installed if necessary. Moreover, we assume that only a limited number of locations can be source/destination of MPLS demands (colored locations). As required, nodes are equipped with BV-Ts so as to provide connectivity between the electronic and optical layer. The remaining locations (intermediate locations) are candidate spots where network equipment is installed according to the functionality required.

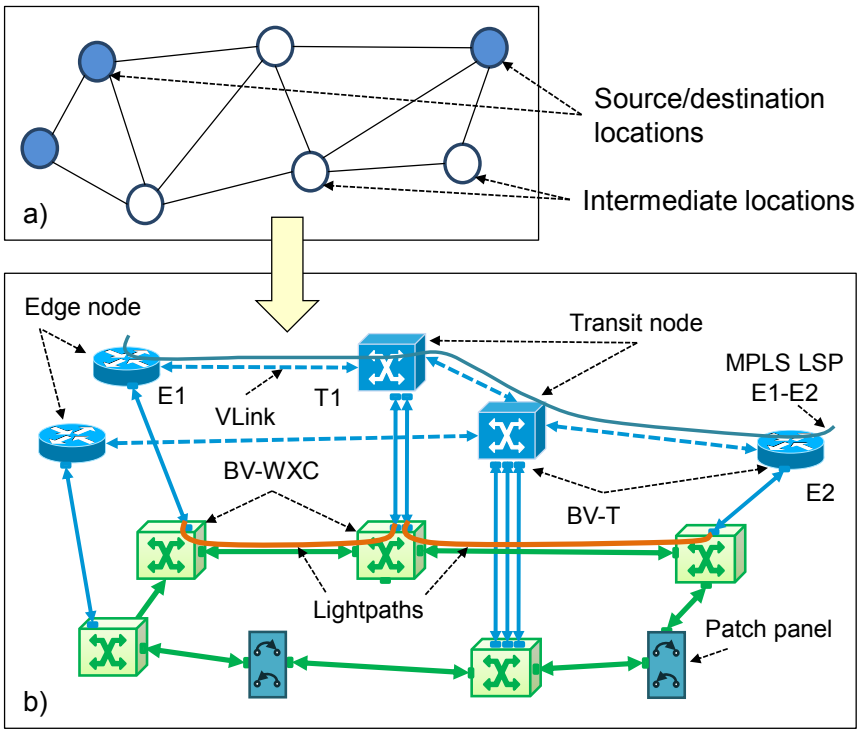

Fig 4-1 Example of a possible solution for the MIFO problem.

Specifically, and given a set of traffic demands to be accommodated, intermediate locations can be either:

- a multilayer node with both MPLS and BV-OXC functionality.
- a BV-OXC node if no MPLS operation is required.
- a patch panel connecting optical fibers if neither MPLS nor BV-OXC is required at such location.
- an empty location if no demand traverses such location.

In Fig 4-1(b), a multilayer network exemplifying a possible solution to the MIFO problem applied to the topology of Fig 4-1(a) is shown. One can observe locations that are equipped with multilayer nodes providing either client flow aggregation (*edge nodes*) or routing flexibility (*transit nodes*). Other locations, however, only operate as patch panels for fiber connectivity purposes, thereby minimizing network CAPEX.

As an example, Fig 4-1(b), illustrates one of the MPLS LSPs established in the network, that is, LSP E1-E2. In this case, this LSP entails setting up two lightpaths to support virtual links E1-T1 and T1-E2. Recall that one lightpath, which is associated to two BV-Ts (one at each end), can be used to transport several traffic demands, and, by this means, provide the grooming functionality required to optimize the use of network resources. It is worth pointing out that the solution obtained by solving the MIFO problem (i.e., the location, type and quantity of

network equipment deployed) will vary in accordance to the input traffic demands considered.

In the next section, the MIFO problem is first formally stated, and an ILP-based formulation of the problem is afterwards provided.

4.3 Mathematical Formulation

4.3.1 MIFO problem statement

The MIFO problem can be formally stated as follows:

Given:

- a network topology represented by a graph $G_o(V, L)$, being V the set of locations and L the set of bidirectional fiber links connecting two locations; each link consists of two unidirectional optical fibers,
- a set S of available frequency slots of a given spectral width in each link in L,
- the virtual network represented by a graph $G_v(V_v, E)$, being V_v the subset of locations V where MPLS nodes can be placed, and E the set of virtual links defining the connectivity among the MPLS locations,
- a set D of MPLS demands to be transported,
- the MPLS equipment cost, specified by a fixed cost for every type of MPLS node and BV-T,
- the BV-OXC nodes cost, which includes a fixed cost for the base system, and a variable cost as a function of the nodal degree including the cost of BV-WSSs (note that the relevant nodal degree, i.e. the number of incident links, is that of the final solution). See figure 4, in [Ji09], for further details on the BV-OXC architecture,
- a cost for every intermediate optical amplifier to be equipped in the fiber links and a cost per km and GHz for using the deployed fiber. It should be emphasized that the fiber costs considered in this chapter only relate to its use, and not to its deployment. Consequently, the cost per spectral bandwidth is also accounted for to have a better estimation of cost of the fiber resources used. Note that if only half of the spectral bandwidth of a given fiber is used, the remaining free capacity can be rented to other clients.

Output:

- the optical network, including patch panels, optical nodes with the number of optical interfaces, and the used fiber links,

- the configuration of MPLS nodes in terms of capacity and number and bit-rate of BV-Ts.

Objective: minimize the total CAPEX of the designed multilayer network for the given set of demands.

The problem can be tackled by solving the following ILP model.

4.3.2 ILP model

The ILP model for the MIFO problem designs both the optical and the MPLS layers using two node-link formulations [Pi04], that is, one per network layer. Note that although the link-path formulation could be used for the optical layer, the amount of routes to be pre-computed should be large enough to overcome the fact that some locations would not be equipped in the optimal solution. Instead, we compute a set of virtual links connecting every pair of locations where MPLS nodes can be installed. For each virtual link, a set of lightpaths is available but its route on the optical topology is determined during the resolution of the problem.

As to how the spectrum allocation is performed, channels are used to ensure frequency slot contiguity in the input data [Ve12.2], thereby alleviating to some extent the problem complexity. The characteristics of the considered modulation format are also embedded in the input data. To be precise, the slot's capacity in Gb/s is pre-computed (parameter *sk*) and different optical signal reaches (parameter *len(r)*) as a function of both the bitrate and modulation format are considered. Aiming at simplifying the dimensioning of the MPLS nodes, a set of port slots is available at each location. The solution of the problem provides the characteristics of the specific BV-T installed in each port slot, if any.

The following sets and parameters have been defined:

Topology:

V	Set of locations, index v.
L	Set of fiber links, index l.
$len(l)$	Length of fiber link l in km.
$L(v)$	Subset of fiber links incidents to location v.
V_v	Set of locations where MPLS nodes can be placed.
E	Set of virtual links, index e.
$K(e)$	Set of lightpaths to support virtual link e, index k.
$E(v)$	Subset of virtual links incidents to node v.
$V(e)$	Set of end nodes of virtual link e.
$P(v)$	Set of port slots of location v, index p.

as	Amplifier span for fiber links in kilometers.

Spectrum and modulation formats:

S	Set of frequency slots, index s.
sw	Frequency slot width in GHz.
sk	Frequency slot capacity in Gb/s.
C	Set of channels, index c. Each channel c contains a subset of contiguous frequency slots.
n^c	Number of frequency slots included in channel c.
$h^c{}_s$	Equal to 1 if channel c includes frequency slot s, 0 otherwise.
R	Set of bitrate-reach pairs (Gb/s, km), index r.
$len(r)$	Reach of a lightpath using bitrate-reach pair r in km.
$bw(r)$	Maximum bitrate of a lightpath using bitrate-reach pair r in Gb/s.

Demands:

D	Set of MPLS demands, index d.
$SD(d)$	Set of source and destination nodes of demand d.
b_d	Bandwidth of demand d in Gb/s.

Equipment, costs and others:

NT	Set of node types, index n ({Patch, BV-OXC, MPLS}). Note that type MPLS includes one BV-OXC.
C_{OXC}	Fixed cost of one BV-OXC. Includes common circuitry and BV-WSSs.
C_{FO}	Cost per km and GHz of using the optical fiber.
C_{Trunk}	Cost of one trunk, which includes one BV-WSS, one optical amplifier, and one optical splitter.
$max\varphi$	Maximum optical nodal degree (number of trunks) of a BV-OXC.
C_{OA}	Cost of each intermediate optical amplifier.
RT	Set of MPLS node classes, index j. Each class defined by a switching capacity and a number of BV-Ts slots.
rk_j	Switching capacity of a MPLS node class j in Gb/s.
rpk_j	Number of BV-Ts port slots available in a MPLS node class j.
rc_j	Cost of one MPLS node of class j.
PT	Set of BV-Ts bit-rates, index i.

pk_i	Capacity of a BV-Ts of bitrate i in Gb/s.
mpc_i	Cost of one BV-Ts of bitrate i in an MPLS node.
M	A large positive constant.

The decision variables are:

ω_{dek}	Binary. Equal to 1 if demand d is routed through lightpath k of virtual link e, 0 otherwise.
$\delta^c{}_{ek}$	Binary. Equal to 1 if lightpath k of virtual link e uses channel c, 0 otherwise.
$\lambda^c{}_{ekl}$	Binary. Equal to 1 if lightpath k of virtual link e uses channel c in fiber link l, 0 otherwise.
γ_l	Binary. Equal to 1 if fiber link l is used, 0 otherwise.
$\sigma^{e'k'}{}_{ekl}$	Binary. Equal to 1 if lightpath k of virtual link e and lightpath k' of virtual link e' share fiber link l, 0 otherwise.
$\psi^{vp}{}_{ek}$	Binary. Equal to 1 if lightpath k of virtual link e is assigned to port slot p in location v, 0 otherwise.
φ^v	Positive integer with the optical nodal degree of location v.
$\rho_i{}^{vp}$	Binary. Equal to 1 if port slot p of location v is equipped with a BV-T of bitrate i, 0 otherwise.
$\Pi_j{}^v$	Binary. Equal to 1 if location v is equipped with an MPLS node of class j, 0 otherwise.
$\mu_n{}^v$	Binary. Equal to 1 if location v is equipped with a node type n, 0 otherwise.
τ^{vp}	Positive integer with the total amount of traffic (in Gb/s) using port slot p of location v.
$\nu^r{}_{ek}$	Binary. Equal 1 if lightpath k of virtual link e uses bitrate-reach pair r.
α^v	Positive real with the optical cost of location v.
β^v	Positive real with the MPLS cost of location v.

Then, network CAPEX can be computed as the sum of the following expressions, where equation (4.1) computes the cost of the nodes and BV-Ts ports, and equation (4.2), the cost of the optical fiber as a result of both installing intermediate optical amplifiers and using the fiber links:

$$COST_{Equipment} = \sum_{v \in V} \left(\alpha^v + \beta^v \right) \tag{4.1}$$

$$COST_{FO} = \sum_{l \in L} \gamma_l \cdot \left\lceil \frac{len(l)}{as} - 1 \right\rceil \cdot C_{OA} + \sum_{l \in L} \sum_{c \in C} \sum_{e \in E} \sum_{k \in K(e)} \lambda_{ekl}^{c} \cdot n_c \cdot sw \cdot len_l \cdot C_{FO} \tag{4.2}$$

Finally, the ILP for the MIFO problem is as follows:

$$\text{minimize } CAPEX = COST_{Equipment} + COST_{FO} \tag{4.3}$$

subject to:

$$\sum_{e \in E(v)} \sum_{k \in K(e)} \omega_{dek} = 1 \quad \forall d \in D, v \in SD(d) \tag{4.4}$$

$$\sum_{e \in E(v)} \sum_{k \in K(e)} \omega_{dek} \leq 2 \quad \forall d \in D, v \in \overline{SD(d)} \tag{4.5}$$

$$\sum_{\substack{e' \in E(v) \\ e' \neq e}} \sum_{k \in K(e')} \omega_{de'k} \geq \sum_{k \in K(e')} \omega_{dek} \quad \forall d \in D, v \in \overline{SD(d)}, e \in E(v) \tag{4.6}$$

$$\sum_{l \in L(v)} \sum_{c \in C} \lambda_{ekl}^{c} = \sum_{c \in C} \delta_{ek}^{c} \quad \forall e \in E, k \in K(e), v \in V(e) \tag{4.7}$$

$$\sum_{l \in L(v)} \sum_{c \in C} \lambda_{ekl}^{c} \leq 2 \quad \forall e \in E, k \in K(e), v \in \overline{V(e)} \tag{4.8}$$

$$\sum_{\substack{l' \in L(v) \\ l' \neq l}} \sum_{c \in C} \lambda_{ekl'}^{c} \geq \sum_{c \in C} \lambda_{ekl}^{c} \quad \forall e \in E, k \in K(e), v \in \overline{V(e)}, l \in L(v) \tag{4.9}$$

$$\sum_{c \in C} \delta_{ek}^{c} \leq 1 \quad \forall e \in E, k \in K(e) \tag{4.10}$$

$$\sum_{c \in C} sk \cdot n^{c} \cdot \delta_{ek}^{c} \geq \sum_{d \in D} b_d \cdot \omega_{dek} \quad \forall e \in E, k \in K(e) \tag{4.11}$$

$$\sum_{l \in L} \lambda_{ekl}^{c} \leq M \cdot \delta_{ek}^{c} \quad \forall e \in E, k \in K(e), c \in C \tag{4.12}$$

$$\sum_{e \in E} \sum_{k \in K(e)} \sum_{c \in C} h_s^{c} \cdot \lambda_{ekl}^{c} \leq 1 \quad \forall l \in L, s \in S \tag{4.13}$$

$$\sum_{d \in D} b_d \cdot \omega_{dek} \leq \sum_{r \in R} bw(r) \cdot \nu_{ek}^{r} \quad \forall e \in E, k \in K(e) \tag{4.14}$$

$$\sum_{l \in L} \sum_{c \in C} len(l) \cdot \lambda_{ekl}^{c} \leq \sum_{r \in R} len(r) \cdot \nu_{ek}^{r} \quad \forall e \in E, k \in K(e) \tag{4.15}$$

$$\sum_{r \in R} \nu_{ek}^{r} \leq 1 \quad \forall e \in E, k \in K(e) \tag{4.16}$$

$$\sum_{e\in E}\sum_{k\in K(e)}\sum_{c\in C}\lambda_{ekl}^{c} \le M\cdot\gamma_{l} \quad \forall l\in L \tag{4.17}$$

$$\sum_{l\in L(v)}\gamma_{l} = \varphi^{v} \quad \forall v\in V \tag{4.18}$$

$$\varphi^{v} \le M\cdot\sum_{n\in NT}\mu_{n}^{v} \quad \forall v\in V \tag{4.19}$$

$$\sum_{n\in NT}\mu_{n}^{v} \le 1 \quad \forall v\in V \tag{4.20}$$

$$\varphi^{v} \le \max\varphi + M\cdot\mu_{n}^{v} \quad \forall v\in V, n\in\{Patch\} \tag{4.21}$$

$$\sum_{d\in D}\sum_{k\in K(e)}\omega_{dek} \le M\cdot\mu_{n}^{v} \quad \forall e\in E, v\in V(e), n\in\{MPLS\} \tag{4.22}$$

$$\sum_{n\in\{WXC,MPLS\}}\mu_{n}^{v} + \sum_{\substack{l'\in L(v)\\ l'\neq l}}\sigma_{ekl'}^{e'k'} \ge 1 + M\cdot(\sigma_{ekl}^{e'k'} - 1) \quad \begin{array}{l}\forall e,e'\in E, e\neq e',\\ k\in K(e),\\ k'\in K(e'),\\ v\in V, l\in L(v)\end{array} \tag{4.23}$$

$$\sigma_{ekl}^{e'k'} \le \sum_{c\in C}\lambda_{ekl}^{c} \quad \forall e,e'\in E, e\neq e', k\in K(e), k'\in K(e'), l\in L \tag{4.24}$$

$$\sigma_{ekl}^{e'k'} \ge \sum_{c\in C}\lambda_{ekl}^{c} + \sum_{c\in C}\lambda_{e'k'l}^{c} - 1 \quad \begin{array}{l}\forall e,e'\in E, e\neq e',\\ k\in K(e), k'\in K(e'), l\in L\end{array} \tag{4.25}$$

$$\sum_{d\in D}\omega_{dek} \le M\cdot\sum_{p\in P(v)}\psi_{ek}^{vp} \quad \forall e\in E, k\in K(e), v\in V(e) \tag{4.26}$$

$$\sum_{p\in P(v)}\psi_{ek}^{vp} \le 1 \quad \forall e\in E, k\in K(e), v\in V(e) \tag{4.27}$$

$$\sum_{e\in E}\sum_{k\in K(e)}\psi_{ek}^{vp} \le 1 \quad \forall v\in V, p\in P(v) \tag{4.28}$$

$$\sum_{d\in D} b_{d}\cdot\omega_{dek} \le \tau^{vp} + M\cdot(1-\psi_{ek}^{vp}) \quad \begin{array}{l}\forall v\in V, p\in P(v),\\ e\in E, k\in K(e)\end{array} \tag{4.29}$$

$$\tau^{vp} \le \sum_{i\in PT} pk_{i}\cdot\rho_{i}^{vp} \quad \forall v\in V, p\in P(v) \tag{4.30}$$

$$\sum_{i\in PT}\rho_{i}^{vp} \le 1 \quad \forall v\in V, p\in P(v) \tag{4.31}$$

$$\sum_{p\in P(v)}\tau^{vp} \le \sum_{j\in RT} rk_{j}\cdot\pi_{j}^{v} \quad \forall v\in V \tag{4.32}$$

$$\sum_{p\in P(v)}\sum_{i\in PT}\rho_i^{vp} \le \sum_{j\in RT} rpk_j \cdot \pi_j^v \quad \forall v \in V \tag{4.33}$$

$$\sum_{j\in RT}\pi_j^v \le 1 \quad \forall v \in V \tag{4.34}$$

$$\alpha^v + M\cdot\left(1-\sum_{n\in\{OXC,MPLS\}}\mu_n^v\right) \ge C_{OXC} + \varphi^v \cdot C_{Trunk} \quad \forall v \in V \tag{4.35}$$

$$\beta^v + M\cdot\left(1-\sum_{n\in\{MPLS\}}\mu_n^v\right) \ge \sum_{p\in P(v)}\sum_{i\in PT} mpc_i\cdot\rho_i^{vp} + \sum_{j\in RT} rc_j\cdot\pi_j^v \quad \forall v \in V \tag{4.36}$$

The objective function (4.3) minimizes the network CAPEX. Constraints (4.4)-(4.6) compute the route and perform aggregation of demands through the virtual topology. Constraint (4.4) ensures that only one virtual link incident to source and destination nodes is used to transport the demand. Constraints (4.5) and (4.6) perform the routing and aggregation in intermediate nodes. Constraints (4.7)-(4.9) compute the route over the physical topology of those lightpaths transporting demands, and likewise constraints (4.4)-(4.6) do for the demands over the virtual one. Constraints (4.10)-(4.13) perform spectrum allocation. Constraint (4.10) implements the spectrum continuity constraint ensuring that no more than one channel is allocated to one lightpath. Constraint (4.11) dimensions the size of the channel as a function of the aggregated bitrate. Constraint (4.12) guarantees that every lightpath uses the same channel along its route. Constraint (4.13) assures that each frequency slot is used by at most one lightpath. Constraints (4.14)-(4.16) take care of bitrate-reach pair selection. Constraint (4.14) chooses a pair with enough bitrate for the traffic to be transmitted and constraint (4.15) assures that the reach of that pair works for the length of the lightpath. Constrain (4.16) guarantees that only one pair is chosen.

Constraints (4.17)-(4.25) decide what equipment is installed in every location. Constraint (4.17) stores whether a fiber link is used or not and constraint (4.18) the nodal degree of each location. Constraint (4.19) ensures that a location is equipped provided that some incident fiber link is used. Constraint (4.20) guarantees that only one type of node is selected to be installed in any location. Constraint (4.21) limits the nodal degree of any BV-OXC. Constraint (4.22) ensures that an MPLS node (and its underlying BV-OXC) is installed in each location where any of the incident virtual links is used for transporting demands.

Constraints (4.23)-(4.25) decide whether a location must be equipped with a BV-OXC (with or without MPLS functionality) or with a patch panel should fiber connectivity be the only functionality required among pairs of incident fiber links. Note that a BV-OXC must be installed provided that capacity to optically switch lightpaths among the incident fiber links in a location is needed (i.e., there exists a

pair of lightpaths whose routes share only one incident fiber link at such particular location).

Constraints (4.26)-(4.31) deal with port slots. Constraint (4.26) ensures that any used lightpath is assigned to a port slot at its end nodes. Constraint (4.27) ensures that no more than one port is assigned at each end node of a lightpath. Constraint (4.28) ensures that a given port slot is not assigned to more than one lightpaths. Constraint (4.29) stores the total amount of traffic transmitted through a given port slot. Constraint (4.30) equips port slots with BV-Ts of bitrate enough to carry the amount of traffic assigned to the port slot. Constraint (4.31) ensures that only one BV-T is equipped in each port slot.

Constraints (4.32)-(4.34) manage MPLS nodes. Constraint (4.32) equips an MPLS node with switching capacity enough for the amount of traffic being switched at that location. Constraint (4.33) makes sure that the node type selected has enough port slots. Constraint (4.34) guarantees that only one node type is equipped at each location.

Finally, constraints (4.35) and (4.36) compute the cost of the BV-OXCs and MPLS nodes, respectively.

4.3.3 Complexity analysis

The MIFO problem can be considered *NP-hard* since simpler multilayer network planning problems have been proved to be *NP-hard* (e.g., [Zh11]). As to the MIFO problem size, the number of variables is $O(|E|^2 \cdot |K(e)|^2 \cdot |L| + |E| \cdot |K(e)| \cdot (|V| \cdot |P(v)| + |D| + |C| \cdot |L|)$ and the number of constraints $O(|E|^2 \cdot |K(e)|^2 \cdot |L| + |E| \cdot (|K(e)| \cdot |V| \cdot |P(v)| + |D|) + |L| \cdot |S|)$. It is worth highlighting that the number of variables and constraints rise up to approximately $3 \cdot 10^7$and $6 \cdot 10^8$ for the networks used in the results section (4.4).

Although the ILP can be solved for small instances, its exact solving becomes impractical for realistic backbone multilayer networks (under appreciable load) such as those described in the results section, even using commercial solvers such as CPLEX [CPLEX]. Thus, aiming at providing near-optimal solutions within reasonable computational effort, a GRASP heuristic to solve the MIFO problem was developed [Pe12.3]. Next section, reports the obtained results from running the mentioned heuristic.

4.4 MIFO Numerical Results

In this section, we first present the network scenarios that we consider in order to carry out our experiments. Second, we solve the MIFO problem, by using the GRASP heuristic, considering a set of realistic traffic instances.

4.4.1 Network Scenario

In order to conduct all the experiments, we consider the three optical network topologies shown in the Appendix A, these are, the 21-node TEL topology, the 20-node BT topology, and the 21-node DT topology. In these networks, we assume that $V_V=V$, that is, that any location can host an MPLS node. Besides, only locations in Table 4-1 can be source or destination of MPLS demands. The remaining locations (i.e., the intermediate locations) will be equipped according to the MIFO problem solution.

Table 4-1 Source/destination locations of MPLS traffic

	1	**2**	**3**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**16**	**17**	**18**	**20**
TEL	X	X		X	X	X	X		X		X	X	X	X	X	X
BT	X	X	X		X				X	X	X	X	X	X		X
DT	X		X			X		X			X	X		X	X	X

As for the traffic profiles (TP) considered, we make use of TP-2, TP-4 and TP-6 as reported in Appendix A. Although each TP injects into the network the same average amount of Tb/s, the traffic scenarios proposed feature lightly loaded demands in TP-2 (only 24.1 Gb/s on average), medium load demands (52.0 Gb/s) in TP-4, and high bit-rate demands in TP-6 (80Gb/s). Hence, the number of demands served decreases substantially from TP-2 to TP-6 in order to keep constant the total volume of Tb/s injected. These TPs are a realistic representation of the expected evolution of bandwidth necessities for the years to come in increasing order. Since our goal in this chapter is to evaluate the slot width impact on the network CAPEX, we considered a wide enough optical spectrum, taking into account the demands to be served in each TP. To be exact, using 2 THz we

Table 4-2 Cost and characteristics of MPLS nodes

Node	**Class 1**	**Class 2**	**Class 3**	**Class 4**	**Class 5**
Capacity (Gb/s)	160	320	640	1280	2560
Max. ports	4	8	16	32	64
Cost (c.u.)	9	13.5	19.5	67.5	150.57

Table 4-3 Cost and reach of BV-Ts

BV-T	**10Gb/s**	**40Gb/s**	**100Gb/s**	**400Gb/s**
Reach (Km)	2500	2000	1000	400
Cost (c.u.)	2.5	7.625	20.625	65.625

corroborated that for all traffic representations executed a feasible solution could be found. Table 4-2 (MPLS nodes) and Table 4-3 (BV-Ts) provide the characteristics of the network equipment considered as well as their value in cost units (c.u.) that we use to compute the CAPEX [Ra13].

In addition, we assume an optical amplifier cost of 5 c.u., and C_{FO}=0.02, that is, the cost per *km* and *GHz* of using the already deployed optical fiber. We consider QPSK as modulation format. Note that considering QPSK and the network topologies shown in Appendix A, it can be assumed that the number of cascaded BV-OXC traversed by a demand does not need to be limited [Ji10].

4.4.2 CAPEX using relative (grid-dependent) BV-WSS costs

In this subsection, we solve the MIFO problem with the aim of finding, given a target CAPEX investment and a set of relative cost values for the different bandwidth-variable wavelength selective switches (BV-WSSs), the maximum affordable cost for each of the (grid-dependent) BV-WSS. Although the actual cost for these enhanced optical devices (also involving higher costs due to a more complex control plane) is still not available, we assume that the finer the grid, the higher the relative cost for a BV-WSS device should be.

In Fig 4-2, Fig 4-3 and Fig 4-4, we provide for each network topology, TP, and frequency grid, the network CAPEX for the MIFO problem solution. Note that CAPEX here only accounts for the network equipment costs (i.e., eq. (4.2)). Each of the points in the plots corresponds to an average over 10 independent runs (each lasting for 40 iterations) of the heuristic algorithm. For the sake of a comprehensive analysis, we consider two traffic scenarios for each TP, these are, a highly loaded scenario (4.5 Tb/s are injected into the network), and a medium one (3.5 Tb/s). Thus, out of the 10 runs, 5 correspond to the highly loaded scenario and 5 to the medium one.

The plots in Fig 4-2, Fig 4-3 and Fig 4-4, clearly illustrate the effectiveness of narrower grids in grooming data directly at the optical layer, thus reducing network CAPEX. Besides, such a positive effect is clearly dependent on the TP considered. For the TP-2, one can observe that the introduction of finer grids allows for the spectrum to be better exploited, and hence, to achieve further benefit. In TP-4, by contrast, the 6.25 GHz grid provides the same performance as the 12.5 GHz in both the TEL and DT networks, and the 25 GHz in the BT network. Eventually, in TP-6, the main benefit is obtained just by considering a 25 GHz grid.

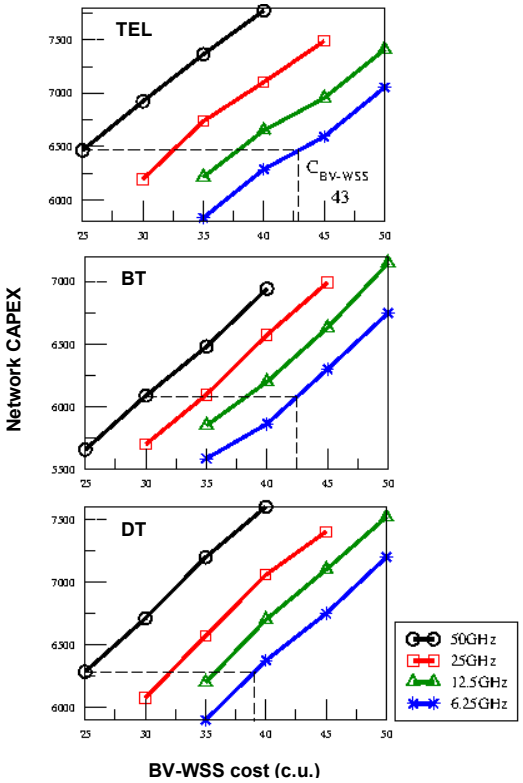

Fig 4-2 Network CAPEX versus the relative cost for one BV-WSS for TP-2.

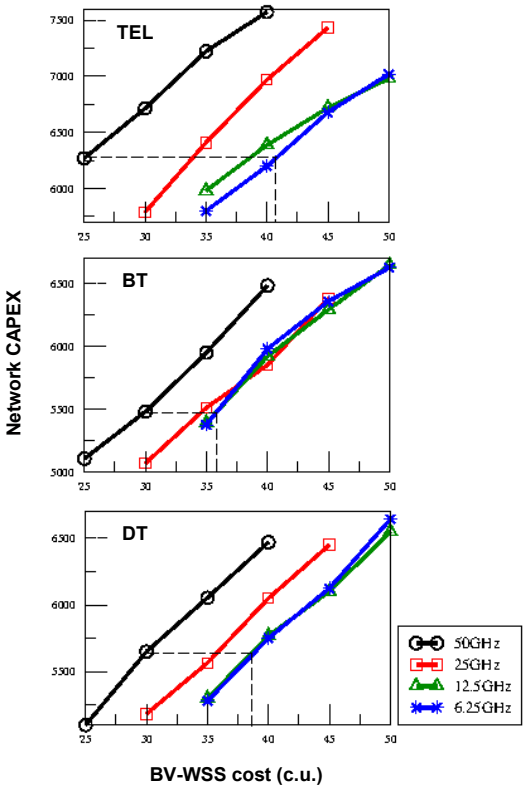

Fig 4-3 Network CAPEX versus the relative cost for one BV-WSS for TP-4.

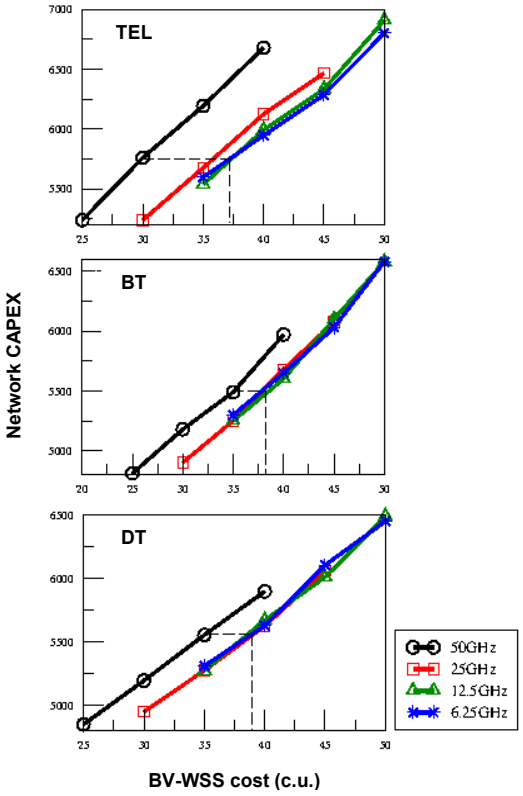

Fig 4-4 Network CAPEX versus the relative cost for one BV-WSS for TP-6.

Complementing these results, Table 4-4 reports the average number and bitrate of the installed BV-Ts, and Table 4-5 provides the average reduction (with respect to the 50 GHz grid) in both MPLS node switching capacity and actual amount of traffic switched (flow switched). As expected, these values are strongly dependent on both the frequency grid and TP evaluated. As long as the TP analyzed allows for it, the use of finer frequency grids entails a higher number of BV-Ts due to the more fragmented spectrum, which eventually results in a different grooming of demands into lightpaths (according to the mapping shown in Table 2-1 and the equipment costs provided in this section). However, this increase comes at the benefit of having a considerably much lower average bit-rate per BV-T, a fact which leads to lower switching capacity, and therefore, to cheaper MPLS equipment.

Finally, in order to estimate the maximum affordable cost increment for a BV-WSS in the 6.25 GHz grid, we use as benchmark reference a BV-WSS cost in the 50 GHz grid (see dotted lines in Fig 4-2, Fig 4-3 and Fig 4-4). In the TEL network under TP-2, the cost of the BV-WSSs can be, for the same network CAPEX, as high as 43 c.u., that is, 72% more expensive than the one used in the 50 GHz grid (25 c.u.). However, when the on-average bit-rate of the demands increases in both TP-4 and TP-6, the cost of a BV-WSS decreases to about 41 c.u. (64%) and 37 c.u. (23%), respectively.

Table 4-4 Avg. BV-T number and bitrate (Gb/s)

Network	Grid	#BV-T			BV-T bitrate		
		TP-2	TP-4	TP-6	TP-2	TP-4	TP-6
TEL	**50**	158	148	119	64	70	70
	25	190	179	118	50	49	67
	12.5	229	219	117	37	39	66
	6.25	277	260	118	30	32	66
BT	**50**	136	121	111	55	59	64
	25	170	150	112	42	47	63
	12.5	215	160	111	32	41	58
	6.25	255	164	110	26	39	57
DT	**50**	147	117	115	59	58	62
	25	183	140	117	49	50	61
	12.5	235	157	116	41	45	60
	6.25	257	170	117	31	41	61

Table 4-5 Avg. Reduction w.r.t. the 50GHz frequency grid

Network	Grid	Switching Capacity (%)			Flow Switched (%)		
		TP-2	TP-4	TP-6	TP-2	TP-4	TP-6
TEL	**25**	13	11	8	10	8	4
	12.5	20	13	8	16	9	4
	6.25	24	13	8	24	11	4
BT	**25**	5	4	1	5	5	1
	12.5	8	7	2	7	7	2
	6.25	14	7	2	14	8	3
DT	**25**	4	6	1	4	12	1
	12.5	11	10	1	11	13	2
	6.25	17	14	1	17	14	2

In Table 4-6, the average BV-WSS affordable cost increment provided by each frequency grid is reported. In light of these results, which represent an average over the three network topologies, it is clear that from a temporal perspective given by the on-average bit-rate of demands, high cost increments can be assumed for a 6.25GHz grid BV-WSS in the near future (57.6%). However, considering the expected traffic evolution, which for the long-term estimates a TP similar to the

Table 4-6 Avg. BV-WSS affordable cost increment per frequency grid (%)

Grid	TP-2	TP-4	TP-6	Avg.
25	25.6	27.3	16.6	23.2
12.5	41.3	36.1	23	33.5
6.25	57.6	38.9	23.4	39.9

TP-6 analyzed in this chapter, these investments will not be profitable. Therefore, it can be concluded that investments in flexgrid optical networks using the 12.5GHz or even the 25GHz grid (considering the increased management complexity of the network in finer frequency grids), are cheaper in the short-term and more appropriated for medium and long-term scenarios.

4.5 Conclusions

This chapter addressed the design of multilayer MPLS-over-flexgrid networks. To this end, an ILP formulation has been presented. Through extensive numerical experiments, the cost implications of several frequency grids have been analyzed. For the sake of a comprehensive study, a set of realistic network topologies, equipment costs, and traffic instances was considered.

Obtained results showed that the benefits that can be achieved using finer slot widths strongly depend on the actual TP under which the network is operating. Whilst investments in costly BV-WSS (finer grid) devices are very well motivated under traffic conditions reporting a high number of light bit-rate demands, which represent short-term traffic scenarios, they do not seem profitable in the long-term, where a reduced number of higher bit-rate demands are expected. Consequently, this study reports both the 12.5GHz and the 25GHz slot widths as potential candidates for the deployment of future multilayer networks based on flexgrid technology.

Once the use of flexgrid has been clearly justified, the next chapter focuses on proposing recovery mechanism specifically tailored for this optical technology.

Chapter 5

Recovery Schemes for flexgrid optical networks

In this chapter, we propose a new recovery scheme, called multi-path recovery (MPR), specifically designed for flexgrid-based optical networks. It combines protection and restoration schemes to jointly recover, in part or totally, the bitrate requested by client demands in case of failure. We define the bitrate squeezed recovery optimization (BRASERO) problem to maximize the amount of bitrate which is recovered in case of failure of any single fiber link; a MILP formulation for the BRASERO problem is provided. However, since their exact solution becomes impractical when real-sized network and traffic instances are considered, we develop a heuristic algorithm, which provides a much better trade-off between optimality and complexity. Exhaustive numerical experiments carried out over realistic network topologies and traffic scenarios show that the efficiency of the proposed MPR scheme approaches that of restoration while providing recovery times as short as protection schemes.

5.1 Bitrate Squeezed Recovery Optimization (BRASERO) problem

We propose to extend bitrate squeezing to protection schemes such as DPP and SPP so that backup paths guarantee only part of the requested bitrate. We propose the MPR scheme which combines protection and restoration schemes to provide the best from both, i.e. short recovery times with the highest efficiency. The proposed MPR scheme consists in ensuring part of the minimum bitrate by protection and complementing the rest by restoration.

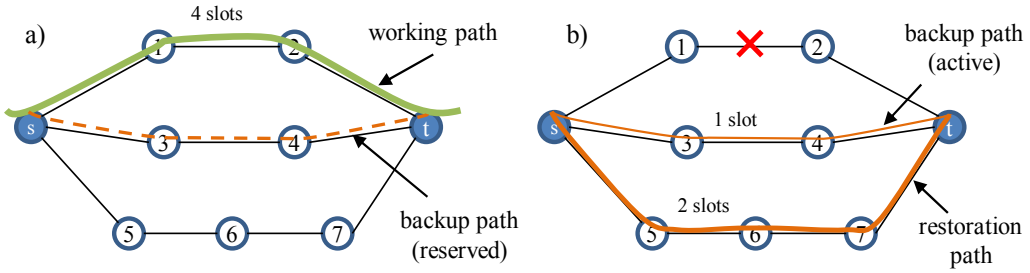

Fig. 5-1 MPR in action.

For illustrative purposes, Fig. 5-1 shows how MPR works. In Fig. 5-1a one client demand is requesting some amount of bitrate that translates into 4 slots. The working path serves all the requested bitrate and there are some slots reserved along the backup route to guarantee that some minimum bitrate is recovered. After a failure has impacted the working path, the backup path is activated and a restoration path is established in Fig. 5-1b. Note that the two recovery paths jointly guarantee that, at least, 75% of the requested bitrate is served for that failure scenario.

Fig. 5-2 presents the evolution of the bitrate actually served (γ^d) for a demand d as a function of time when only restoration (Fig. 5-2a), only protection (Fig. 5-2b) and a mixed of protection and restoration (Fig. 5-2c) is used for recovery. In the period $t<t_1$ all the requested bitrate is served. In $t=t_1$ a failure impacts the working path of demand d and in $t=t_3$ the failure is repaired. When only restoration is used for recovery in Fig. 5-2a, no bitrate is served until the restoration path is established for the demand in $t=t_2$. On the opposite, when only protection is used, the recovered bitrate is served just after the failure is detected in $t=t_1$, as shown in Fig. 5-2b. Since the outage inherent to restoration might not be desirable for some clients and lower efficiency might not be desirable for the network operator, the proposed MPR scheme serves the minimum bitrate ($q^d \cdot b^d$) just after the failure is detected in $t=t_1$ using protection, while complementing some bitrate using restoration later in $t=t_2$, as shown in Fig. 5-2c. In $t=t_3$ the failure is eventually repaired and all the requested bitrate is served again in all the three schemes.

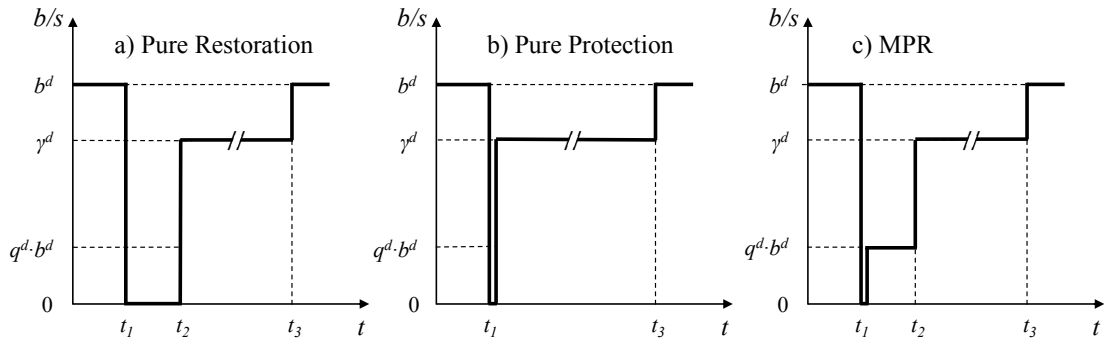

Fig. 5-2 Evolution of the served bitrate.

In the next section, the BRASERO problem is formally stated and a MILP-based formulation that uses each of the above-described schemes is proposed.

5.2 Mathematical Formulation

5.2.1 BRASERO Problem Statement

The problem can be formally stated as follows:

Given:

- a network topology represented by a graph G(N, E), where N is the set of optical nodes and E is the set of fiber links connecting two optical nodes,
- a set S of available frequency slots of a given spectral width in each fiber link in E,
- a set D of demands to be transported, each requesting a fixed bitrate b^d, and the minimum bitrate to be guaranteed. For the latter, it can be assumed that $m \cdot \Delta_S$ Gb/s must be ensured for each demand.

Output: the routing and spectrum assignment for each $d \in D$, including those scenarios where a failure in a fiber link $e \in E$ impacts the working route of d.

Objective: maximize the total recovered bitrate transported in case of failure of any single fiber link $e \in E$ provided that all demands are served in the non-failure scenario.

As previously discussed, the problem can be faced using protection (DPP or SPP), restoration, or our proposal of mixing of protection and restoration. A MILP-based model which includes the above schemes is presented next.

5.2.2 MILP model

The MILP model is based on the link-path formulation [Pi04], where a set of routes are computed beforehand for each demand. It is worth highlighting that the term path is used to describe the set of resources used or reserved to convey a given demand, i.e. a set of contiguous slots along with one route. In addition, the model uses a set of failure scenarios F where one fiber link fails at a time in each of them.

Regarding spectrum allocation, pre-computed channels are used to ensure frequency slot contiguity in the input data, thereby alleviating to some extent the problem complexity [Ve12.2]. The characteristics of the considered modulation format are also embedded in the input data. To be precise, different optical signal reach values, as a function of both the bitrate and modulation format, are considered.

The following sets and parameters have been defined:

Topology:

N	Set of optical nodes, index n.
E	Set of fiber links, index e.
K	Set of pre-computed routes, index k.
h^k_e	Equal to 1 if route k uses link e, 0 otherwise.
$len(k)$	Length of route k in Km.

Spectrum:

S	Set of frequency slots available in each link, index s.
Δ_S	Spectral width (GHz) of each frequency slot.
m	Spectral efficiency of the used modulation format (m=2 bit/s/Hz for QPSK).
C	Set of channels, index c. Each channel c contains a subset of contiguous slots.
l_{cs}	Equal to 1 if channel c uses slot s, 0 otherwise.
n_c	Number of slots of channel c.
R	Set of bitrate-reach pairs (Gb/s, km), index r.
$len(r)$	Reach of a path using bitrate-reach pair r in km.
$b(r)$	Maximum bitrate of a path using bitrate-reach pair r in Gb/s.

Demands:

D	Set of optical demands, index d.
b^d	Bitrate requested by demand d in Gb/s.
q^d	Percentage of bitrate to be assured in case of failure.
$n^d_{recovery}$	Number of frequency slots needed to convey the guaranteed bitrate in case of failure.
dpp	Binary, equal to 1 if DPP is used, 0 otherwise.
spp	Binary, equal to 1 if SPP is used, 0 otherwise.
$rest$	Binary, equal to 1 if restoration is used, 0 otherwise.
$Kw(d)$	Set of working routes for demand d (includes reach constraint).
$C(d)$	Set of feasible channels to convey the bitrate requested by demand d.
$Cp(d)$	Set of feasible channels for demand d so to protect some amount of bitrate in the range $[\Delta_S \cdot m, max(\Delta_S \cdot m, b^d)]$.

$Cr(d)$	Set of feasible channels for demand d so to restore some amount of bitrate in the range $[0, b^d]$.

Failures and others:

F	Set of failure scenarios, index f.
$Kp(d)$	Set of backup routes for demand d.
$Kp^k(d)$	Subset of $Kp(d)$ link disjoint to working route k.
a_{ef}	Equal to 1 if fiber link e is available under failure scenario f.
a^k_f	Equal to 1 if route k is available under failure scenario f ($a^k_f = \Pi_{\{e:\, h^k_e = 1\}} a_{ef}$).
a^d_f	Equal to 1 if demand d is available under failure scenario f.
u_f	Probability of failure scenario f ($u_f \in [0,1]$).

The decision variables are:

ω^{dk}_c	Binary. Equal to 1 if demand d uses route k and channel c for working, 0 otherwise.
κ^{dk}_c	Binary. Equal to 1 if demand d uses route k and channel c for protection, 0 otherwise.
κ^{dk}_{cf}	Binary. Equal to 1 if demand d uses route k and channel c for restoration in failure scenario f, 0 otherwise.
τ^d	Positive real. Protected bitrate for demand d.
τ^d_f	Positive real. Protected bitrate for demand d in failure scenario f.
φ^d_f	Positive real. Restored bitrate for demand d in failure scenario f.
γ^d_f	Positive real. Recovered (protected + restored) bitrate for demand d in failure scenario f.
ν^d_r	Binary. Equal to 1 if demand d uses bitrate-reach pair r, 0 otherwise.
ν^d_{rf}	Binary. Equal to 1 if demand d uses bitrate-reach pair r in failure scenario f, 0 otherwise.
δ_{es}	Binary. Equal to 1 if slot s in fiber link e is used by any backup route, 0 otherwise.
$\sigma^{dd'}$	Binary. Equal to 1 if working routes of two demands d and d' share at least one fiber link, 0 otherwise.

