Universidade de Aveiro
2013
Departamento de
Electrónica, Telecomunicações e Informática

Tiago Ferreira Grade
Dinis Trota

Planeamento e Optimização de Redes Ópticas Elásticas

Design and Optimization of Elastic Optical Networks

Universidade de Aveiro Departamento de Electrónica, Telecomunicações e Informática
2013

Tiago Ferreira Grade Dinis Trota

Design and Optimization of Elastic Optical Networks

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Electrónica e Telecomunicações, realizada sob a orientação científica do Prof. Dr. Paulo Miguel Nepomuceno Pereira Monteiro, Professor Associado do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro, e do Prof. Dr. Amaro Fernandes de Sousa, Professor Auxiliar do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro.

O júri / The jury

Presidente / President	**Professor Doutor António Luís Jesus Teixeira** Professor Associado do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro
Vogais / Examiners Committee	**Professor Doutor João José de Oliveira Pires** Professor Auxiliar do Departamento de Engenharia Electrotécnica e de Computadores do Instituto Superior Técnico de Lisboa (arguente externo)
	Professor Doutor Paulo Miguel Nepomuceno Pereira Monteiro Professor Associado do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro (orientador)

Agradecimentos / Acknowledgements

Agradeço em primeiro lugar aos meus pais e ao meu avô pelo carinho e suporte que sempre me deram. Sem eles nada disto era possível.

Depois, tenho a agradecer ao Prof. Dr. Paulo Monteiro e ao Prof. Dr. Amaro de Sousa por toda a disponibilidade, orientação e conhecimento transmitido ao longo deste último ano. Ao Marco, pelas sugestões e discussões que foram valiosas para o desenrolar do trabalho.

Por último, um especial agradecimento a todos os meus amigos, em especial ao Gonçalo, Loução, Miguel, Rafael e Silva pela amizade, companheirismo, e pelos momentos vividos ao longo destes anos que ficarão na memória.

Palavras-chave Redes Ópticas Elásticas, Optimização de Redes, Alocação de Espectro, Grelhas Flexíveis de Frequência

Resumo O volume de tráfego nas redes ópticas de transporte tem aumentado nos últimos anos e espera-se que continue a apresentar um crescimento elevado, com os serviços de largura de banda intensivos a serem os principais impulsionadores desse crescimento. Adicionalmente, o tráfego está a tornar-se mais dinâmico, e sendo caracterizado por uma forte heterogeneidade. Assim, de forma a lidar com estas alterações, as redes ópticas de transporte estão a evoluir no sentido de operar com taxas de linha mistas e a utilizar o espectro de forma elástica através do uso de grelhas flexíveis na frequência. Face as esta mudança de paradigma, os operadores de redes de telecomunicações procuram avaliar soluções que permitam às redes de transporte fornecer maior capacidade e escalabilidade. No entanto, devido à redução das margens de lucro, outro parâmetro tido em consideração é o custo da rede, quer de investimento quer operacional. Consequentemente, as ferramentas de planeamento e optimização surgem como instrumentos valiosos para explorar as funcionalidades das redes ópticas de próxima geração, e quantificar o seu impacto na relação custo-benefício.

O objectivo desta dissertação é estudar o problema de dimensionamento de redes ópticas. Para este fim, foi desenvolvido um algoritmo de optimização cuja função objectivo minimiza o custo da rede. Este algoritmo permite considerar a utilização de diferentes equipamentos terminais com diferentes taxas de linha e inclui as funcionalidades de colocação de regeneradores, de *multi-hop grooming*, de *inverse-multiplexing* e de alocação de espectro com baixa granularidade. Além disso, de forma a melhorar a eficiência do algoritmo, utilizou-se uma abordagem *multi-thread* para aproveitar ao máximo os recursos computationais das plataformas actuais.

O algoritmo foi aplicado e avaliado em diferentes cenários de tráfego e de topologias de rede. Os resultados de dimensionamento permitem estudar o compromisso entre os preços dos equipamentos. Também permitem interpretar a importância de cada uma das funcionalidades previamente mencionadas em cada um dos cenários propostos. Os resultados computacionais obtidos revelam ganhos na abordagem *multi-thread*, em particular quando é implementado um mecanismo de partilha de informação entre todas as *threads*.

Keywords

Elastic Optical Networks, Network Optimization, Spectrum Assignment, Flexible Frequency Grid

Abstract

The traffic volume in backbone optical networks has increased in the last few years and it is expected to continue to exhibit a high growth, with bandwidth-hungry services being the key drivers of this growth. Furthermore, the traffic is becoming more dynamic, being characterized by a strong heterogeneity. Thus, in order to cope with these changes, optical transport networks are developing towards the operation with mixed-line rates and the elastic usage of spectrum through flexible frequency grids.
Regarding this paradigm shift, telecommunication network operators seek to assess solutions that enable transport networks to provide capacity enhancement and