The bitrate recovered in each failure scenario weighted by the probability of that scenario can be computed as

$$\Phi = \frac{1}{\sum_{f \in F} u_f} \cdot \sum_{f \in F} \sum_{d \in D} u_f \cdot \gamma_f^d \tag{5.1}$$

Then, the MILP formulation for the BRASERO problem, consisting in maximizing the total bitrate recovered Φ, is as follows:

$$\text{(BRASERO)} \ \ Max \ \ \Phi \tag{5.2}$$

subject to:

Common constraints:

$$\sum_{k \in Kw(d)} \sum_{c \in C(d)} \omega_c^{dk} = 1 \quad \forall d \in D \tag{5.3}$$

$$\gamma_f^d \le b^d \quad \forall d \in D, f \in F \tag{5.4}$$

$$\gamma_f^d \ge n_{recovery}^d \cdot \Delta_S \cdot m \quad \forall d \in D, f \in F \tag{5.5}$$

$$\gamma_f^d \le \tau_f^d + \varphi_f^d \quad \forall d \in D, f \in F \tag{5.6}$$

Constraints for path protection:

$$\sum_{d \in D} \sum_{k \in Kp(d)} \sum_{c \in Cp(d)} \kappa_c^{dk} \le (spp + dpp) \cdot |D| \tag{5.7}$$

$$\sum_{k' \in Kp^k(d)} \sum_{c \in Cp(d)} \kappa_c^{dk'} \le \sum_{c \in C(d)} \omega_c^{dk} \quad \forall d \in D, k \in Kw(d) \tag{5.8}$$

$$\tau^d \le \sum_{k \in Kp(d)} \sum_{c \in Cp(d)} n_c \cdot \Delta_S \cdot m \cdot \kappa_c^{dk} \quad \forall d \in D \tag{5.9}$$

$$\tau^d \le \sum_{r \in R} b(r) \cdot \nu_r^d \quad \forall d \in D \tag{5.10}$$

$$\tau^d \ge \Delta_S \cdot m \cdot (spp + dpp) \quad \forall d \in D \tag{5.11}$$

$$\sum_{k \in Kp(d)} \sum_{c \in Cp(d)} len(k) \cdot \kappa_c^{dk} \le \sum_{r \in R} len(r) \cdot \nu_r^d \quad \forall d \in D \tag{5.12}$$

$$\sum_{r \in R} \nu_r^d \le 1 \quad \forall d \in D \tag{5.13}$$

$$spp \cdot \sum_{d \in D} \sum_{k \in Kp(d)} \sum_{c \in Cp(d)} h_e^k \cdot l_{cs} \cdot \kappa_c^{dk} \le \delta_{es} \cdot |D| \quad \forall e \in E, s \in S \tag{5.14}$$

$$spp \cdot \sum_{k \in Kw(d)} \sum_{c \in C(d)} h_e^k \cdot \omega_c^{dk} + \sum_{k \in Kw(d')} \sum_{c \in C(d')} h_e^k \cdot \omega_c^{d'k} \le 1 + \sigma^{dd'} \qquad \forall d, d' \in D, d \ne d', e \in E \tag{5.15}$$

$$spp \cdot \sum_{k \in Kp(d)} \sum_{c \in Cp(d)} h_e^k \cdot l_{cs} \cdot \kappa_c^{dk} + \sum_{k \in Kp(d')} \sum_{c \in Cp(d')} h_e^k \cdot l_{cs} \cdot \kappa_c^{d'k} \le 2 - \sigma^{dd'} \qquad \forall d, d' \in D, d \ne d', e \in E, s \in S \tag{5.16}$$

$$dpp \cdot \sum_{d \in D} \sum_{k \in Kp(d)} \sum_{c \in Cp(d)} h_e^k \cdot l_{cs} \cdot \kappa_c^{dk} \le \delta_{es} \quad \forall e \in E, s \in S \tag{5.17}$$

$$\sum_{d \in D} \sum_{k \in Kw(d)} \sum_{c \in C(d)} h_e^k \cdot l_{cs} \cdot \omega_c^{dk} \le (1 - \delta_{es}) \quad \forall e \in E, s \in S \tag{5.18}$$

$$\tau_f^d \le a_f^k \cdot b^d + (1 - a_f^k) \cdot \tau^d + M \cdot (1 - \omega_c^{dk}) \qquad \forall d \in D, k \in Kw(d), c \in C(d), f \in F \tag{5.19}$$

Constraints for path restoration:

$$\sum_{d \in D} \sum_{k \in Kp(d)} \sum_{c \in Cr(d)} \sum_{f \in F} \kappa_{cf}^{dk} \le rest \cdot |D| \cdot |F| \tag{5.20}$$

$$\sum_{k \in Kp(d)} \sum_{c \in Cr(d)} \kappa_{cf}^{dk} \le \sum_{k \in Kw(d)} \sum_{c \in C(d)} (1 - a_f^k) \cdot \omega_c^{dk} \ \forall d \in D, f \in F \tag{5.21}$$

$$\varphi_f^d \le \sum_{k \in Kp(d)} \sum_{c \in Cr(d)} n_c \cdot \Delta_S \cdot m \cdot \kappa_{cf}^{dk} + \sum_{k \in Kw(d)} \sum_{c \in C(d)} a_f^k \cdot b^d \cdot \omega_c^{dk} \quad \forall d \in D, f \in F \tag{5.22}$$

$$\varphi_f^d \le \sum_{r \in R} b(r) \cdot \nu_{rf}^d \quad \forall d \in D, f \in F \tag{5.23}$$

$$\sum_{k \in Kp(d)} \sum_{c \in Cr(d)} len(k) \cdot \kappa_{cf}^{dk} \le \sum_{r \in R} len(r) \cdot \nu_{rf}^d \quad \forall d \in D, f \in F \tag{5.24}$$

$$\sum_{r \in R} \nu_{rf}^d \le 1 \quad \forall d \in D, f \in F \tag{5.25}$$

$$\sum_{d \in D} \sum_{k \in Kw(d)} \sum_{c \in C(d)} a_f^k \cdot h_e^k \cdot l_{cs} \cdot \omega_c^{dk} + \sum_{d \in D} \sum_{k \in Kp(d)} \sum_{c \in Cr(d)} h_e^k \cdot l_{cs} \cdot \kappa_{cf}^{dk} \le a_{ef} \cdot (1 - \delta_{es}) \qquad \forall e \in E, s \in S, f \in F \tag{5.26}$$

Constraint (5.3) ensures that the bitrate requested by each demand is served in the non-failure scenario, i.e. one working path (route and channel) is assigned to each demand. Constraints (5.4)-(5.5) limit the recovered bitrate of each demand in each failure scenario to some amount of bitrate in between the requested and the guaranteed ones. Constraint (5.6) computes the bitrate conveyed for each demand in each failure scenario, which can be simultaneously protected and restored when a failure impacts the working path for that demand.

Constraints (5.7) to (5.13) deal with the protection of demands, i.e. guaranteeing a minimum protected bandwidth per demand subject to path protection constraints. Constraint (5.7) makes sure that protection paths are allowed if and only if protection is permitted. Constraint (5.8) selects, at most, one backup route, link disjoint to the working one, for each demand. In addition, one channel is chosen. Constraints (5.9)-(5.11) compute the protected bitrate for each demand, which is limited by the size of the selected channel (5.9), and the maximum bitrate allowed by the bitrate-reach pair selected (5.10), ensuring that, at least, one slot is restored by protection (5.11). Constraint (15.12) guarantees that the length of the backup route chosen does not exceed the reach selected. Constraint (5.13) assures that one and only one bitrate-reach pair is selected for each demand.

Constraints (5.14)-(5.18) deal with the use of slots. When SPP is selected, constraint (5.14) allows slots to be shared by a number of backup paths, whereas constraints (5.15)-(5.16) manage slot sharing; constraint (5.15) checks whether the working routes of two demands share at least one fiber link and constraint (5.16) prevents the backup paths of two demands whose working routes share some fiber link from sharing any slot in common links. In contrast, when DPP is selected, constraint (5.17) makes sure that a slot is used by no more than one backup path. Constraint (5.18) guarantees that each slot is used at most by one working path when it is not used for protection.

Constraint (5.19) together with (5.3)-(5.6) computes the bitrate conveyed for each demand in each failure scenario, which equals τ^d when a failure impacts the working path for that demand and b^d on the contrary.

Constraints (5.20) to (5.26) deal with restoration. Constraint (5.20) controls whether restoration is allowed. Constraint (5.21) allows finding a restoration path for each demand and failure scenario if and only if the failure affects the working path of the demand. Constraints (5.22)-(5.23) compute the restored bitrate for each demand, which is limited when the failure impacts the demand by the size of the selected channel (5.22) and the maximum bitrate allowed by the bitrate-reach pair selected (5.23). Constraint (5.24) ensures that the length of each restoration route chosen do not exceed the reach selected. Constraint (5.25) guarantees that one and only one bitrate-reach pair is selected for each demand and failure scenario. Constraint (5.26) deals with the use of slots in case of failure; it guarantees that each slot is used at most by either one working path or one restoration path, provided that it is not already in use for protection and the fiber link is not involved in the current failure scenario.

5.2.3 Complexity analysis

The BRASERO problem is *NP-hard* since simpler network routing problems have been proved to be *NP-hard* (e.g., [Ra95]). Regarding the BRASERO problem size, the number of variables and constraints are depicted in Table 5-1 for each

Table 5-1 Size of the BRASERO problem

	Constraints	Variables
Common	$O(\|D\|\cdot\|F\|)$ $(4\cdot10^3)$	$O(\|D\|\cdot\|K\|\cdot\|C\|+\|D\|\cdot\|F\|)$ (10^7)
DPP	$O(\|D\|\cdot\|K\|\cdot\|C\|\cdot\|F\|+\|E\|\cdot\|S\|)$ $(2\cdot10^8)$	$O(\|D\|\cdot\|K\|\cdot\|C\|+\|D\|\cdot\|R\|+\|E\|\cdot\|S\|)$ (10^7)
SPP	$O(\|D\|\cdot\|K\|\cdot\|C\|\cdot\|F\|+\|D\|^2\cdot\|E\|\cdot\|S\|)$ $(3\cdot10^8)$	$O(\|D\|^2+\|D\|\cdot\|K\|\cdot\|C\|+\|D\|\cdot\|R\|+\|E\|\cdot\|S\|)$ (10^7)
Path Restoration	$O(\|D\|\cdot\|F\|+\|E\|\cdot\|S\|\cdot\|F\|)$ $(5\cdot10^4)$	$O(\|D\|\cdot\|K\|\cdot\|C\|\cdot\|F\|+\|D\|\cdot\|R\|\cdot\|F\|)+\|E\|\cdot\|S\|)$ $(2\cdot10^8)$

constraint group defined in the model above; the complete size of the problem is obtained adding the size of each group of constraints. Additionally, an estimation of the BRASERO problem size is calculated for the 10-node and 18-link DT network topology (Fig. A-8).

Although the MILP model can be solved for small instances, its exact solving becomes impractical for realistic backbone networks (under appreciable load) such as those described in Appendix A, even using commercial solvers such as CPLEX [CPLEX]. Thus, aiming to provide near-optimal solutions within reasonable computational effort, the next Section presents a BRKGA-based metaheuristic algorithm to solve the BRASERO problem which provides a much better trade-off between optimality and complexity.

5.3 Heuristic Algorithm

In this section, we provide a detailed description of the heuristic algorithm that was developed so as to efficiently solve the BRASERO problem. Although many metaheuristics have been lately proposed, the algorithm presented is based on the recently proposed BRKGA metaheuristic [Go11], which has proved to effectively solve optimization problems, in particular, network related problems such as routing in single layer and multilayer optical networks [Ru11].

As described in the Background chapter, the only problem-dependent parts to specify a BRKGA heuristic are the decoder and the chromosome internal structure. Since the order in which the demands are routed influences the goodness of the solution, one gene for each demand to specify the order in which the demands are routed is needed. Therefore, each individual is represented by an array of $n=|D|$ genes.

The algorithm in Table 5-2 specifies the decoder pseudo-code. Essentially, the decoder is divided into three phases. First, demand's order is initialized using the assigned gene of the input chromosome and demands are sorted afterwards (lines 1-4).

Second, demands are routed and a working path, including route and spectrum allocation, is found (lines 5-22). When restoration is used, a shortest path algorithm adapted to the flexgrid technology (see Chapter 6) is used to solve the routing and spectrum assignment problem (lines 5-9). In addition, spectrum allocation must guarantee both continuity and contiguity, and routes length must to be consistent with reach limitations used for the requested bitrate. For the spectrum allocation, we use channels as defined in [Ve12.2] and we assume that the QPSK modulation format is used. Then, the number of frequency slots in the set of channels for working lightpaths of each demand d is computed. When the recovery scheme is DPP or SPP a backup path (route and channel) is calculated along with the working path (lines 10-22). In particular, the Suurballe algorithm [Su84] is used for DPP (lines 11-15) whereas for SPP the CAFES algorithm, proposed in [Ou04] and adapted to the flexgrid technology, is used (lines 16-20); specifically, modifications in CAFES algorithm consist in considering frequency slots as the shared resources. It is worth noting that the spectrum assignment for the backup paths is fixed to 1 slot; once a working and backup path are computed for each demand, an improvement stage tries to increase the protected bitrate of every demand (lines 21-22).

Third, all failures scenarios are processed (lines 23-38). Initially, for every failure all the affected demands are grouped and sorted (lines 24-29). Next, each demand's backup path, if exist, is activated (lines 31-32). Afterwards, the shortest path algorithm is used to found a restoration path (lines 33-36), provided that the recovery scheme is restoration or MPR. Note that, once more the spectrum assignment is fixed to 1 slot. Finally, an improvement stage tries to increase the restored bitrate in each failure scenario (lines 37-38).

The decoder algorithm ensures the minimum protected bitrate allocating a number of frequency slots $n_{recovery}$ so that $n_{recovery} \cdot m \cdot \Delta_S \geq q^d \cdot b^d$. Once, the minimum bitrate is ensured, the improvement phase tries to increase the number of slots allocated for protection (lines 21-22) and restoration (lines 37-38). To this end, the algorithm in Table 5-3 is used to reallocate the used resources; it first tries to increase the assigned spectrum of each demand in steps of one slot from $n_{recovery}$ until reaching, at most, $[b^d/(m \cdot \Delta_S)]$-1. Next, the last slot of each demand (which capacity might not

Table 5-2 Decoder Algorithm

```
IN: N, E, D, scheme, mpr, Chromosome ch;
OUT: Φ
 1:  i ← 0
 2:  for each d∈ D do
 3:    d.order ← ch[i++]
 4:  sort (D, D.order, ascending)
 5:  if scheme = rest then
 6:    for each d∈ D do
 7:      {d.kw, d.cw}← shortestPath(d, d.b)
 8:      if not exist d.kw then
 9:        return NOT FEASIBLE
10:  else
11:    if scheme = dpp then
12:      for each d∈ D do
13:        {{d.kw, d.cw},{d.kp, d.cp}}←Suurballe(d, d.b, m·Δ_S)
14:        if not exist d.kw or d.kp then
15:          return NOT FEASIBLE
16:    else if scheme = spp then
17:      for each d∈ D do
18:        {{d.kw, d.cw},{d.kp, d.cp}}←CAFES(d, d.b, m·Δ_S)
19:        if not exist d.kw or d.kp then
20:          return NOT FEASIBLE
21:    reAllocateBackupPaths(N, E, D, "diffSlots", scheme)
22:    reAllocateBackupPaths(N, E, D, "diffBW", scheme)
23:  for each failure scenario f ∈ F do
24:    for each route k∈ K do
25:      compute a^k_f = availability of route k under failure f
26:    for each d∈ D do
27:      if d.a^kw_f = 0 then
28:        D[f] ← D[f] ∪ {d}
29:    sort(D[f], D.order, ascending)
30:    for each d ∈ D[f] do
31:      if scheme = dpp or scheme = spp then
32:        activeBackupPath(d)
33:      if scheme = rest or mpr then
34:        {d.kr,d.cr}←shortestPath(d, m·Δ_S)
35:      if not mpr and not exist d.kr then
36:        return NOT FEASIBLE
37:    reAllocateRestPaths(N, E, D[f], "diffSlots")
38:    reAllocateRestPaths(N, E, D[f], "diffBW")
39:  return Φ
```

be completely used) is established in the last round. The fitness value Φ, eq. (5.1), is eventually returned.

The performance of the proposed heuristic was compared against the optimal solution obtained solving the developed MILP model over small topologies. In all the tests performed, the optimal solution was found within running times of few minutes, in contrast to several hours needed to find the optimal solution with the model. Thus, we use the heuristic algorithm to solve the larger instances presented in the next section.

Table 5-3 Reallocate Backup Paths Algorithm

```
IN: N, E, D, criteria; scheme
OUT: none
 1: repeat
 2:   changeDone ← false
 3:   if criteria="diffSlots" then
 4:     sort(D, D.n - D.np, descending)
 5:   else
 6:     sort(D, D.b - D.np·m·Δs, descending)
 7:   for each d ∈ D' do
 8:     if (criteria = "diffSlots" and (d.n - d.np) ≤ 1) or
 9:     (criteria = "diffBW" and d.np·m·Δs ≥ d.b) then
10:       continue
11:     d' = d
12:     d.np ← d.np + 1
13:     if scheme = dpp then
14:       {{d.kw, d.cw},{d.kp, d.cp}} ← Suurballe(d, d.n, d.np)
15:     else /* scheme = spp */
16:       {{d.kw, d.cw},{d.kp, d.cp}} ← CAFES(d, d.n, d.np)
17:     if not exist d.kw or d.kp then
18:       d = d'
19:     else
20:       changeDone ← true
21: until changeDone = false
```

5.4 BRASERO Numerical Results

In this section, we first present the network scenarios that were considered in experiments. Next, we show the results of solving the BRASERO problem considering a set of realistic traffic instances.

5.4.1 Network Scenario

In order to conduct all the experiments, we consider the following optical network topologies: the 10-node and 16-link TEL (Fig. A-3), the 10-node and 16-link BT (Fig. A-6), and the 10-node and 18-link DT (Fig. A-8). In addition, two frequency slot widths are considered, namely 25 GHz and 12.5 GHz. In all the experiments the entire spectrum is set to 1.6 THz, i.e. 64 25GHz slots or 128 12.5GHz slots. Table 5-4 reports the number of slots that each demand requires under the different slot widths evaluated.

Table 5-4 Number of slots required for each bitrate

Slot width (Δ_S) (GHz)	10 Gb/s	40 Gb/s	100 Gb/s	400 Gb/s
25	1	1	2	8
12.5	1	2	4	16

With respect to the traffic we make use of three different TPS, namely, TP-2, TP-4 and TP-6 from Table A-1. Indeed, these TPs are a realistic representation of the expected evolution of bandwidth necessities for the years to come. It is worth noting, however, that the average amount of Tb/s offered to the network is equal for all TPs. Finally, we assume a q^d value so that $n^d_{recovery} = 1$ for all the demands.

5.4.2 Performance evaluation of protection schemes

Fig. 5-3, Fig. 5-4 and Fig. 5-5 plot the percentage of unrecovered bitrate against the total network load (Tb/s) for the considered TPs and slot widths, for the TEL, BT, and DT network topologies, respectively. Each point in the plots represents the average value from five independent instances, where the demand matrix of each instance consists in uniformly distributed origin-destination pairs which requested bitrate follows the above-defined TPs. Six graphs per network scenario are represented, one per each TP and frequency slot width pair. Each graph plots the obtained performance of every studied recovery schemes.

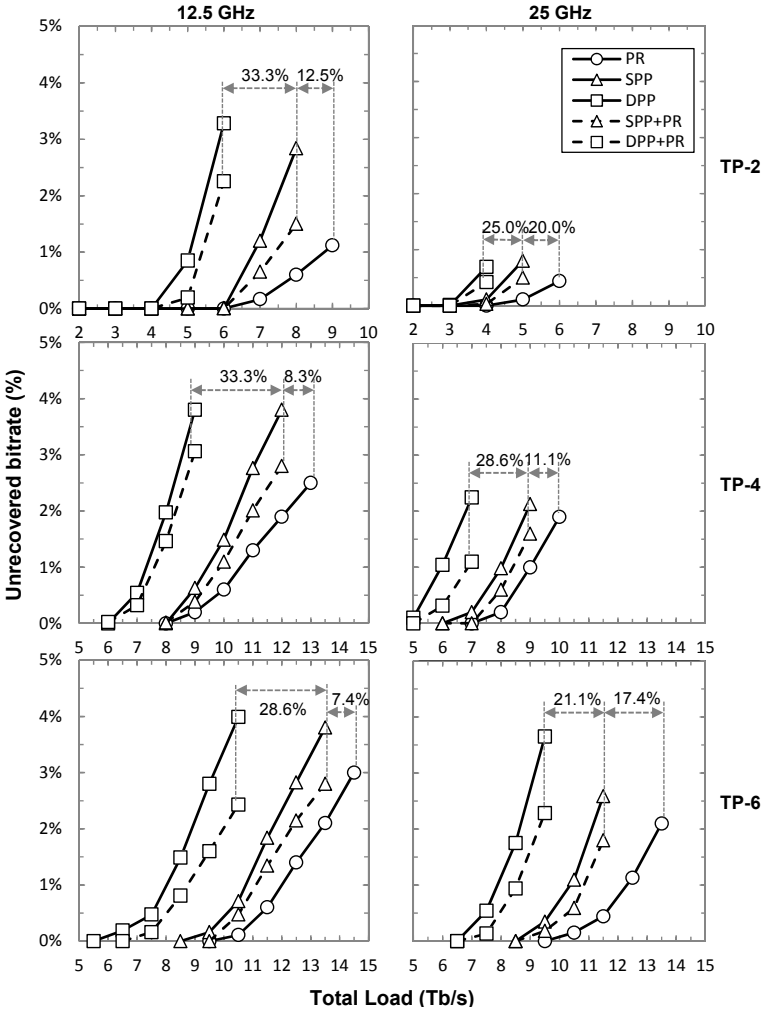

Fig. 5-3 Unrecovered bitrate vs. load (TEL).

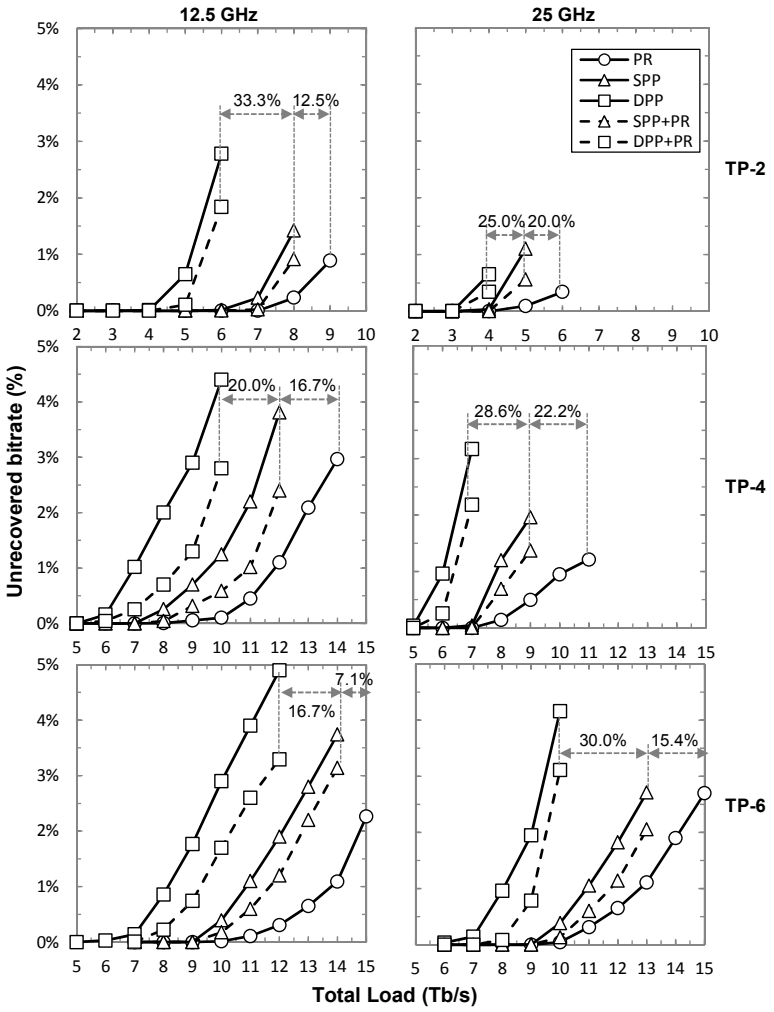

Fig. 5-4 Unrecovered bitrate vs. load (BT).

It is worth noting that when the network load increases, the amount of unrecovered also increases until the problem becomes infeasible as a result of:

i. one or more demands could not be provisioned, or,

ii. the minimum bitrate to be recovered could not be guaranteed for one or more demands in one or more failure scenarios.

Hence, feasible solutions serve all the traffic and guarantee the minimum bitrate in all the failure scenarios.

As expected, the restoration approach (PR) clearly supports more traffic load than the protection ones in all the network scenarios studied. In addition, SPP outperforms DPP; while the traffic increment of restoration with respect to DPP is meaningful, 24% on average, that ratio for SPP is only 14%. This is as a result of a better spare resources usage of SPP. Obviously MPR-based schemes behave the

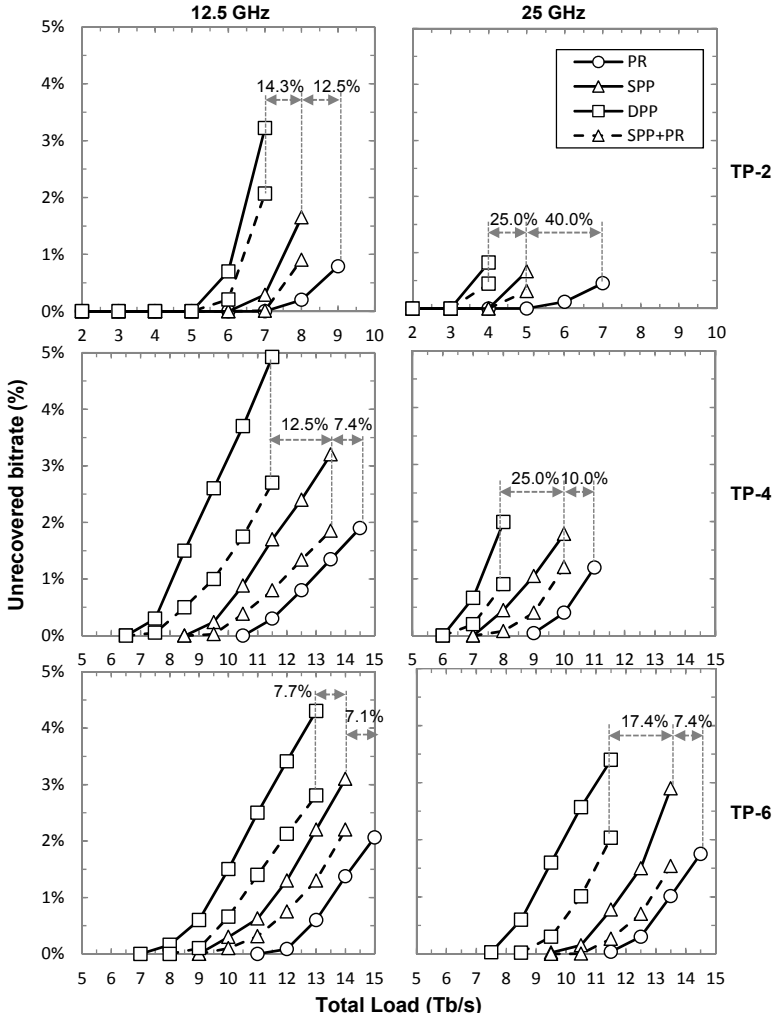

Fig. 5-5 Unrecovered bitrate vs. load (DT).

same as their pure protection counterpart schemes, since guaranteed bitrate is recovered also by protection (DPP or SPP).

Interestingly, the maximum amount of served traffic is higher when the 12.5GHz slot width is implemented when compared to that of the 25GHz. The rationale behind that is the higher spectral efficiency of the narrower slot width.

Comparing mixed schemes against pure protection schemes we observe, for the same traffic load, a clear gain in the amount of unrecovered bitrate when restoration is added in MPR-based schemes. Note that, due to the higher efficiency of restoration, more bitrate can be recovered. Table 5-5 summarizes the gains obtained by MPR-based with respect to pure protection schemes for the highest load supported, for each network topology and TP considered. Gains as high as 54% are observed, clearly demonstrating the goodness of the proposed scheme.

Table 5-5 Bitrate gain at highest load when MPR-based schemes are used

Slot Width (GHz)	TP	TEL		BT		DT	
		DPP + PR	SPP + PR	DPP + PR	SPP + PR	DPP + PR	SPP + PR
12.5	2	31.2%	47.1%	33.9%	35.7%	35.8%	45.1%
	4	19.4%	26.3%	36.4%	37.0%	45.2%	42.1%
	6	39.1%	26.3%	32.8%	16.1%	34.7%	29.0%
25	2	39.1%	37.5%	47.7%	49.1%	45.9%	53.3%
	4	51.0%	24.8%	31.2%	30.0%	54.7%	32.7%
	6	37.4%	30.5%	25.2%	24.2%	40.1%	46.9%

Let us now study effective recovery times when the different recovery schemes are applied. Aiming at comparing the recovery times, we concentrate on the times to recover the bitrate for each demand, irrespective of the amount of bitrate recovered. Assuming average protection (t_p) and restoration (t_r) times, the average recovery time ($t_{recovery}$) is t_p when the pure protection and t_r when the pure restoration schemes is applied. However, when our MPR scheme is applied two recovery time components need to be considered: first, the time to protect the minimum bitrate, and second, the time to restore the rest recovered bitrate. Equation (5.27) presents the way to compute $t_{recovery}$ for every recovery scheme. Note that τ^d_f and φ^d_f contains values greater than zero when protection and restoration, respectively is applied.

$$t_{recovery} = \frac{1}{|F|} \cdot \sum_{f \in F} \sum_{d \in D} \left(1 - a^d_f\right) \cdot \frac{\tau^d_f \cdot t_p + \varphi^d_f \cdot t_r}{\gamma^d_f} \tag{5.27}$$

Thus, average recovery time for the MPR scheme will be in the range [t_p, t_r). Since we assumed $n^d_{recovery} = 1$, those demands which bitrate can be conveyed into one single frequency slot will experience $t_{recovery} = t_p$, whereas those needing a large amount of frequency slots will experience $t_{recovery} \approx t_r$.

Table 5-6 shows the computed $t_{recovery}$ for the mixed schemes in milliseconds; we assumed t_p = 100 ms and t_r = 2 s. Looking at the values for TP-2, $t_{recovery}$ is virtually t_p as a result of the low average bitrate of the demands in this TP. In contrast, a clear trend towards time to restore values is observed for the recovery values for TP-6; the narrower the slot width, the closer the recovery value to t_r. Notwithstanding the tendency, the highest recovery time is 1.4 s, 30% better than using pure restoration. The effect of the ratio between demands bitrate and slot width can be clearly appreciated for TP-4. When the slot 25 GHz is used, the large majority of demands can be conveyed using one slot, and thus recovery times are just over t_p. In contrast, when the slot 12.5 GHz is used, demands need just over

Table 5-6 Mean recovery time for the mixed schemes (ms)

Slot Width (GHz)	TP	TEL		BT		DT	
		DPP + PR	SPP + PR	DPP + PR	SPP + PR	DPP + PR	SPP + PR
12.5	2	120	100	118	100	122	105
	4	1,055	1,078	1,058	1,083	1,052	1,071
	6	1,386	1,400	1,386	1,400	1,393	1,404
25	2	100	100	100	100	107	100
	4	147	168	127	168	155	168
	6	793	809	768	811	800	807

two slots on average, which translates into restoration times just above 1 s, since half of the bitrate is protected and the other half is restored. Note that these times are 50% lower than the obtained using pure restoration.

5.5 Conclusions

In this chapter a novel recovery scheme, named multi-path recovery (MPR), specifically designed for flexgrid optical networks was proposed. MPR can be applied to recover the requested bitrate of client connection demands in the event of single-link failures. It is important to highlight that partial recovery, i.e. recovery some proportion of the total requested bitrate, was assumed to be included in the SLA contracts between the network operator and the clients.

To compare the performance of MPR against existing single-path pure recovery schemes, such as DPP, SPP, and path restoration, the BRASERO problem was stated and modeled as a MILP problem.

The performance of all recovery schemes was analyzed through intensive numerical experiments. For the sake of a comprehensive study, three different network topologies, two frequency slot widths, and three traffic profiles were considered. The results clearly showed the benefits of the proposed MPR recovery scheme when compared to pure protection ones. In addition, recovery times were shorter when MPR was used compared to pure path restoration. Therefore, MPR provides a good trade-off between efficiency and recovery times.

In this and previous chapters, we studied different architectures for optical networks using off-line planning techniques. In the next chapters, we assume that the network is in operation and propose solutions taking into account that fact.

Chapter 6

Dynamic Provisioning

In this chapter we present two provisioning algorithms. Firstly, we investigate lightpath provisioning in flexgrid, where spectrum allocation must satisfy the contiguity and the continuity constraints, which results in the need to devise algorithms for solving the RSA problem. Targeting at dynamic traffic scenarios, we introduce a novel RSA algorithm based on the Dijkstra algorithm. The algorithm is exhaustively tested through simulation. Secondly, we focus on cross-layer provisioning problems and devise a novel dynamic impairment-aware RWA algorithm for translucent DWDM networks that minimizes regenerator usage. Experimental evaluation shows that significant improvements are attained in terms of the offered traffic load.

6.1 Routing and Spectrum Allocation

6.1.1 RSA algorithm

In this section, we propose an approach for the RSA problem facing routing and spectrum allocation problems separately. Without loss of generality, we include the spectrum availability in an adapted Dijkstra's shortest path routing algorithm. Note, however, that our approach can be adapted for any other shortest path algorithm, such as the breath-first search shortest path algorithm [Bh09]. Regarding the spectrum allocation, any heuristics similar to those used for wavelength assignment can be applied.

Some notation needs to be introduced first. Let graph G (N, E) represent the topology of a flexgrid optical network, where N is the set of optical nodes and E the set of optical links connecting two nodes, and let S be the set of frequency slots

available in each link $e \epsilon E$. Then, let $C(d)$ be the set of frequency channels that fits the required spectrum width for a given connection request d and $S(c)$ be the set of contiguous slots in channel c. Finally, let η_{es} be equal to 1 if slot s in optical link e is free, 0 otherwise.

In the Dijkstra's algorithm, each node i is labeled with the aggregated metric $m(i)$ from the source node o and with its predecessor $pre(i)$. Consequently, the route o-i, defined by the subset of links $E(o,i) \subseteq E$, can be computed repetitively visiting the predecessor node starting from i, until source node o is reached. The label also contains $\eta_s(i)$, the aggregated state of frequency slot s ($\eta_s(i) = \prod_{e \epsilon E(o,i)} \eta_{es}$) for each $s \epsilon S$. The downstream node j of node i updates the label only if at least one channel is available as described in equation (6.1), i.e. only if $\pi(i,j) = 1$.

$$\pi(e=(i,j)) = \begin{cases} 1 & \exists\, c \in C(d) : \eta_s(i) \cdot \eta_{es} = 1 \; \forall s \in S(c) \\ 0 & \text{otherwise} \end{cases} \tag{6.1}$$

Since the length in hops and the distance of optical connections using the 16-QAM modulation format is limited, the routing algorithm looks first for a solution considering 16-QAM, but as soon as one of these constraints is not satisfied, the QPSK modulation format is applied.

Additionally, we account for guard bands including them in the requested spectrum $S(d)^* = S(d)+GB$, where GB is the number of slots required between two spectrum contiguous allocations. The guard band is assumed to occupy the right-hand side of the requested spectrum only; such a representation ensures that the guard band will separate any two spectrum allocations. Concurrently, in order to allow allocating GB extreme slots of S, we add GB slack slots to S, resulting in the augmented set of slots S^*.

The benefits of the above RSA algorithm are:

i. Only one graph G (N, E) containing the availability of every frequency slot needs to be stored and maintained.

ii. The routing algorithm runs once for each connection request.

iii. The complexity of the proposed extension for spectrum availability to the routing algorithm is negligible.

iv. The spectrum assignment is performed after the shortest route is found, adding flexibility to use any heuristic.

After implementing the above described RSA algorithm, some simulations were run to test its performance using different widths of the frequency slots, e.g. 50, 25, 12.5, and 6.25 GHz. We used the first fit heuristic for spectrum allocation, where the spectrum is explored sequentially and the first available channel is allocated.

6.1.2 Performance vs. frequency slots width

This section evaluates the performance, in terms of GoS, of implementing frequency slots of the considered spectrum widths (50 GHz, 25 GHz, 12.5 GHz, and 6.25 GHz). To this end, the RSA algorithm described above was developed and integrated into an ad-hoc event driven simulator based on OMNeT++ [OMNet]. A dynamic network environment is simulated in realistic network scenarios, namely, three national optical network topologies shown in the Appendix A are considered: the 21-node and 35-link TEL topology, the 20-node and 32-link BT topology, and the 21-node and 31-link DT topology. In all the experiments, the size of the entire spectrum is set to 800 GHz and guard bands are neglected.

Since the studies performed in this section might be influenced by the mix of bit rates demanded by connection requests, TP-1 to TP-6, shown in Appendix A, are used. When TP-1 is selected, 400 Gb/s connections are not considered. Profiles TP-2 to TP-6 successively increase the requested on-average bit rate. This is performed at the expense of decreasing the amount of 10 Gb/s traffic while keeping constant the amount requested for each of the rest bit rates, and so, the amount of 400 Gb/s traffic increases. The weighted bit rate columns detail the contribution of each bit rate to the on-average bit rate of each profile.

Incoming connection requests arrive following a Poisson process and are sequentially served without prior knowledge of future incoming connection requests. The holding time of connections is exponentially distributed with the mean value equal to 2 hours. Source/destination pairs are randomly chosen with equal probability (uniform distribution) among all network nodes. Different values of offered network load are considered by changing the arrival rate while keeping the mean holding time constant. The bit rate demanded by each connection request is 10, 40, 100 or 400 Gb/s and it is chosen following one of the traffic profiles previously described.

We conducted a number of simulations using the previously described configuration and traffic profiles. To integrate blocking of different requested bit rates into a single figure of merit we use a bandwidth-weighted blocking probability (Pb_{bw}) that accounts for the total amount of blocked bit rate with respect to the total requested. For all the bandwidth-weighted blocking probability results, the simulation stopped when a confidence interval of at least 5% and a confidence level of 95% were achieved.

Fig. 6-1 shows the influence of the slot width on the performance, in terms of bandwidth-weighted blocking probability, of each considered network topology as a function of the network load expressed in Erlangs using traffic profile TP-1. As illustrated, as soon as the frequency slot width is reduced, the amount of traffic served for a given Pb_{bw} notably increases. This is a consequence of the fact that more efficient spectrum utilization is achieved by reducing the slot width for this

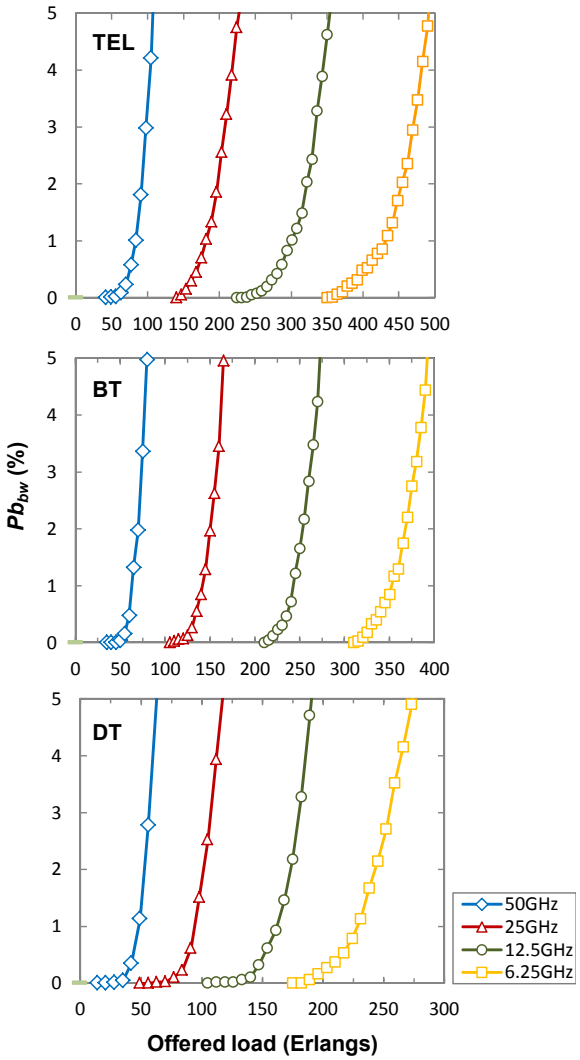

Fig. 6-1 Pb_{bw} vs. offered load for different slot widths using TP-1.

traffic profile. Note that most of the connections request 10Gb/s, which can be served using a single slot even in the narrowest slot width scenario.

Graphs in Fig. 6-2 and Fig. 6-3 study the influence of the traffic profile on the performance of the TEL network. Four traffic profiles are used (TP-2 to TP-5), each increasing the amount of 400 Gb/s traffic at the expense of that of 10 Gb/s thus increasing the on-average bit rate requested. Plots in Fig. 6-2 show the bandwidth-weighted blocking probability as a function of the offered load to the network. Pb_{bw} values remain constant for each of these traffic profiles when the wider slot widths (50 GHz and 25 GHz) are used. In contrast, they clearly decrease when the narrower slot widths (12.5 GHz and 6.25 GHz) are used. The opposite can be observed in Fig. 6-3, where the offered load to the network is expressed in terms of total bit rate offered to the network. In general, an increased amount of bit rate is served in any slot width option when the traffic increases from TP-2 to TP-5.

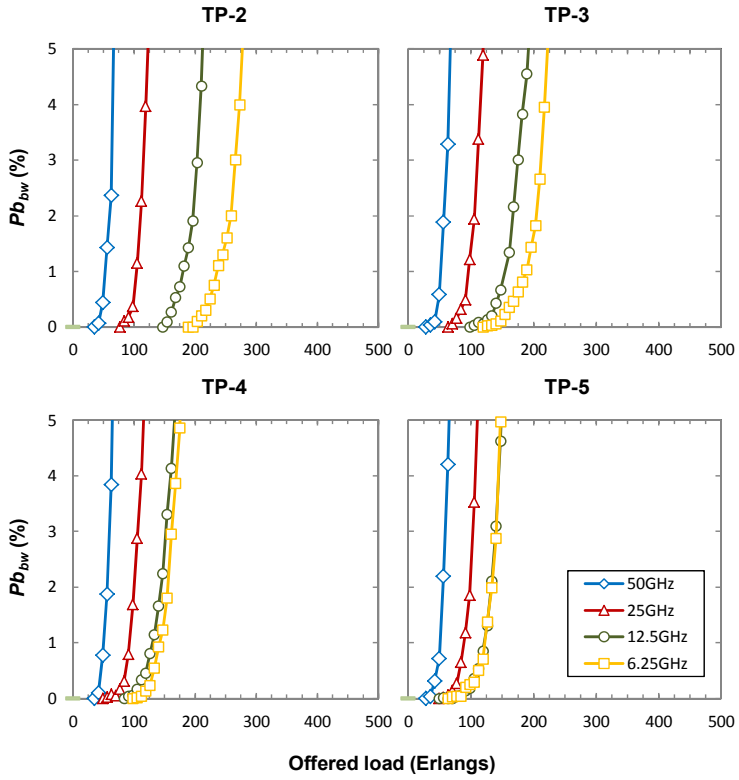

Fig. 6-2 Influence of the TP on Pb_{bw} vs. offered load in Erlangs.

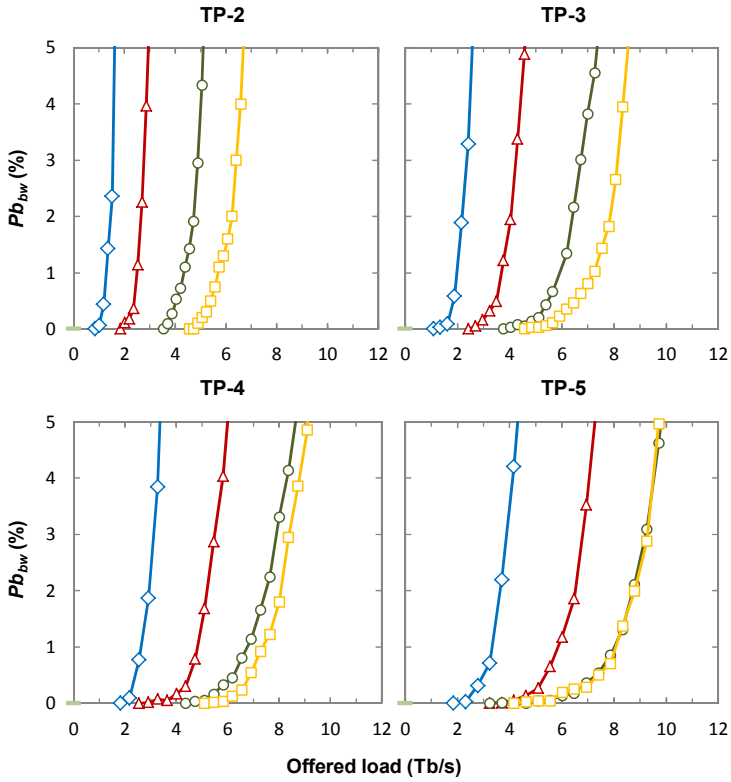

Fig. 6-3 Influence of the TP on the Pb_{bw} vs. offered bit rate.

The reasons for that are: on the one hand, the effect of the more efficient spectrum utilization obtained for TP-1 is gradually reduced as the on-average requested bit rate is increased; on the other hand, spectrum fragmentation increases as a consequence of requesting connections for a larger amount of frequency slots (up to 32 slots). In fact, as shown in Fig. 6-3, in the case of requesting TP-5, the advantage of using the narrowest slot width (6.25 GHz) observed under other traffic profiles is canceled and the same results as using 12.5 GHz are obtained.

Finally, from Fig. 6-4 we can see that under TP-6 the performance when using 12.5 GHz or 6.25 GHz slot widths is quite close to that of the 25 GHz. This is as a consequence of the larger amount of 400 Gb/s connection requests considered in that traffic profile.

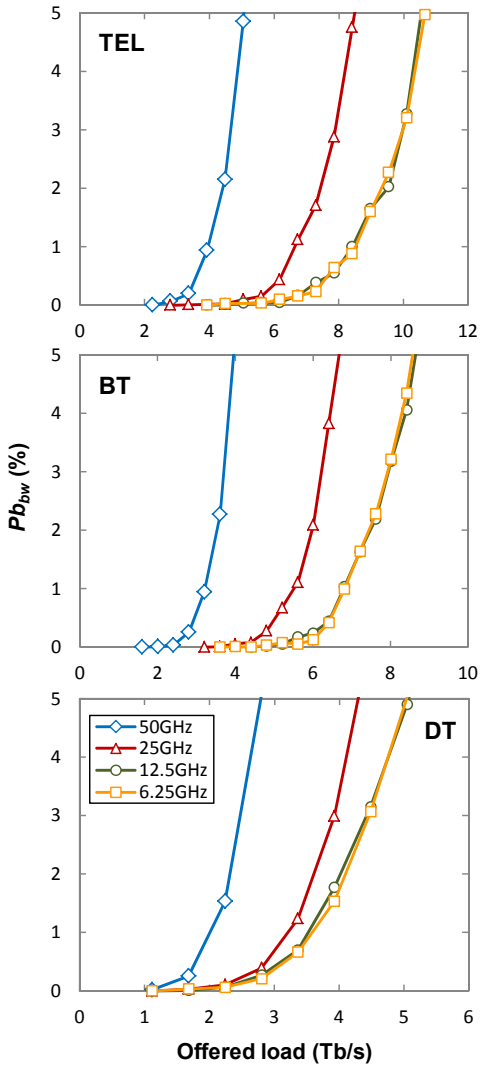

Fig. 6-4 Pb_{bw} vs. offered load for different slot widths using TP-6.

6.2 Translucent IA-RWA proposed algorithm

In this section, we consider dynamic translucent DWDM networks where a number of regenerators are available and study the RWA problem taking into account physical impairments. In particular, we assume a distributed regeneration approach, wherein hybrid nodes constitute the translucent WSON [Ya05.1]. Each hybrid node is made up of an all-optical switching core and a pool of 3R regenerators at some optical ports. Consequently, a lightpath traversing a hybrid node can be optically switched to the outgoing port or electronically regenerated before going to the output port.

Before describing the IA-RWA problem, we need to introduce some notation. We are given a graph describing the network topology and the state of the resources (wavelength channels at every link and regenerators availability at every node). Similarly as in the previous section, we represent that network topology by the graph $G(N, E)$. In addition, let W be the set of available wavelengths in each link $e \in E$. Let $N_R \subseteq N$ be the subset of nodes with regeneration capability, and conversely $N_T \subseteq N$ the subset of nodes without regeneration capability. Thus $N=N_R \cup N_T$. Additionally, we are given a pair of source and destination nodes $\{s, t\}$ for the connection being requested.

In our model, the OSNR represents the QoT parameter which allows evaluating the feasibility of the transparent segments. Specifically, the computed *Total_OSNR* is compared to a threshold OSNR level (*OSNR_thr*) to determine whether the physical impairment constraint is fulfilled at the receiver node:

$$Total\_OSNR_N \geq OSNR\_thr \tag{6.2}$$

If equation (6.2) is not satisfied, an available 3R regenerator is allocated at an intermediate node to improve the signal quality.

6.2.1 Proposed algorithm

The IA-RWA problem consists in finding a feasible route and wavelength assignment for the requested connection, so as to minimize the number of regenerators needed to guarantee the QoT for that connection. As a secondary objective, the length of the route, in terms of number of hops, should be minimized.

To fulfill the regenerators minimization objective, we build an auxiliary directed graph (digraph) $\mathcal{D}_{st}(\mathcal{N}_{st}, \mathcal{A})$, where $\mathcal{N}_{st}=N_R \cup \{s,t\}$ and $\mathcal{A}$ is the set of directed arcs. Each arc $a=(u,v) \in \mathcal{A}$ connects two nodes $u,v \in N_{st}$, where $u \in N_R \cup \{s\}$ and $v \in N_R \cup \{t\}$. An arc $a=(u,v)$ exists only if a feasible lightpath can be found between u and v in G, where the route has an acceptable OSNR level and the WCC is satisfied.

For illustrative purposes, Fig. 6-5 shows an example of auxiliary digraph construction. There, digraph $\mathcal{D}_{st}$ is built upon the reception of a connection request

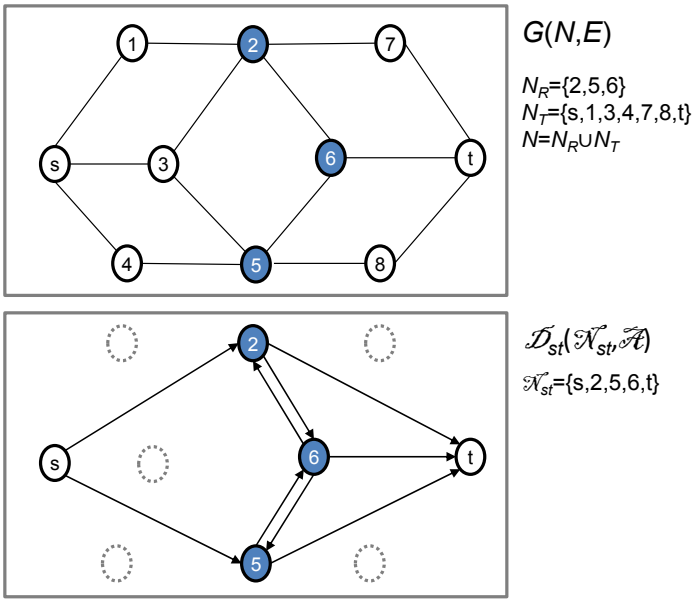

Fig. 6-5 Original (top) and auxiliary (bottom) graphs.

between nodes s and t. To this end, the set N_R with those nodes with regeneration capabilities currently available (colored circles) and nodes s and t belong to $\mathcal{N}_{st}$. Directed arcs are created connecting s and nodes in N_R to t and other nodes in N_R, provided that a feasible lightpath exists in G. In such scenario, routes {s,2,t} and {s,5,t} minimize the number of regenerators used. Note that other routes, such as {s,2,6,t}, being feasible, need more regenerators than the previous ones.

To compute those routes, we label each node and link in N with the inverse of its linear OSNR level. Then, the RWA problem can be solved by using the Dijkstra's algorithm to find the best route between two nodes in terms of OSNR in graph G, and the first-fit heuristic for assigning a wavelength.

Finally, to solve the IA-RWA problem we use the proposed Translucent IA-RWA (TrIA-RWA) algorithm described in Table 6-1. We first compute a set of k-shortest paths between s and t in the auxiliary graph $\mathcal{D}_{st}$, so that the number of hops ($|\mathcal{A}(k)|$) equals that of the computed shortest path. This ensures that the number of regenerators is minimized (first objective). Next, each k-route is transformed to a translucent optical connection in graph G, expanding each transparent optical segment and using regenerators to stitch the segments. The translucent connection minimizing the total number of hops, $|E(x)|$, (second objective) is eventually returned.

A major concern regarding the above algorithm is scalability. Note that an auxiliary digraph $\mathcal{D}_{st}$ needs to be built to solve the IA-RWA problem for each single connection request. To solve that issue, a set of auxiliary digraphs $\mathcal{D}_s(N, \mathcal{A})$ are computed off-line and stored to be used upon the reception of connection requests.

Table 6-1 Proposed TrIA-RWA algorithm

```
INPUT s, t
OUTPUT lightpath
 1: SP ← Shortest Path (in terms of hops) in 𝒟_st
 2: if SP not found then
 3:    return BLOCK_REQUEST
 4: K← {k-shortest paths (hops) in 𝒟_st: |𝒜(k)|=|𝒜(SP)|}
 5: Q ← ∅
 6: for each k∈K do
 7:    q ← ∅
 8:    for each a=(u,v)∈𝒜(k) do
 9:       q ← q ∪ {expand a into a lightpath in G}
10:       if v≠t then
11:          use one regenerator in v
12:    Q ← Q ∪ {q}
13: return x∈Q: ∀q∈Q |E(x)|≤|E(q)|
```

Note that $|N|$ digraphs are built, one $\mathcal{D}_s$ for each node s in N. In addition, although each digraph $\mathcal{D}_s$ contains all the nodes, being s the source one, the rules to construct each of the directed arcs are the same as above; only s and nodes in N_R can be source of arcs. At this stage, N_R contains the set of nodes where regeneration capabilities have been installed.

When a connection request (s, t) arrives, the algorithm gets the pre-computed digraph $\mathcal{D}_s$ and updates regeneration availability removing those arcs (u, v) in $\mathcal{A}$ where $u \neq s$ and u has not regenerators currently available. Next, algorithm in Table is executed taking into account that solving the RWA problem for transparent segments may result into unfeasible routes (due to accumulated impairments) as a consequence of the current state of the resources.

The next section reports the conducted experimental performance evaluation of the proposed TrIA-RWA algorithm.

6.2.2 Experimental results

This work was done in collaboration with the CTTC, and the evaluation of the TrIA-RWA algorithm was carried out in the [Mu11]. Emulated optical nodes are controlled by a GMPLS control plane implemented on Linux-based routers with Intel Xeon 3 GHz processors. The connection requests are dynamically served by a dedicated centralized stateless PCE.

The TrIA-RWA algorithm was implemented in C++ and deployed as a callable algorithm in the PCE. Aiming at comparing the performance of the proposed TrIA-RWA algorithm, the IA-RWA algorithm proposed in [Ma10] was also deployed in the PCE as a benchmark.

To experimentally evaluate the performance of both algorithms the 22-node and 34-link European Optical Network (EON) topology depicted in Fig. A-10 was

deployed, where each optical link supports 32 wavelength channels and each node was equipped with 3 regenerators. Fig. A-10 details the OSNR levels for each optical link which is computed according to the fiber distance and the placed optical amplifier segments. In addition, nodes contribute with a fixed OSNR level set to 41dB.

Uniformly distributed connection requests arrive to the network following a Poisson process whereas connection holding time follows a negative exponential distribution. Different offered traffic loads to the network were used by fixing the mean inter-arrival time of the requests to 4s, so to ensure TED updating convergence, and changing the mean holding time. We assume an OSNR threshold equal to 19dB.

Graphs in Fig. 6-6, Fig. 6-7 and Fig. 6-8 evaluate the performance of the proposed algorithm, where each data point was obtained after requesting 10,000 connections.

Fig. 6-6 plots the blocking probability as a function of the offered traffic load, where traffic load has been normalized to the load where the TrIA-RWA algorithm unleashes a blocking probability equal to 10% (250 Erlangs). We can observe that TrIA-RWA clearly outperforms IA-RWA in the whole range of traffic loads. In fact, comparing the offered traffic load at 1% of blocking probability a gain of 344% is attained. This is due to the fact that TrIA-RWA uses regenerators more efficiently which in turn leads to increase the accepted traffic when compared to IA-RWA.

Fig. 6-7 shows the average length (in hops) of the routes for the established connections. We observe that using the IA-RWA shorter routes are achieved compared to the TrIA-RWA. The rationale behind that is that, when IA-RWA is used, connections with long routes (which needs regenerators) are frequently blocked resulting in higher blocking probability. In the case of TrIA-RWA, the length of the routes is constant until the load increases and blocking probability appears (load>0.7), where the mean length decreases as a consequence of that connections with longest routes are blocked due to the lack of free regenerators.

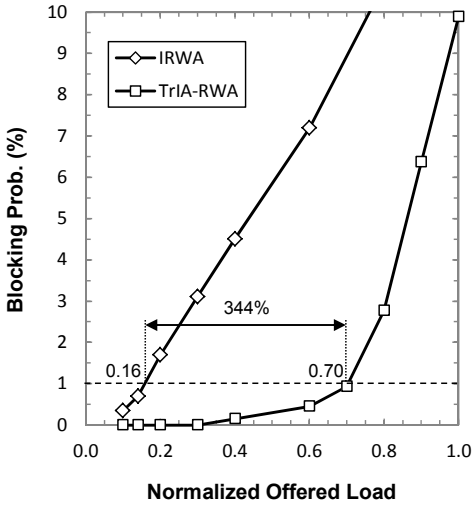

Fig. 6-6 Blocking probability against normalized load.

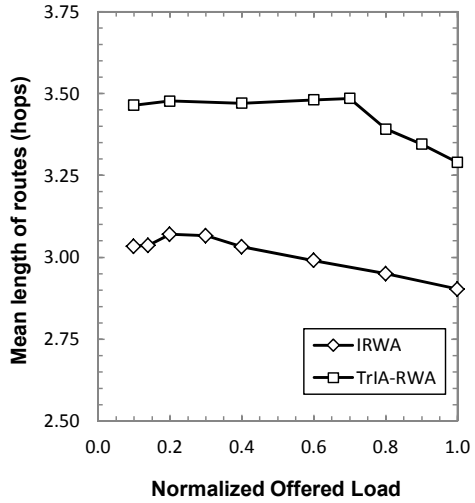

Fig. 6-7 Route length against normalized load.

Finally, Fig. 6-8 examines the path computation and set-up times. As shown, TrIA-RWA takes slightly longer computation times, in the order of 6.5%, than that of IA-RWA. Regarding set-up times, which includes path computation and signaling, TrIA-RWA needs 7% more time than IA-RWA. The difference is as a result of the longer routes to be established when using the former.

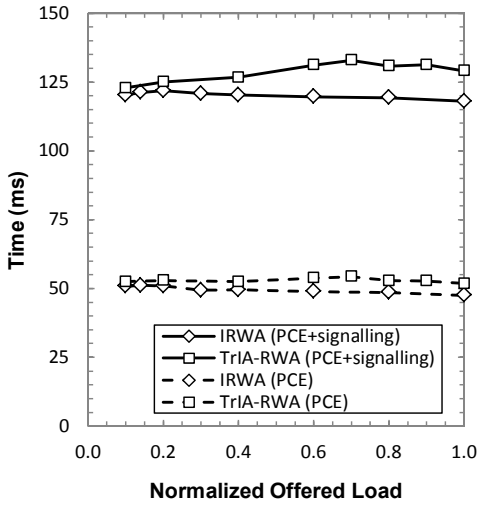

Fig. 6-8 Route computation and set-up times against normalized load.

6.3 Conclusions

Two provisioning algorithms were presented in this chapter. Firstly, a RSA algorithm to be used in dynamic flexgrid network scenarios was proposed. Its design included adaptive considerations involving the optical signal to be conveyed by the optical connection, such as the use of modulation formats, distance constraints, cascade filters, and guard bands.

The RSA algorithm was integrated into an event driven simulator, where a dynamic network environment was simulated for three networks under study. The optimal frequency slot width was analyzed as a function of the traffic profile to be served by each of the optical networks under consideration. Interestingly, the narrowest slot width (6.25GHz) provided the highest performance when the traffic profile requested to the network included high amount of low bit rate connection requests. However, when the relative weight of low bit rate (10Gb/s) connection requests decreased with respect to the weight of high bit rate (>100Gb/s) connection requests, the effectiveness of the narrowest slot width was canceled, providing then the same performance than that of 12.5GHz. Spectrum fragmentation as a consequence of requesting connections for a larger amount of frequency slots together with connections requesting lower amounts was identified as the main reason of that performance behavior.

Secondly, an impairment aware RWA in translucent DWDM networks, named TrIA-RWA, was presented. The algorithm made use of auxiliary digraphs to minimize regeneration usage and path length in terms of number of hops. Results showed that the TrIA-RWA algorithm allows increasing the traffic load that can be offered to the network; compared with other existing algorithms, a gain of more

than 340% in terms of offered load can be obtained. In addition, path computation and set-up times were in line with that of existing algorithms.

Once provisioning algorithms for dynamic network scenarios have been presented, the next chapter introduces link failure events. To cope with this new challenge, we devise algorithms that face dynamic restoration scenarios.

Chapter 7

Dynamic Bulk Path Restoration

In this chapter we focus on MLN, which combining packet and optical switching leads to jointly leverage intrinsic per-layer benefits such as statistical multiplexing and huge transport capacity. Efficient network resource utilization is attained by grooming together packet LSPs into a single optical LSP. In this context, an optical link failure may cause the disruption of multiple groomed packets LSPs. Thereby, efficient recovery schemes, such as multilayer restoration, are required.

When multilayer networks are operated using a control plane provided with a centralized PCE, that element is in charge of computing the route of connection requests. In the event of a failure, the PCE sequentially computes backup paths for the set of failed packet LSPs. Since the TED in the PCE is not updated until an LSP is actually set up, it is very likely that the PCE assigns the same network resources to different backup paths. This does increase resource contention and not fully exploits the potential grooming opportunities among the backup LSPs; consequently, the restorability metric performs poorly. To improve this, a designed PCE-based GCO architecture is implemented favoring grooming and lowering resource contention. We address this problem in two different network technology scenarios: the MPLS over DWDM and MPLS over flexgrid.

In the MPLS-over-WSON scenario we face the Bulk pAth REstoration in Multi-layer Optical networks (BAREMO) problem, which is formally modeled and stated using a MILP formulation; a heuristic algorithm solving the BAREMO problem is devised. The performance evaluation is comprehensively conducted by comparing the devised heuristic against the sequential restoration for several traffic loads and failure rates, in a simulation environment. The results show that the bulk restoration improves remarkably restorability compared to the sequential one at the expenses, however, of increasing the restoration time.

In the MPLS-over-flexgrid networks scenario we take advantage of MF-TPs, which provide additional flexibility allowing reconfiguration of optical connections to be performed. We introduce the DYNamic restorAtion in Multi-layer IP/MPLS-over-flexgrid Optical networks (DYNAMO) problem. The DYNAMO problem is modeled by using mathematical programming. However, as a consequence of its complexity and the stringent times within which the problem has to be solved, a GRASP-based heuristic is developed. Exhaustive simulation results performed on two national core topologies show that a PCE with a GCO module solving DYNAMO highly improves restorability and reduces remarkably the number and capacity of MF-TPs, at the expense, as in the previous scenario, of some increment in restoration times.

7.1 Dynamic Bulk Restoration

For illustrative purposes, Fig. 7-1(a) shows a simple physical network consisting of five OXCs and four MPLS routers. OXCs are connected by bidirectional fiber links. Let us assume that each MPLS router is connected to the collocated OXC. Finally, two MPLS client flows are already being served. We assume that the bitrate of each flow is 1 Gb/s. Two lightpaths were established in the physical topology to support an equal number of virtual links in the virtual topology. The route of each MPLS flow over the virtual topology is given in the adjacent table.

At this stage, let us assume that a new MPLS request #3 between R1 and R3 need to be served. After requesting a route to the PCE it computes R1-R2-R3, where the existing virtual link R1-R2 is reused and a new virtual link R2-R3 must be created, which triggers the new lightpath R2-R3 to be established. Later, another MPLS request #4 between R2 and R3 arrives and it is served through the route R2-R3, using capacity available in virtual link R2-R3. Fig. 7-1(b) describes the configuration of both the physical and the virtual topologies once all four MPLS flows, also named Packet Switching Capable (PSC) LSPs, have been routed.

Next, a failure in fiber link X1-X2 has triggered each of the affected flows (flows #1 and #3) to request a restoration route to the centralized PCE.

In Fig. 7-2(a) the restoration route has been computed sequentially by the PCE and signaled afterwards. Since the TED in the PCE is only updated when the resources have been effectively allocated in the network, the signaling of the restoration routes of both MPLS flows have triggered two parallel lightpaths to be set-up so as to create the virtual links needed to route the MPLS flows. In the example, both lightpaths could be created because enough resources, i.e. frequency slots in the links and ports in the MPLS routers, were available. Frequently, nonetheless, that is not the case and *resource contention* may arise. In that regard, note that both restoration routes reused the virtual link R1-R4; again resource contention could

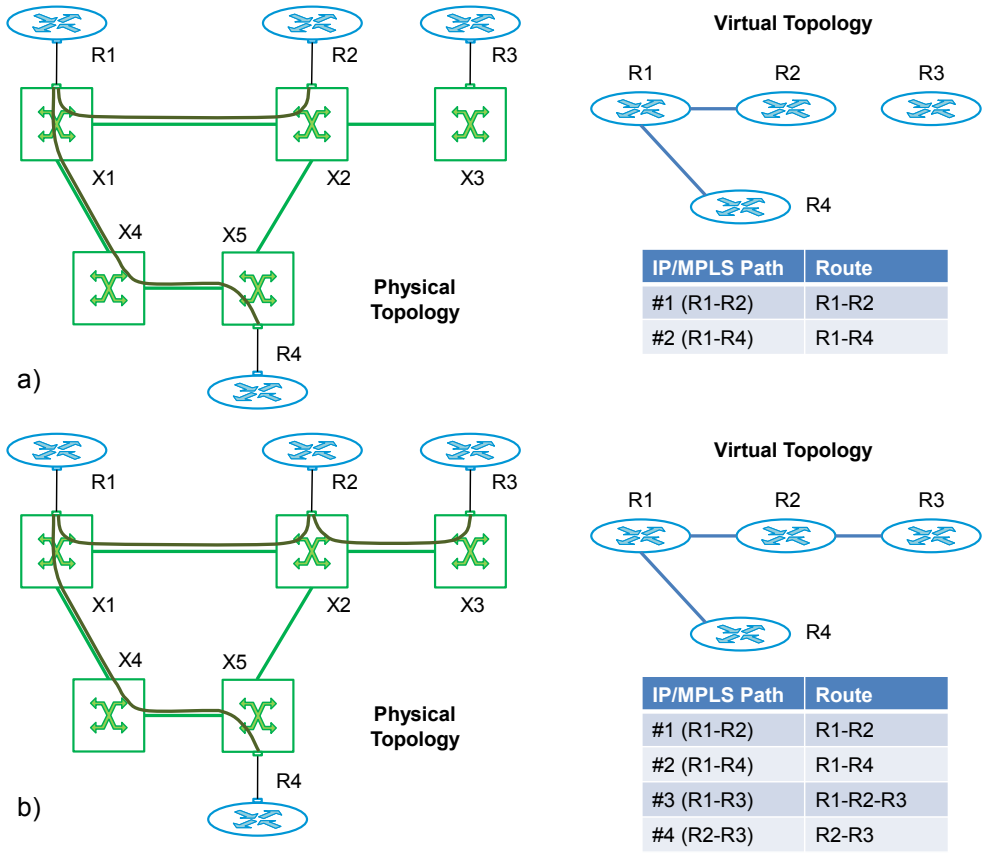

Fig. 7-1 Example of multilayer network set-up.

arise as a consequence of not enough capacity being available for both MPLS flows in that virtual link.

In Fig. 7-2(b) the PCE has grouped all restoration requests and performed bulk route computation. In the example, the restoration route of both MPLS flows has been computed. The bulk routing algorithm decides to create virtual link R4-R2 using that for both restoration routes, thus reducing the amount of resources used compared to the sequential approach. However, for the bulk restoration to work restoration routes need to be sequenced: one of the routes must be signaled first, so as to trigger actual virtual link creation; after waiting enough time, virtual link R4-R2 is effectively created and the second route reusing it can be signaled.

As anticipated above, for the bulk restoration to work properly, computed routes must be sequenced for signaling so as to allow that new virtual links are firstly created, and their lightpaths established, by one MPLS demand; after that their capacity is available to be reused by other MPLS flows. In our example, flows #1 and #3 need virtual link R4-R2 to be created. Then, one of the demands is rerouted and the other one must be delayed enough to allow the virtual link R4-R2 to be effectively created. This fact introduces a set of *dependences* among the demands that must be considered so as to minimize recovery times.

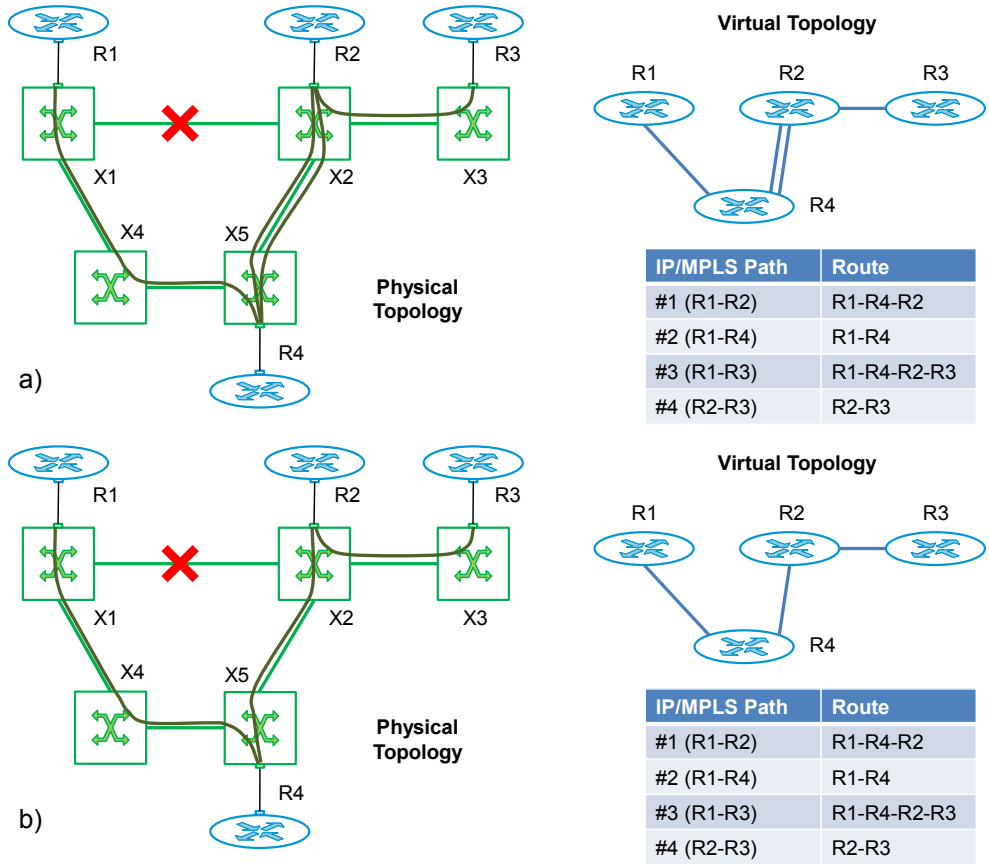

Fig. 7-2 Sequential (a) and bulk (b) restoration after a failure in link X1-X2.

In the following section, we present two different scenarios that reveal the benefits of applying bulk restoration: the MPLS-over-WSON and the MPLS-over-flexgrid multilayer restoration scenarios.

7.2 MPLS-over-WSON: The BAREMO Problem

7.2.1 Problem Statement

The bulk path restoration in multi-layer optical networks (BAREMO) problem can be formally stated as follows:

Given:

- an optical network topology represented by a graph $G_o(N, L)$, being N the set of optical cross-connects (OXCs) and L the set of unidirectional fiber links connecting two OXCs (excluding failed ones),
- a set W of wavelength channels for each link in L. The channel status in each link (i.e., free or used) is also given,

- the virtual network represented by a graph $G_v(V, E)$, being V the set of MPLS routers, and E the set of FA TE links defining the connectivity among the MPLS routers. For each existing FA TE link e, the unused capacity is also given,
- a set D of PSC LSPs to be recovered. Each demand d is defined by the tuple $\{s_d, t_d, b_d\}$, where s_d and t_d are the source and destination MPLS nodes and b_d its bitrate.

Output:

- the routing in the virtual topology of every demand which is recovered,
- the route and wavelength assignment of new optical LSPs (lightpaths) created,
- the order in which the demands must be set up.

Objective: maximize the total amount of bitrate recovered whilst minimizing the amount of optical ports used and the recovery time.

We have modeled the BAREMO problem by means of a MILP formulation. Next subsection presents the proposed formulation.

7.2.2 Mathematical Model

The MILP model for the BAREMO problem performs routing in both the MPLS and the WSON layers combining node-link and arc-path formulations respectively [Pi04]. A set of FA TE links is pre-computed beforehand; each FA TE link connects two MPLS routers provided that, at least, one feasible lightpath in the optical network can be found. A set of lightpaths for each FA TE link, including its route on the optical topology, is pre-computed.

The following sets and parameters have been defined:

Optical Topology:

N	Set of OXC nodes, index n.
L	Set of unidirectional fiber links, index l.
W	Set of wavelengths, index w.
α^{lw}	Equal to 1 if wavelength channel w in fiber link l is being used.

Virtual Topology:

V	Set of MPLS routers, index v.
E	Set of FA TE links, index e.
$K(e)$	Set of lightpaths to support FA TE link e, index k.

$K_1(e)$	Subset of $K(e)$ already deployed in the optical topology.
$K_2(e)$	Subset of $K(e)$ not currently deployed in the optical topology.
$E(v)$	Subset of FA TE links incident to router v.
p_v	Amount of optical ports available in router v.
b_{ek}	Available capacity in FA TE link e using lightpath k (Gb/s).
r^l_k	Equal to 1 if route k uses fiber link l.

Demands to be recovered:

D	Set of PSC LSPs to be recovered, index d.
$SD(d)$	Set of $\{s_d, t_d\}$ MPLS routers of demand d.
b_d	Bitrate of demand d (Gb/s).

The decision variables are:

ω_{dek}	Binary. Equal to 1 if demand d is routed through FA TE links e using lightpath k, 0 otherwise.
δ^w_{ek}	Binary. Equal to 1 if FA TE link e uses lightpath k and wavelength w, 0 otherwise.
σ_d	Binary. Equal to 1 if demand d is recovered, 0 otherwise.
γ_v	Positive integer. Amount of optical ports used for restoration in MPLS router v.
υ_{dekt}	Binary. Equal to 1 if demand d is routed through FA TE links e using lightpath k and is established in time interval t.
χ_{dt}	Binary. Equal 1 if demand d is established in time interval t.
φ_{ekt}	Binary. Equal 1 if lightpath k of FA TE links e is active in time interval t.
C_{ek}	Positive integer. Completion time of lightpath k of FA TE links e.

Finally, the mathematical programming formulation for the BAREMO problem is as follows:

$$\text{maximize } A_1 \cdot \sum_{d \in D} b_d \cdot \sigma_d - A_2 \cdot \sum_{v \in V} \gamma_v - A_3 \cdot \sum_{e \in E} \sum_{k \in K(e)} C_{ek} \tag{7.1}$$

subject to:

$$\sum_{e \in E(v)} \sum_{k \in K(e)} \omega_{dek} = \sigma_d \quad \forall d \in D, v \in SD(d) \tag{7.2}$$

$$\sum_{e\in E(v)}\sum_{k\in K(e)}\omega_{dek}\le 2 \quad \forall d\in D, v\in \overline{SD(d)} \tag{7.3}$$

$$\sum_{\substack{e'\in E(v)\\ e'\neq e}}\sum_{k\in K(e')}\omega_{de'k}\ge \sum_{k\in K(e)}\omega_{dek} \quad \forall d\in D, v\in \overline{SD(d)}, e\in E(v) \tag{7.4}$$

$$\sum_{d\in D} b_d\cdot\omega_{dek}\le b_{ek} \quad \forall e\in E, k\in K_1(e) \tag{7.5}$$

$$\sum_{d\in D} b_d\cdot\omega_{dek}\le b_{ek}\cdot\sum_{w\in W}\delta_{ek}^{w} \quad \forall e\in E, k\in K_2(e) \tag{7.6}$$

$$\sum_{w\in W}\delta_{ek}^{w}\le 1 \quad \forall e\in E, k\in K_2(e) \tag{7.7}$$

$$\sum_{e\in E}\sum_{k\in K_2(e)} r_k^l\cdot\delta_{ek}^{w}+a^{lw}\le 1 \quad \forall l\in L, w\in W \tag{7.8}$$

$$\gamma_v=\sum_{e\in E(v)}\sum_{k\in K_2(e)}\sum_{w\in W}\delta_{ek}^{w} \quad \forall v\in V \tag{7.9}$$

$$\gamma_v\le p_v \quad \forall v\in V \tag{7.10}$$

$$\upsilon_{dekt}\le\omega_{dek} \quad \forall d\in D, e\in E, k\in K_2(e), t\in T \tag{7.11}$$

$$\upsilon_{dekt}\le\chi_{dt} \quad \forall d\in D, e\in E, k\in K_2(e), t\in T \tag{7.12}$$

$$\upsilon_{dekt}\ge\omega_{dek}+\chi_{dt}-1 \quad \forall d\in D, e\in E, k\in K_2(e), t\in T \tag{7.13}$$

$$\sum_{d\in D}\upsilon_{dekt}\le 1+|D|\cdot\sum_{\substack{t'\in T\\ t'<t}}\varphi_{ekt'} \quad \forall e\in E, k\in K_2(e), t\in T \tag{7.14}$$

$$\sum_{d\in D}\sum_{\substack{t'\in T\\ t'\le t}}\upsilon_{dekt'}\ge\varphi_{ekt} \quad \forall e\in E, k\in K_2(e), t\in T \tag{7.15}$$

$$\sum_{t\in T}\chi_{dt'}=\sum_{e\in E}\sum_{k\in K_2(e)}\omega_{dek} \quad \forall d\in D \tag{7.16}$$

$$C_{ek}=\sum_{t\in T}\left(1-\varphi_{ekt}\right)-|T|\cdot\left(1-\sum_{w\in W}\delta_{ek}^{w}\right) \quad \forall e\in E, k\in K_2(e) \tag{7.17}$$

The objective function (7.1) maximizes the total bitrate recovered, whilst minimizing the use of optical ports and the restoration time. A_1, A_2, and A_3 are constants.

Constraints (7.2)-(7.6) compute the route and perform aggregation of demands through the virtual topology. Constraint (7.2) defines whether a demand is to be

restored by selecting one FA TE link incident to source and destination routers. Constraints (7.3) and (7.4) perform routing and aggregation of demands in intermediate routers. Constraints (7.5)-(7.6) allow demands to use FA TE links ensuring that their capacity is not exceeded. When a new FA TE link has to be created, constraint (7.6) ensures that a wavelength is assigned.

Constraints (7.7) and (7.8) deal with wavelength assignment. Constraint (7.7) implements the wavelength continuity constraint and constraint (7.8) ensures that each wavelength is used only once.

Constraints (7.9)-(7.10) take care of the amount of optical ports used in each router. Constraint (7.9) counts the number of new optical ports used and constraint (7.10) guarantees that the number of optical ports used does not exceed the quantity available in the router.

Constraints (7.11)-(7.17) perform sequencing assigning new FA TE links set up to time intervals. Constraints (7.11)-(7.13) determine in which period each FA TE link must be established. Constraint (7.14) guarantees that each new FA TE link to be established is triggered by exactly one demand. Once the FA TE link is established, several demands using it can be set up simultaneously, i.e. these latter demands depend upon the former to be established. Constraint (7.15) ensures that a given FA TE link becomes available once the first demand using it has been established. Constraint (7.16) assures that each demand with assigned restoration resources is assigned to a time period. Finally, constraint (7.17) accounts for the completion time for each FA TE link. Note that the completion time of any FA TE link not to be established is set to zero.

Concerning the BAREMO problem complexity, it can be considered NP-hard since simpler MLN routing problems have been proved to be NP-hard (e.g., [Zh11]). As to the BAREMO problem size, the number of variables is $O(|D|\cdot|E|\cdot|K| + |E|\cdot|K|\cdot|W| + |V| + |D|\cdot|E|\cdot|K|\cdot|T|)$ and the number of constraints $O(|D|\cdot|V|\cdot|E| + |E|\cdot|K| + |L|\cdot|W| + |D|\cdot|E|\cdot|K|\cdot|T|)$. To illustrate the size of the problem, it is worth highlighting that the number of variables and constraints rise up to approximately 10^6 for the network scenario presented in the BAREMO results section (7.2.4).

Although the MILP can be solved for really small instances, its size and complexity is so high that solving the problem to optimality becomes impractical for the stringent times required for restoration, even using commercial solvers such as CPLEX [CPLEX]. Hence, heuristic algorithms are needed to provide good solutions within the required time.

7.2.3 Heuristic Algorithm

In this subsection, we describe the heuristic algorithm developed to efficiently solve the BAREMO problem. Recall that the computation time is a key performance

Table 7-1 Randomized Algorithm

```
INPUT TED_copy, D, maxIter
OUTPUT Sol
 1:  Sol ← Ø
 2:  for 1..maxIter do
 3:      Q ← sortRandom(D); R ← Ø
 4:      for each d in Q do
 5:         d.route ← shortestPath(TED_copy, d)
 6:         if d.route ≠ Ø then
 7:            allocate(TED_copy, d)
 8:            R ← R ∪ {d}
 9:      R.fitness ← computeFitness(R) (eq. 7.1)
10:      if Sol = Ø or R.fitness > Sol.fitness then
11:         Sol ← R
12:      resetAllocations(TED_copy, R)
13:  return Sol
```

metric for the proposed algorithm. To cope with this requirement, it is essential to produce solutions as diverse as possible to reduce the amount of iterations, and thus the computation time needed to find high quality solutions.

In view of the above, we developed the randomized algorithm, specified in Table 7-1. It consists in performing a number of iterations to construct solutions by rerouting every demand to be restored in different order. As stated in the previous section, the algorithm receives the *TED_copy*, containing $G_o(N, L)$ and $G_v(V, E)$, and the set of demands to restore. The number of iterations to be performed is a fixed parameter.

At each iteration, the set of demands is sorted randomly (line 3). Next, the shortest route is computed for each demand (line 5). If the route exists, the demand is allocated (i.e., resources are used in *TED_copy*) and added to the solution set *R* (lines 6-8). In case that new FA TE links are created during the routing process, they are also added to *TED_copy* and its remaining capacity becomes immediately available for the rest of the demands. Once a solution is built, its fitness value is evaluated using eq. (1). That solution replaces the best solution (*Sol*) provided that it improves *Sol* (lines 9-11). All new used resources are afterwards released from *TED_copy*.

7.2.4 BAREMO Numerical Results

In this section, we present the scenario considered in our simulations and we show the results obtained when using sequential versus bulk restoration.

Fig. A-9 shows the topology of a European MLN network consisting of 12 LSC nodes. Colored nodes have a PSC node attached. Each PSC node is physically connected to its associated LSC node with 16 bidirectional ports/transceivers. Each optical link (i.e., between LSC nodes) supports 16 DWDM channels per direction operating at 10 Gb/s.

Aiming at validating the heuristic algorithm proposed in the previous subsection (7.2.3), we used the scenario described above; the network was loaded and the specific configuration, including TED and established LSPs, stored at some random times. After that, we used those recorded configurations to produce individual problem instances to be solved by both, the MILP formulation and the heuristic algorithm proposed for the BAREMO problem. The MILP formulation was implemented in Matlab and solved using the CPLEX optimizer [CPLEX] on a 2.4 GHz Quad-Core machine with 8 GB RAM memory running Linux. The heuristic, implemented in C++, was run as a standalone application in the same computer as before.

Table 7-2 shows the obtained solving times and the amount of LPSs at each dependence level (labeled as 0..3) for several sizes of path restorations bulks. Note that when no dependences were found all the paths could be established in parallel using already set-up FA links, whereas when 1 or more dependences are found a delay (*t_lsp*) needs to be introduced to guarantee that the required logical links are actually induced and created. In the table, for each level of dependence, the amount of LSPs to be established in parallel resulting from solving the MILP model or from using the heuristic is shown. Regarding solving times, the time to generate each instance was not considered in the case of MILP. In all the tests performed, the proposed heuristic restored all the PSC LSPs. Nonetheless, the amount of LSPs in each dependence level was slightly different, which increases average times to restore; the maximum level of dependence remain constant, and so the maximum restoration times. The obtained solving times and results clearly illustrate the effectiveness of the proposed algorithm.

To compare the performance of sequential and bulk restoration in dynamic scenarios, different offered traffic loads were generated. Incoming PSC LSP requests arrive to the system following a Poisson process and are sequentially served without prior knowledge of future incoming connection requests. The holding time of the PSC LSPs is exponentially distributed with a mean value equal

Table 7-2 MILP-Heuristic Comparison

LSPs to restore	**MILP time (s)**	**Heuristic time (ms)**	**Dependence level (MILP/Heuristic)**			
			0	**1**	**2**	**3**
5	2.3	39	3/3	2/2		
12	5.2	97	6/6	3/2	3/4	
16	7.6	139	6/6	15/15		
29	17.9	207	14/14	14/14	1/1	
33	20.7	268	12/12	13/12	3/3	5/6

to 8 hours. Source/destination pairs are randomly chosen with equal probability (uniform distribution) among all MPLS nodes. Different values of the offered traffic load are created by changing the inter-arrival rate while keeping the mean holding time constant. Furthermore, it is assumed that the bandwidth demand of each PSC LSP request is equal to 1 Gb/s. In our experiments, we assume that no retrial is performed; if a request cannot be served, it is immediately blocked. Besides, optical link failures follow a Poisson process with a MTTF equal to 51 hours. Link failures are randomly chosen with equal probability and the MTTR is fixed to 12 hours. Finally, all the results were obtained after requesting 10,000 PSC LSPs.

Plots in Fig. 7-3 depict the blocking probability and restorability as a function of the offered traffic load when the sequential and the bulk approaches are used. The obtained blocking probability performs similarly in both approaches (Fig. 7-3 top). It is worth noting that the range of offered traffic loads considered unleash blocking probabilities not higher than just over 1% (similar to those expected in realistic scenarios). In contrast, restorability reveals remarkable differences. When the sequential restoration approach is used, restorability is below 70% even under low traffic loads. That restorability value decreases dramatically as soon as the traffic grows, showing a value around 36% for the highest traffic load. Interestingly, when using the bulk approach, restorability remains almost constant, being as high as 91% even under high offered loads.

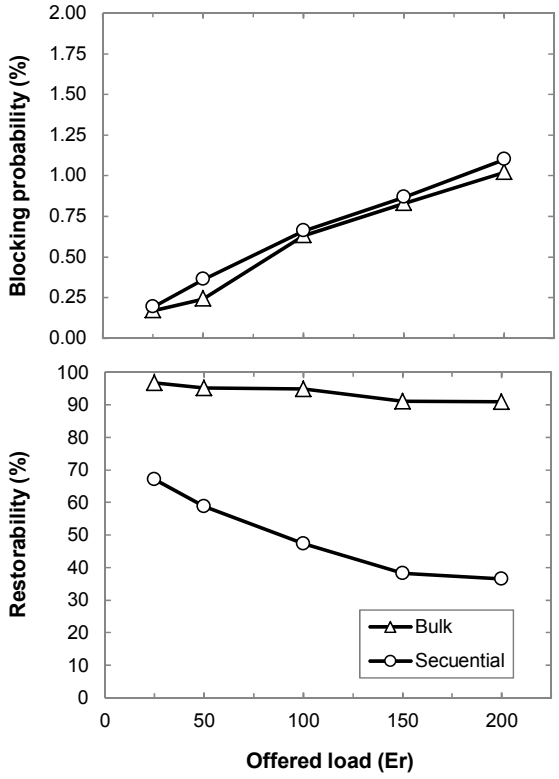

Fig. 7-3 Blocking probability (top) and restorability (bottom) against offered load.

Table 7-3 Un-Restorability vs. Offered Load

Load (Er)	Sequential		Bulk	
	Contention	Computation	Contention	Computation
25	32.6%	0.3%	3.3%	0.0%
50	41.2%	0.0%	4.0%	0.9%
100	52.4%	0.2%	4.2%	1.0%
150	61.4%	0.3%	7.6%	1.2%
200	63.5%	0.0%	8.0%	1.1%

It has to be highlighted that, being restorability significantly better when using the bulk restoration approach, more resources are used after the restoration process ends. In contrast, using the sequential restoration approach, many backup LSPs could not be restored, so no resources were used. However, blocking probabilities are similar under both approaches. It is clear that the sequential approach wastes many resources for restoration that cannot be reused, whereas the bulk approach optimizes the use of resources resulting in similar blocking probabilities.

Table 7-3 gives insight on the causes for *un-restorability* (computed as 1 - *restorability*) as a function of the offered load. Two main causes of un-restorability are analyzed: path computation, i.e. no sufficient resources to restore a given LSP are found during path computation, and resource contention, i.e., resources assigned during path computation to restore a given LSP have been occupied by another LSP during signaling phase. In other words, resource contention is produced when trying to allocate a wavelength channel already in use or when reserving bandwidth on a fully used FA TE link.

The high un-restorability observed when applying the sequential approach is mainly due to the resource contention when signaling the computed backup LSPs, as observed in Table 7-3. Recall that in the sequential restoration approach, the PCE uses the same TED vision to compute paths for all the requests. Thus, since no TED update is done between consecutive path computations, the same resources may be assigned to different backup paths. In this context, observe that the path computation failure (due to lack of resources) is negligible compared to the resource contention occurrences. On the contrary, when using the bulk approach, the BAREMO algorithm tries to globally optimize the use of the network resources. The marginal resource contention percentage obtained is due to the fact that specific computed backup LSPs actually required a setup time longer than the selected *t_lsp*. A slightly higher percentage of path computation failures are shown using the bulk approach; this is caused by the lack of network resources. Note that these path computation failures illustrate the real un-restorability level that could be reached provided that longer *t_lsp* times were considered.

Fig. 7-4 depicts the influence of each cause of un-restorability under both approaches as a function of the offered load. The causes of un-restorability are categorized into three groups:

i. *Algorithm*: no available resources are found during the path computation phase,

ii. *LSC-LSP*: a wavelength channel for a LSC LSP segment to transport the PSC LSP being restored could not be allocated during the signaling phase,

iii. *PSC-LSP*: the required available bandwidth on a FA TE link could not be allocated.

A noticeable increase is shown for the *PSC-LSP* cause under the sequential approach. In fact, *PSC-LSP* cause is responsible for more than 95% of un-restorability under all the considered offered loads. The reason behind this is that most of the backup LSPs tend to occupy resources on the existing logical TE links causing the aforementioned resource contention among concurrent signaled backup LSPs. Under the bulk approach, the *PSC-LSP* cause is responsible for virtually all the obtained un-restorability. In this approach, *PSC-LSP* cause refers to the situation where the required FA TE link does not yet exist at signaling time. As stated above, the effects of this un-restorability cause can be reduced by increasing the value of *t_lsp* at a cost of increasing total restoration times.

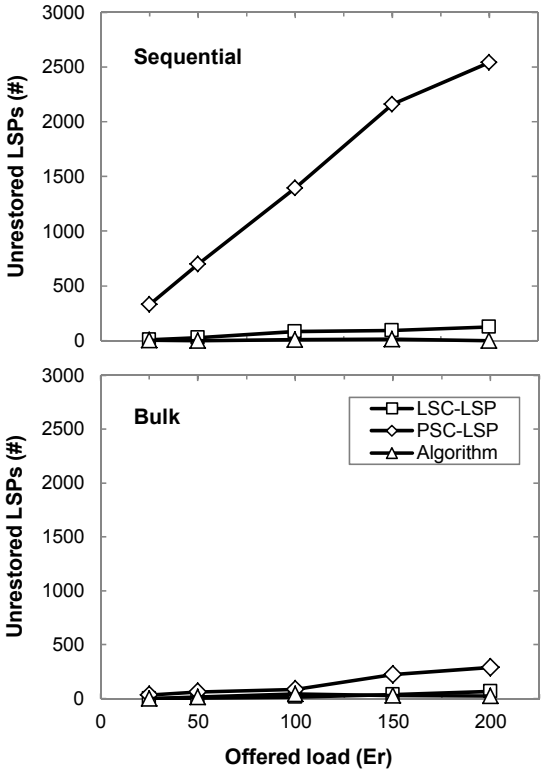

Fig. 7-4 Cause of un-restorability: Unrestored PSC LSPs against offered load

Finally, Fig. 7-5 evaluates the influence of the MTTF on the blocking probability and on the restorability. Aiming at establishing a fair comparison between both approaches, we selected the lowest offered traffic load (25 Er). At this point the blocking performance of the sequential approach is closest to that of the bulk. Different MTTF values were selected, ranging from 1 failure every 51 hours to 1 failure every 6 hours on average. We observe that the blocking probability increases exponentially under both approaches as soon as MTTF values are closer to both, the MTTR values and the LSPs' holding time. The reason of this is twofold: firstly, multiple failures may coexist; and secondly, the influence of a failure persists for a considerable time on the network, causing that the restored LSPs have longer routes, and thus, more resources are temporarily used. When MTTF decreases, the high resources usage does not disappear completely resulting in the observed increased blocking probability. Interestingly, however, the sequential approach performs much worse than the bulk approach under lower MTTFs. This is a consequence of the poor and low reuse of created FA TE links. Conversely, the FA TE links created under the bulk approach can be more frequently reused reducing up to 45% blocking probability. As for the restorability, its performance remains constant under both approaches, since the offered traffic load is fixed and the main cause of un-restorability is resource contention.

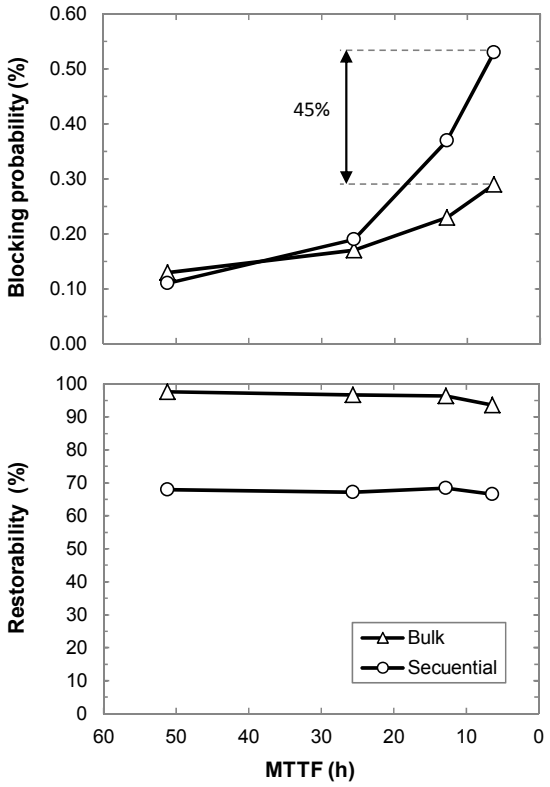

Fig. 7-5 Blocking probability (top) and restorability (bottom) vs. MTTF.

7.3 MPLS-over-Flexgrid: The DYNAMO Problem

As a consequence of the efficiency that bulk restoration reaches by reusing virtual links, the number of ports connecting MPLS routers and BV-OXCs can be reduced; let as assume that those ports are MF-TPs. Fig. 7-6 is intended for illustrating that reduction. Fig. 7-6(a) shows how the MPLS flows have been routed before the failure; each router is equipped with a number of client ports were the MPLS flows arrive. One MF-TP allowing for five lightpaths to be ended is connected to each router. For instance, in Fig. 7-6(a) two lightpaths are ended in the MF-TP, were flows #1 and #3 are groomed together into a single lightpath whereas flow #2 uses a different one. Flow #3 is routed through intermediate Router 2, so that flow enters in Router 2 by one of the lightpaths terminating in the MF-TP in that router and leaves it using the same MF-TP but aggregated together with flow #4 into lightpath R2-R3.

Fig. 7-6(b) and Fig. 7-6(c) show the routing at each router after the restoration has been done sequentially and bulk, respectively. The main difference can be shown in routers 2 and 4 were more resources in each MF-TP have been used in the case of sequential restoration as a consequence of the increased number of lightpaths created. Hence, efficiency in the use of the resources can reduce notably the amount of resources needed for the same grade of service, thus reducing remarkably CAPEX costs when dealing with expensive resources such as MF-TPs.

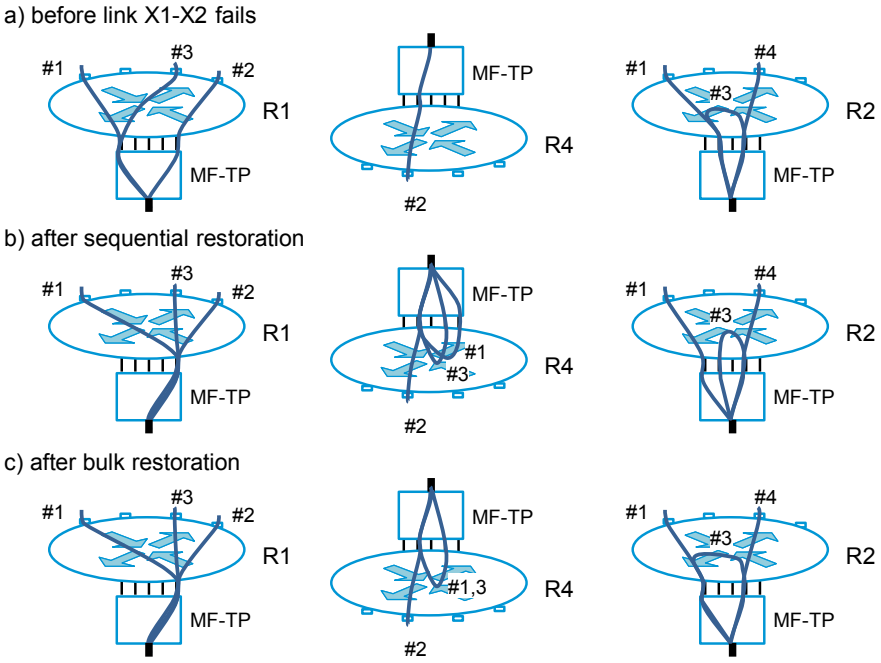

Fig. 7-6 Routing of MPLS demands.

7.3.1 Problem Statement

The DYNAMO problem can be formally stated as follows:

Given:

- a network topology represented by a graph $G_o(N, L)$, being N the set of BV-OXC nodes and L the set of bidirectional fiber links connecting two BV-OXC nodes, excluding failed ones; each link consists of two unidirectional optical fibers,
- a set S of available slots of a given spectral width for each link in L,
- the virtual network represented by a graph $G_v(V, E)$, being V the subset of N where MPLS routers are placed, and E the set of virtual links defining the connectivity among the MPLS nodes,
- a set D of MPLS demands to be recovered. Each demand d is defined by the tuple $\{s_d, t_d, b_d\}$, where s_d and t_d represent demand's source and destination MPLS routers and b_d its bitrate.

Output:

- the routing in the virtual topology of every recovered demand,
- the routing of the new lightpaths used to serve new virtual links to be created.

Objective: maximize the total amount of bitrate recovered whilst minimizing the amount of resources used (i.e., slots and MF-TPs) and the total recovery time.

We have modeled the DYNAMO problem by means of a mathematical programming formulation based on pre-computing channels to ensure spectrum contiguity as described in [Ve12.2]. The next subsection presents the formulation proposed.

7.3.2 Mathematical Model

The mathematical programming model for the DYNAMO problem performs routing in both the optical and the MPLS layers using node-link formulations for each network layer [Pi04]. A set of virtual links is pre-computed beforehand; each virtual link connects two locations with MPLS nodes provided that a feasible optical route can be found. A set of lightpaths is available for each virtual link, although its actual route on the optical topology is determined during the resolution of the problem.

The following sets and parameters have been defined:

Optical Topology:

N	Set of BV-OXC nodes, index n.
L	Set of fiber links, index l.
$L(n)$	Subset of fiber links incidents to BV-OXC node n.
$len(l)$	Length of fiber link l (km).
R	Set of bitrate-reach pairs (Gb/s, km), index r.
$len(r)$	Reach of a path using bitrate-reach pair r in km.
$b(r)$	Maximum bitrate of a path using bitrate-reach pair r.

Optical Spectrum:

S	Set of frequency slots, index s.
C	Set of channels, index c. Each channel c contains a subset of contiguous slots.
a^{ls}	Equal to 1 if slot s in fiber link l is being used.
u^{cs}	Equal to 1 if channel c includes slot s.
b_c	Capacity of channel c (Gb/s).

Virtual Topology:

V	Set of MPLS routers ($V \subseteq N$), index v ($v=n$ provided that BV-OXC node with index n is physically connected to the MPLS router with index v).
E	Set of virtual links, index e.
$P(v)$	Set of MF-TPs in MPLS router v, index p.
$K(e)$	Set of routes to support virtual link e, index k.
$K_1(e)$	Subset of $K(e)$ already deployed in the optical topology.
$K_2(e)$	Subset of $K(e)$ not currently deployed in the optical topology.
$E(v)$	Subset of virtual links incident to MPLS router v.
$N(e)$	Set of end BV-OXC nodes (nodes connected to the correspondent MPLS router) of virtual link e.
b_{ek}	Available capacity in virtual link e using lightpath k (Gb/s).
b_{pv}	Available capacity in MF-TP p in MPLS router v (Gb/s).
f_{pv}	Number of lightpaths that can be assigned to MF-TP p in MPLS router v.

g_{ekpv}	Equal to 1 if virtual link e using lightpath k ends in MF-TP p in MPLS router v

Demands to be recovered:

D	Set of MPLS demands to be recovered, index d.
$SD(d)$	Set of $\{s_d, t_d\}$ MPLS routers of demand d.
b_d	Bitrate of demand d (Gb/s).

The decision variables are:

ω_{dek}	Binary. Equal to 1 if demand d is routed through virtual link e using lightpath k, 0 otherwise.
δ^c_{ek}	Binary. Equal to 1 if lightpath k of virtual link e uses channel c, 0 otherwise.
λ^{lc}_{ek}	Binary. Equal to 1 if lightpath k of virtual link e uses channel c in fiber link l, 0 otherwise.
σ_d	Binary. Equal to 1 if demand d is recovered, 0 otherwise.
γ_{pv}	Binary. Equal 1 if MF-TP p in MPLS node v is allocated.
ν^r_{ek}	Binary. Equal 1 if lightpath k of virtual link e uses bitrate-reach pair r.
χ_{dt}	Binary. Equal 1 if demand d is established in time interval t.
φ_{ekt}	Binary. Equal 1 if lightpath k of virtual link e is active in time interval t.
C_{ekt}	Positive integer. Completion time of lightpath k of virtual link e.
C_{max}	Positive integer with the total recovery time.

Finally, the mathematical programming formulation for the DYNAMO problem is as follows:

$$\text{maximize } A_1 \cdot \sum_{d \in D} b_d \cdot \sigma_d - A_2 \cdot \sum_{v \in V} \sum_{p \in P(v)} \gamma_{pv} - A_3 \cdot C_{\max} \tag{7.18}$$

subject to:

$$\sum_{e \in E(v)} \sum_{k \in K(e)} \omega_{dek} = \sigma_d \quad \forall d \in D, v \in SD(d) \tag{7.19}$$

$$\sum_{e \in E(v)} \sum_{k \in K(e)} \omega_{dek} \leq 2 \quad \forall d \in D, v \in \overline{SD(d)} \tag{7.20}$$

$$\sum_{\substack{e' \in E(v) \\ e' \neq e}} \sum_{k \in K(e')} \omega_{de'k} \geq \sum_{k \in K(e)} \omega_{dek} \quad \forall d \in D, v \in \overline{SD(d)}, e \in E(v) \tag{7.21}$$

$$\sum_{d \in D} b_d \cdot \omega_{dek} \leq b_{ek} \quad \forall e \in E, k \in K_1(e) \tag{7.22}$$

$$\sum_{d \in D} b_d \cdot \omega_{dek} \leq \sum_{c \in C} b_c \cdot \delta_{ek}^c \quad \forall e \in E, k \in K_2(e) \tag{7.23}$$

$$\sum_{l \in L(n)} \sum_{c \in C} \lambda_{ek}^{lc} = \sum_{c \in C} \delta_{ek}^c \quad \forall e \in E, k \in K_2(e), n \in N(e) \tag{7.24}$$

$$\sum_{l \in L(n)} \sum_{c \in C} \lambda_{ek}^{lc} \leq 2 \quad \forall e \in E, k \in K_2(e), n \in \overline{N(e)} \tag{7.25}$$

$$\sum_{\substack{l' \in L(n) \\ l' \neq l}} \sum_{c \in C} \lambda_{ek}^{l'c} \geq \sum_{c \in C} \lambda_{ek}^{lc} \quad \forall e \in E, k \in K_2(e), n \in \overline{N(e)}, l \in L(n) \tag{7.26}$$

$$\sum_{l \in L} \lambda_{ek}^{lc} \leq |L| \cdot \delta_{ek}^c \quad \forall e \in E, k \in K_2(e), c \in C \tag{7.27}$$

$$\sum_{c \in C} \delta_{ek}^c \leq 1 \quad \forall e \in E, k \in K_2(e) \tag{7.28}$$

$$\sum_{e \in E} \sum_{k \in K_2(e)} \sum_{c \in C} \lambda_{ek}^{lc} \cdot u^{cs} + a^{ls} \leq 1 \quad \forall l \in L, s \in S \tag{7.29}$$

$$\sum_{e \in E(v)} \sum_{k \in K_2(e)} \sum_{c \in C} \delta_{ek}^c \cdot g_{ekpv} \leq f_{pv} \cdot \gamma_{pv} \quad \forall v \in V, p \in P(v) \tag{7.30}$$

$$\sum_{d \in D} \sum_{e \in E(v)} \sum_{k \in K_2(e)} \omega_{dek} \cdot b_d \cdot g_{ekpv} \leq b_{pv} \quad \forall v \in V, p \in P(v) \tag{7.31}$$

$$\sum_{d \in D} b_d \cdot \omega_{dek} \leq \sum_{r \in R} b(r) \cdot \nu_{ek}^r \quad \forall e \in E, k \in K_2(e) \tag{7.32}$$

$$\sum_{l \in L} \sum_{c \in C} len(l) \cdot \lambda_{ek}^{lc} \leq \sum_{r \in R} len(r) \cdot \nu_{ek}^r \quad \forall e \in E, k \in K_2(e) \tag{7.33}$$

$$\sum_{r \in R} \nu_{ek}^r \leq 1 \quad \forall e \in E, k \in K(e) \tag{7.34}$$

$$\sum_{d \in D} \omega_{dek} \cdot \chi_{dt} \leq 1 + |D| \cdot \sum_{\substack{t' \in T \\ t' < t}} \varphi_{ekt'} \quad \forall e \in E, k \in K_2(e), t \in T \tag{7.35}$$

$$\sum_{d \in D} \sum_{\substack{t' \in T \\ t' \leq t}} \chi_{dt'} \geq \varphi_{ekt} \quad \forall e \in E, k \in K_2(e), t \in T \tag{7.36}$$

$$\sum_{t\in T}\chi_{dt'}=\sum_{e\in E}\sum_{k\in K_2(e)}\omega_{dek}\quad \forall d\in D \tag{7.37}$$

$$C_{ek}=\sum_{t\in T}\left(1-\varphi_{ekt}\right)-\left|T\right|\cdot\left(1-\sum_{c\in C}\delta_{ek}^{c}\right)\quad \forall e\in E,k\in K_2(e) \tag{7.38}$$

$$C_{\max}\geq C_{ek}\quad \forall e\in E,k\in K_2(e) \tag{7.39}$$

The objective function (7.18) maximizes the total bitrate recovered, whilst minimizing the use of MF-TPs and the total restoration time. A_1, A_2, and A_3 are constants.

Constraints (7.19)-(7.21) compute the route and perform aggregation of demands through the virtual topology. Constraint (7.19) defines whether a demand is to be restored by selecting one virtual link incident to source and destination routers. Constraints (7.20) and (7.21) perform routing and aggregation in the intermediate routers.

Constraints (7.22)-(7.23) allow the demands to restore for using existing virtual links or new ones, in which case new lightpaths need to be created. Constraint (7.22) ensures that demands use existing virtual links with enough capacity, whereas constraint (7.23) ensures a channel with enough capacity is allocated for the amount of bitrate assigned to each new lightpath to be created.

Constraints (7.24)-(7.29) compute both route and channel assignment over the optical topology for those lightpaths supporting new virtual links. Note that constraints (7.24)-(7.26) compute the route in a similar way as constraints (7.19)-(7.21) do. Constraint (7.27) guarantees that every lightpath uses the same channel along its route. Constraint (7.28) implements the channel continuity constraint ensuring that no more than one channel is allocated to one lightpath. Constraint (7.29) allows each slot in each optical link to be used by only one lightpath provided that it was not previously used in the network.

Constraints (7.30)-(7.31) ensure that MF-TPs capacity is not exceeded. Constraint (7.30) guarantees that the number of new lightpaths assigned to each MP-TP does not exceed the given availability. Constraint (7.31) ensures that the bitrate associated to each MF-TP is under the given maximum.

Constraints (7.32)-(7.34) take care of bitrate-reach pair selection. Constraint (7.32) chooses a pair with enough bitrate for the traffic to be transmitted and constraint (7.33) assures that the reach of that pair works for the length of the lightpath. Constraint (7.34) ensures that only one pair is chosen.

Constraints (7.35)-(7.39) perform demand sequencing assigning demand establishing, and so new virtual links, to time intervals. Constraint (7.35) guarantees that each new virtual link to be established is triggered by exactly one demand. Once the virtual link is established, several demands using it can be set-

up simultaneously, i.e. these latter demands depend upon the former to be established. Constraint (7.36) ensures that a given virtual link becomes available once the first demand using it has been established. Constraint (7.37) assures that each demand with assigned restoration resources is assigned to a time period. Constraint (7.38) accounts for the completion time for each of the virtual links that are established. Note that the completion time of any virtual link not to be established is set to zero. Finally, constraint (7.39) computes the total completion time defined as the max function of the completion time for every single virtual link.

Note that constraint (7.35) entails multiplying two binary variables thus converting the mathematical model into a non-linear one. Notwithstanding, variable multiplication can be easily solved as the expense of introducing additional binary variables and constraints. Even though, the DYNAMO problem can be considered *NP-hard* since simpler multilayer network problems have been proved to be *NP-hard* (e.g. [Zh11]); hence its exact solving becomes impractical for the stringent times required for restoration and, as a result, an heuristic algorithm is needed so as to provide good near optimal solutions in the time periods required for recovering.

7.3.3 Heuristic Algorithm

In this section we propose a GRASP-based heuristic to solve the DYNAMO problem. Table 7-4 describes the proposed greedy randomized constructive algorithm, where the α parameter controls the size of the RCL. Equation (7.40) is used to quantify the quality of recovering a given demand d, in line with the objective function for the mathematical model.

$$q(d) = A_1 \cdot d.bw - A_2 \cdot d.newResourœs - A_3 \cdot d.depend \tag{7.40}$$

During the local search, the route of the demands is changed so as to try to avoid new virtual links to be created.

The heuristic was validated, for really small instances, against the mathematical formulation; in all the instances checked, the heuristic provided the optimal solution, i.e. the same solution than the one obtained from solving the mathematical model with CPLEX. In light of this, the heuristic was used to obtain the results presented next.

Table 7-4 Greedy Randomized Constructive Algorithm

```
INPUT G_o(N, L), G_V(V, E), D, α
OUTPUT Sol
 1:  Sol ← Ø
 2:  Q ← D
 3:  while Q ≠ Ø do
 4:    for each d ∈ Q do
 5:      d.route = shortestPath(G_V, d)
 6:      if d.route = Ø then
 7:        q(d) ← -INF
 8:      else
 9:        evaluate the quality q(d) using eq. (7.40)
10:    q^min ← min{q(d) : d∈Q}
11:    q^max ← max{q(d) : d∈Q}
12:    if q^max = -INF then
13:      break
14:    RCL ← {d∈Q : q(d) ≥ q^max-α(q^max - q^min)}
15:    Select an element d from RCL at random
16:    for each e ∈ d.route do
17:      if not e.isImplemented then
18:        implement(G_o, e)
19:    implement(G_V, d)
20:    Q ← Q \ {d}
21:    Sol ← Sol ∪ {d}
22:  return Sol
```

7.3.4 DYNAMO Numerical Results

The performance of the considered restoration approaches was compared on two national network topologies: the 21-node TEL and the 21-node DT topologies, reproduced in Appendix A, where each location contained one MPLS router and one BV-OXC.

Evaluation of the restoration approaches was performed by using the simulation algorithm presented in Table 7-5. To load the network (line 2), we developed an ad-hoc event-driven simulator in OMNet++ [OMNet]; a dynamic network environment was simulated where incoming MPLS connection requests arrive to the system following a Poisson process and are sequentially served without prior knowledge of future incoming connection requests.

To compute the RSA of the lightpaths, we used the algorithm described in Chapter 6. The holding time of the connection requests is exponentially distributed with a mean value equal to 2 hours. Source/destination pairs are randomly chosen with equal probability (uniform distribution) among all MPLS nodes. Different values of the offered network load are created by changing the inter-arrival rate while keeping the mean holding time constant. Finally, note that each point in the results is the average of $10 \cdot |L|$ runs and that sequential and bulk restoration approaches are executed using identical input data.

Table 7-5 Simulation Algorithm

```
 1:  for i=1..10 do
 2:      Load the network to the desired level
 3:      store the state of the network
 4:      for each l ∈ L do
 5:          cut l
 6:          perform sequential restoration
 7:          repair l
 8:          restore network state
 9:          cut l
10:          perform bulk restoration
11:          repair l
12:          restore network state
```

In our experiments, the bitrate of each MPLS flow was set to 1 Gb/s, the QPSK modulation format was used for the optical signals, the optical spectrum width was set to 1 THz, the slot width was fixed to 6.25GHz, and each MPLS router was equipped with one MF-TP, whose capacity for terminating lightpaths ranged from 2 to 5. Regarding bitrate-reach pairs, we used the values reproduced in Table 7-6.

To find the appropriate loads, we first run the simulator without cutting links and store the resulting blocking probabilities. Five traffic loads unleashing blocking probabilities ranging from 0.1% to 5% for each of the networks and MF-TP capacities considered were found.

Fig. 7-7 presents the percentage of un-restorability of MPLS flows as a function of the amount of flows to restore for the TEL and DT networks. Five plots for both sequential and bulk restoration approaches are presented, one for each of the found traffic loads, where each point corresponds to one MF-TP capacity value. As anticipated, the sequential approach produces un-restorability values as high as 27% to above 50% as a function of the traffic load. In contrast, the bulk approach achieves un-restorability values of almost 0%, i.e. virtually all MPLS flows are restored for even the most stringent traffic load. The behavior is the same for the TEL network as for the DT, as shown in Fig. 7-7.

Table 7-6 Bitrate-reach Pairs

Bitrate (Gb/s)	Max reach (km)
400	400
100	1,000
40	2,000
10	2,500

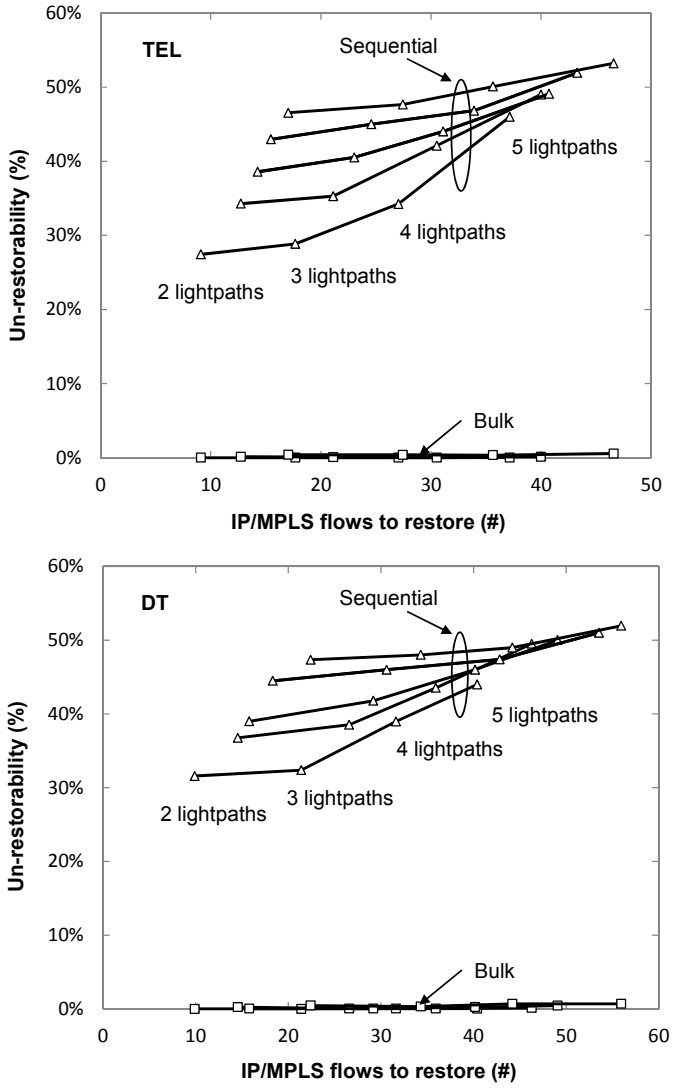

Fig. 7-7 Unrestored MPLS flows against the total amount of flows to restore.

Table 7-7 gives insight on the results for the TEL network using MF-TP with capacity for 5 lightpaths. There, the amount of MPLS flows to be restored ranges, on average, from 37 to 46 as a function of the load offered to the network. Un-restorability values are given for both, the sequential and the bulk approach. Two main causes behind unrestored flows are detailed:

i. no route could be found during path computation,

ii. resource contention, i.e. resources where already in use during the signaling phase.

The latter gets together frequency slots and existing virtual links that were available in the TED when the route was computed, or MF-TPs resources that are actually allocated during lightpaths' set-up.

Table 7-7 Restoration Results for the TEL Network Using 5-Lightpath MF-TPs

Offered Load	# flows to restore	Un-restorability Sequential			Un-restorability Bulk		
		Total	Path Comp.	Resource contention	Total	Path Comp.	Resource contention
357	37.14	47.72%	0.00%	47.72%	0.02%	0.02%	0.00%
378	39.98	50.10%	0.00%	50.10%	0.13%	0.13%	0.00%
383	40.71	49.11%	0.00%	49.11%	0.13%	0.13%	0.00%
395	43.28	51.95%	0.00%	51.95%	0.21%	0.21%	0.00%
409	46.59	53.21%	0.00%	53.21%	0.56%	0.56%	0.00%

As detailed, the reason for the high un-restorability of the sequential approach is resource contention; restoration routes are computed using the state of the resources in the TED, however, as a result of the number of path computation requests arriving at the PCE, the TED becomes immediately outdated and thus both, the same resources could be assigned to several routes, and ports availability decreases notably so no new lightpaths could be established. The bulk restoration approach, in contrast, reaches negligible un-restorability values since network resources are globally optimized. Once, restoration routes are computed for a bulk of requests, resource contention disappears completely.

Fig. 7-8 explores the causes of un-restorability for the sequential restoration approach when the TEL network was loaded with the intensity unleashing 1% blocking probability. As shown, when the capacity of the MF-TPs is low, the percentage of resource contention as a consequence of the lack of the resources in MF-TPs is the dominant cause of un- restorability. However, as soon as higher capacity MF-TPs are used, the main cause rapidly changes to contention in the use

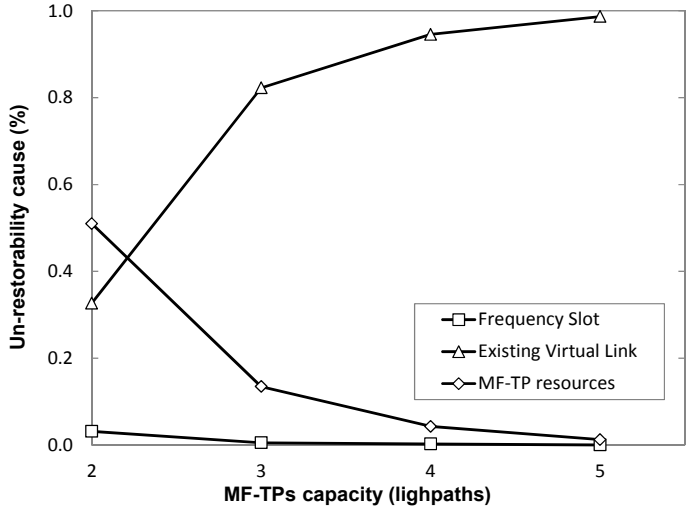

Fig. 7-8 Un-restorability by cause for the sequential approach.

of existing virtual links capacity. The same results and conclusions are valid for any other load on both TEL and DT networks.

The advantages of bulk restoration come at the cost of increasing restoration times when compared to that of the sequential approach. That is particularly noticeable for high traffic loads where large number of MPLS flows need to be restored as illustrated in Fig. 7-9. Plots for maximum bulk restoration computation times for each network and capacity of the installed MF-TPs are presented; an almost linear trend with the amount of flows to restore can be observed.

Those computation times translate into restoration times that include both, sequencing restoration route signaling to allow new virtual links to be created prior being reused, and actual signaling. Thus, the time to restore an MPLS flow depends on the depth of the dependences list for that flow and the length of the route to be signaled.

As described in the DYNAMO mathematical model above, dependence depth needs to be minimized so as to minimize restoration times in bulk restoration, and as such, it was included in the heuristic algorithm. Table 7-8 presents the dependence depth values (max, min and average) as a function of the offered load. As shown, the maximum value is only 5, which in turn introduces a considerable delay for the last set of restoration routes to be signaled.

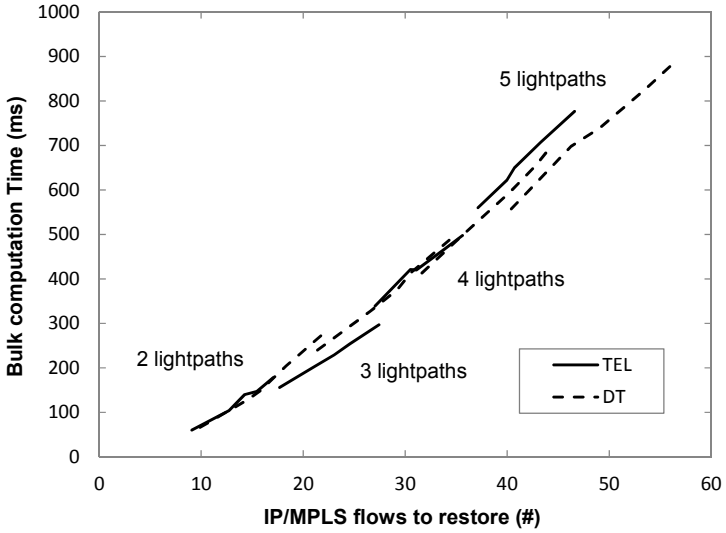

Fig. 7-9 Max. bulk computation times vs. number of flows to restore.

Table 7-8 Bulk Computation for the TEL Network Using 5-Lightpath MF-TPs

Offered Load	Dependences			Max. Computation Time (ms)
	Avg	Max	Min	
357	0.8	3.0	0.0	560
378	1.0	4.0	0.0	622
383	1.0	3.0	0.0	649
395	1.1	5.0	0.0	706
409	1.4	5.0	0.0	776

The histograms in Fig. 7-10 represent the restoration times distribution when the TEL network was loaded with the medium intensity. Cumulative distributions are also plotted. Signaling times computation were performed using equations and experimental times described in [Ve10]. The main conclusion is that restoration times lower than 1s can be achieved even when the number of MPLS flows to be restored is as high as 50; more than 50% of them being restored in less than 500ms.

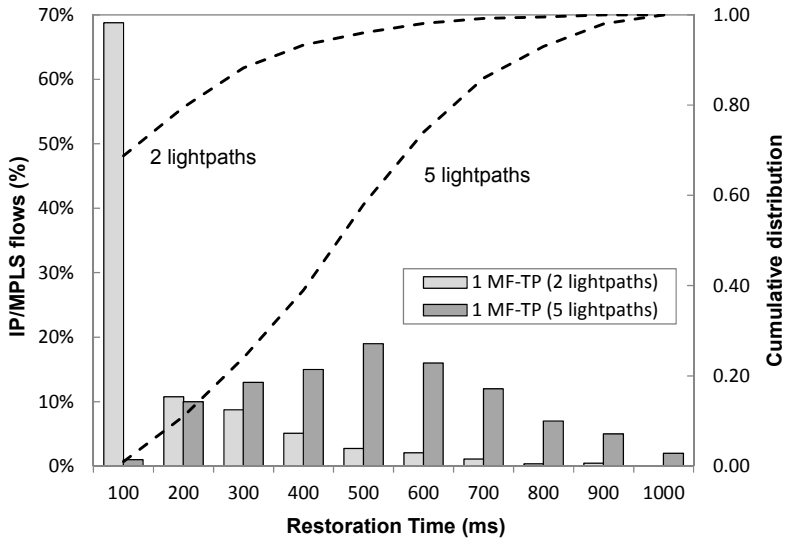

Fig. 7-10 Restoration times distribution for different MF-TP capacities.

7.4 Conclusions

The problem of dynamic restoration within multilayer networks using a centralized PCE for the path computation was studied. In such scenario, an optical link failure may disrupt multiple MPLS flows, generating a large set of path requests to be served by the PCE. Applying a regular sequential PCE approach, the attained

restorability was rather poor due to resource contention when signaling MPLS connections, as well as the scarce exploitation of the grooming. To improve such restorability metric, reducing resource contention and leveraging grooming objectives, a PCE GCO architecture was designed and implemented to groupe all the requests to perform bulk path computation.

Firstly, we focused on the MPLS-over-WSON restoration problem, which was formally stated and modeled using a MILP formulation. Due to the stringent requirements in terms of restoration time, a randomized heuristic algorithm was conceived.

Both approaches (sequential vs. bulk) were experimentally validated and compared by means of three figures of merit: blocking probability, restorability and restoration time. For the sake of completeness, different traffic loads and failure rates were considered. In light of the results, we can conclude that bulk restoration using the GCO architecture increases noticeable the restorability attained by the sequential approach: 91% and 36% at the highest traffic load when using bulk and sequential approaches, respectively.

That enhancement was, however, achieved at the expense of increasing the restoration time due to fixed timers required to both create the bulk of requests and delay the establishment of those LSPs having dependencies with former backup paths. Notwithstanding, the results obtained showed that this restoration time is even lower for the bulk than for the sequential approach when the failure rate increases. Last but not least, we observed that as the MTTF is decreased, thus increasing the failure rate, the bulk approach performs better with regard to the blocking probability. This was achieved thanks to its intrinsic more efficient use of the overall network resources.

Secondly, the bulk restoration approach in multilayer MPLS-over-flexgrid networks was proposed. To this end, the DYNamic restorAtion in Multi-layer MPLS-over-flexgrid Optical networks (DYNAMO) problem was stated and a mathematical model was developed. Similarly to the previous problem, as a consequence of the stringent times needed in on-line restoration scenarios, a heuristic algorithm that provides near-optimal solutions with computation times lower than one second was proposed. The focus of that heuristic was in obtaining the highest effectiveness in terms of the objective function (maximize restorability, minimizing the amount of resources used and dependence depth). The performance of sequential and bulk approaches was extensively assessed on two national network topologies, using an ad-hoc network simulator.

The main causes of the high un-restorability of the sequential approach were studied resulting in contention in the use of capacity in existing virtual links and resource availability in MF-TPs.

Increased restoration times were identified as the main disadvantage of the bulk restoration. Two main causes for those longer times were:

i. longer computation times;

ii. dependence depth.

The first cause can be notably improved using simplified heuristics such as the one used for BAREMO and computation times remarkably shorter can be achieved. However, the second cause was identified as the really limiting factor for restoration times, since it involves long waiting times. There, several approaches can be devised such as using restoration classes to give priority to some flows. Notwithstanding, sub-second restoration times were achieved even for high traffic loads, where as many as 50 restoration paths were computed and signaled.

In the next chapter, we continue designing restoration schemes for flexgrid single layer networks.

Chapter 8

Dynamic Multipath Restoration

Continuing with dynamic restoration, in the bitrate squeezing and multipath restoration scenario we take advantage of MF-TPs, which provide additional flexibility allowing reconfiguration of optical connections to be performed. Recall that using multiple paths to restore each individual demand was found advantageous in Chapter 5. In this chapter, we propose the bitrate squeezing and multipath restoration (BATIDO) problem to improve restorability of large bitrate connections. The BATIDO problem aims at maximizing the amount of restored bitrate by exploiting the available spectrum resources along multiple routes. BATIDO is formally stated and next modeled using an ILP formulation. As a result of the stringent time to computing a solution, a heuristic algorithm providing better trade-off between optimality and complexity is proposed to solve the problem.

8.1 Bitrate squeezing and multipath restoration: The BATIDO Problem

To illustrate the different restoration schemes that can be applied to the set of connections affected by a failure in the context of flexgrid networks, Fig. 8-1 shows a simple network topology were a lightpath is set-up between nodes *s* and *t*. In normal conditions (Fig. 8-1a), let us assume that the lightpath uses a slot consisting of 16 frequency slots to convey the requested bitrate, in our example 400 Gb/s (see Table A-3).

Let us imagine that a link failure occurs and the above lightpath is affected. If a restoration lightpath, including route and spectrum allocation, can be found in the network for the required 16 slots, the disrupted lightpath is obviously restored

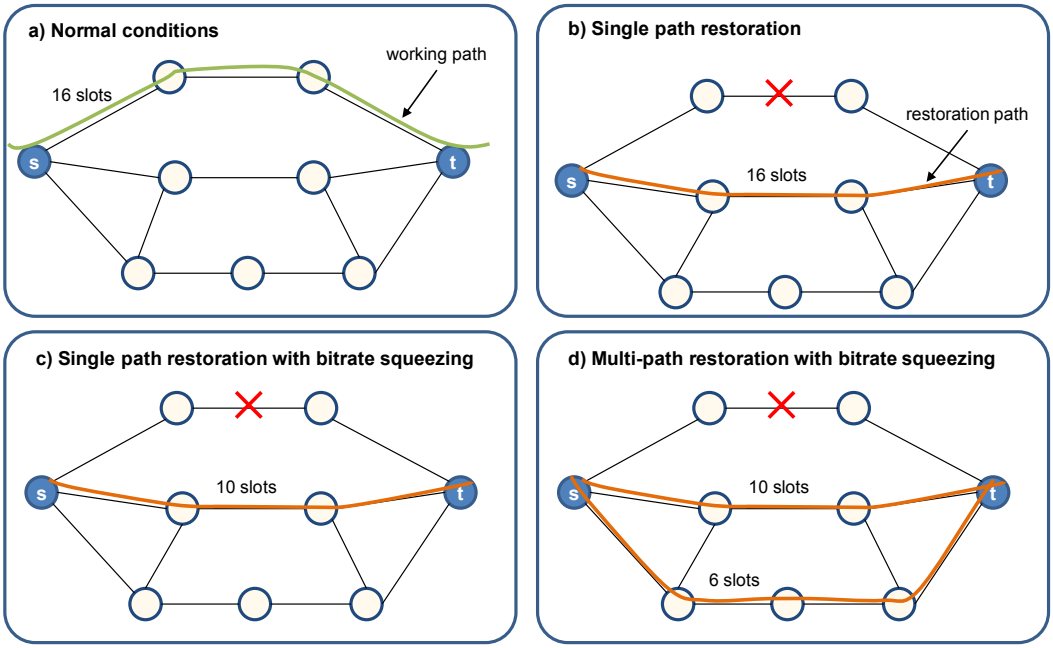

Fig. 8-1 Bitrate squeezing and multipath restoration.

using the restoration path. This is the normal restoration scheme that has been traditionally used in optical networking; we call this scheme as single path restoration (Fig. 8-1b).

However, in contrast to protection schemes, the availability of 16 contiguous frequency slots at failure time is not generally guaranteed in restoration schemes. In such case, the disrupted lightpath can be restored in part. This case, named single path restoration with bitrate squeezing in this thesis, is illustrated in Fig. 8-1c. Note that the restoration lightpath uses just 10 frequency slots and thus, the conveyed bitrate has been squeezed to 200Gb/s.

Another possibility is to use several parallel lightpaths, each conveying part of the total bitrate, so the original bitrate of the disrupted lightpath is fully restored (Fig. 8-1d). Note that in this case, although restoration lightpaths use 16 frequency slots, the total bitrate cannot be recovered, since 200Gb/s can be conveyed within 10 slots and only 100Gb/s within 6 slots. This illustrates the fact that spectral efficiency decreases when multiple lightpaths are used. Although for that very reason network operators prefer not using multipath for provisioning, it can be exploited to improve restorability, provided that the number of parallel lightpaths is kept limited.

After stating the bitrate squeezing and multipath restoration problem, next subsections present two alternative ways to solve it.

8.1.1 Problem Statement

The problem can be formally stated as follows:

Given:

- a network topology represented by a graph G(N, E), where N is the set of optical nodes and E is the set of fiber links connecting two optical nodes,
- a set S of available frequency slots of a given spectral width in each fiber link in E,
- a set D of failed demands to be restored, each requesting a fixed bitrate b^d.

Output: the routing and spectrum assignment for each restored demand $d \in D$.

Objective: maximize the total restored bitrate.

As previously discussed, the BATIDO problem can be faced using three different approaches:

i. single path restoration, where the total requested bitrate is either restored using a single lightpath or blocked,

ii. bitrate squeezing restoration, where part of the originally requested bitrate is restored whilst the rest is blocked,

iii. multipath restoration with bitrate squeezing, where several restoration lightpaths can be established to recover partially or totally one single demand.

An ILP-based model, which includes the above schemes, is presented next.

8.1.2 Mathematical Model

The ILP model is based on the *link-path* formulation for RSA in [Ve12.2], where a set of routes are computed beforehand for each demand (excluding the failed fiber link). It is worth highlighting that the term lightpath includes both, the physical route and the allocated slot. Pre-computed slots are used to ensure frequency slot contiguity in the input data, thereby alleviating to some extent the problem complexity. The characteristics of the considered modulation format are also embedded in the input data.

The following sets and parameters have been defined:

Topology:

N	Set of optical nodes, index n.
E	Set of fiber links, index e.
K	Set of pre-computed routes excluding the failed fiber link, index k.

P Set of MF-TPs, index p.

$N(k)$ Subset of optical nodes that are source or destination of route k.

$K(n)$ Subset of routes which source or destination optical node is n.

$P(n)$ Subset of MF-TPs of optical node n.

h^k_e Equal to 1 if route k uses link e, 0 otherwise.

b^k Maximum bitrate (in Gb/s) that route k can convey (due to physical impairments).

fF^n_p Number of free flows in MF-TP p of optical node n. If $p \notin P(n)$, fF^n_p is equal to 0.

b^n_p Unreserved capacity of MF-TP p of optical node n in Gb/s.

Spectrum:

S Set of frequency slots available in each link, index s.

u_{es} Equal to 1 if the slot s in fiber link e is free, 0 otherwise. To compute this parameter, only non-failed lightpaths are considered.

C Set of slots, index c. Each slot c contains a subset of contiguous slots.

l_{cs} Equal to 1 if slot c uses slot s, 0 otherwise.

b_c Bitrate capacity of slot c in Gb/s (see Table A-3).

Failed demands:

D Set of failed optical demands to be restored, index d.

b^d Bitrate requested by demand d in Gb/s.

$K(d)$ Subset of routes for demand d (includes reach constraint).

$C(d)$ Subset of feasible slots for demand d so to restore some amount of bitrate in the range $(0, b^d]$.

sQ Equal to 1 if squeezing is used, 0 otherwise.

mP Equal to 1 if multipath approach is used, 0 otherwise.

mk^d Maximum number of lightpaths that can be used to restore demand d, when multipath approach is selected.

The decision variables are:

x^{dk}_c Binary. Equal to 1 if demand d uses route k and slot c for restoration, 0 otherwise.

x^{kn}_{cp} Binary. Equal to 1 if restoration lightpath with route k and slot c uses MF-TP p in optical node n, 0 otherwise.

y^d Positive real. Restored bitrate for demand d.

y^k_c Positive real. Bitrate conveyed by restoration lightpath with route k and slot c.

y^{kn}_{cp} Positive real. Bitrate conveyed by restoration lightpath with route k and slot c using MF-TP p in optical node n.

w^d Binary. Equal to 1 if demand d is restored (total or partially), 0 otherwise.

z^d Positive integer accounting for the number of lightpaths used to restore demand d.

The ILP formulation for the BATIDO problem is as follows:

$$\text{(BATIDO)} \quad Max \quad \Phi = \sum_{d \in D} y^d \tag{8.1}$$

subject to:

$$\sum_{k \in K(d)} \sum_{c \in C(d)} x^{dk}_c \leq mP \cdot \left(mk^d - 1 \right) + 1 \quad \forall d \in D \tag{8.2}$$

$$\sum_{k \in K(d)} \sum_{c \in C(d)} x^{dk}_c = z^d \quad \forall d \in D \tag{8.3}$$

$$z^d \leq mk^d \cdot w^d \quad \forall d \in D \tag{8.4}$$

$$y^d \leq \sum_{k \in K(d)} \sum_{c \in C(d)} b_c \cdot x^{dk}_c \quad \forall d \in D \tag{8.5}$$

$$y^d \leq b^d \quad \forall d \in D \tag{8.6}$$

$$B \cdot sQ + y^d \geq b^d \cdot w^d \quad \forall d \in D \tag{8.7}$$

$$\sum_{d \in D} x^{dk}_c \leq \sum_{p \in P(n)} x^{kn}_{cp} \quad \forall k \in K, c \in C, n \in N(k) \tag{8.8}$$

$$\sum_{p \in P(n)} x^{kn}_{cp} \leq 1 \quad \forall k \in K, c \in C, n \in N(k) \tag{8.9}$$

$$\sum_{k \in K} \sum_{c \in C} x^{kn}_{cp} \leq fF^n_p \quad \forall n \in N, p \in P(n) \tag{8.10}$$

$$\sum_{d \in D} b_c \cdot x^{dk}_c \leq y^k_c \quad \forall k \in K, c \in C \tag{8.11}$$

$$y^k_c \leq b^k \quad \forall k \in K, c \in C \tag{8.12}$$

$$y_{cp}^{kn} \le b^k \cdot x_{cp}^{kn} \quad \forall n \in N, p \in P(n), k \in K(n), c \in C \tag{8.13}$$

$$y_{cp}^{kn} \le y_c^k \quad \forall n \in N, p \in P(n), k \in K(n), c \in C \tag{8.14}$$

$$y_{cp}^{kn} \ge y_c^k - b^k \cdot \left(1 - x_{cp}^{kn}\right) \quad \forall n \in N, p \in P(n), k \in K(n), c \in C \tag{8.15}$$

$$\sum_{k \in K} \sum_{c \in C} y_{cp}^{kn} \le b_p^n \quad \forall n \in N, p \in P(n) \tag{8.16}$$

$$\sum_{d \in D} \sum_{k \in K(d)} \sum_{c \in C(d)} h_e^k \cdot l_{cs} \cdot x_c^{dk} \le u_{es} \quad \forall e \in E, s \in S \tag{8.17}$$

Equation (8.1) maximizes the total restored bitrate Φ. Constraint (8.2) accounts the number of lightpaths assigned to each failed demand. Constraint (8.3) stores the number of lightpaths used to restore each demand, and constraint (8.4) limits that number to the maximum value allowed while keeping track of the restored demands. Constraint (8.5) accounts the restored bitrate of each demand, which is limited by demand's bitrate in constraint (8.6). In case bitrate squeezing is not allowed, constraint (8.7) ensures that all its bitrate is restored or blocked. Parameter B represents a large integer number that can be computed as the larger bitrate of all demands.

Constraints (8.8)-(8.17) deals with lightpath allocation. Constraints (8.8) and (8.9) assign one MF-TP at each end of the lightpath, whilst constraint (8.10) ensures that the available number of flows in each MF-TP is not exceeded. Constraint (8.11) accounts the bitrate conveyed by a lightpath, whereas constraint (8.12) makes sure that the maximum bitrate for the assigned route k (b^k) is not exceeded. Constraints (8.13)-(8.15) force $y^{kn}{}_{cp}$ to take the value of $x^{kn}{}_{cp} \cdot y^k{}_c$, and constraint (8.16) limits the total bitrate that can be allocated in each MF-TP. Constraint (8.17) guarantees that at most by one lightpath uses each slice in each link.

Regarding the complexity of the BATIDO problem, it is *NP*-hard since simpler network routing problems have been proved to be *NP*-hard (see e.g. [Ra95]). Regarding its size, the number of variables is $O(|D| \cdot |K| \cdot |C| + |N| \cdot |P| \cdot |K| \cdot |C|)$ and the number of constraints is $O(|D| + |N| \cdot |P| \cdot |K| \cdot |C| + |E| \cdot |S|)$. As an example, the number of variables and constraints for the TEL network (Fig. A-) and the scenario presented in next is 10^9 for both of them.

Although the ILP model can be solved for small instances, its exact solving becomes impractical for realistic backbone networks under appreciable load, even using commercial solvers [CPLEX]. Hence, aiming at providing near-optimal solutions for the BATIDO problem within the stringent required times (e.g., hundreds of milliseconds), we present next a heuristic algorithm to solve the problem.

8.1.3 Heuristic Algorithm

To cope with the required computation time constraints, we propose a very simple but efficient heuristic algorithm that generates a number of randomized solutions. The algorithm consists in performing a fixed number of iterations (*maxIterations*) in which one solution is built from a randomly sorted set of demands (lines 2-5 in Table 8-1). Next, k shortest routes are computed for each demand (line 6-7), which are afterwards sorted in decreasing slot width order (line 8). For each route, the larger available slot (considering both continuous free slices and available resources at end nodes) is selected and the restored bitrate is updated (lines 10-13).

In case that the demand can be totally restored, the corresponding lightpath is allocated on graph G to prevent that these resources are used by subsequent restoration lightpaths. The lightpath is finally added to the solution being built (lines 14-18) and the algorithm continues with the next demand. In case that the demand can be only partially restored, we need to consider two options. Firstly, if multipath is allowed, the lightpath is allocated, added to the solution, and new lightpaths are considered, provided that the number of lightpaths already used does not exceeded the given limit (lines 19-25). Secondly, if bitrate squeezing is allowed, the lightpath is allocated and added to the solution (lines 26-30). Otherwise, the demand cannot be restored and it is blocked (lines 31-32).

Once a solution is built, its total restored bitrate is compared against the best solution obtained so far, which it is updated as long as Φ is improved (lines 33-35). Finally, before building new solutions, all the allocated resources are released from graph G (line 36). The best solution is eventually returned (line 37).

The proposed heuristic performs a constant number of iterations (*maxIterations*); on each iteration one permutation, as well as one k-shortest paths computation, based on the Yen's algorithm [Ye70], and one sorting for each demand, are performed. Therefore, the time complexity of the heuristic is polynomial and can be expressed as the $O(maxIterations \cdot |D| \cdot |K(d)| \cdot |N| \cdot (|E| + |N| \cdot \log(|N|)))$.

The performance of the proposed heuristic was compared against the optimal solution obtained from solving the ILP model for small instances. In all the tests performed, the optimal solution was found within running times of few milliseconds, in contrast to just above 1 hour needed to find the optimal solution with CPLEX. Consequently, we use the proposed algorithm to solve the instances presented in next section.

Table 8-1 Heuristic Algorithm pseudo-code

```
IN: N, E, D, mP, sQ, maxIterations
OUT: Φ, Sol
 1:  Φ ← 0
 2:  Sol ← ∅
 3:  for i = 1..maxIterations do
 4:    tempΦ ← 0
 5:    tempSol ← ∅
 6:    randomSort(D)
 7:    for each d ∈ D do
 8:      Kd ← kShortestRoutes(N, E, d)
 9:      sort(Kd, DEC_SLOT_WIDTH)
10:      z^d ← 0
11:      tempB ← 0
12:      for each k ∈ Kd do
13:        z^d ++
14:        c ← getLargerSlot(k, b^d)
15:        tempB ← tempB + getBitrate(c)
16:        if tempB ≥ b^d then
17:          allocate(k, c)
18:          tempΦ ← tempΦ + b^d
19:          tempSol ← tempSol ∪ {d, k, c}
20:          break
21:        else
22:          if mP then
23:            allocate(k, c)
24:            tempΦ ← tempΦ + tempB
25:            tempSol ← tempSol ∪ {d, k, c}
26:            if z^d = mk^d then
27:              break
28:          else if sQ then
29:            allocate(k, c)
30:            tempΦ ← tempΦ + tempB
31:            tempSol ← tempSol U {d, k, c}
32:            break
33:          else
34:            break
35:    if tempΦ > Φ then
36:      Φ ← tempΦ
37:      Sol ← tempSol
38:    resetAllocations(tempSol)
39:  return {Φ, Sol}
```

8.1.4 BATIDO Numerical Results

In this section, we focus on studying the performance of the restoration schemes. To that end, we consider the 30-node and 56-link TEL (Fig. A-), and the 22-node and 35-link BT (Fig. A-4) topologies; each node location is equipped with one single BV-OXC.

For evaluation proposes, we developed an ad-hoc event-driven simulator in OMNet++ [OMNet]. A dynamic network environment was simulated where

incoming connection requests arrive to the system following a Poisson process and are sequentially served without prior knowledge of future incoming connection requests. To compute the RSA of the lightpaths, we used the algorithm described in Chapter 6. The holding time of the connection requests is exponentially distributed with a mean value equal to 2 hours. Source/destination pairs are randomly chosen with equal probability (uniform distribution) among all nodes. Different values of the offered network load are created by changing the inter-arrival rate while keeping the mean holding time constant. We assume that no retrial is performed; if a request cannot be served, it is immediately blocked. Regarding the optical spectrum, the total width was fixed to 4 THz and the slot width to 6.25 GHz.

Besides incoming connection requests, optical link failures follow a Poisson process with a MTTF equal to 50 hours, and link failures are randomly chosen with equal probability. We consider that the link is repaired immediately after the restoration process has ended.

In our experiments, the bitrate of each connection request was selected considering 80% of the connections being 100 Gb/s and the other 20% of 400Gb/s. To convert bitrate into spectrum width, we use the correspondence in Table A-3. Finally, note that each point in the results is the average of 10 runs with 150,000 connection requests each.

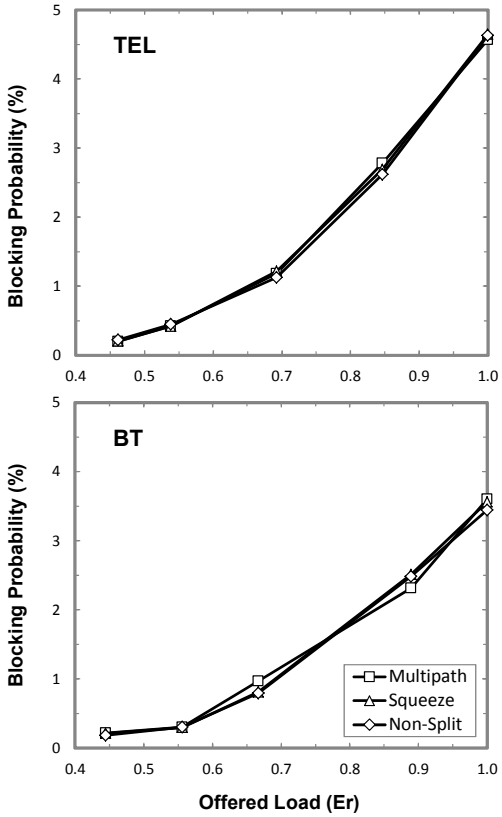

Fig. 8-2 Blocking probability against offered load.

Fig. 8-2 plots blocking probability as a function of the offered load for the restoration approaches and network topologies considered. Note that offered loads have been limited to those unleashing blocking probability in the meaningful range [0.1%-5%]. For the sake of comparability between topologies, offered loads have been normalized to the value of the highest load.

As shown, all three approaches behave similarly, i.e. whatever the restoration approach is selected the blocking probability for provisioning remains unchanged. Note that the considered MTTF value is larger enough to reduce the probability of two lightpaths being affected by two consecutive failures, and hence virtually all the restored lightpaths have been torn down when a new failure occurs.

When we analyze the results for aggregated restorability, i.e. combined values for 100Gb/s and 400Gb/s lightpaths, (Fig. 8-3), we observe that multipath and bitrate squeezing approaches restore almost all the failed bitrate, in contrast to the non-split one. Certainly, aggregated restorability using multipath and bitrate squeezing is better than 95%, even for the highest considered loads, remarkably higher than that obtained using the non-split approach, which values range in the noticeable poor interval [80%-60%].

To get insight into the performance of the different approaches, Fig. 8-4 focuses restorability for 400 Gb/s lightpaths. The performance of multipath and bitrate

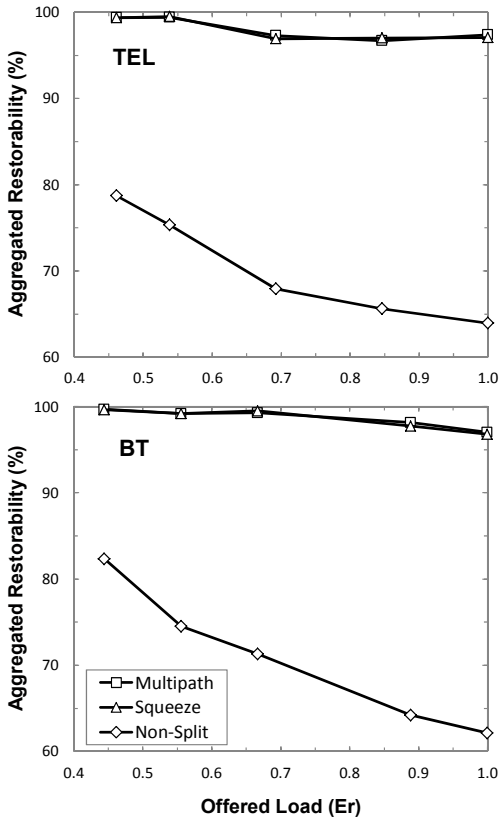

Fig. 8-3 Restorability against offered load.

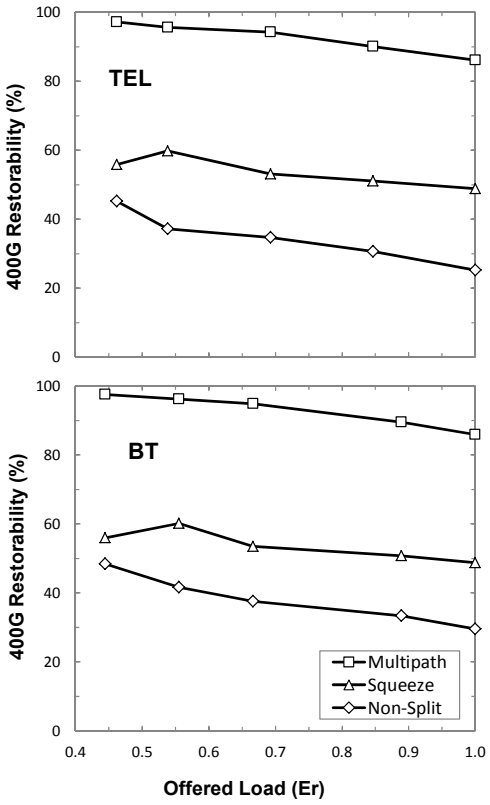

Fig. 8-4 400Gb/s connections restorability against offered load.

squeezing is noticeably divergent: the multipath approach shows significantly better performance than that of the bitrate squeezing. The rationale behind that behavior is that the multipath approach includes the bitrate squeezing one and additionally adds the ability to use several lightpaths to restore one single demand.

To appreciate the way the multipath approach works, let us evaluate the distribution of lightpaths actually used for restoration. Note that since 400 Gb/s demands can be restored using any combination of 400Gb/s, 200Gb/s, and 100Gb/s lightpaths. Thus, 1, 2, 3 or 4 different lightpaths can be used by the restoration approach. Then, Fig. 8-5 depicts number of demands restored using z^d lightpaths and its average value as a function of the offered load.

Fig. 8-5 shows an upwards trend of z^d average that clearly is as a consequence of under heavier loads the probability of finding a single lightpath with enough spectral resources for each demand decreases. That can be clearly observed analyzing the distribution of z^d values; the number of demands restored using one single lightpath decreases as the offered load increases, whereas the demands restored using more than one lightpath increases with the load.

Consequently, as the offered load grows the multipath approach takes advantage from using multiple lightpaths to maximize the total restored bitrate.

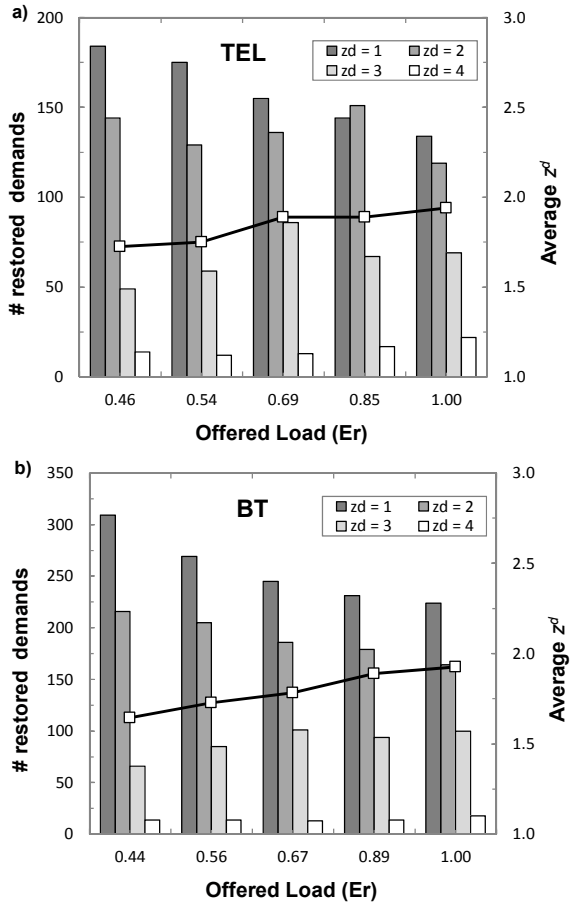

Fig. 8-5 Distribution and average z^d values for restored 400Gb/s demands.

8.2 Conclusions

An effective restoration scheme enabling multipath recovery and bitrate squeezing in elastic optical networks implementing the flexgrid optical technology was proposed. The restoration scheme exploits the advanced flexible capabilities provided by MF-TPs, which support the adaptation of connection parameters in terms of number of sub-carriers, bitrate, transmission parameters and reserved spectrum resources.

To efficiently recover network failures by exploiting limited portions of spectrum resources along multiple routes, the BATIDO problem was stated and an ILP model and heuristic algorithm were proposed.

Illustrative results obtained by simulation showed that the proposed recovery scheme cans even double the percentage of restored bitrate with respect to single path restoration, where no squeezing or multipath are exploited.

Due to networks dynamics coming from provisioning or even from restoration, network resources could be inefficiently used. To cope with this fact, in the next chapter we introduce re-optimization algorithms to reallocate connections so as to improve network performance.

Chapter 9

Re-optimization: Provisioning-triggered Spectrum Defragmentation

In this chapter we address the spectrum defragmentation problem identified in Chapter 6 in the context of flexgrid optical networks. We propose an algorithm triggered before an incoming connection is blocked to reallocate already established optical connections. Exhaustive simulation results reveal that the proposed algorithm improves network GoS.

In a step forward, the defragmentation problem is extended to deal with time-varying traffic. To cope with the latter, the SPRESSO algorithm is enhanced, and the EL-SPRESSO algorithm is proposed. Simulation results to evaluate the performance of EL-SPRESSO are eventually presented.

9.1 Spectrum Reallocation: The SPRESSO Problem

In Chapter 6 we identified that the optical spectrum might be fragmented as a consequence of connections dynamicity and of the different amounts of contiguous frequency slots required for the connections, leading to high blocking probability. Aiming at improving the GoS of flexgrid networks, i.e. reducing blocking probability, we propose a mechanism called *SPectrum REaLLOcation* (SPRE(LLO)→(SSO)). SPRESSO reallocates already established connections in the spectrum without rerouting them, so as to make enough room for incoming connection requests.

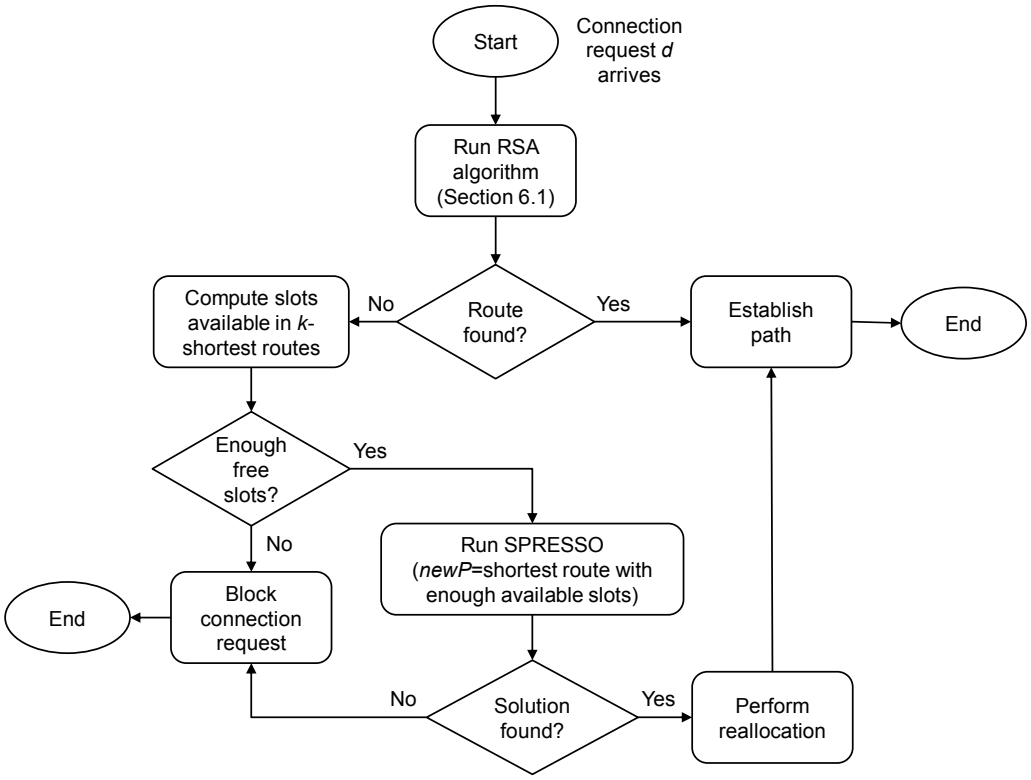

Fig. 9-1 Algorithm for routing and spectrum (re)allocation.

The SPRESSO algorithm follows a path-triggered scheme for spectrum defragmentation in flexgrid optical networks, where the algorithm is triggered whenever not enough resources have been found for an incoming connection request. Every link in the routes from a set of k-shortest routes connecting source and destination nodes is checked to know whether the amount of available frequency slots is equal to or higher than the required amount for the incoming connection request. If enough frequency slots are available in one of the shortest routes, the SPRESSO mechanism is run to find a set of already established path reallocations so to make enough room for the connection request in the selected route (*newP*). Otherwise, the connection request is blocked. Fig. 9-1 summarizes the routing and spectrum (re)allocation algorithm.

Fig. 9-2 illustrates spectrum fragmentation on a test network consisting in 9 nodes and 11 links (Fig. 9-2a). The entire spectrum width corresponds to 16 slots. Fig. 9-2b represents the utilization of each frequency slot in the network, where a number of paths are already established. In this scenario, the connection request between nodes 4 and 7 requesting 4 slots cannot be served. Notwithstanding, each link in the shortest route 4-5-6-7 (links 4-5-6) has at least 4 free slots, and then the request could be established reallocating some of the established paths. In Fig. 9-2c, paths *p4* and *p5* are reallocated making enough room for the new optical connection *newP*.

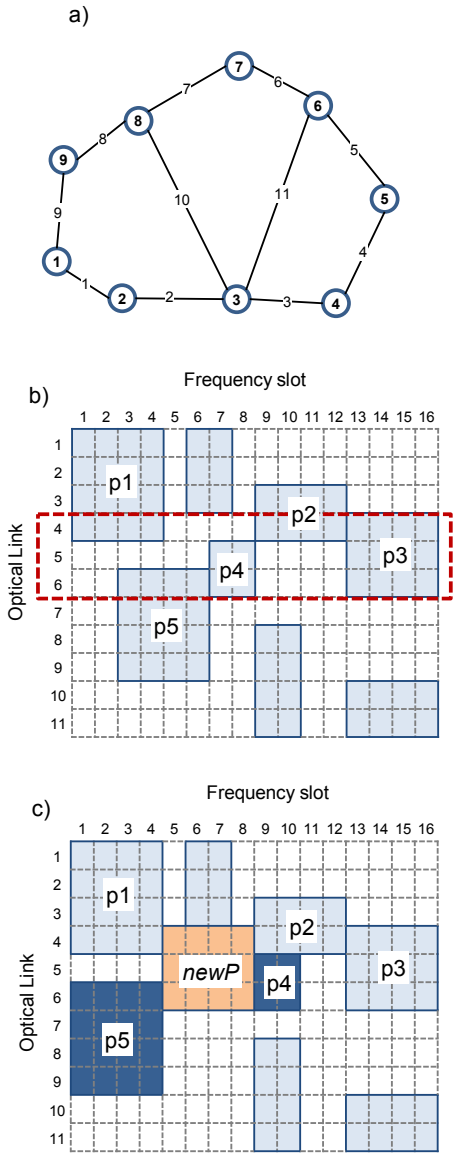

Fig. 9-2 Example of spectrum reallocation.

Aiming at reducing path set-up delays, we impose a number of constraints to the above approach:

i. We use k=1, i.e., only the links in the shortest path of the connection request are checked for slots availability;

ii. Already established paths can be reallocated in the spectrum but their routes are not modified (rerouting is not used);

iii. To reduce the size of the SPRESSO problem, only those paths using any link in the route of *newP* are considered as candidate to be reallocated, preventing thus any other paths to be reallocated;

iv. To avoid complex solutions involving a large number of reallocations, the length of any SPRESSO solution is limited to a given maximum value, i.e. problem instances which optimal solutions involve higher number of reallocations are considered unfeasible.

The problem can be formally stated as follows:

Given:

- an optical network, represented by a graph G (*N*, *E*), being *N* the set of nodes and *E* the set of optical links connecting two nodes,
- a set *S* of frequency slots available in each link $e \epsilon E$,
- a set *P* of already established paths,
- a new path (*newP*) to be established in the network. A route for the path has been already selected but there is no feasible spectrum allocation,
- the threshold number of paths to be reallocated.

Output:

- for each path to be reallocated, its new spectrum allocation,
- the spectrum allocation for *newP*.

Objective: Minimize the amount of paths to be reallocated so to fit *newP* in.

An ILP model for the SPRESSO problem is presented next. The model takes as input the set of already established paths that share common links with *newP*. The following sets and parameters are defined:

E	Set of optical links, index *e*.
P	Set of already established paths, index *p*.
E(*p*)	Subset of *E* with those links in the route of path *p*.
P(*e*)	Subset of *P* with those paths using optical link *e*.
P(*p*)	Subset of *P* with those paths sharing at least one link with path *p*. $P(p) = \cup_{e \epsilon E(p)} P(e)$
Pm	Subset of *P* with the candidate paths, including *newP*. $Pm = P(newP) \cup \{newP\}$.
S	Set of frequency slots, index *s*.
C	Set of channels, index c. Each channel c contains a subset of contiguous slots.
C(*p*)	Subset of *C* with those channels that can be assigned to path *p*. Note that each path *p* and its originating demand *d* are associated. Thus $C(p) \equiv C(d)$.
S(*p*)	Subset of *S* with those frequency slots allocated for path *p*.

δ_{cs} 1 if channel c uses slot s, 0 otherwise.

ω_{pc} 1 if path p was using channel c, 0 otherwise.

η_{es} 1 if slot s in optical link e is free, 0 otherwise.

$\eta_{es}(p,c)$ If reallocation path p to channel c is feasible, it returns η_{es} after reallocating path p to channel c, otherwise it returns 0.

$maxR$ The threshold number of paths to be reallocated.

Additionally, the decision variables are:

x_{pc} Binary, 1 if channel c is assigned to path p, 0 otherwise.

y_p Binary, 1 if path p is reallocated, 0 otherwise.

r Integer with the number of already established paths to reallocate.

Then, the ILP model for the SPRESSO problem is as follows:

$$(\text{SPRESSO})\,\text{minimize}\,r \tag{9.1}$$

subject to:

$$r = \sum_{p \in Pm \setminus \{\text{newP}\}} y_p \tag{9.2}$$

$$\sum_{c \in C(p)} x_{pc} = 1 \;\forall p \in Pm \tag{9.3}$$

$$x_{pc} - \omega_{pc} \le y_p \;\forall p \in Pm, c \in C(p) \tag{9.4}$$

$$\sum_{p \in P(e)} \sum_{c \in C(p)} \delta_{cs} \cdot x_{pc} \le \eta_{es} \quad \forall e \in E, \quad s \in S \tag{9.5}$$

$$r \le \text{maxR} \tag{9.6}$$

The objective function (9.1) minimizes the number of paths that need to be reallocated in the spectrum so *newP* can be allocated. Constraint (9.2) counts the number of paths that need to be reallocated. Note that only *newP* and those paths sharing at least one common optical link with *newP* are in the set of candidate paths. Constraint (9.3) assigns one channel to every candidate path. Constraint (9.4) stores whether path p is reallocated. Constraint (9.5) guarantees that each frequency slot in a given optical link is assigned to one path at most. Finally, constraint (9.6) limits the number of paths to be reallocated.

Although the size of the problem is limited –the number of variables and constraints is $O(|Pm| \cdot |C|)$ and $O(|Pm| \cdot |C| + |E| \cdot |S|)$, respectively– it must be solved in real time (in the order of milliseconds) to minimize the optical connections set-up delay. In our experiments over the networks described in section 6.1.2, the

problem takes several seconds on average to be solved; more than 1 minute in the worst case. As a consequence, in the next section we propose a heuristic algorithm that provides near-optimal solutions in practical computations times, short enough to be used in the management plane of real flexgrid optical networks.

9.1.1 GRASP+PR-Based Heuristic Algorithm

In this section we propose a GRASP+PR-based heuristic. In our implementation, a multi-greedy heuristic consisting in three different constructive algorithms is used to provide diversification. A multi-start local search procedure [Fe95] is used to find local optimal solutions. An *ES* storing the best solutions found, is used for the PR procedure. *ES* is initially populated with three GRASP solutions resulting of applying the three greedy constructive algorithms plus the corresponding local search. Each sub-sequent iteration of the GRASP algorithm produces a new feasible solution using one randomly-selected greedy constructive algorithm.

Table 9-1 reproduces the GRASP+PR algorithm for the SPRESSO problem, where a set of values in the interval [0,1], named *GreedyProportions*, is used to run the required proportion of each constructive algorithm.

To cope with the stringent computation time requirement, it is essential to produce as diverse GRASP solutions as possible with the purpose of reducing the amount of iterations, and thus the computation time needed to find high-quality near-optimal solutions. In view of that, we have developed three different greedy randomized constructive algorithms, each building really different feasible solutions.

Table 9-2 describes the greedy randomized constructive Algorithm-1. It consists in reallocating those already established paths whose route intersects that of *newP* so to generate enough room for the latter. Equation (9.7) is used to quantify the quality of reallocating a given path *i*; it explores every feasible spectrum

Table 9-1 GRASP+PR heuristic algorithm for the SPRESSO problem

```
INPUT newP, α, β, maxR, GreedyProportions
OUTPUT Sol
 1:  Incumbent ← Ø
 2:  ES ← Ø
 3:  for iter=1 ... Iter do
 4:      r ← getRandom()
 5:      if r < GreedyProportions.alg1 then
 6:          Sol ← Algorithm-1(newP, α, maxR)
 7:      else if r < GreedyProportions.alg2 then
 8:          Sol ← Algorithm-2(newP, α, maxR)
 9:      else
10:          Sol ← Algorithm-3(newP, α, β, maxR)
11:      Sol ← LocalSearch(Sol)
12:      {Sol, ES} ← PathRelinking(Sol, ES)
13:      if |Sol| < |Incumbent| then
14:          Incumbent ← Sol
15:  return Incumbent
```

Table 9-2 Greedy Randomized Constructive Algorithm-1

```
INPUT newP, α, maxR
OUTPUT Sol
 1:  Sol ← Ø
 2:  Initialize the candidate set: Q ← P(newP)
 3:  spaceNeeded ← |E(newP)| · |S(newP)|
 4:  while Q ≠ Ø and |Sol| ≤ maxR do
 5:      Evaluate the quality q(i) for all i∈Q using eq. (9.7)
 6:      q^min ← min{q(i) : i∈Q, i feasible}
 7:      q^max ← max{q(i) : i∈Q, i feasible}
 8:      RCL ← {i∈Q : q(i) ≥ q^max-α(q^max - q^min)}
 9:      if RCL = Ø then
10:         Sol ← Ø
11:         break
12:     if q^max ≥ spaceNeeded then
13:         i ← arg max (RCL)
14:         Sol ← Sol ∪ {i}
15:         Perform the spectrum reallocation defined by element i
16:         break
17:     Select an element i from RCL at random
18:     Q ← Q \ {i}
19:     Sol ← Sol ∪ {i}
20:     Perform the spectrum reallocation defined by element i
21: if |Sol| ≤ maxR then
22:     return Sol
23: return Ø
```

reallocation for path i and accounts the greatest amount of contiguous space generated in the specific sets of optical links and channels for *newP*. An RCL containing those paths with the best quality is used. Parameter α in the real interval [0, 1] determines the size of RCL.

$$q(i) = \max_{c1 \in C(i)} \max_{c2 \in C(newP)} \sum_{e \in E(newP)} \sum_{s \in S(c2)} \eta_{es}(i, c_1) \tag{9.7}$$

As stated before, Algorithm-1 builds solutions by successively reallocating already established paths. Note that this is in contrast with removing all paths intersecting *newP* and reallocating them next so to minimize the amount of paths effectively reallocated, which is the approach adopted to solve the *SPRESSO* ILP model previously introduced.

To avoid up to some extent blocking among reallocations that the sequentiality of the algorithm might cause, we allow paths to be reallocated multiple times. Note however, that aiming at minimizing disruption we could take advantage from sequentiality to implement the make-before-break mechanism.

However, special attention must be paid when paths are allowed to be reallocated multiple times to avoid cycling, i.e. repeating the same sequence of reallocations periodically. In this regard, in addition to limit the length of the solutions as previously stated, two measures have been adopted: firstly, a single path cannot be reallocated in two consecutive iterations. This is implemented by storing the path

Table 9-3 Greedy Randomized Constructive Algorithm-2

```
INPUT newP, α, maxR
OUTPUT Sol
 1:  Sol ← Ø
 2:  Initialize the candidate set: Q ← P(newP)
 3:  for each p∈Q do p.lastChannel ← p.channel
 4:  spaceNeeded ← |E(newP)| · |S(newP)|
 5:  lastRealloc ← Ø
 6:  while |Sol| ≤ maxR do
 7:     Evaluate the quality q(i) for all i∈Q using eq. (9.8)
 8:     q^min ← min{q(i,i.lastChannel): i∈Q, i feasible ∩ i ≠ lastRealloc}
 9:     q^max ← max{q(i,i.lastChannel): i∈Q, i feasible ∩ i ≠ lastRealloc}
10:     RCL ← {i∈Q : q(i) ≥ q^max - α(q^max - q^min)}
11:     if RCL = Ø then
12:        Sol ← Ø
13:        break
14:     if q^max ≥ spaceNeeded then
15:        i ← arg max (RCL)
16:        Sol ← Sol ∪ {i}
17:        break
18:     Select an element i from RCL at random
19:     Sol ← Sol ∪ {i}
20:     lastRealloc ← i
21:     i.lastChannel ← i.channel
22:     Perform the spectrum reallocation defined by element i
23:  if |Sol| ≤ maxR then
24:  return Sol
25:  return Ø
26:
```

that has been reallocated in the current iteration so that it remains in the candidate list Q but it can be added to RCL in the next iteration; Secondly, any path is prevented from being reallocated back to its previous channel, and then the last channel allocated to each path needs to be stored.

Moreover, quality quantification has been slightly modified as shown in eq. (9.8) to avoid a given channel c to be considered as a feasible reallocation option for path i. Table 9-3 shows the so updated constructive algorithm.

$$q(i,c) = \max_{\substack{c1 \in C(i) \\ c1 \neq c}} \max_{c2 \in C(newP)} \sum_{e \in E(newP)} \sum_{s \in S(c2)} \eta_{es}(i,c_1) \tag{9.8}$$

Finally, the more complex constructive Algorithm-3 illustrated in Table 9-4, exploits a totally different method; path reallocation is confined to incremental sizes of spectrum, starting from the amount of slots required by *newP*. For a given number of slots n, eq. (9.9) is used to initialize candidate set $Q_1(n)$ with the sets of already established paths which route intersects that of *newP* and are using, at least in part, the area of the spectrum considered.

$$Q_1(n) = \left\{ \left\{ p \in P(newP) : S(c) \cap S(p) \neq \emptyset \right\}, \forall c \in C : |S(c)| = n \right\} \tag{9.9}$$

Table 9-4 Greedy Randomized Constructive Algorithm-3

```
INPUT newP, α, β, maxR
OUTPUT Sol
 1:  Sol ← Ø
 2:  n← |S(newP)|
 3:  while n≤ |S| do
 4:     Initialize the candidate set Q1(n) as defined in eq. (9.9)
 5:     Evaluate the quality q(i)=|i| for all i∈Q1(n)
 6:     while Q1(n) ≠ Ø do
 7:        q1^min ← min{q(i) : i∈Q1(n)}
 8:        q1^max ← max{q(i) : i∈Q1(n)}
 9:        RCL1 ← {i∈Q1(n) : q(i) ≤ q1^min + α(q1^max - q1^min)}
10:        Select an element i from RCL1 at random
11:        Q1(n) ← Q1(n) \ {i}
12:        for each p∈ i do
13:          Release network resources used by p
14:        Initialize the candidate set Q2← i ∪ {newP}
15:        while Q2 ≠ Ø and |Sol| ≤ maxR do
16:            Evaluate the quality q(p)=|E(p)|·|S(p)| for all p∈Q2
17:            if ∃ p∈Q2 : p not feasible then
18:               Sol ← Ø
19:               break
20:            q2^min ← min{q(p) : p∈Q2}
21:            q2^max ← max{q(p) : p∈Q2}
22:            RCL2 ← {p∈Q2 : q(p) ≥ q2^max - β(q2^max - q2^min)}
23:            Select an element p from RCL2 at random
24:            Q2 ← Q2 \ {p}
25:            if ∃ c∈C(p) : ηes=1 ∀e∈E(p), s∈S(c) then
26:               Sol ← Sol ∪ {p}
27:               Perform the spectrum reallocation
28:            else
29:               Sol ← Ø
30:               break
31:        if |Sol| >maxR then return Ø
32:        if Sol ≠ Ø then return Sol
33:        Restore network resources
34:     n++
35:  return Ø
```

Algorithm-3 uses two nested RCLs. An outer RCL is built from $Q_1(n)$ using the size of each element to quantify its quality. After selecting at random one element i from *RCL1*, every path in that element is removed from the network and its allocated resources are released. Next, a second candidate set is initialized with those paths in i plus *newP*. All these paths must be reallocated, and the inner *RCL2* is used to find a reallocation order; the required spectrum space is used as criterion in this case. If some paths cannot be reallocated, another element i is selected from *RCL1* until $Q_1(n)$ is empty, when a new iteration of the algorithm starts and the size of the spectrum area is increased.

Due to the fact that a feasible solution *Sol* resulting from the constructive phase has no guarantee of being locally optimal, GRASP heuristics apply a local search

procedure starting at *Sol* in the hope of finding a better solution in its neighborhood.

In our problem, a feasible solution is defined as the set of paths to be reallocated and its reallocation position so to make enough room for *newP*. To avoid the sequentiality problem inherent to the constructive algorithms, all paths with multiple reallocations in *Sol* are removed. Next, every path is checked to be moved back to its original allocation, reducing thus the size of the solution and resulting in a solution *x*.

As introduced above, a multi-start local search procedure to find the optimal solution in $N(x)$ is started. $N(x)$ contains those feasible solutions that can be reached from *x* by one move, defined as an exchange between a path p_2 not in *x* and another path p_1 in *x*, such that after doing it some other path *p* in *x* can be moved back to its original allocation. The local search procedure ends when no feasible movement can be done in *x*.

Finally, regarding the PR phase, we apply the ES management criteria introduced in Chapter 2.

9.1.2 Performance Evaluation

9.1.2.1 Heuristics Tuning

Aiming at analyzing the performance of the GRASP+PR heuristic proposed to solve SPRESSO compared to the ILP model, 2400 problem instances were saved from the simulator (one hundred instances for the 21-node TEL, the 20-node BT, and the 21-node DT topologies, for each slot width, and for that network load unleashing Pb_{bw} equal to 1% and 5% under TP-6 (see Appendix A)).

To find the optimal value of parameter α, the three different greedy randomized constructive algorithms described in section 9.1.1 were compared. Each of the heuristics was followed by the local search phase; path-relinking was disabled in these tests. The number of iterations was limited to 30. A subset of 1000 different instances was used and each heuristic was run 10 times. The next table shows the on average number of movements in the solution. As shown in Table 9-5, the

Table 9-5 On-average number of reallocations as a function of α

α	Algorithm 1	Algorithm 2	Algorithm 3	Multi-greedy
0.1	7.43	6.78	6.83	6.28
0.3	7.88	6.99	7.04	6.01
0.5	6.99	7.36	7.04	6.12
0.7	7.48	7.64	7.04	6.76

optimal value of α ranges from 0.5 for greedy algorithm 1 to 0.1 for algorithms 2 and 3. We assumed $\beta=\alpha$ in Algorithm-3.

Next, the proposed multi-greedy heuristic with path relinking was run for the previous set of problem instances using the same value of α for all the greedy algorithms. As shown in Table 9-5, the best solutions were obtained when α=0.3 was used. It is worth noticing that when PR is enabled, the proposed multi-greedy algorithm provides much better solutions than each of the greedy algorithms alone; this is the consequence of increasing the diversity of the solutions.

9.1.2.2 Heuristics vs ILP Model

Finally, the above problem instances were solved using the ILP model implemented in iLog-OPL and using the CPLEX v.12 optimizer [CPLEX] on a 2.4 GHz Quad-Core machine with 4 GB RAM memory running Linux. Table 9-6 shows the on-average results.

As shown, the proposed heuristic provides solutions only 5% over the optimal values on average in times in the order of one hundred milliseconds and thus it can be used for real time applications. To reach these shorter solving times, the number of iterations of the heuristic algorithm was limited to only 10, which illustrates the effectiveness of the proposed constructive algorithms in building diversification to the GRASP+PR-based heuristic.

Table 9-6 ILP vs. Heuristic Performance Comparison

Pb_{bw}	Number of realloc.	Heuristic vs. ILP performance gap	Average ILP time (s)	Max. ILP time (s)	Average heuristic time (s)	Max. heuristic time (s)
1%	5.9	3.7%	28	72	0.098	0.145
5%	6.1	4.6%	33	87	0.137	0.178

9.1.2.3 Performance of the SPRESSO algorithm

After evaluating the proposed heuristic algorithm, we integrated it into the general algorithm for routing and spectrum (re)allocation depicted in Fig. 9-1. To evaluate the performance of the SPRESSO algorithm, we use the network topologies and traffic profiles defined in the Appendix A.

Fig. 9-3 shows the performance of the three network topologies under traffic profiles TP-2, TP-4, and TP-6. To compare the performance of SPRESSO, plots obtained for provisioning in section 6.1 are also reproduces here. We observe that remarkable gains are achieved when SPRESSO algorithm is applied.

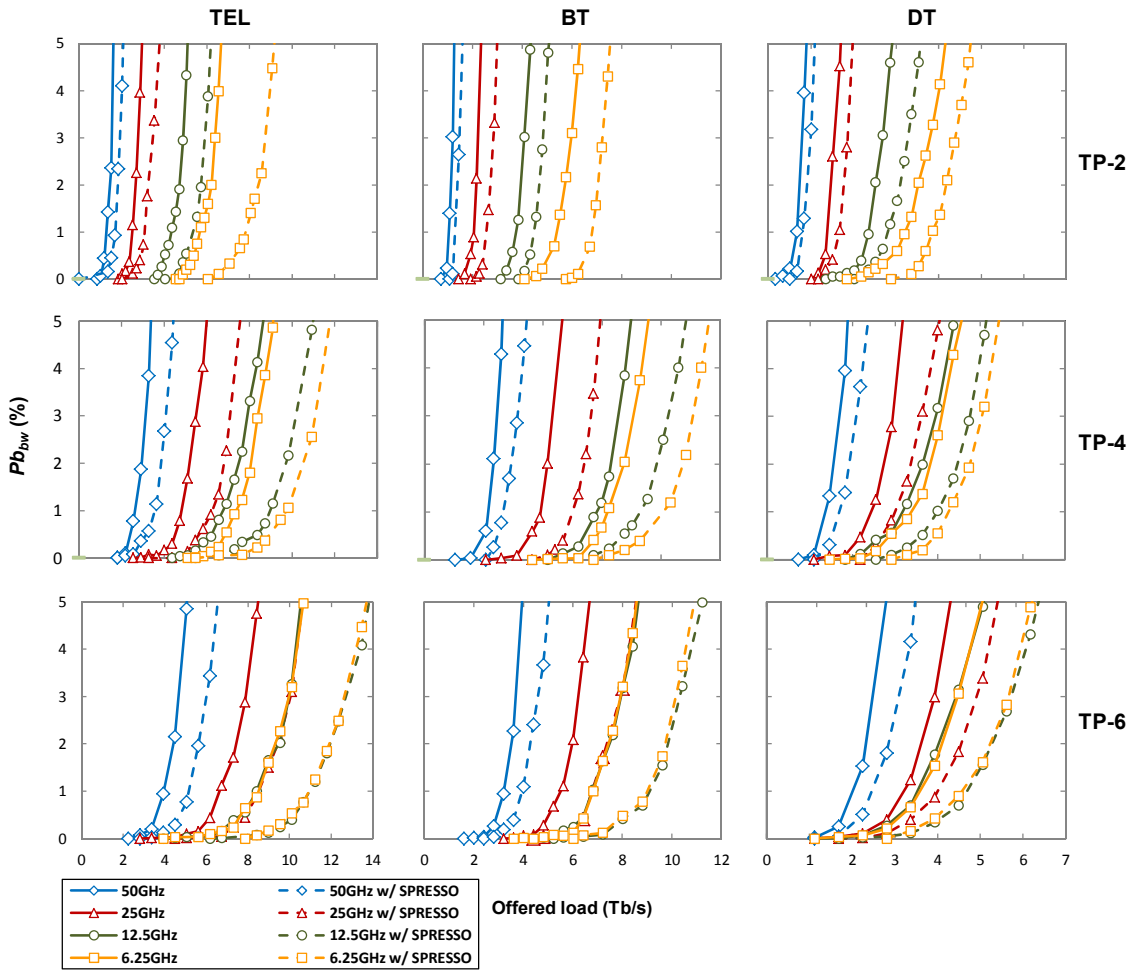

Fig. 9-3 Blocking vs. load for different slot widths and TPs.

Table 9-7 summarizes the gain of using SPRESSO for each network topology, traffic profile, and slot width. Presented values are for Pb_{bw}=1%. Interestingly, spectrum reallocation provides only a gain in the order of 10% for all the network topologies under TP-1 when slots of 50GHz are used; recall that in such a scenario, all connections request only one slot and then spectrum fragmentation is as a consequence of the spectrum continuity constraint (similarly to DWDM). As soon as the slot width is decreased, the diversity in the number of contiguous slots requested increases and, since also slot contiguity is required, the spectrum fragmentation increases. In these scenarios, especially in TPs 2-6 where the amount of requests for 400Gb/s is important, SPRESSO provides improvements in the range between 20% and 31%. These results verify the goodness of our proposed mechanism.

Table 9-7 Gain of using SPRESSO (Pb_{bw}=1%)

	Network	Average TP	50 GHz	25 GHz	12.5 GHz	6.25 GHz
	TEL	21.6%	10.7%	17.8%	28.2%	29.7%
TP-1	**BT**	16.3%	12.4%	11.3%	20.8%	20.6%
	DT	13.6%	6.7%	16.8%	15.9%	15.0%
	TEL	29.4%	32.8%	23.3%	23.4%	38.1%
TP-2	**BT**	22.9%	23.6%	22.5%	19.1%	26.3%
	DT	20.0%	19.7%	20.6%	19.8%	19.9%
	TEL	30.9%	33.3%	29.6%	29.5%	31.2%
TP-3	**BT**	23.6%	24.0%	26.1%	19.2%	25.1%
	DT	25.3%	24.9%	24.0%	29.0%	23.3%
	TEL	31.7%	35.5%	26.7%	32.7%	32.1%
TP-4	**BT**	26.1%	24.0%	24.8%	25.3%	30.3%
	DT	24.6%	24.0%	23.1%	25.6%	25.8%
	TEL	28.0%	32.2%	28.6%	25.0%	26.3%
TP-5	**BT**	29.2%	25.5%	27.3%	35.3%	28.5%
	DT	25.2%	24.1%	25.1%	24.1%	27.4%
	TEL	29.4%	30.4%	28.4%	30.3%	28.5%
TP-6	**BT**	28.0%	22.9%	23.0%	34.1%	32.1%
	DT	26.9%	22.2%	24.8%	33.0%	27.4%

9.2 Time-varying Traffic with defragmentation: The EL-SPRESSO Problem

As already stated, one of the main characteristics of flexgrid networks is their inherent capacity to allocate spectrum to connections so as to transport the desired bitrate. When flexgrid networks are provided with a control plane, spectrum allocations can be dynamically modified to adapt to traffic variations [As13]; we call those as elastic spectrum operations. Then, when the traffic conveyed by a connection increases, an elastic operation can be performed to increase the spectrum allocated to it. However, some of the required slots might be allocated to other connections at that time, which would prevent the operation to be eventually implemented and some traffic rejected. A solution to the above problem is to perform defragmentation on neighboring connections to allow the elastic operation to be performed.

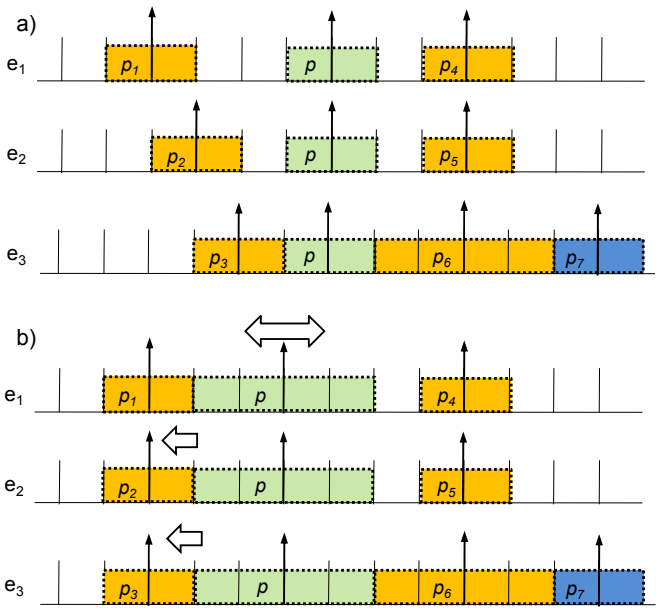

Fig. 9-4 Example of elastic provisioning with reallocation.

For illustrative purposes, Fig. 9-4 shows a scenario with a number of already established connections. An elastic operation is requested to increase the spectrum allocation of connection p in two slots. As shown in Fig. 9-4a, the request cannot be fulfilled since other established connections in the route of p are currently using neighboring slots. Note, however, that enough free spectrum to fulfill p requirement exist; the central frequency of some connections can be shifted thus, releasing frequency slots that can be used to increase p spectral allocation. For instance, the central frequency of paths p_2 and p_3 can to be shifted to make enough room for p elastic request (Fig. 9-4b). Once paths p_2 and p_3 have been reallocated, the spectrum allocated to p can be increased.

We can extend further the SPRESSO algorithm previously defined, to be run not only to make enough room for new connection requests, but also when elastic operations over already established connections are requested. To that end, we enhance the SPRESSO algorithm adding elastic capabilities. Therefore, in this section we are facing the problem of elastic provisioning with spectrum reallocation (EL-SPRESSO).

9.2.1 Problem statement and algorithm

The EL-SPRESSO problem can be formally stated as:

Given:

- a core network topology represented by a graph G(N, E), being N the set of optical nodes and E the set of bidirectional fiber links connecting two nodes,

- a set S of available slots of a given spectral width for every link in E,
- a set L of LSPs already established on the network; each LSP l is defined by the tuple $\{R_l, fc_l, m_l\}$, where the ordered set $R_l \subseteq E$ represents its physical route, fc_l its central frequency and m_l the amount of slots,
- a LSP $p \in L$ for which a spectrum adaptation request has arrived to increase the spectrum allocation to number of slots, m_p.

Output:

- the list of LSPs to be reallocated and the new values for the spectrum allocation for the given LSP p: $\{R_p, fc_p, m_p\}$.

Objective: minimize the number of reallocations to be performed.

Table 9-8 describes the EL-SPRESSO algorithm, which combines in a single algorithm the SPRESSO algorithm and the algorithms described in [As13] for elastic provisioning, to perform elastic provisioning with reallocation. It is worth noting that, as SPRESSO, the EL-SPRESSO algorithm aims to reallocate optical resources, while keeping the paths' route unchanged (i.e., no rerouting is performed). Moreover, even in the case that a path releases all its optical resources, its route is preserved. Eventually, when this path requests an elastic operation to increase its bitrate, a spectrum assignment is performed over its route (i.e., no new route is computed).

The algorithm first tries to find a new channel $\{fc, m\}$ for the already established connection p with enough capacity for the requested bitrate b (line 2). To perform that computation, the graph $G(N, E)$ and the set of connections in the network, P, are considered. Second, if a channel is found without moving already established connections (lines 3-5), the *Solution* is populated. In case that no suitable channel is found, the defragmentation process is triggered (lines 6-16). Initially, among all the connections whose route shares links with p, only those that are allocated besides p in the spectrum are selected; the distance between the connection and p, in terms of slots, is calculated (lines 7-11). Next, the SPRESSO algorithm is used to compute the minimum number of connections to be moved to create a channel with the requested capacity (line 12). If a feasible reallocation can be done, *Solution* is initialized with the list of reallocations to be performed. Subsequently, a new channel is found taking into account the reallocations in *Solution* (line 14). The new reallocation is added to *Solution* (line 15), which is eventually returned (line 16).

Table 9-8 EL-SPRESSO Algorithm

```
INPUT G, P, p, b
OUTPUT Solution
 1: Solution ← ∅
 2: {fc, m} ← findNewChannel(G, P, p, b) (see [As13])
 3: if m > 0 then
 4:    Solution ← {realloc(p, fc, m)}
 5:    return Solution
 6: Q⁻ ← ∅, Q⁺ ← ∅
 7: for all e ∈ p.route do
 8:    neighP ← getNeighboringLeft(P, p.fc, e)
 9:    Q⁻ ← {(p, maxShiftLeft(G, neighP))}
10:    neighP ← getNeighboringRight(P, p.fc, e)
11:    Q⁺ ←{(p, maxShiftRight(G, neighP))}
12: Solution ← findSolMinMoves(Q⁻ , Q⁺, p.fc, b) (see section 8.1)
13: if Solution ≠ ∅ then
14:    {fc, m} ← findNewChannel(G, P, p, b, Solution) (see [As13])
15:    Solution ← Solution ∪ {realloc(p, fc, m)}
16: return Solution
```

9.2.2 Performance Evaluation

In this section, we focus on studying the performance of the EL-SPRESSO algorithm. To that end, we consider the network topologies presented in the Appendix A, these are the 21-node and 35-link TEL topology, the 20-node and 32-link BT topology, and the 21-node and 31-link DT topology.

We developed the EL-SPRESSO algorithm in C++ and deployed in an ad-hoc event-driven simulator based on the OMNet++ framework [OMNet]. Aiming at performing elastic operations, we first established a single connection between each node pair. Next, a dynamic network environment was simulated, where incoming elastic operations requests over the established connections arrive to the system following a Poisson process; requests are sequentially served without prior knowledge of future incoming requests. A percentage (30%) of the requests requires transporting no traffic, whereas the rest of the requests (70%) follow the traffic profile specified in Table A-2.

Although the EL-SPRESSO algorithm is executed upon the reception of every request, those requests asking to decrease the current spectrum allocation will find a new spectrum allocation, whilst for those asking for an increment, part of that spectral increment might be rejected.

Already established connections are randomly chosen with equal probability (uniform distribution). Different values of the offered network load are created by changing the offered average bitrate (see Table A-2). We assume that no retrial is performed; the spectrum that cannot be served, it is immediately blocked.

Regarding optical spectrum, the total width was fixed to 4 THz and the slot width to 6.25 GHz. We use the correspondence in Table A-3 to convert bitrate into

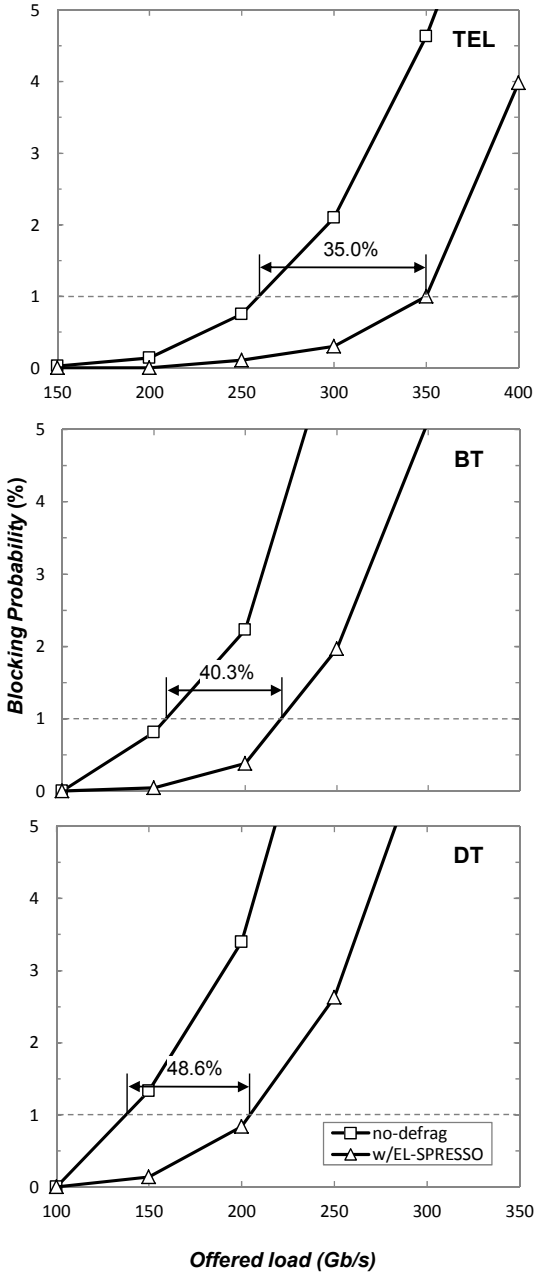

Fig. 9-5 Blocking probability vs. offered load in Gb/s.

spectrum width. Finally, it is worth highlighting that each point in the results is the average of 10 runs with 20,000 elastic operation requests each.

Graphs in Fig. 9-5 present blocking probability values against the offered load, in Gb/s, for the three networks considered. It can be observed that the EL-SPRESSO algorithm allows increasing remarkably the amount of bitrate transported; the gain achieved using EL-SPRESSO algorithm ranges from 35 to 48%.

9.3 Conclusions

Looking to make enough room for an incoming connection request, in this chapter the SPRESSO re-optimization mechanism was proposed to reallocate already established paths in the spectrum. The SPRESSO mechanism was modeled as an ILP problem. As a consequence of the real time computation requirements, a GRASP+PR-based heuristic algorithm was proposed to obtain near-optimal solutions for SPRESSO in practical computation times. The SPRESSO mechanism was implemented and results obtained showed that the performance of the networks in terms of blocking probability can be improved in a range from 20 to 31%.

Next, the SPRESSO algorithm was extended to include elastic operations. The new algorithm, named EL-SPRESSO, was designed and its performance evaluated by simulation. The performance results showed that a gain over 35% in terms of blocking probability could be achieved.

Once the main algorithms developed in this thesis have been developed and evaluated by simulation, the next chapter is devoted to experimentally validate their feasibility and evaluate their performance deploying them in real test-beds.

Chapter 10

Towards In-Operation Planning

This chapter starts experimentally assessing some of the algorithms previously presented: *i*) the BAREMO heuristic is evaluated on a PCE-GCO architecture where the algorithm is deployed within the GCO module. BAREMO performance is compared against the sequential approach in terms of restoration time; *ii*) the stateful PCE architecture is used to experimentally validate the feasibility of the EL-SPRESSO algorithm.

In view of PCE architectures do not facilitate network reconfiguration we propose a control and management architecture to allow the network to be dynamically operated, reconfigured and re-optimized in response to traffic changes in an automatic fashion; hence, resource over-provisioning can be minimized reducing overall network costs.

10.1 Experimental assessment of dynamic restoration

10.1.1 PCE architecture for the BAREMO problem

Fig. 10-1 depicts the architecture of the implemented GCO module for solving the BAREMO problem (defined in section 7.2), inside a stateless PCE. For both regular LSP provisioning and restoring disrupted LSPs, the ingress PCC is responsible for sending a path request to the PCE using the PCEP (step 1 in Fig. 10-1).

At the PCE, the incoming request is stored in either the provisioning or restoration queues. An eXclude Route Object (XRO) [Ok09.2] received in the path computation request, conveying the failed physical link with the fail-bit active, allows the PCE to decide storing it in the restoration queue; otherwise, the request is assumed to

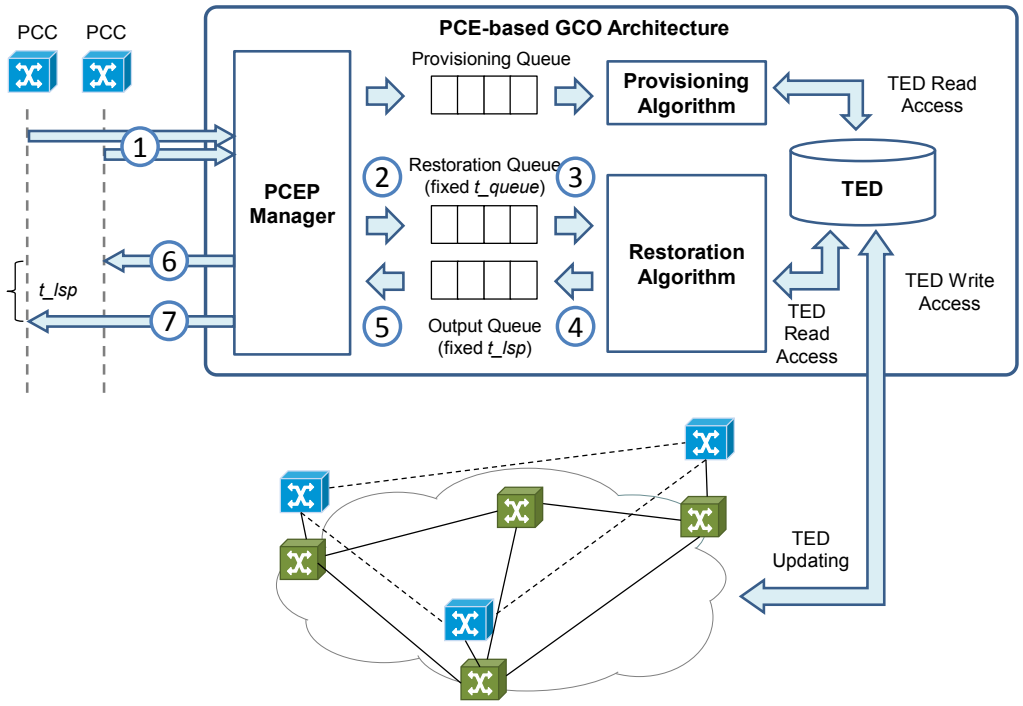

Fig. 10-1 Designed and implemented PCE-based GCO architecture.

be a regular working path provisioning and is stored in a priority queue attended/served by a thread pool. The selected routing algorithm is determined by the objective function (OF) object included in the PCEP request. Such OF must match with one of the algorithms available in the Algorithm API.

Path computation requests originated from restoration processes are grouped in a bulk (step 2) and processed after a configurable timer (*t_queue*, step 3) expires. This allows synchronizing such bulk of backup paths to be computed. Once *t_queue* has elapsed, the path computation for regular LSP provisioning is stopped to avoid any interference on the restoration procedure, being incoming requests temporary stored in the provisioning queue until that procedure ends. Next, the selected restoration algorithm available in the Algorithm API is triggered (step 4). Before the restoration algorithm is called, a copy of the TED is created and both failed physical and logical links specified in the XRO are removed from that copy. In addition, the restoration algorithm receives the set D of pairs of source and destination endpoints and the required bitrates.

Besides computing the path for each request, the algorithm specifies the ordering, sequence and timing of the PCEP responses (see proposed algorithm in section 7.2.3) (step 5). In other words, the PCE decides whether a PCEP response is immediately sent back to the PCC or delayed. The latter is forced to occur when the computed path has some dependencies (e.g., creation of a new logical link) with respect to a formerly computed path (e.g., Fig. 7-2(b)). In this situation, the PCEP

response is artificially delayed by *t_lsp* to guarantee that, for instance, the required logical links are actually induced and created (steps 6 and 7).

Finally, let us analyze the influence of TED updating on path computation. LSA flooding starts just after any resource in the network is either allocated or released. When a TED update message is received at the PCE, the TED needs to be updated; to this end the TED is firstly blocked, which results in any process trying to access it being blocked until the updating process ends. This can highly increase restoration times in the sequential restoration when the number of PSC LSPs to be restored is high, since some updating LSA could arrive at the PCE before all the restoration paths have been computed. Note that TED updating does not overlap when bulk restoration is used.

10.1.2 BAREMO Experimental Results

In this section, we present the scenario considered in our experiments and we show the results obtained when using sequential (no GCO) versus bulk PCE GCO restoration.

A GMPLS/PCE control plane platform [Mu11] is used for experimentally validating the implemented PCE-based dynamic restoration with GCO for MLN. Fig. A-9 shows the topology of a European MLN network consisting of 12 LSC nodes. Colored nodes have a PSC node attached. Each PSC node is physically connected to its associated LSC node with 16 bidirectional ports/transceivers. Each optical link (i.e., between LSC nodes) supports 16 DWDM channels per direction operating at 10 Gb/s. The PCE managing provisioning as well as restoration path requests is co-located at node 9.

We performed some tests to experimentally retrieve the values for *t_queue* and *t_lsp* without excessively penalizing the restoration time. We finally found that, in the testbed, at least 100 ms are needed to guarantee that every PCEP requests arrive to the PCE, whereas 150 ms are needed for *t_lsp* to delay PCEP replies ensuring that LSPs are actually established in the network. For illustrative purposes, let us assume that two PSC LSPs have been affected by a failure. Exchanged PCEP request/reply messages for MLN restoration are shown in Fig. 10-2. Two path computation requests *PCReq1* and *PCReq2* are received in the PCE. Since those requests contain an XRO object they are placed into the PCEP restoration queue. After the timer *t_queue* (started upon the reception of *PCReq1)* expires, the PCE invokes the algorithm in the GCO to solve the BAREMO problem to simultaneously compute the routes for all the received backup path requests. Next, the PCE sends a response message to those LSPs that can be immediately established, since they do not depend on new FA TE links. In Fig. 10-2, the PCE sends the response to *PCReq1* and waits a time *t_lsp* before sending the response to *PCReq2*.

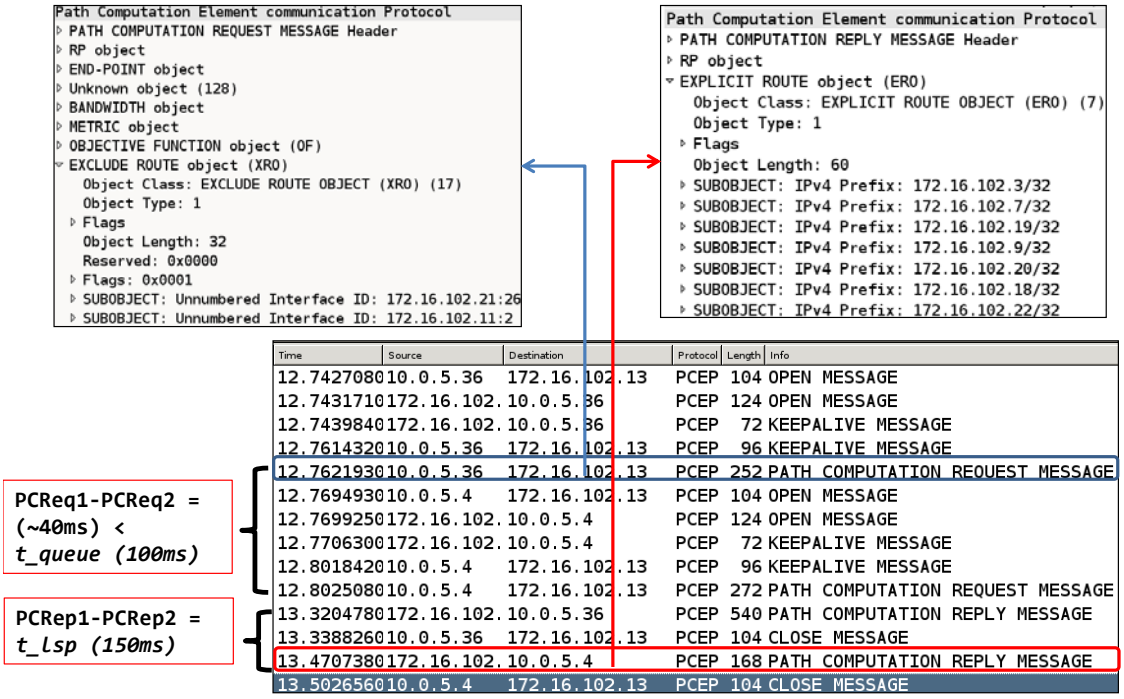

Fig. 10-2 Messages exchange between PCC and PCE applying PCE-based GCO

Fig. 10-3 shows the obtained restoration times as a function of the traffic load. Average times in Fig. 10-3 show that the bulk approach increases the restoration time up to 26% with respect to the sequential approach, being lower than 350 ms even under the highest offered load. This is the cost/penalty of the bulk approach, which comes from waiting *t_queue* (100 ms) to create the request bulk, and *k* times *t_lsp* (150 ms) for those PSC LSPs which need that *k* FA TE links are previously established (i.e., LSP dependencies). Notwithstanding, in view of the obtained times, the value of *k* is low (not higher than 2 on average). When focusing on the maximum restoration times, they increase remarkably for the sequential approach.

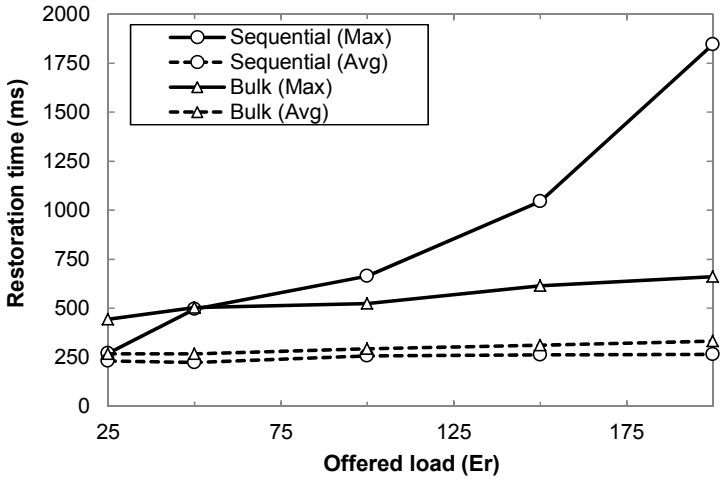

Fig. 10-3 Restoration times vs. offered load.

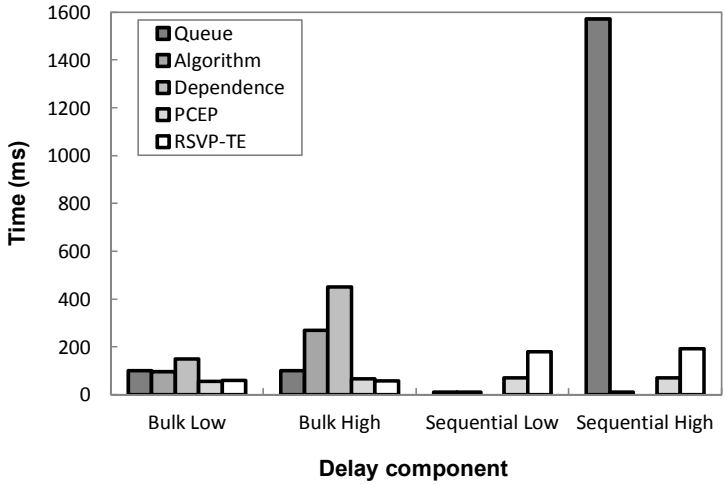

Fig. 10-4 Disaggregated maximum values of time components.

Fig. 10-4 gives insight of maximum restoration times disaggregating those into their five components: input queue, computations time, delay waiting for dependences, and times for PCEP and GMPLS signaling. Remarkably high values can be observed for the PCE input queue, which is as a consequence of TED updating messages sent immediately after the first LSP is signaled, thus blocking the TED for the update, as stated in the previous section. In contrast, the increment in the bulk approach is almost linear with the load; starting from 445 ms under low load, increases to only 662 ms for the highest one. Note that in the bulk approach, TED updating starts after the whole bulk computation is performed.

10.1.3 PCE architectures comparison

To solve the BAREMO problem (as well as in Chapter 7), we assumed a stateless PCE architecture together with a GCO module for the Dynamic Path Bulk Restoration problem. Let us now, compare the stateless PCE architecture with the active stateful PCE architecture in the context of restoration.

The stateless PCE architecture (Fig. 10-1) includes the TED and the GCO module where the bulk restoration algorithm runs. Two queues are also needed: the input queue is used to create the bulk of requests, whereas the output queue is used to deal with dependences.

The sequence diagram in Fig. 10-5 illustrates the process: when the source PCC of an affected LSP is informed about the failure, it sends a PCReq message to the PCE to request a restoration path computation, which also includes information about the failure (step 1). Note that a failure in a fiber link might disconnect a large amount of MPLS connections, so that the same number of PCReq messages arrives at the PCE within a period of time. Hence, the input queue is used to store all the PCReq arriving in such period, denoted as *t_queue* (2). In the previous

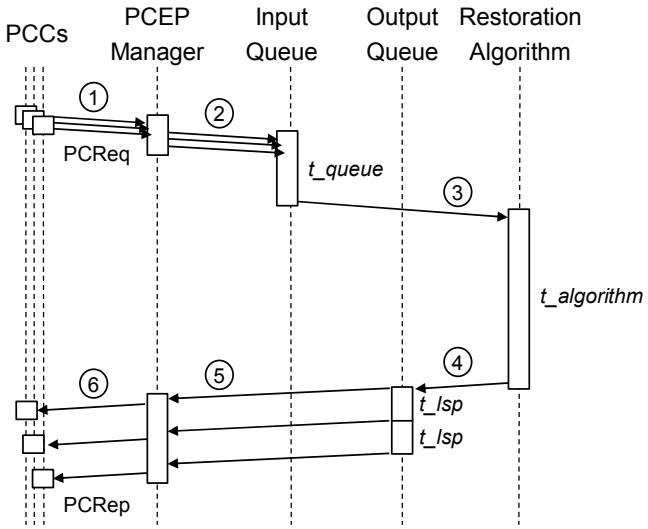

Fig. 10-5 Restoration workflow for the stateless PCE architecture.

section, we experimentally found that 100ms is enough to receive the large majority of PCReq messages for restoration, starting from the arrival of the first message. When *t_queue* expires, the bulk path restoration algorithm is invoked with the set of requests in the input queue (3). The solution from the algorithm is stored in the output queue (4) grouped by each dependence level, from where PCRep messages with the computed restoration path are sent back to each PCC (5,6). Note that a time (*t_lsp*) needs to be waited for each dependence level so as virtual links to be actually created in the data plane. As for *t_lsp*, in the previous section we experimentally found that 150ms are needed to be sure that any connection involving virtual link creation is completely set-up and thus their resources can be reused. This is the reason behind minimizing the dependence level.

The main drawback of the stateless PCE architecture is that two queues with a fixed delay are needed as a result of the lack of control about connections in the network. That can be solved, in part, by using an active stateful PCE architecture (Fig. 10-6); note that the provisioning queue remains the same in this architecture. In such architecture, the LSP-DB stores the connections established in the network. Because of connection knowledge, when a PCReq message for a restoration path computation arrives at the PCE (1), a process finds all the connections that might be affected using the information about the failure received (2). Therefore, the input queue is not needed to create the bulk and the path restoration algorithm can be immediately invoked (3), thus saving *t_queue* time. However, the input queue cannot be removed and it is still used to store the subsequent PCReq messages arriving for path restoration (4). This is as a consequence of PCRep messages need to be sent back in response to incoming PCReq ones, so the input queue is used to match requests and responses. When the algorithm finishes, the solution is stored in the output queue (5) and that matching

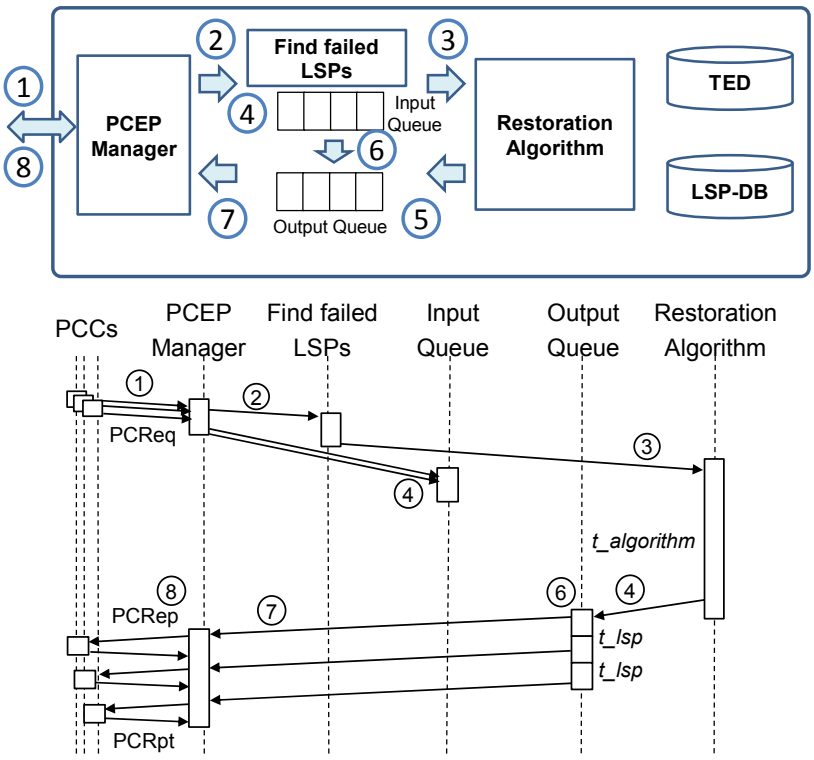

Fig. 10-6 Stateful PCE architecture.

is performed (6). Responses are send then back to each PCC taking into account the dependence level (7,8).

In this PCE architecture, PCCs asynchronously notify the PCE upon any connection has been actually set up or modified in the network using PCE Report (PCRpt) messages. Therefore, the delay between dependence levels in the output queue corresponds to the set up time of each specific connection, in contrast to the fixed delay time of the stateless PCE.

10.1.4 Illustrative results

For evaluation purposes, we developed an ad-hoc event-driven simulator in OMNeT++ [OMNet] implementing the stateless and the active stateful PCE architectures. The 30-node and 56-link TEL (Fig. A-), and the 22-node and 35-link BT (Fig. A-4) topologies were considered, where every node was equipped with a BV-OXC and an IP/MPLS router on top.

Aiming at selecting an algorithm for the evaluation, the heuristic algorithms proposed in Table 7-1 and Table 7-4 were compared. We implemented them in C++ and run on a the same computer as in the previous use case, a 2.4 GHz Quad-Core machine with 8 GB RAM memory under the Linux operating system. Individual problem instances were solved using both algorithms to compare the obtained solutions and the computation times.

Table 10-1 Heuristics Performance Comparison

Connections to restore	Computation time (s)		GRASP / Random			
			Un-restored	Dependence level		
	GRASP	Random		0	1	2
13	0.15	0.019	0/0	13/13		
93	6.05	0.16	0/0	86/86	7/7	
278	71.88	0.596	0/0	220/226	58/50	0/2
305	93.78	0.685	1/2	258/266	46/36	0/1

Table 10-1 shows the obtained solving times and the amount of IP/MPLS connections at each dependence level (labeled as 0..2) for several sizes of connections-to-restore bulks. Note that when no dependences were found, all the paths could be established in parallel using already set-up virtual links, whereas when 1 or more dependences are found connections are delayed in the output queue using the fixed *t_lsp* time in the case of the stateless PCE or a variable time in the case of the stateful PCE architecture.

In the table, for each level of dependence, the amount of connections to be established in parallel resulting from solving the GRASP or the Random heuristic is shown. Only in one of the tests performed, the proposed heuristics could not restore all the connections, which illustrates its performance in terms of restorability. Regarding dependence, the performance of both heuristics was slightly different; the Random heuristic provides solutions with one additional dependence level with respect to the GRASP-based algorithm. Note that the connections in the last level will suffer the largest restoration times. To improve that, the Random heuristic can be complemented with a local search phase, which would slightly increase computation times.

Finally, note that, as anticipated in light of the complexity analysis performed, GRASP computation times are very long, over 1 minute when the number of connections to restore is about 300. The solving times obtained with the Random heuristic, however, are in the order of 600 ms clearly illustrating the effectiveness of the proposed Random algorithm.

In view of the results in Table 10-1, we can select the Random heuristic to compare the performance of the stateless and the stateful PCE architectures.

A dynamic network environment was simulated where incoming IP/MPLS connection requests arrive to the system following a Poisson process and are sequentially served without prior knowledge of future incoming connection requests. The holding time of the connection requests is exponentially distributed with a mean value equal to 2 hours. Source/destination pairs are randomly chosen

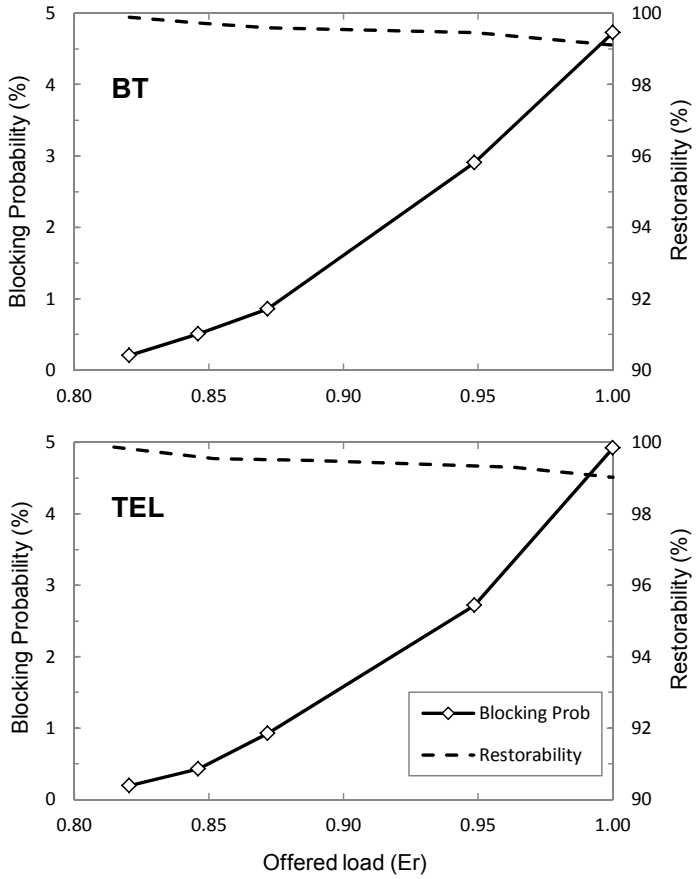

Fig. 10-7 Blocking Probability and Restorability against offered load.

with equal probability (uniform distribution) among all IP/MPLS nodes. Different values of the offered network load are created by changing the inter-arrival rate while keeping the mean holding time constant. In our experiments, the bitrate of each IP/MPLS flow was set to 1 Gb/s, the QPSK modulation format was used for the optical signals, the optical spectrum width was set to 4 THz, and the slice width was fixed to 6.25GHz. We assume that no retrial is performed; if a request cannot be served, it is immediately blocked.

Besides connections, optical link failures also follow a Poisson process with a mean time to failure equal to 50 hours. Link failures are randomly chosen with equal probability and the mean time to repair is fixed to 6 hours. Finally, all the results were obtained after requesting 150,000 IP/MPLS connections. Each point in the results is the average of 10 runs and both PCE architectures are executed using identical input data.

Plots in Fig. 10-7 show blocking probability and restorability results as a function of the offered load for the network topologies considered. Note that offered loads have been limited to those unleashing blocking probability in the meaningful range [0.1%-5%]. For the sake of comparability between topologies, offered loads have

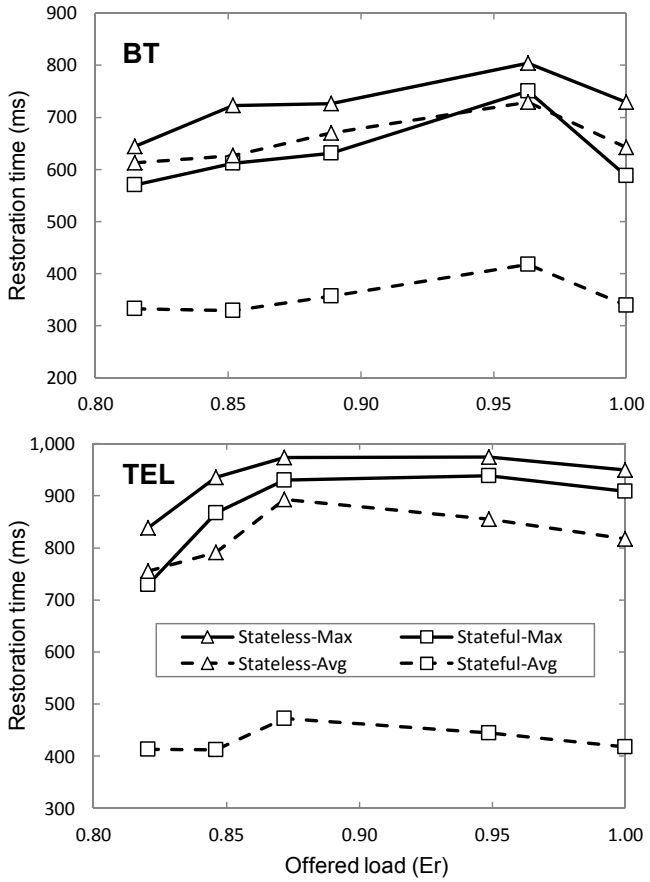

Fig. 10-8 Restoration time vs. offered load.

been normalized to the value of the highest load. As anticipated, restorability values are higher than 99% even when provisioning blocking is as high as 5%.

Maximum and average restoration times as a function of the offered load are plotted in Fig. 10-8. Interestingly, maximum and average plots for the stateless PCE architecture are really close one to the other, which reveals little difference among restoration times of individual connections. In contrast, the difference between maximum and average plots for the stateful PCE architecture is much higher, indicating higher dispersion among restoration times of individual connections. Moreover, although plots for maximum times are close for stateless and stateful PCE architectures, average plots are almost double in the case of the stateless architecture. Note that average restoration times are in the order of 300-400ms when the stateful PCE architecture is used, in contrast to 650-800ms for the stateless architecture.

When total restoration times are disaggregated (Fig. 10-9), we realize that the components that change between both PCE architectures are the time that restoration requests remain on average in the input queues and the time that, as result of the dependences, responses remain in the output queues. Regarding input

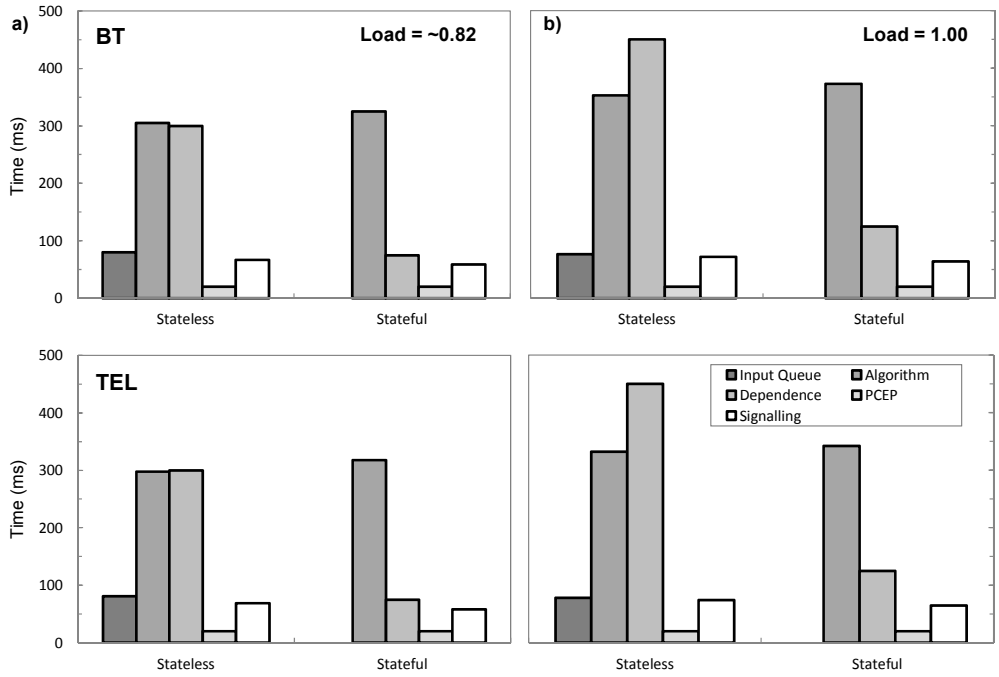

Fig. 10-9 Disaggregated values of maximum restoration time components.

queue, note that the maximum time is limited by *t_queue* in the stateless PCE architecture, so time in the input queue is, not surprisingly, close to that value. However, time in the input queue is zero in the stateful PCE architecture, since as soon as the first restoration request arrives, all the connections that might have been impacted by the failure are computed and the restoration algorithm is invoked. This is the reason why the restoration algorithm takes slightly longer in the case of the stateful PCE architecture; that time includes both, the time to find all the affected connections and the restoration heuristic.

As for the dependences, note that in the stateless architecture, each dependence level is delayed a fixed *t_lsp* time. Although *t_lsp* needs to be experimentally found for the specific network, its value reflects the maximum connection set-up time. In contrast, in the stateful architecture dependences are solved as soon the previous connection has been actually set up, fact that is notified through the PCEP interface to the PCE. Therefore, on average, the dependence time drops to the mean set up time.

10.2 Experimental assessment of EL-SPRESSO

In this section, we take advantage of the active stateful PCE architecture capabilities to experimentally address EL-SPRESSO problem.

Before entering into the details of the proposed active stateful PCE architecture, it is worth noting that elastic operations are usually performed increasing spectrum allocation symmetrically with respect to the central frequency. Therefore, the elastic operation performed over path p in the example shown in Fig. 9-4 actually entails two operations performed sequentially: first the central frequency is shifted to the desired value (we call this a *move_LSP* operation) and then, the spectrum is symmetrically expanded (*update_LSP* operation). It is clear that all the operations need to be performed in a strict sequence, which raises the necessity of defining an entity to coordinate a list of actions to be performed in a defined order.

For this very reason, we first present an updated version of the EL-SPRESSO algorithm (first introduced in Table 9-8) that copes with the sequencing requirement in Table 10-2 (changes are highlighted).

The algorithm first tries to find a new channel {*fc*, *m*} for the already established LSP p with enough capacity for the requested bitrate b (line 2). To perform that computation, the TED ($G(N, E)$, where N represents the set of nodes and E the set of TE links) and the LSP-DB containing the set of LSPs in the network, P , are considered. Second, if a channel is found without moving already established LSPs (lines 3-8), the *Solution* is populated with the list of actions to be performed; in case the solution entails shifting p central frequency, a *notification_flag* is set to

Table 10-2 EL-SPRESSO Experimental Algorithm

```
INPUT G, P, p, b
OUTPUT Solution
 1:  Solution ← ∅
 2:  {fc, m} ← findNewChannel(G, P, p, b) (see [As13])
 3:  if m > 0 then
 4:     if fc ≠ p.fc then
 5:        Solution ← {move(p, fc)} ∪ {notification_flag} ∪
           {update(p, m)}
 6:     else
 7:        Solution ← {update(p, m)}
 8:  return Solution
 9:  Q⁻ ← ∅, Q⁺ ← ∅
10:  for all e ∈ p.route do
11:     neighP ← getNeighboringLeft(P, p.fc, e)
12:     Q⁻ ← {(p, maxShiftLeft(G, neighP))}
13:     neighP ← getNeighboringRight(P, p.fc, e)
14:     Q⁺ ←{(p, maxShiftRight(G, neighP))}
15:  Solution ← findSolMinMoves(Q⁻ , Q⁺, p.fc, b) (see section 8.1)
16:  if Solution = ∅ then
17:     return {no_solution}
18:  {fc, m} ← findNewChannel(G, P, p, b, Solution) (see [As13])
19:  if fc ≠ p.fc then
20:     Solution ← Solution ∪ {move(p, fc)} ∪ {notification_flag} ∪
        {update(p, m)}
21:  else
22:     Solution ← Solution ∪ {update(p, m)}
23:  return Solution
```

synchronize move and update actions. In case that no suitable channel is found, the defragmentation process is triggered (lines 9-23). Initially, among all the LSPs whose route shares links with *p*, only those that are allocated besides *p* in the spectrum are selected; the distance between the LSP and *p*, in terms of slots, is calculated (lines 10-14). Next, the SPRESSO algorithm is used to compute the minimum number of LSPs to be moved to create a channel with the requested capacity (line 15). If no feasible reallocation can be done, a *no_solution* action is returned (lines 16-17). On the opposite, *Solution* is initialized with the list of reallocations to be performed. Subsequently, a new channel is found taking into account the reallocations in *Solution* (line 18). The appropriate actions are finally added to *Solution*.

The EL-SPRESSO algorithm is designed so as to be deployed in an active stateful PCE. Fig. 10-10 details the implemented PCE architecture. In addition to the topological information contained in the TED, the PCE stores the established LSPs' information in the stateful LSP database in Fig. 10-10.

A specialized module inside the PCE, the Online Active Solver, was developed for solving PCCs requests. This module implements the EL-SPRESSO algorithm. In this regard, a controller named Action Handler between the PCE and the specialized module was defined. Through that controller, the PCE forwards the requests received from a PCC to the Online Active Solver and receives back the solution. Two types of requests were implemented from PCC:

i. *new_LSP* setup is used to request a new LSP along with routing, spectrum allocation, and modulation format;

ii. *update_LSP* is used to request an increase/decrease of the bitrate (i.e., elastic operation) of an already established LSP.

Note that the *update_LSP* request does not imply rerouting the LSP but only modifying its spectrum allocation. The solution returned by the algorithm consists of a list of actions to be performed by the PCE which are queued in the Action queue (see Fig. 10-10). Four action types were defined:

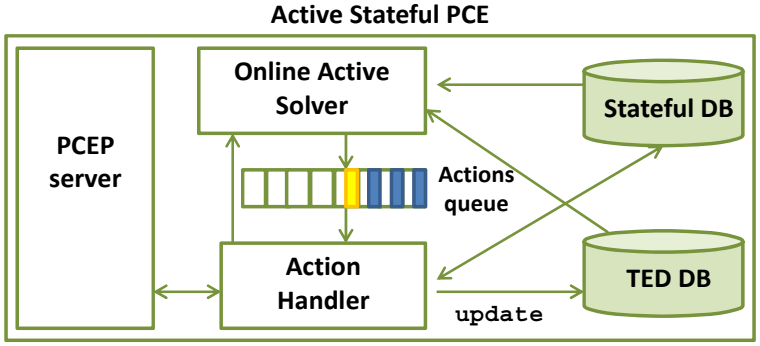

Fig. 10-10 PCE Architecture.

i. *new_LSP* for provisioning of a new LSP, that contains the route, the spectrum allocation and the modulation format;

ii. *update_LSP* (*p*, *m*) for elastic operations on LSP *p* which contains the new spectrum allocation, *m*. Note that the central frequency is kept constant;

iii. *move_LSP* (*p*, *fc*) for spectrum shifting of LSP *p* which contains the new central frequency, *fc*;

iv. *no_solution* returned if a feasible solution is not found for the received request.

For example, if enough resources are available in the network, an *update_LSP* request could entail a solution list with just one action, i.e., an *update_LSP* action. However, if some defragmentation is needed to make room for the given LSP, a list of actions with the necessary *move_LSP* actions, in addition to the *update_LSP* one, should be provided. Note that actions in the queue may need to be serially executed since some dependence among them could exist, i.e. some actions need to wait until some previous ones are actually implemented on the data plane. For instance, an LSP can only be updated once all LSPs have been moved to make the required room. For this reason, each action has a *notification_flag* indicating if the PCE must wait till it is completed or not. Consequently, if one action has the *notification_flag* set, the PCE waits the action confirmation message (PCEP PCNtf messages) before executing the next action in the queue. This mechanism allows considering dependence between actions.

To experimentally assess EL-SPRESSO algorithm a C++ active stateful PCE implementation running on Linux has been used (Fig. 10-11). The active PCE architecture, described above, has been implemented by extending the existing multi-bitrate impairment-aware PCE software tool, already demonstrated to run complex path computation algorithms while providing good scalability performances [Pa11]. The PCE communicates with PCCs co-located within GMPLS-enabled node controllers supporting flexible label grid extensions.

A request for an elastic operation of an established 100 Gb/s LSP is considered in this evaluation. Fig. 10-12 illustrates initial spectrum allocation in solid lines. Thus, to increase the LSP bitrate to 200 Gb/s the allocated spectrum has to be doubled and a shift of the adjacent superchannel is required. Specifically, upon receiving the PCReq message, event (1) in Fig. 10-12 and Fig. 10-13, the Online Active Solver returns the set of actions required to perform the elastic operation. In particular, the EL- SPRESSO algorithm returns the action list within 3 ms. As a result, the 400 Gb/s superchannel is shifted of 100 GHz. Fig. 10-13 captures the relevant control messages, collected at the PCE used to complete the process. A PCUpd message (2) is sent to the superchannel ingress node to trigger the superchannel shifting which is realized with GMPLS make-before-break mechanism and hitless continuous spectral shift [Cu13.1].

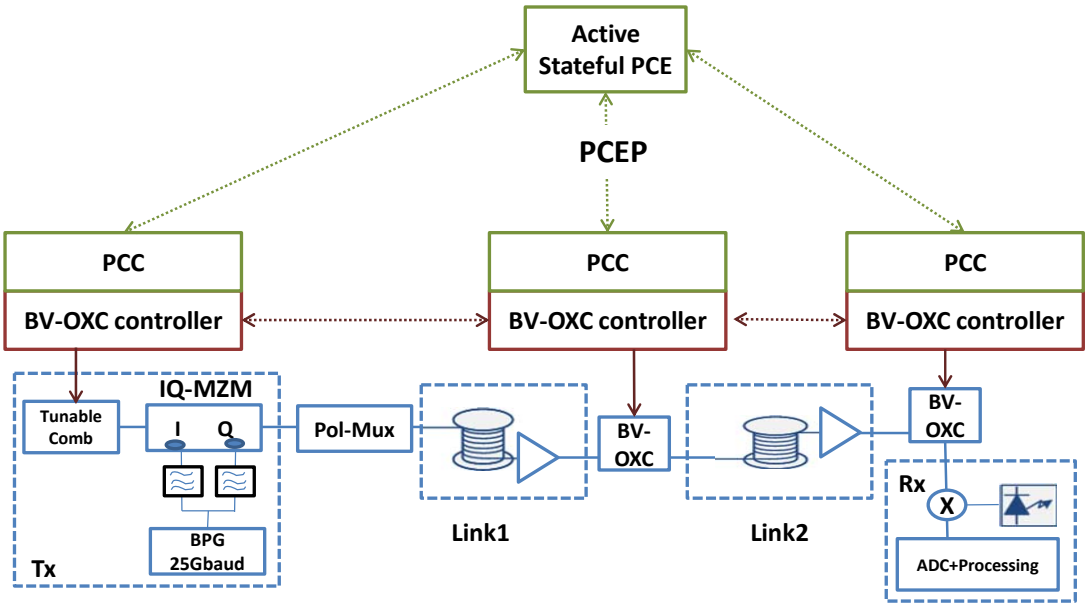

Fig. 10-11 Experimental scenario.

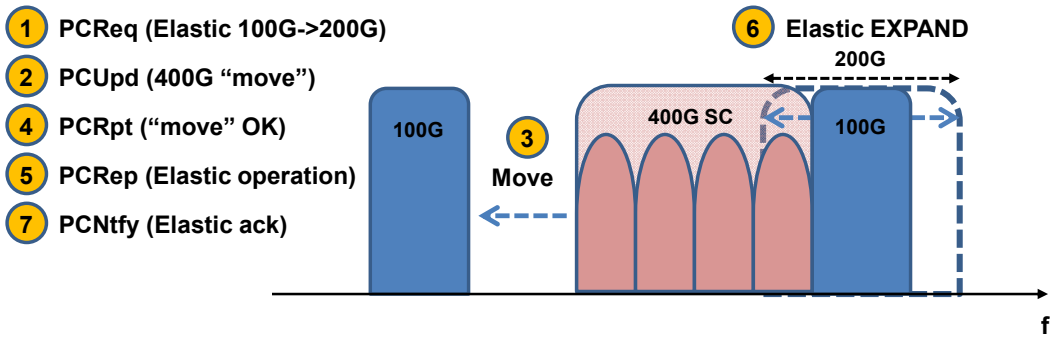

Fig. 10-12 Control and data plane operations.

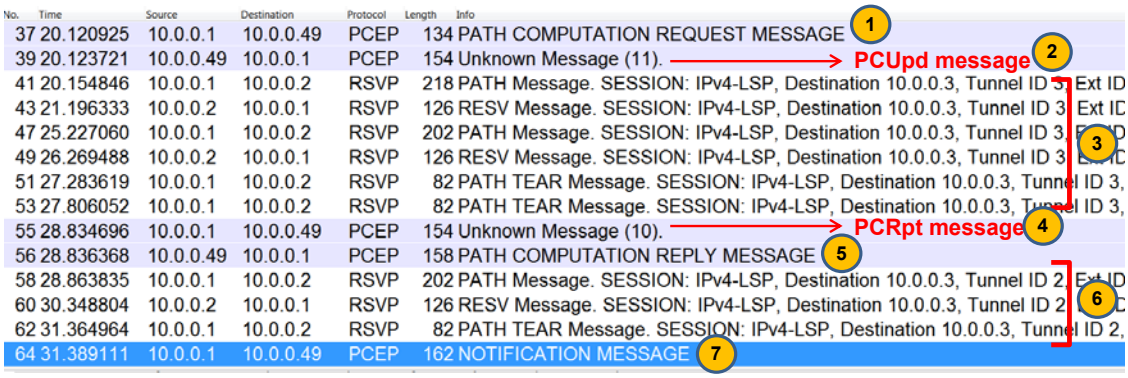

Fig. 10-13 Control messages at ingress PCC.

Once all shifting operations are concluded (3), the source PCC notifies the PCE with a PCRpt message (4). Then, the PCE updates its Stateful DB and triggers the elastic operation by sending a PCRep message (5) to increase LSP bitrate to 200

Gb/s over the newly available spectrum resources. Once the elastic operation is performed (6), the PCC notifies the PCE with PCNtf message (7).

10.3 In-Operation Planning

Despite the active stateful PCE, network reconfiguration requires executing complex workflows including solving optimization problems, where the obtained solutions are immediately implemented in the network. To solve this, the ABNO architecture, based on standard modules is being defined by the IETF, as introduced in section 2.4. In this section, we extend the ABNO architecture with a planning tool able to solve complex re-optimization problems.

Note that this is in contrast to traditional network planning, where results and recommendations require from manual intervention and hence long time is required until they are implemented in the network.

10.3.1 Migration towards in-operation network planning

Assuming the benefits of operating the network in a dynamic way are proven, the classical network life-cycle (Fig. 2-2) has to be augmented to include a new step focused on reconfiguring and re-optimizing the network, as represented in Fig. 10-14. We call that step *in-operation planning* and, in contrast to the traditional network planning, the results and recommendations can be immediately implemented on the network.

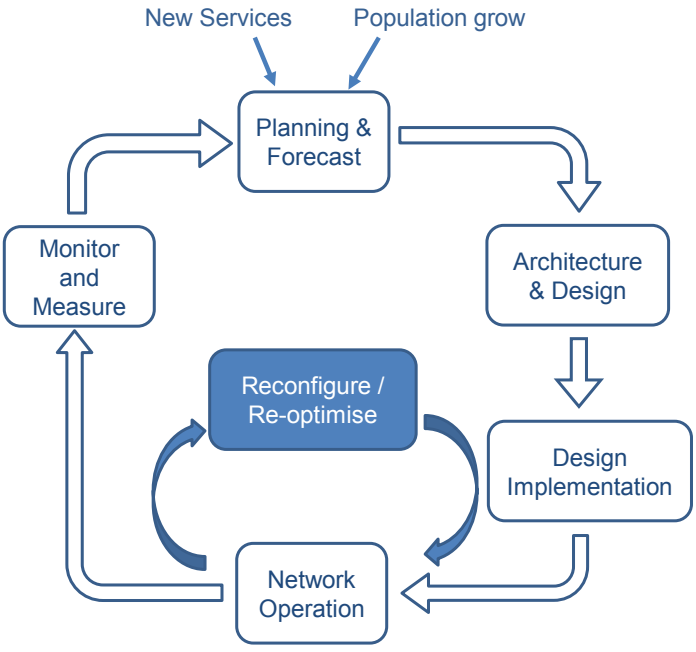

Fig. 10-14 Augmented networks life-cycle.

To support dynamicity, however, the current network architecture depicted in Fig. 2-3 will need evolve to include a functional block between the service layer and the network elements to support multi-service provisioning in multi-vendor and multi-technology scenarios; two standard interfaces are required. Firstly, the *north bound* interface that, among other tasks, gives an abstracted view of the network, enabling a common entry point to provision multiple services and to provision the planned configuration for the network. Moreover, this interface allows coordinating network and service layers according to service requirements. Secondly, the *south bound interface* covering provisioning, monitoring, and information retrieval.

Finally, operators will typically require human-machine interaction, this is to ensure new configurations and network impact are reviewed and acknowledged, before being implemented in the network.

10.3.2 ABNO Architecture and required functionalities for in-operation planning

Standardization bodies, especially the IETF, have been working to address all the above requirements, and as a result, the ABNO architecture is now being proposed as a candidate solution. As introduced in section 2.4, the ABNO architecture consists of a number of standard components and interfaces which, when combined together, provide a method for controlling and operating the network. The ABNO controller is the entrance point to the network for NMS/OSS and the service layer for provisioning and advanced network coordination, and it is responsible for the establishment of LSPs using Network Configuration Protocol (NetConf) or acting as an OpenFlow controller. Fig. 10-15 presents a future dynamic architecture based on ABNO.

Within the ABNO architecture, the in-operation planning tool can be deployed as a dedicated back-end PCE for performance improvements and optimizations. The

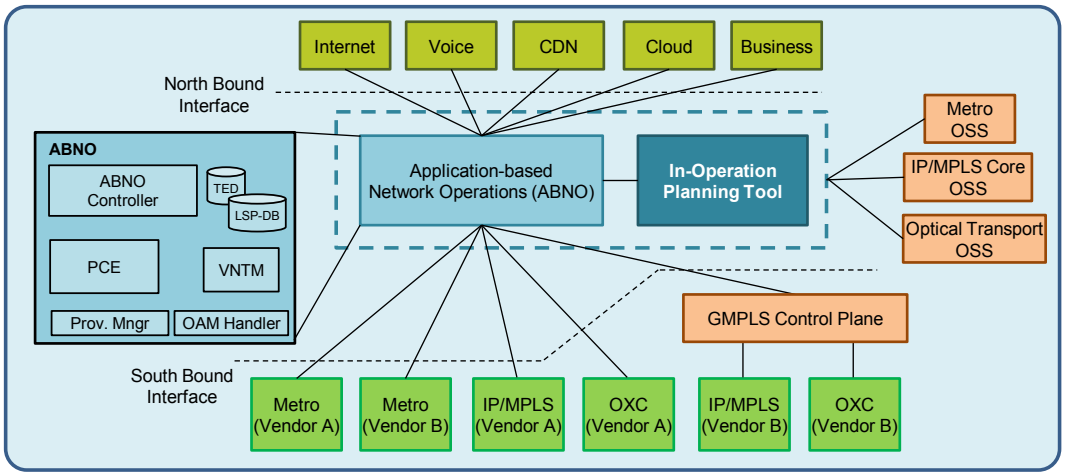

Fig. 10-15 Future dynamic architecture based on ABNO.

back-end PCE is accessible via the PCEP interface, so the ABNO components can forward requests to the planning tool.

Furthermore, in-operation network planning can only be achievable if planning tools are synchronized with the state of network resources, so new configurations can be computed with updated information, and those configurations can be easily deployed in the network. In the proposed architecture, the back-end PCE gathers network topology and current state of network resources, via the ABNO components, using protocols designed to convey link-state and traffic engineering

Table 10-3 Architectures supporting in-operation planning.

Architecture	Strengths	Weaknesses
Stateless PCE	• Path computation can be off-loaded into a dedicated entity capable of complex computations with customized algorithms and functions. • Has a standard and mature interface and protocol. • Supports simple optimization, such as bulk path computation [Le09.2].	• Is unaware of existing LSPs and has no view of the current network resource utilization and key choke points. • Cannot configure by itself any LSP in the network. • Delays need to be introduced to sequence LSP set-up [Ma13].
Stateful PCE	• Requests can be more efficiently placed optimizing network resources Supports optimization involving already established LSPs.	• More complex than a stateless PCE, requires additional database and synchronization. • No existing LSPs can be modified, e.g., for network re-configuration purposes.
Active Stateful PCE	• Supports complex reconfiguration and re-optimization, even in multilayer networks.	• No new LSPs can be created, e.g., for VNT re-optimization purposes. • Requires protocol extensions to modify and/or instantiate (if the capability is available) LSPs.
ABNO	• New LSPs can be created for in-operation planning. VNTM in charge of VNT re-configuration. • Supports deployment of solutions in multi-technology scenarios (NetConf, OpenFlow, control plane, etc.).	• Requires implementation of a number of key components in addition to the PCE function. • Some interfaces still need to be defined and standardized.

information, such as BGP-LS [Gr12]. First implementations showing this architecture are already available [Cu13.2].

ABNO provides a network control system for coordinating OSS and NMS requests to compute paths, enforce policies, and manage network resources for the benefit of the applications that use the network. The strengths and weaknesses of PCE architectures and ABNO for providing in-operation network planning are described in Table 10-3.

10.3.3 Use Case I: Virtual Topology reconfiguration

In our multilayer network scenario, the virtual network topology is supported by a set of optical connections. Thus, the IP/MPLS layer can operate independently from the optical one. However, in some cases the virtual topology needs to be reconfigured based on changing network conditions and demands. Many network operators prefer that such reconfiguration is supervised and approved by a person in charge before it is implemented in the network. This human intervention allows the application of additional policies and considerations and for integration with existing business policy and the OSS, e.g., new services not completely integrated with the tool.

In this section we analyze the virtual topology reconfiguration under two of such scenarios: *i*) after a failure and *ii*) disaster recovery.

10.3.3.1 Virtual topology reconfiguration after a failure

Fig. 10-16a represents a multilayer network consisting of four OXC in the optical layer and three routers in the IP/MPLS layer. IP/MPLS routers are connected through 10 Gb/s lightpaths, which create a VNT. Three bidirectional IP/MPLS LSPs have been set-up on the virtual topology. After a failure occurs, either in the optical or in the IP/MPLS layer, Fast Reroute (FRR) [Pa05] can be used to recover part of the affected IP/MPLS LSPs immediately after the failure. In addition, the state of the network after the failure can be updated in the control plane also within seconds. However, the capacity of some IP/MPLS LSPs might be reduced or even remain disconnected as a consequence of high congestion in some links in the virtual topology and a virtual topology reconfiguration needs to be performed to groom traffic and distribute traffic away from choke points and heavily utilized resources. An example is presented in Fig. 10-16b, where the LSP R2-R3 gets its capacity reduced from 8Gb/s to only 2Gb/s. To cope with virtual topology reconfigurations our approach relies on the ABNO architecture presented in Fig. 10-17, where the process is also represented.

When the network operator wishes to perform a network-wide virtual topology reconfiguration, a request is sent from the NMS/OSS to the ABNO controller (1), who then forwards it to the PCE via the VNTM (2). To alleviate PCE workload, a back-end PCE which contains an active solver is responsible for computing new

virtual topology layout taking into account the current state of the network. Therefore, the PCE sends a request towards the in-operation planning tool running in that back-end PCE to compute the new virtual topology (3). The tool considers all the surviving resources, which may include router interfaces and transponders connected to failed optical connections, spare interfaces that typically exist in the network for normal growth, and possibly some spare routers that have been installed ahead of time; all those resources can be stored in an inventory database. The tool must consider how to implement the desired IP/MPLS connectivity over the optical layer. To this end, it needs to know which optical links and nodes are up and which connections are optically feasible, considering optical impairments. When a result is obtained (4), the set of lightpaths is replied in a Path Computation Reply (PCRep) message (5) towards the originating PCE. In case that an operator needs to approve the new (virtual) layout, the PCE forwards it to the ABNO controller (6). The computed layout is then presented to the operator for final

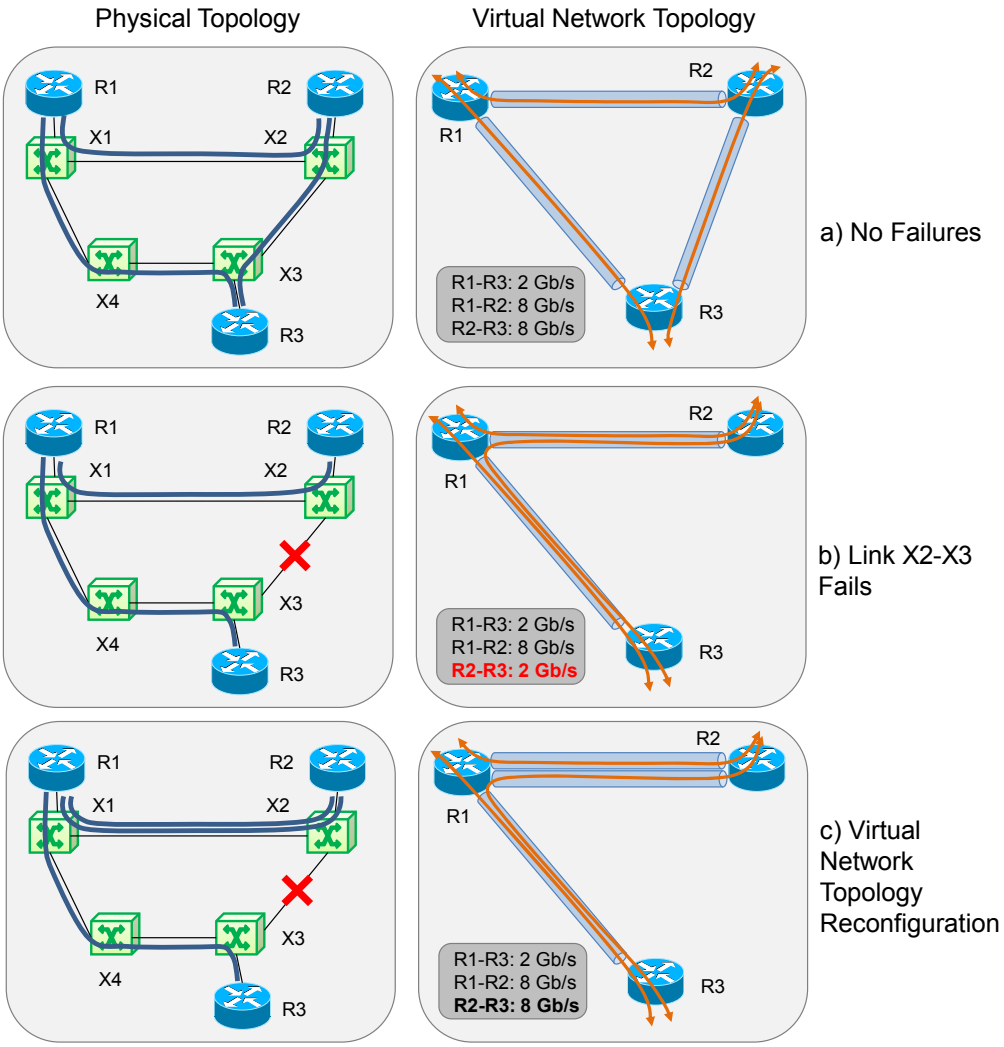

Fig. 10-16 An example of reconfiguration.

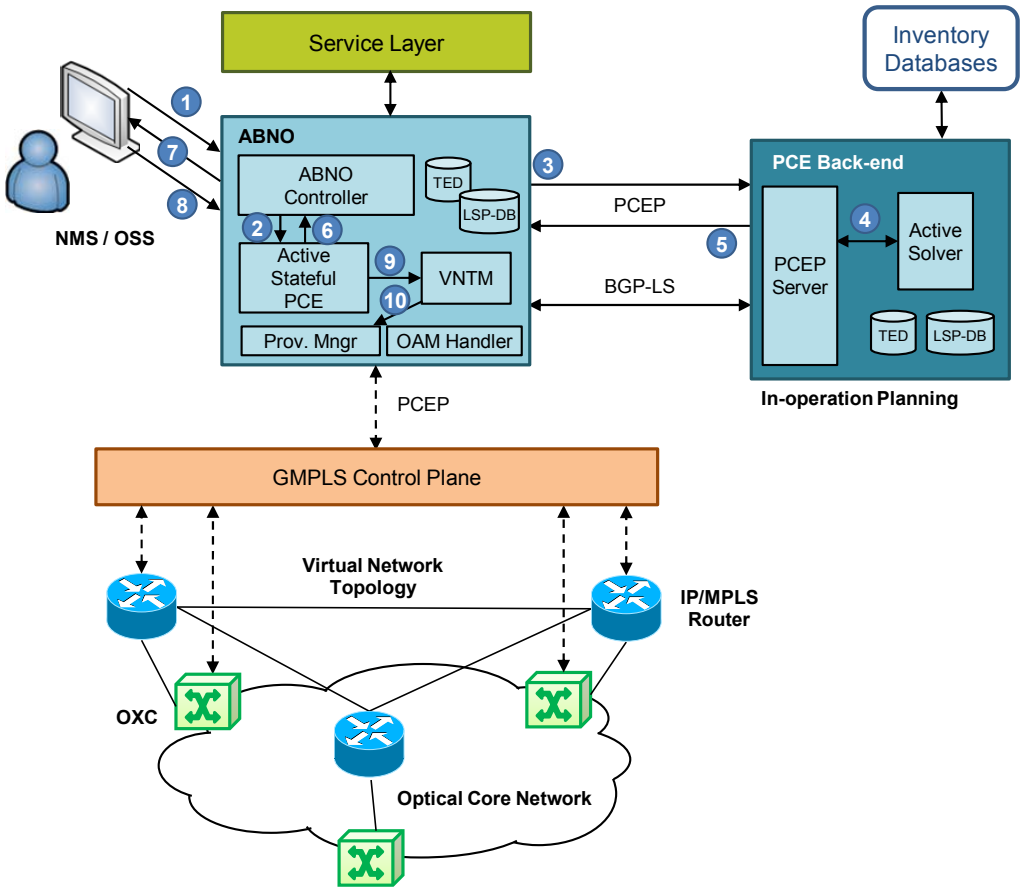

Fig. 10-17 VNT reconfiguration process.

approval (7). When the operator acknowledges the new optimized layout (8), it is passed to the VNTM (9) which computes the sequence of operations to carry out in terms of re-routing existing new LSPs minimizing disruption.

The sorted sequence of actions on existing LSPs is passed to the provisioning manager (10), which is able to interact with each head-end node. The provisioning interface, by which the provisioning manager is able to suggest re-routing of LSPs is based on the PCE Protocol (PCEP) and interface, using PCUpdate messages [Cr13]. The new allocated resources are reported back to the provisioning manager and ultimately the VNTM, using PCReport messages. Note that after a successful re-optimization the LSP-DB is updated accordingly. In our example in Fig. 10-16c, a new lightpath is created between R1 and R2 and as a result the IP/MPLS LSP R2-R3 can be rerouted and its initial capacity restored.

10.3.3.2 Disaster Recovery

While most networks are typically designed to survive single failures without affecting customer Service Level Agreements (SLA), they are not designed to survive large scale disasters, such as earthquakes, floods, wars or terrorist acts, simply because of their low failure probability and the high cost of over-

provisioning to address such events in today's network. Since many systems might be affected, large network reconfigurations are necessary during large scale disaster recovery.

The disaster recovery process is similar to that of the virtual topology reconfiguration after a failure. However multiple optical systems, IP links, and possible routers and OXCs (assuming central offices are affected) may be taken offline during the disaster. Several additional planning and operation requirements in response to large scale disasters are highlighted below:

- Consideration of potential IP layer traffic distribution changes, either using MPLS-TE tunnels, or by modification of IP routing metrics, and evaluating benefits based on the candidate topology.
- It may be impossible to reach the desired network end state with one step optimizations. Therefore, it may be necessary two or more step optimizations. For example, to reroute some other optical connections to make room for some of the new connections.
- The system must verify that the intermediate configuration after each such step is robust and can support the current traffic and possibly withstand additional outages.
- Based on pre-emption and traffic priorities, it might be desirable to disconnect some virtual links, so as to reuse the resources for post-disaster priority connections and traffic.

We have described the creation of one disaster recovery plan, but in a real network there may be several possible plans, each with their pros and cons. The tool must present all these plans to the operator so that the operator can select the best plan, and possibly modify it and understand how it will be behave.

To summarize, the above process consists of several steps:

1. Immediate action by the network to recover some of the traffic.
2. Dissemination of the new network state.
3. Root cause analysis to understand what failed, and why.
4. Operator-assisted planning process to come up with a disaster recovery plan.
5. Execution of the plan - possible in multiple steps.
6. Re-convergence of the network after each step and in its final state.

10.3.4 Use Case II: Re-optimization

Algorithms in the control or management planes compute routes and find feasible spectrum allocations for connection requests taking into account the state of

network resources at the time each connection is requested. Nonetheless, as a consequence of network dynamics, some resources may not be released so that better routes could be computed and thus, re-optimization could not be applied to improve network efficiency. For example, imagine an optical connection that due network congestion, is required to circumnavigate optimal nodes and links, so that the end-to-end connection requires intermediate regenerators; at some point additional paths become available and the service could be rerouted to use the shorter route and eliminate regeneration. Additionally, other existing services could be rerouted to remove the bottlenecks and avoid network congestion. Or even allow some connections to increase their capacity when needed.

In this use case we study a specific problem that arises in flexi-grid networks and where re-optimization could bring clear benefits. In such networks, lightpaths can be allocated using variable-sized frequency slots, whose width (usually a multiple of a basic width such as 12.5 GHz) is a function of the requested bit rate, FEC and modulation format. Such frequency slots must be contiguous in the spectrum and the same along the links in its route. As a consequence of the unavailability of spectrum converters, spectrum fragmentation appears increasing the blocking probability of connection requests, making worse the network grade of service.

An example is shown in Fig. 10-18a where the optical spectrum of a link is represented. Three already established lightpaths share that link; each lightpath uses a different frequency slot width. If a new lightpath needing 37.5 GHz is requested, it would be blocked as a consequence of lack of spectrum contiguity. In such scenario, re-optimization could be applied to the network before a connection request is blocked, by re-allocating already established lightpaths in the spectrum (Fig. 10-18b) to make enough room for the triggering connection requested (Fig. 10-18c). The SPRESSO algorithm (section 9.1) efficiently computes the set of connections to be reallocated.

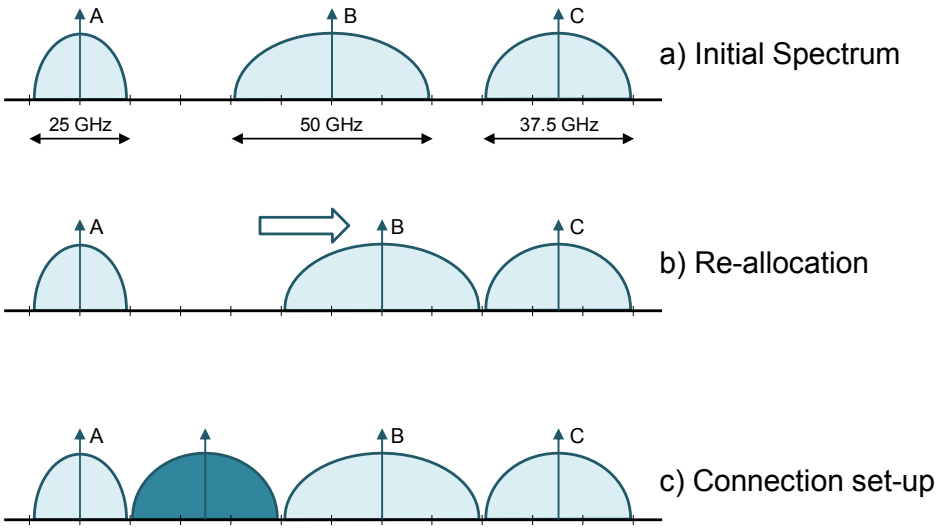

Fig. 10-18 Example of re-optimization.

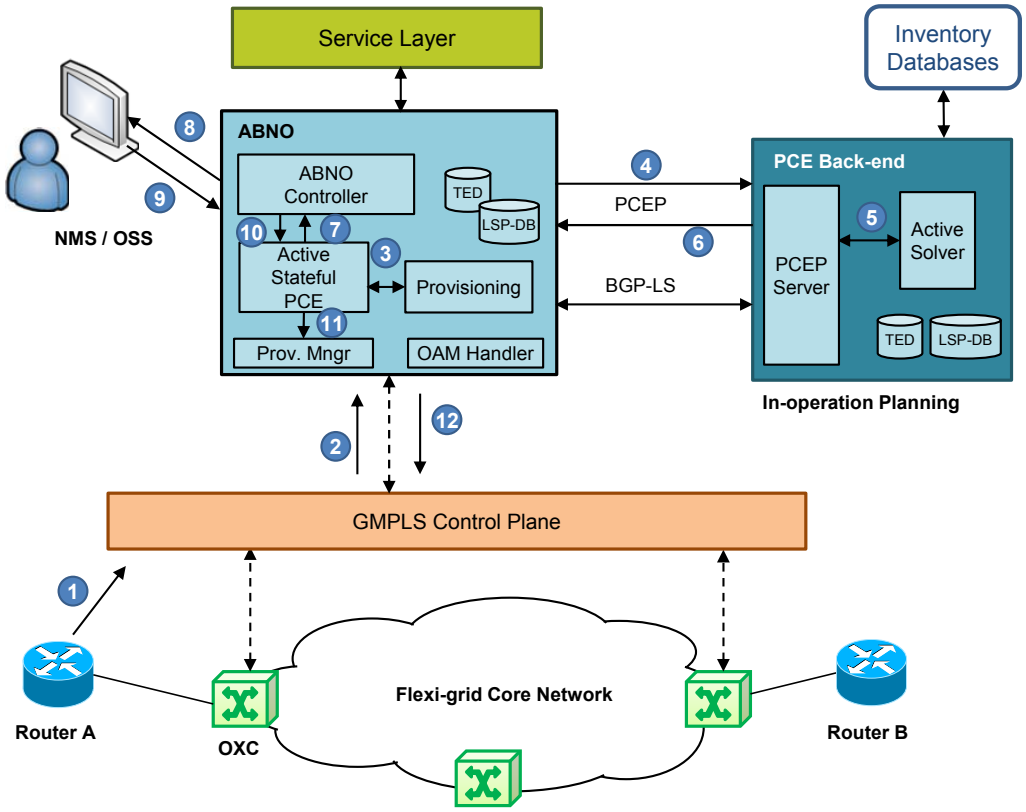

Fig. 10-19 Example of re-optimization process.

Fig. 10-19 illustrates the proposed control plane architecture to support flexi-grid network re-optimization, which also facilitates human verification and acknowledgement of network changes. When Router A needs a new connection to Router B, it sends a request to the control plane of the optical network (1). After checking admission policies, a Path Computation Request (PCReq) message is sent to the PCE (2), which invokes its local provisioning algorithm (3). In the event of insufficient resources being available due to spectrum fragmentation, the active PCE recommends the defragmentation of relevant nodes and connections, utilizing the right algorithm to provide such re-optimizations. Similarly, as in the previous use case, let us assume that the back-end PCE providing such algorithm will perform the computation (4) upon receipt of a request. When a result is obtained (5), it is sent back to the front-end PCE (6). In case that an operator need to approve implementing the computed solution in the network, a request is sent to the NMS/OSS via the ABNO controller (7, 8). When the solution has been verified and acknowledged by the operator, the NMS/OSS informs the PCE via the ABNO controller (9, 10), which forwards the solution towards the provisioning manager. Existing connection reallocations are requested using PCUpdate messages (11). Once the dependent connections have been setup, the responsible PCE will invoke the local provisioning algorithm for the original connection request between routers A and B and sends a PCRep message to the originating control plane node (12).

10.4 Conclusions

This chapter was devoted to experimentally assess some of the algorithms presented in previous chapters. Firstly, the BAREMO heuristic was evaluated on a PCE-GCO architecture where the algorithm was deployed within the GCO module; its performance was compared against the sequential approach in terms of restoration time. Secondly, the stateful PCE architecture was used to experimentally validate the feasibility of the EL-SPRESSO algorithm.

In view of re-optimization problems need to be solved in stringent computation times and immediately implemented in the network, a control and management architecture of transport networks was proposed to support *in-operation planning*. The architecture is based on ABNO and allows carriers to operate the network in a dynamic way and to reconfigure and re-optimize the network near real-time in response to changes. Networks life-cycle was extended achieving better resource utilization, thus reducing network CAPEX. Moreover, process automation reduces manual interventions and, consequently, OPEX.

Finally, two use cases were used to illustrate how the proposed in-operation planning and control and management architecture work together. In a multilayer network scenario, VNT reconfiguration after a failure and for disaster recovery was analyzed. In a flexi-grid network scenario, LSP re-allocation to reduce connection blocking was studied.

Chapter 11

Closing Discussion

11.1 Main contributions

The main contributions of this thesis are:

- In chapter 4 the benefits that can be achieved using finer slot widths as a function of the actual TP under which the network is operating were showed. Both the 12.5GHz and the 25GHz slot widths are potential candidates for the deployment of future multilayer networks based on flexgrid technology. (G1.1)
- In chapter 5 a novel recovery scheme called MPR, specifically designed for flexgrid optical networks, was proposed. It can be applied to recover the requested bitrate of client connection demands in the event of single-link failures. To compare the performance of MPR against existing single-path pure recovery schemes, such as DPP, SPP, and path restoration, the BRASERO problem was stated and modeled as a MILP problem. (G1.2)
- In chapter 6 we presented two provisioning algorithms. Firstly, a RSA algorithm to be used in dynamic flexgrid network scenarios was proposed. Its design includes adaptive considerations involving the optical signal to be conveyed by the optical connection, such as the use of modulation formats, distance constraints, cascade filters, and guard bands. Secondly, an impairment aware RWA in translucent DWDM networks, named TrIA-RWA, was presented. The algorithm uses auxiliary digraphs to minimize regeneration usage and path length in terms of number of hops. (G2.1)
- In chapter 7 the problem of dynamic restoration within multilayer networks using a centralized PCE for the path computation was studied. To improve restorability, a PCE GCO architecture was designed and implemented to

allow grouping all the requests, performing bulk path computation, reducing resource contention and leveraging grooming objectives. Firstly, focusing on the MPLS-over-WSON restoration problem, the BAREMO problem was formally stated and modeled using a MILP formulation. Due to the stringent requirements in terms of restoration time, a randomized heuristic algorithm was conceived. Secondly, the bulk restoration approach in multilayer MPLS-over-flexgrid networks was proposed. To this end, the DYNAMO problem was stated and a mathematical model was developed. (G2.2)

- In chapter 8 an effective restoration scheme enabling multipath recovery and bitrate squeezing in elastic optical networks implementing the flexgrid optical technology was proposed. The restoration scheme exploits the advanced flexible capabilities provided by MF-TPs. To efficiently recover network failures by exploiting limited portions of spectrum resources along multiple routes, the BATIDO problem was stated and an ILP model and heuristic algorithm were proposed. (G2.2)
- In chapter 9 the SPRESSO re-optimization mechanism was first proposed to reallocate already established paths in the spectrum, making enough room for an incoming connection request. The SPRESSO mechanism was modeled as an ILP problem. As a consequence of the real time computation requirements, a GRASP+PR-based heuristic algorithm to obtain near-optimal solutions for SPRESSO in practical computation times was proposed. Next, the SPRESSO algorithm was extended to include elastic operations, and the new algorithm EL-SPRESSO was designed. (G2.3)
- Chapter 10 was devoted to experimentally assess some of the algorithms previously presented. Firstly, the BAREMO heuristic algorithm was evaluated on a PCE-GCO architecture where the heuristic algorithm was deployed within the GCO module. Secondly, the stateful PCE architecture is used to experimentally validate the feasibility of the EL-SPRESSO algorithm. Furthermore, a control and management architecture of transport networks was proposed to support *in-operation planning*. (G2.4)

11.2 List of Publications

11.2.1 Publications in Journals

[Ca14.1] **A. Castro**, R. Martínez, R. Casellas, L. Velasco, R. Muñoz, R. Vilalta, and J. Comellas, "Experimental Assessment of Bulk Path Restoration in Multi-layer Networks using PCE-based Global Concurrent Optimization," IEEE/OSA Journal of Lightwave Technology (JLT), vol. 32, pp. 81-90, 2014.

[Ca14.2] **A. Castro**, L. Velasco, J. Comellas, and G. Junyent, "On the benefits of Multi-path Recovery in Flexgrid Optical Networks," accepted in Springer Photonic Network Communications (PNET), DOI 10.1007/s11107-014-0443-5, 2014.

[Pa14.1] F. Paolucci, **A. Castro**, F. Fresi, M. Imran, A. Giorgetti, BB. Bhownik, G. Berrettini, G. Meloni, F. Cugini, L. Velasco, L. Potì, and P. Castoldi, "Active PCE Demonstration performing Elastic Operations and Hitless Defragmentation in Flexible Optical Networks," accepted in Springer Photonic Network Communications (PNET), 2014.

[Pa14.2] F. Paolucci, **A. Castro**, F. Cugini, L. Velasco, and P. Castoldi, "Multipath Restoration and Bitrate Squeezing in SDN-based Elastic Optical Networks," [Invited] accepted in Springer Photonic Network Communications (PNET), DOI 10.1007/s11107-014-0444-4, 2014.

[Ve14.1] L. Velasco, **A. Castro**, D. King, O. Gerstel, R. Casellas, and V. Lopez, "In-Operation Network Planning," IEEE Communications Magazine (ComMag), vol 52, pp. 52-60, 2014.

[Ve14.2] L. Velasco, **A. Castro**, M. Ruiz, and G. Junyent, "Solving Routing and Spectrum Allocation Related Optimization Problems: from Off-Line to In-Operation Flexgrid Network Planning," [Invited] accepted in IEEE/OSA Journal of Lightwave Technology (JLT), DOI 10.1109/JLT.2014.2315041, 2014.

[Ca12.1] **A. Castro**, L. Velasco, M. Ruiz, M. Klinkowski, J. P. Fernández-Palacios, and D. Careglio, "Dynamic Routing and Spectrum (Re)Allocation in Future Flexgrid Optical Networks," Elsevier Computers Networks (ComNet), vol. 56, pp. 2869-2883, 2012.

[Pe12.1] O. Pedrola, **A. Castro**, L. Velasco, M. Ruiz, J. P. Fernández-Palacios, and D. Careglio, "CAPEX study for Multilayer IP/MPLS over Flexgrid Optical Network," IEEE/OSA Journal of Optical Communications and Networking (JOCN), vol. 4, pp. 639-650, 2012.

11.2.2 Publications in Conferences

[Ca13.1] **A. Castro**, L. Velasco, J. Comellas, and G. Junyent, "Dynamic Restoration in Multi-layer IP/MPLS-over-Flexgrid Networks," in Proc. IEEE Design of Reliable Communication Networks (DRCN), 2013.

[Ca13.2] **A. Castro**, L. Velasco, J. Comellas, and G. Junyent, "Global Concurrent Optimization: Advantages and Opportunities in Flexgrid-based networks," in Proc. IEEE International Conference on Transparent Optical Networks (ICTON), 2013.

[Ca13.3] **A. Castro**, F. Paolucci, F. Fresi, M. Imran, B. Bhowmik, G. Berrettini, G. Meloni, A. Giorgetti, F. Cugini, L. Velasco, L. Poti, and P. Castoldi, "Experimental Demonstration of an Active Stateful PCE Performing Elastic Operations and Hitless Defragmentation," in Proc. European Conference on Optical Communication (ECOC), 2013.

[Ma13] R. Martínez, **A. Castro**, R. Casellas, R. Muñoz, L. Velasco, R. Vilalta, and J. Comellas, "Experimental Validation of Dynamic Restoration in GMPLS-controlled Multi-layer Networks using PCE-based Global Concurrent Optimization," in Proc. IEEE/OSA Optical Fiber Communication Conference (OFC), 2013.

[Ve13] L. Velasco, **A. Castro**, M. Ruiz, "Solving Routing and Spectrum Allocation Related Optimization Problems," Invited tutorial in European Conference on Optical Communication (ECOC), 2013.

[Ca12.2] **A. Castro**, M. Ruiz, L. Velasco, G. Junyent, and J. Comellas, "Path-based Recovery in Flexgrid Optical Networks," in Proc. IEEE International Conference on Transparent Optical Networks (ICTON), 2012.

[Ca12.3] **A. Castro**, R. Martínez, L. Velasco, R. Casellas, R. Muñoz, and J. Comellas, "Experimental Evaluation of a Dynamic PCE-Based Regenerator-Efficient IA-RWA Algorithm in Translucent WSON," in Proc. European Conference on Optical Communication (ECOC), 2012.

[Ca12.4] **A. Castro**, L. Velasco, M. Ruiz, and J. Comellas, "Single-path Provisioning with Multi-path Recovery in Flexgrid Optical Networks," in Proc. International Workshop on Reliable Networks Design and Modeling (RNDM), 2012.

[Pe12.2] O. Pedrola, L. Velasco, **A. Castro**, J. Fernández-Palacios, D. Careglio, and G. Junyent, "CAPEX study for grid dependent multi-layer IP/MPLS-over-EON using relative BV-WSS costs," in Proc IEEE/OSA Optical Fiber Communication Conference (OFC), 2012.

[Ve12.1] L. Velasco, M. Ruiz, **A. Castro**, O. Pedrola, M. Klinkowski, D. Careglio, and J. Comellas, "On the Performance of Flexgrid-based Optical Networks," [Invited] in Proc. IEEE International Conference on Transparent Optical Networks (ICTON), 2012.

11.3 List of Research Projects

11.3.1 European funded projects

GEANT: "Pan-European research and education network that interconnects Europe's National Research and Education Networks", open call "Research and Experimental Assessment of Control Plane Architectures for In-Operation Flexgrid Network Re-Optimization (**REACTION**)", Ref: 238875, 2013-2015.

IDEALIST: "Industry-Driven Elastic and Adaptive Lambda Infrastructure for Service and Transport Networks", Ref: FP7-ICT-2011-8, 2012-2015.

STRONGEST: "Scalable, Tunable and Resilient Optical Networks Guaranteeing Extremely-high Speed Transport", Ref: FP7-247674, 2010-2012.

11.3.2 Spanish funded projects

ELASTIC: "Enhanced Optical Networks Featuring Adaptable and Highly Scalable Multi-Granular Transport", Ref: TEC2011-27310, 2012-2014.

ENGINE: "Engineering Next Generation Optical Transport Networks", Ref: TEC2008-02634, 2009-2011.

11.4 Topics for further research

The algorithms devised in this thesis are being deployed in the pre-commercial UPC's PLAnning Tool for Optical Networks (PLATON). PLATON is currently under development by the PhD thesis of Lluis Gifre in the context of the IDEALIST project. The objective of PLATON is to provide an environment where a defined set of optical network planning algorithms can be executed using high performance and state-of-the-art hardware and software technologies. Furthermore, PLATON focuses on off-line and in-operation network planning. For off-line planning, PLATON can rely on up-to-date computation hardware, such as GPUs, which enable the design of highly parallel algorithms (Fig. 11-1). As for the in-operation planning, PLATON includes a PCEP interface so as to receive PCEP requests from a PCE (Fig. 11-2).

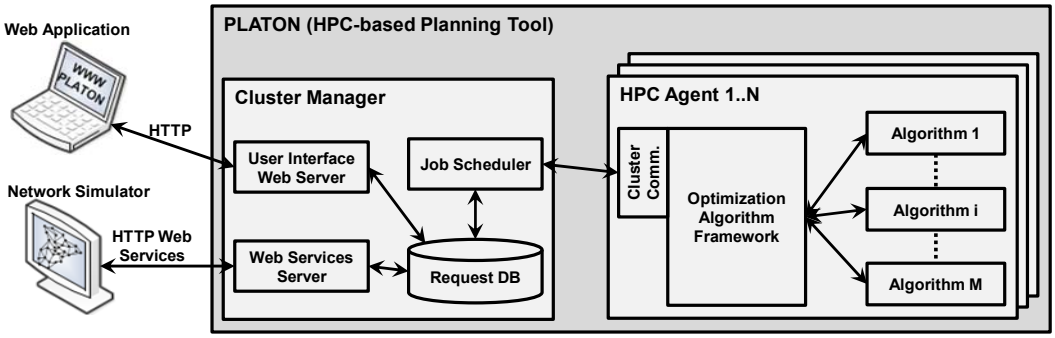

Fig. 11-1 PLATON Architecture – Off-line planning scenario.

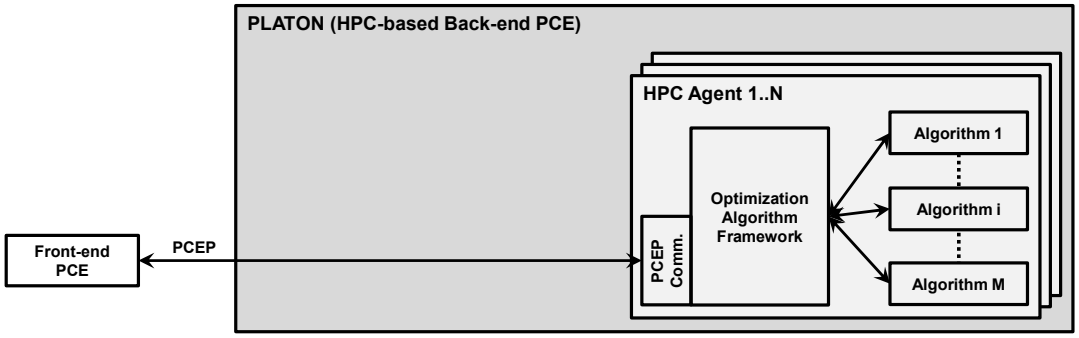

Fig. 11-2 PLATON Architecture – In-operation planning scenario.

This work in progress has already produced two papers:

- Ll. Gifre, F. Paolucci, A. Aguado, R. Casellas, **A. Castro**, F. Cugini, P. Castoldi, L. Velasco, and V. Lopez, "Experimental Assessment of In-Operation Spectrum Defragmentation," Springer Photonic Network Communications, vol. 27, pp. 128-140, 2014.
- Ll. Gifre, **A. Castro**, M. Ruiz, N. Navarro, and L. Velasco, "An In-Operation Planning Tool Architecture for Flexgrid Network Re-Optimization," accepted in IEEE International Conference on Transparent Optical Networks (ICTON), 2014.

Appendix A. **Network Topologies and Traffic Profiles**

This appendix compiles the main network topologies and traffic used along the thesis.

A.1 Network Topologies

Telefonica (TEL)

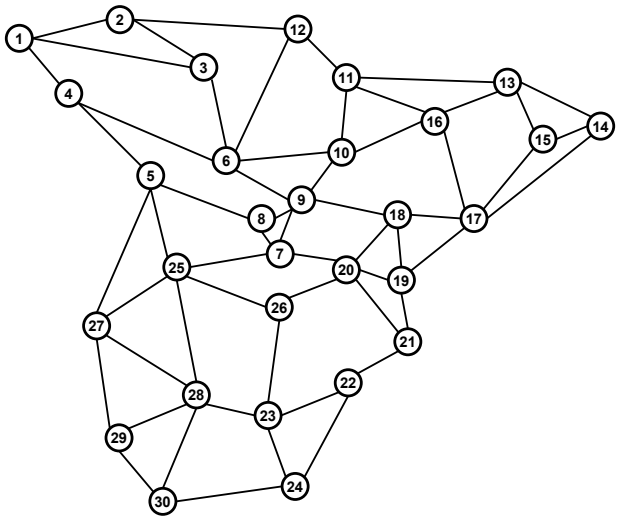

Fig. A-1 30-node and 56-link TEL network topology.

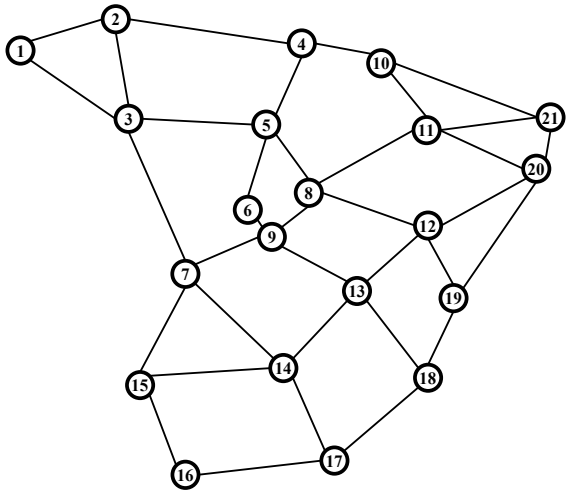

Fig. A-2 21-node and 35-link TEL network topology.

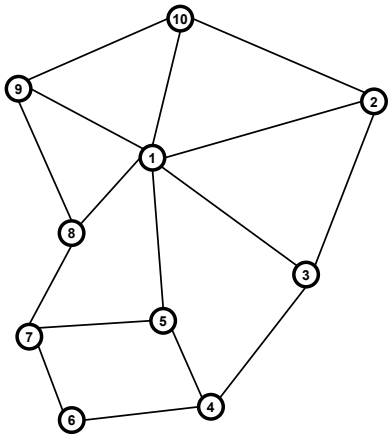

Fig. A-3 10-node and 16-link TEL network topology.

British Telecom (BT)

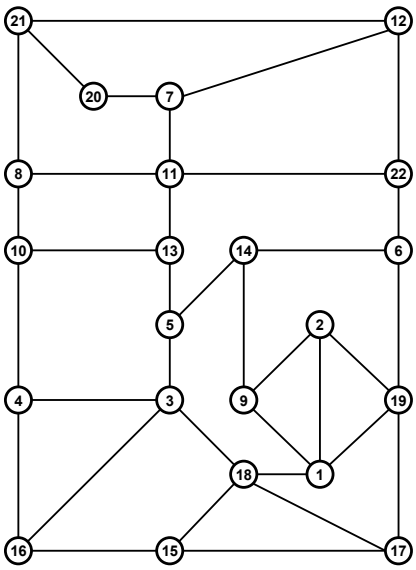

Fig. A-4 22-node and 35-link BT network topology.

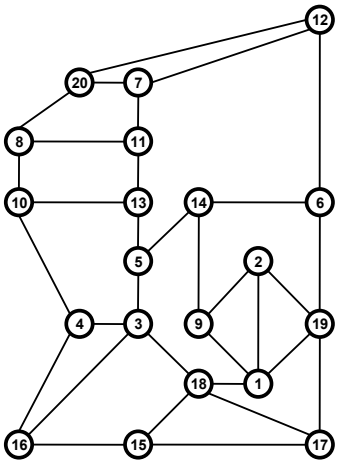

Fig. A-5 20-node and 32-link BT network topology.

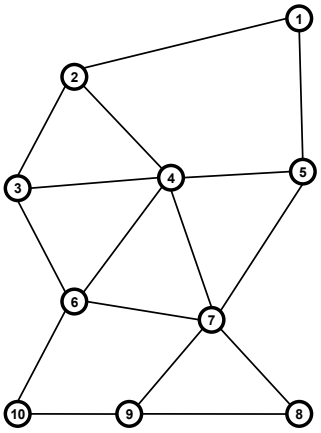

Fig. A-6 10-node 16-link BT network topology.

Deutsche Telekom (DT)

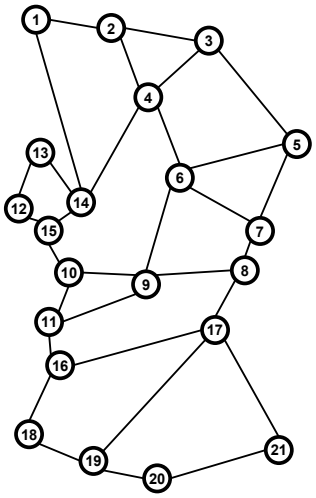

Fig. A-7 21-node and 31-link DT network topology.

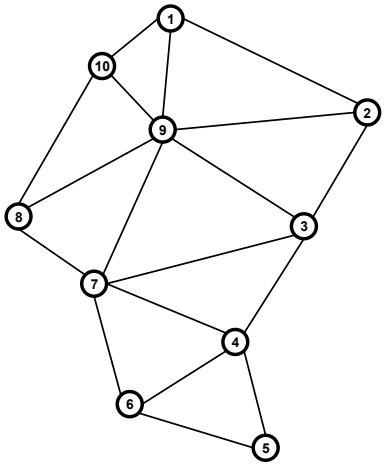

Fig. A-8 10-node and 18-link DT network topology.

European Networks

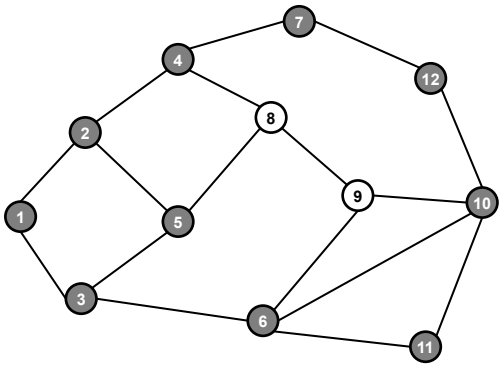

Fig. A-9 EON MLN topology. Colored circles include routers and optical nodes.

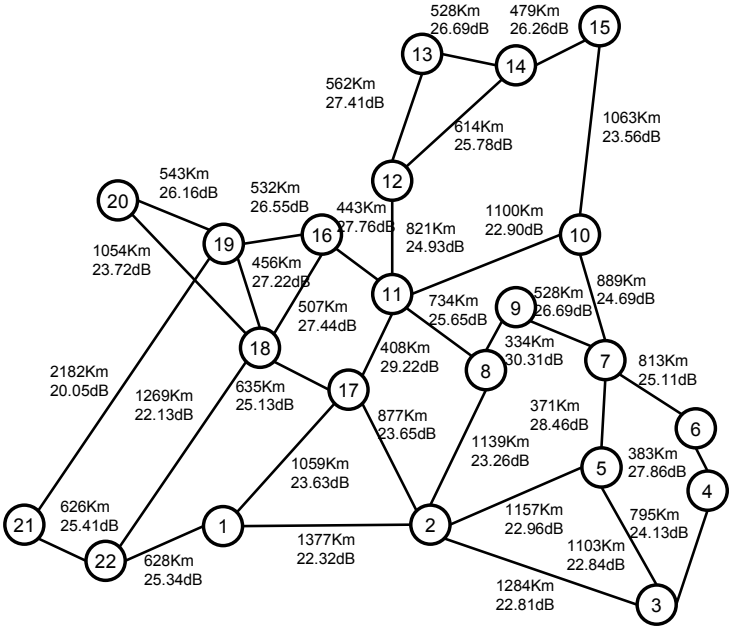

Fig. A-10 22-node and 34-link EON topology.

A.2 Traffic Profiles (TPs)

Table A-1 Static Traffic Profiles

Traffic Profile	Average bit rate (Gb/s)	10 Gb/s		40 Gb/s	
		Conn. (%)	Weighted bit rate	Conn. (%)	Weighted bit rate
TP-1	22.2	74.1	7.4	18.5	7.4
TP-2	24.1	80.0	8.0	13.4	5.4
TP-3	38.4	60.0	6.0	27.0	10.8
TP-4	52.0	40.0	4.0	40.0	16.0
TP-5	66.1	20.0	2.0	53.4	21.4
TP-6	80.0	0.0	0.0	66.7	26.7

Traffic Profile	Average bit rate (Gb/s)	100 Gb/s		400 Gb/s	
		Conn. (%)	Weighted bit rate	Conn. (%)	Weighted bit rate
TP-1	22.2	7.4	7.4	0.0	0.0
TP-2	24.1	5.4	5.4	1.3	5.4
TP-3	38.4	10.8	10.8	2.7	10.8
TP-4	52.0	16.0	16.0	4.0	16.0
TP-5	66.1	21.4	21.4	5.3	21.4
TP-6	80.0	26.7	26.7	6.7	26.7

Table A-2 Time-varying Traffic Profile

Offered average bitrate (Gb/s)	100Gb/s (%)	200Gb/s (%)	400Gb/s (%)	1000Gb/s (%)
100	100	0	0	0
150	70	20	10	0
200	40	40	20	0
250	30	30	40	0
300	10	35	55	0
350	30	10	50	10
400	10	15	65	10

A.3 Bitrate-Spectrum width Conversion for demands above 100 Gb/s

Table A-3 Bitrate-Spectrum width

Bitrate (Gb/s)	Bandwidth (GHz)	#Slots
100	37.5	6
200	62.5	10
400	100	16
1000	200	32

List of Acronyms

ABNO	Application-Based Network Operations
BAREMO	Bulk pAth REstoration in Multi-layer Optical networks
BATIDO	Bitrate squeezing and multipath restoration
BER	Bit Error Rate
BRASERO	BitRate squEezed Recovery Optimization problem
BRKGA	Biased Random Key Genetic Algorithm
BV-OXC	Bandwidth-Variable Wavelength Cross-connect
BV-T	Bandwidth-Variable Transponders
BV-WSS	Bandwidth-Variable Wavelength Selective Switches
CAFES	Compute A FEasible Solution
CAPEX	CAPital EXpenditures
DPP	Dedicated Path Protection
DT	Deutsche Telekom
DWDM	Dense Wavelength Division Multiplexing
DYNAMO	DYNamic restorAtion in Multi-layer MPLS-over-flexgrid Optical networks
ELASTIC	Enhanced Optical Networks Featuring Adaptable and Highly Scalable Multi-Granular Transport
EL-SPRESSO	Elastic-SPRESSO
ENGINE	Engineering Next Generation Optical Transport Networks
EON	European Optical Network
ERO	Explicit Route Object

FA	Forwarding Adjacency
FF	First Fit
Gb/s	Gigabit per second
GCO	Global Concurrent Optimization
GHz	Gigahertz
GMPLS	Generalized Multiprotocol Label Switching
GoS	Grade of Service
GRASP	Greedy Randomized Adaptative Search Procedure
IA-RWA	Impairment Aware RWA
IDEALIST	Industry-Driven Elastic and Adaptive Lambda Infrastructure for Service and Transport Networks
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
ILP	Integer Linear Programming
IP	Internet Protocol
ITU-T	International Telecommunication Union - Telecommunication Standardization Sector
LSC	Lambda Switched Capable
LSP	Label Switched Path
LSP-DB	LSP database
MF-TP	Multiflow Transponder
MIFO	Multilayer MPLS-over-Flexgrid Optimization problem
MILP	Mixed Integer Linear Programming
MLN	Multilayer Network
MPLS	Multiprotocol Label Switching
MPR	Multipath Restoration
NMS	Network Management System
O-OFDM	Optical OFDM
OPEX	Operational EXpenditures
OSA	Optical Society of America
OSNR	Optical Signal-Noise Ratio

OTN	Optical Transport Network
OXC	Optical Cross-Connect
PCC	Path Computation Client
PCE	Path Computation Element
PCEP	Path Computation Element Protocol
PR	Path Relinking
PSC	Packet Switched Capable
QAM	Quadrature amplitude modulation
QoS	Quality of Service
QoT	Quality of Transmission
QPSK	Quadrature Phase Shift Keying
RCL	Restricted Candidate List
REACTION	Research and Experimental Assessment of Control Plane Architectures for In-Operation Flexgrid Network Re-Optimization
RSA	Routing and Spectrum Allocation
RWA	Routing and Wavelength Assignment
SBVT	Sliceable Bandwidth Variable Transponder
SLA	Service Level Agreement
SPP	Shared-Path Protection
SPP-MPR	Single-Path Provisioning Multi-Path Recovery
SPRESSO	SPectrum REaLLOcation (SPRE (LLO) → (SSO)) problem
STRONGEST	Scalable, Tunable and Resilient Optical Networks Guaranteeing Extremely-high Speed Transport
TE	Traffic Engineering
TED	Traffic Engineering Database
TP	Traffic Profile
UPC	Universitat Politècnica de Catalunya (BarcelonaTech)
WCC	Wavelength Continuity Constraint
WSON	Wavelength Switched Optical Network

References

[Ag09] F. Agraz, L. Velasco, J. Perelló, M. Ruiz, S. Spadaro, G. Junyent, and J. Comellas, "Design and Implementation of a GMPLS-Controlled Grooming-capable Optical Transport Network," IEEE/OSA Journal of Optical Communications and Networking (JOCN), vol. 1, pp. A258-A269, 2009.

[Ah12] J. Ahmed, C. Cavdar, P. Monti, and L. Wosinska, "A Dynamic Bulk Provisioning Framework for Concurrent Optimization in PCE-Based DWDM Networks," IEEE/OSA Journal of Lightwave Technology (JLT), vol. 30, pp. 2229-2239, 2012.

[Al11] D. Alvarez, V. Lopez, J. Anamuro, J. Lopez, O. Gonzalez, and J. Aracil, "Utilization of temporary reservation of path computed resources for multi-domain path computation element protocols in DWDM networks" in Proc. Network of the Future (NOF), 2011.

[Am11] N. Amaya, M. Irfan, G. Zervas, K. Banias, M. Garrich, I. Henning, D. Simeonidou, Y. Zhou, A. Lord, K. Smith, V. Rancano, S. Liu, P. Petropoulos, and D. Richardson, "Gridless Optical Networking Field Trial: Flexible Spectrum Switching, Defragmentation and Transport of 10G/40G/100G/555G over 620-km Field Fiber," in Proc. European Conference on Optical Communication (ECOC), 2011.

[As13] A. Asensio, M. Klinkowski, M. Ruiz, V. Lopez, A. Castro, L. Velasco, and J. Comellas, "Impact of aggregation level on the performance of dynamic lightpath adaptation under time-varying traffic," in Proc. Conference on Optical Networking Design and Modeling (ONDM), 2013.

[Bh99] R. Bhandari, "Survivable Networks: Algorithms for Diverse Routing," Kluwer Academic Publishers, 1999.

[Ch08] X. Chu, H. Yin, and X. Li, "Lightpath rerouting in wavelength-routed DWDM networks," OSA Journal of Optical Networking (JON), vol. 7, pp. 721-735, 2008.

[Ch09] X. Chen, M. Chamania, A. Jukan, A. Drummond, and N. Fonseca, "A Multipath Routing Mechanism in Optical Networks with Extremely High Bandwidth Requests," in Proc. IEEE Global Communication Conference (GLOBECOM), 2009.

[Ch10] B. Chen, G. Rouskas, and R.Dutta, "Clustering Methods for Hierarchical Traffic Grooming in Large-Scale Mesh DWDM Networks," IEEE/OSA Journal of Optical Communications and Networking (JOCN), vol 2, pp. 502-514, 2010.

[Ch11] K. Christodoulopoulos, I. Tomkos, and E. Varvarigos, "Elastic bandwidth allocation in flexible OFDM based optical networks," IEEE/OSA Journal of Lightwave Technology (JLT), vol. 29, pp. 1354-1366, 2011.

[Cl05] M. Clouqueur, and W. Grover, "Mesh-restorable networks with enhanced dual-failure restorability properties," Springer Photonic Network Communications (PNET), vol. 9, pp. 7-18, 2005.

[CPLEX] CPLEX, [Online] Available: http://www-01.ibm.com/software/commerce/optimization/ cplex-optimizer/

[Cr13] E. Crabbe, I. Minei, S. Sivabalan, R. Varga, "PCEP Extensions for PCE-initiated LSP Setup in a Stateful PCE Model," IETF draft, work in progress, 2013.

[Cr14] E. Crabbe, J. Medved, I. Minei, and R. Varga, "PCEP Extensions for Stateful PCE," IETF draft, work in progress, 2014.

[Cu08] F. Cugini, N. Sambo, N. Andriolli, A. Giorgetti, L. Valcarenghi, P. Castoldi, E. L. Rouzic, and J. Poirrier, "Enhancing GMPLS signaling protocol for encompassing quality of transmission (QoT) in all-optical networks," IEEE/OSA Journal of Lightwave Technology (JLT), vol. 26, pp. 3318-3328, 2008.

[Cu13.1] F. Cugini, F. Paolucci, G. Meloni, G. Berrettini, M. Secondini, F. Fresi, N. Sambo, L. Pot`ı, and P. Castoldi, "Push-pull defragmentation without traffic disruption in flexible grid optical networks," IEEE/OSA Journal of Lightwave Technology (JLT), vol. 31, no. 1, pp. 125–133, Jan 2013.

[Cu13.2] M. Cuaresma, et al., "Experimental Demonstration of H-PCE with BPG-LS in Elastic Optical Networks," in Proc. European Conference on Optical Communication (ECOC), 2013.

[Da09] A. Das, C. Martel, and B. Mukherjee, "A Partial-Protection Approach Using Multipath Provisioning," in Proc. IEEE International Conference on Communications (ICC), 2009.

[Ei11] M. Eiger, H. Luss and D. Shallcross, "Network restoration under a single link or node failure using Preconfigured Virtual Cycles," Telecommunication Systems Journal, 46, 17-30, 2011.

[Fa06] A. Farrel, J.P. Vasseur, and J. Ash, "A Path Computation Element (PCE)-Based Architecture," IETF RFC-4655, 2006.

[Fe09.1] P. Festa, and M. Resende, "An annotated bibliography of GRASP—part I: algorithms," International Transactions in Operational Research, vol. 16, pp. 1-24, 2009.

[Fe09.2] P. Festa, and M. Resende, "An annotated bibliography of GRASP—part II: applications," International Transactions in Operational Research, vol. 16, pp. 131-172, 2009.

[Fe89] T. Feo, and M. Resende, "A probabilistic heuristic for a computationally difficult set covering problem," Operations Research Letters, vol. 8, pp. 67-71, 1989.

[Fe94] T. Feo, M. Resende, and S. Smith, "A greedy randomized adaptive search procedure for maximum independent set," Operations Research, vol. 42, pp. 860-878, 1994.

[Fe95] T. Feo, and M. Resende, "Greedy randomized adaptive search procedures," Springer Journal of Global Optimization, vol. 6, pp. 109-133, 1995.

[Finisar] Finisar, "Programmable narrow-band filtering using the WaveShaper 1000E and WaveShaper 4000E," white paper, http://www.finisar.com.

[G.694.1] ITU-T G.694.1, "Spectral grids for DWDM applications: DWDM frequency grid," May 2002.

[G.8080] ITU-T G.8080, "Architecture for the automatically switched optical network," 2012.

[Ge10] M. Gendreau, and J.Y. Potvin, "Handbook of Metaheuristics" Springer International Series in Operations Research & Management Science, 2010.

[Ge12] O. Gerstel, M. Jinno, A. Lord, and S. Ben Yoo, "Elastic Optical Networking: A New Dawn for the Optical Layer?" IEEE Communications Magazine (ComMag), vol. 50, pp. s12-s20, 2012.

[Gh08] R. Ghimire, S. Mohan, M. Leary, T. Tidwell, "Concurrent multi-layer restoration scheme for GMPLS based DWDM networks," in Proc. International Conference on Ant Colony and Swarm Intelligence (ANTS), 2008.

[Gi10.1] A. Giorgetti, L. Valcarenghi, F. Cugini, and P. Castoldi, "PCE-based Dynamic Restoration in Wavelength Switched Optical Networks," in Proc. IEEE International Conference on Communications (ICC), 2010.

[Gl96] F. Glover, "Tabu search and adaptive memory programming—advances, applications and challenges," Interfaces in computer science and operations research, Kluwer Academic Publishers, 1996.

[Gl97] F. Glover, and M. Laguna, "Tabu search," Kluwer Academic Publishers, 1997.

[Go10] J. Gonçalves and M. Resende, "A parallel multi population genetic algorithm for a constrained two dimensional orthogonal packing problem," Journal of Combinatorial Optimization, vol. 22, pp. 180-201, 2010.

[Go11] J. Gonçalves, and M. Resende, "Biased random-key genetic algorithms for combinatorial optimization," Springer Journal of Heuristics, vol. 17, pp. 487-525, 2011.

[Gr03] W. Grover, "Mesh-based Survivable Transport Networks: Options and Strategies for Optical, MPLS, SONET and ATM Networking," Prentice Hall, 2003.

[Gr12] H. Gredler, J. Medved, S. Previdi, A. Farrel, and S. Ray et al., "North-Bound Distribution of Link-State and TE Information using BGP," IETF draft, work in progress, 2013.

[Ho06] H. Höller, and S. Voß, "A heuristic approach for combined equipment-planning and routing in multi-layer SDH/DWDM networks," Elsevier European Journal of Operational Research (EJOR), vol. 171, pp. 787-796, 2006.

[Hu05] Y. Huang, J. P. Heritage, and B. Mukherjee, "Connection provisioning with transmission impairment consideration in optical DWDM networks with high-speed channels," IEEE/OSA Journal of Lightwave Technology (JLT), vol. 23, pp. 982-993, 2005.

[Hu11] S. Huang, C. Martel, and B. Mukherjee, "Survivable Multipath Provisioning With Differential Delay Constraint in Telecom Mesh Networks," IEEE/ACM Transactions on Networking (ToN), vol. 19, pp. 657-669, 2011.

[Ji09] M. Jinno, H. Takara, B. Kozicki, Y. Tsukishima, Y. Sone, and S. Matsuoka, "Spectrum-efficient and scalable elastic optical path network: architecture, benefits, and enabling technologies," IEEE Communications Magazine (ComMag), vol. 47, pp. 66-73, 2009.

[Ji10] M. Jinno, B. Kozicki, H. Takara, A. Watanabe, Y. Sone, T. Tanaka, and A. Hirano, "Distance-adaptive spectrum resource allocation in spectrum-sliced elastic optical path network," IEEE Communications Magazine (ComMag), vol.48, pp. 138-145, 2010.

[Ji12] M. Jinno, H. Takara, Y. Sone, K. Yonenaga, and A. Hirano, "Multiflow Optical Transponder for Efficient Multilayer Optical Networking," IEEE Communication Magazine (ComMag), vol. 50, pp. 56 -65, 2012.

[Ki14] D. King, and A. Farrel, "A PCE-based Architecture for Application-based Network Operations," IETF draft, work in progress, 2014.

[Kl11] M. Klinkowski and K. Walkowiak, "Routing and Spectrum Assignment in Spectrum Sliced Elastic Optical Path Network," IEEE Communications Letters, vol. 15, pp. 884-886, 2011.

[Ko06] S. Koo, G. Sahin, and S. Subramaniam, "Dynamic LSP routing in IP/MPLS over DWDM networks," IEEE Journal of Selected Areas in Communications (JSAC), vol. 24, pp. 45–55, 2006.

[Kr95] K. Krishnan, R. Doverspike, and C. Pack, "Improved survivability with multi-layer dynamic routing," IEEE Communications Magazine (ComMag), vol. 33, pp. 62-68, 1995.

[La99] M. Laguna, and R. Martí, "GRASP and path relinking for 2-layer straight line crossing minimization," INFORMS Journal on Computing, vol. 11, pp. 44-52, 1999.

[Le09.1] F. Leplingard, A. Morea, T. Zami, and N. Brogard, "Interest of an adaptive margin for the quality of transmission estimation for lightpath establishment," in Proc. IEEE/OSA Optical Fiber Communication Conference (OFC), 2009.

[Le09.2] Y. Lee, J. Le Roux, D. King, and E. Oki, "Path Computation Element Communication Protocol (PCEP) Requirements and Protocol Extensions in Support of Global Concurrent Optimization," IETF RFC-5557, 2009.

[Le96] K. Lee, and V. Li, "A wavelength rerouting algorithm in wide-area all-optical networks," IEEE/OSA Journal of Lightwave Technology (JLT), vol. 14, pp. 1218–1229, 1996.

[Lo14] V. Lopez, B. Cruz, O. González de Dios, O. Gerstel, N. Amaya, G. Zervas, D. Simeonidou, and J.P. Fernandez-Palacios, "Finding the Target Cost for Sliceable Bandwidth Variable Transponders," IEEE/OSA Journal of Optical Communications and Networking (JOCN), vol. 6, pp. 476-485, 2014.

[Lu05] K. Lu, G. Xiao, and I. Chlamtac, "Analysis of blocking probability for distributed lightpath establishment in DWDM optical networks," IEEE/ACM Transactions on Networking (ToN), vol. 13, pp. 187-197, 2005.

[Ma10] R. Martínez, R. Casellas, R. Muñoz, and T. Tsuritani, "Experimental Translucent-Oriented Routing for Dynamic Lightpath Provisioning in GMPLS-Enabled Wavelength Switched Optical Networks," IEEE/OSA Journal of Lightwave Technology (JLT), vol 28, pp 1241-1255, 2010.

[Mo99] G. Mohan, and C. Murthy, "A time optimal wavelength rerouting algorithm for dynamic traffic in DWDM networks," IEEE/OSA Journal Lightwave Technology (JLT), vol. 17, pp. 406 - 417, 1999.

[Mu11] R. Muñoz, R. Casellas, and R. Martínez, "An Open GMPLS-enabled Control Plane testbed for remote development and experimentation of PCE-based path computation algorithms," in Proc Future Network & Mobile Summit (FUNEMS), 2011.

[No10] T. Noronha, M. Resende, and C. Ribeiro, "A biased random-key genetic algorithm for routing and wavelength assignment," Springer Journal of Global Optimization, vol. 50, pp. 503-518, 2010.

[Ok09.1] E. Oki, T. Takeda, JL. Le Roux, and A. Farrel, "Framework for PCE-Based Inter-Layer MPLS and GMPLS Traffic Engineering," IETF RFC-5623, 2009.

[Ok09.2] E. Oki, T. Takeda and A. Farrel, "Extensions to the Path Computation Element Communication Protocol (PCEP) for Route Exclusions," IETF RFC 5521, 2009.

[OMNet] OMNet++, [Online] Available: http://www.omnetpp.org/

[OpenFlow] Open Networking Foundation, [Online] Available: https://www.opennetworking.org/

[Ou04] C. Ou, J. Zhang, H. Zang, L. Sahasrabuddhe, and B. Mukherjee, "New and Improved Approaches for Shared-Path Protection in DWDM Mesh Networks," IEEE/OSA Journal of Lightwave Technology, vol. 22, pp. 1223-1232, 2004.

[Pa08] S. Pachnicke, T. Paschenda, and P. M. Krummrich, "Physical impairment based regenerator placement and routing in translucent optical networks," in Proc. IEEE/OSA Optical Fiber Communication Conference (OFC), 2008.

[Pa10] F. Palmieri, U. Fiore, and S Ricciardi, "A GRASP-based network re-optimization strategy for improving RWA in multi-constrained optical transport infrastructures," Computer Communications, vol. 33, pp. 1809-1822, 2010.

[Pa12] A. Pages, J. Perello, and S. Spadaro, "Lightpath Fragmentation for Efficient Spectrum Utilization in Dynamic Elastic Optical Networks," in Proc. International Conference on Optical Network Design and Modeling (ONDM), 2012.

[Pa13] F. Paolucci, F. Cugini, A. Giorgetti, N. Sambo, and P. Castoldi, "A survey on the path computation element (PCE) architecture," IEEE Communications Surveys Tutorials, vol. PP, no. 99, pp. 1–23, 2013.

[Pe10] L. Pessoa, M. Resende, and C. Ribeiro, "A hybrid Lagrangean heuristic with GRASP and path relinking for set-k covering," AT&T Labs Research Technical Report; 2010.

[Pe12.3] O. Pedrola, "Cross-Layer Modeling and Optimization of Next-Generation Internet Networks", PhD. Dissertation, B. 16924-2013, 2012.

[Pi04] M. Pióro, and D. Medhi, "Routing, Flow, and Capacity Design in Communication and Computer Networks," Morgan Kaufmann, 2004.

[Po08] Y. Pointurier, M. Brandt-Pearce, S. Subramaniam, and B. Xu, "Crosslayer adaptive routing and wavelength assignment in all-optical networks," IEEE Journal on Selected Areas in Communications (JSAC), vol. 26, pp. 1-13, 2008.

[Ra13] F. Rambach, B. Konrad, L. Dembeck, U. Gebhard, M. Gunkel, M. Quagliotti, L. Serra, and V. Lopez, "A multilayer cost model for metro/core networks," IEEE/OSA Journal of Optical Communications and Networking (JOCN), vol. 5, pp. 210-225, 2013.

[Ra95] R. Ramaswami, and K.N. Sivarajan, "Routing and wavelength assignment in all-optical networks," IEEE/ACM Transactions on Networking, vol. 3, pp. 489-500, 1995.

[Ra99] B. Ramamurthy, H. Feng, D. Datta, J. P. Heritage, and B. Mukherjee, "Transparent versus opaque versus translucent wavelength-routed optical networks," in Proc. IEEE/OSA Optical Fiber Communication Conference (OFC), 1999.

[Re03] M. Resende, and C. Ribeiro, "Greedy randomized adaptive search procedures," Handbook of metaheuristics, Kluwer, 2003.

[Re05] M. Resende, and C. Ribeiro, "GRASP with path-relinking: Recent advances and applications," Metaheuristics: Progress as Real Problem Solvers, Springer, pp. 29-63, 2005.

[Re08] M. Resende, and C. Ribeiro, "GRASP," AT&T Labs Research Technical Report, 2008.

[Re10.1] M. Resende, R. Martí, M. Gallego, and A. Duarte, "GRASP and path relinking for the max-min diversity problem," Elsevier Computers and Operations Research, vol. 37, pp. 498-508, 2010.

[Re10.2] M. Resende, and C. Ribeiro, "Greedy randomized adaptive search procedures: advances and applications," Handbook of metaheuristics, Springer Science+Business Media, 2010.

[Re11] R. Reis, M. Ritt, L. Buriol, and M. Resende, "A biased random-key genetic algorithm for OSPF and DEFT routing to minimize network congestion," International Transactions in Operational Research, vol. 18, pp. 401-423, 2011.

[Ri11] O. Ribal, and A. Morea, "Cost-efficiency of mixed 10-40-100Gb/s networks and elastic optical networks," in Proc. IEEE/OSA Optical Fiber Communication Conference (OFC), 2011.

[Ru11] M. Ruiz, O. Pedrola, L. Velasco, D. Careglio, J. Fernández-Palacios, and G. Junyent, "Survivable IP/MPLS-Over-WSON Multilayer Network Optimization," IEEE/OSA Journal of Optical Communications and Networking (JOCN), vol. 3, pp. 629-640, 2011.

[Sa07] N. Sambo, A. Giorgetti, I. Cerutti, and P. Castoldi, "A contention detection scheme for lightpath restoration in GMPLS networks," IEEE Communication Letters, vol. 11, pp. 820-822, 2007.

[Sa09] E. Salvadori, Y. Ye, C. V. Saradhi, A. Zanardi, H. Woesner, M. Cargagni, G. Galimberti, G. Martinelli, A. Tanzi, and D. L. Fauci, "Distributed optical control plane architectures for handling transmission impairments in transparent optical networks," IEEE/OSA Journal Lightwave Technology (JLT), vol. 27, pp. 2224-2239, 2009.

[Sa11.1] N. Sambo, F. Cugini, G. Bottari, G. Bruno, P. Iovanna, and P. Castoldi, "Lightpath provisioning in wavelength switched optical networks with flexible grid," in Proc. European Conference on Optical Communication (ECOC), 2011.

[Sa11.2] N. Sambo, M. Secondini, F. Cugini, G. Bottari, P. Iovanna, F. Cavaliere, and P. Castoldi, "Modeling and Distributed Provisioning in 10–40–100-Gb/s Multirate Wavelength Switched Optical Networks," IEEE/OSA Journal of Lightwave Technology (JLT), vol. 29, pp. 1248-1257, 2011.

[Sh02] G. Shen, W. Grover, T. H. Cheng, and S. K. Bose, "Sparse placement of electronic switching node for low blocking in translucent optical networks," OSA Journal of Optical Networking (JON), vol. 1, pp. 424-441, 2002.

[Sh07] G. Shen, and R. S. Tucker, "Translucent optical networks: The way forward," IEEE Communication Magazine (ComMag), vol. 45, pp. 48-54, 2007.

[So11] Y. Sone, A. Watanabe, W. Imajuku, Y. Tsukishima, B. Kozicki, H. Takara, and M. Jinno, "Bandwidth Squeezed Restoration in Spectrum-Sliced Elastic Optical Path Networks (SLICE)," IEEE/OSA Journal of Optical Communications and Networking (JOCN), vol. 3, pp. 223-233, 2011.

[St01] J. Strand, and A. Chiu, "Issues for routing in the optical layer," IEEE Communication Magazine (ComMag), vol. 39, pp. 81-87, 2001.

[Su84] J.W. Suurballe, and R. Tarjan, "A Quick Method for Finding Shortest Pairs of Disjoint Paths," Networks, vol. 14, pp. 325-336, 1984.

[Ta07] S. Tak and E. Park, "Restoration performance study of k-shortest disjoint paths in DWDM optical networks," Telecommunication Systems Journal, 34, 13-26, 2007.

[Ta11] T. Takagi, H. Hasegawa, K. Sato, Y. Sone, A. Hirano, and M. Jinno, "Disruption Minimized Spectrum Defragmentation in Elastic Optical Path Networks that Adopt Distance Adaptive Modulation," in Proc. European Conference on Optical Communication (ECOC), 2011.

[Ts08] T. Tsuritani, M. Miyazawa, S. Kashihara, and T. Otani, "Optical path computation element interworking with network management system for transparent mesh networks," in Proc. IEEE/OSA Optical Fiber Communication Conference (OFC), 2008.

[Va09] J.P. Vasseur and JL. Le Roux, "Path Computation Element (PCE) Communication Protocol," IETF RFC 5540, March 2009.

[Va11] C. Vadrevu, R. Wang, and B. Mukherjee, “Degraded Services in Mixed-Line-Rate Networks Using Multipath Routing,” in Proc. IEEE International Conference on Advance Networks and Telecommunications Systems (ANTS), 2011.

[Va12] C. Vadrevu, R. Wang, M. Tornatore, C. Martel, and B. Mukherjee, “Survivable Provisioning in Mixed-Line-Rate Networks Using Multipath Routing,” in Proc. IEEE/OSA Optical Fiber Communication Conference (OFC), 2012.

[Ve09] L. Velasco, S. Spadaro, J. Comellas, and G. Junyent, "Shared-path protection with extra-traffic in ASON/GMPLS ring networks,” OSA Journal of Optical Networking, vol. 8, pp. 130-145, 2009.

[Ve10] L. Velasco, F. Agraz, R. Martínez, R. Casellas, S. Spadaro, R. Muñoz, and G. Junyent, “GMPLS-based Multi-domain Restoration: Analysis, Strategies, Policies and Experimental Assessment,” IEEE/OSA Journal of Optical Communications and Networking (JOCN), vol. 2, pp. 427-441, 2010.

[Ve12.2] L. Velasco, M. Klinkowski, M. Ruiz, and J. Comellas, “Modeling the Routing and Spectrum Allocation Problem for Flexgrid Optical Networks,” Springer Photonic Network Communications (PNET), vol. 24, pp. 177-186, 2012.

[Vi11] J. Villegasa, C. Prinsa, C. Prodhona, A. Medaglia, and N. Velasco, “A GRASP with evolutionary path relinking for the truck and trailer routing problem,” Computers & Operations Research, vol. 38, pp. 1319-1334, 2011.

[Wa11.1] X. Wang, Q. Zhang, I. Kim, P. Palacharla, and M. Sekiya, “Blocking Performance in Dynamic Flexible Grid Optical Networks - What is the Ideal Spectrum Granularity?,” in Proc. European Conference on Optical Communication (ECOC), 2011.

[Wa11.2] Y. Wang, X. Cao, and Y. Pan, “A Study of the Routing and Spectrum Allocation in Spectrum-sliced Elastic Optical Path Networks,” In Proc. IEEE International Conference on Computer Communications (INFOCOM), 2011.

[Wa11.3] X. Wan, L. Wang, N. Hua, H. Zhang, and X. Zheng, “Dynamic Routing and Spectrum Assignment in Flexible Optical Path Networks,” in Proc. IEEE/OSA Optical Fiber Communication Conference (OFC), 2011.

[We11] K. Wen, Y. Yin, D. Geisler, S. Chang, and S. Yoo, “Dynamic On-demand Lightpath Provisioning Using Spectral Defragmentation in Flexible Bandwidth Networks,” in Proc. European Conference on Optical Communication (ECOC), 2011.

[Ya05.1] X. Yang, and B. Ramamurthy, “Dynamic routing in translucent DWDM optical networks: The intradomain case,” Journal of Lightwave Technology (JLT), vol. 23, pp. 955-971, 2005.

[Ya05.2] X. Yang, L. Shen, and B. Ramamurthy, "Survivable lightpath provisioning in DWDM mesh networks under shared path protection and signal quality constraints," IEEE/OSA Journal of Lightwave Technology (JLT), vol. 23, pp. 1556-1567, 2005

[Ya08] W. Yao, and B. Ramamurthy, "Rerouting schemes for dynamic traffic grooming in optical DWDM networks," Elsevier Computer Networks (ComNet), vol. 52, pp. 1891–1904, 2008.

[Ye70] J. Yen, "An algorithm for finding shortest routes from all source nodes to a given destination in general networks", Quarterly of Applied Mathematics, vol. 27, pp. 526–530, 1970.

[Yu05] S. Yuan, and J. Jue, "Dynamic lightpath protection in DWDM mesh networks under wavelength-continuity and risk-disjoint constraints," Elsevier Computer Networks (ComNet), vol. 48, pp. 91-112, 2005.

[Zh00] Y. Zhu, G. Rouskas, and H. Perros, "A Comparison of Allocation Policies in Wavelength Routing Networks," Springer Photonic Networks Communications (PNET), vol. 2, pp. 265-293, 2000.

[Zh02] K. Zhu, and B. Mukherjee, "Traffic Grooming in an Optical DWDM Mesh Network," IEEE Journal of Selected Areas in Communications (JSAC), vol. 20, pp. 122-133, 2002.

[Zh11] X. Zhang, F. Shen, L. Wang, S. Wang, L. Li, and H. Luo, "Two-layer mesh network optimization based on inter-layer decomposition," Springer Photonic Networks Communications (PNET), vol. 21, pp. 310-320, 2011.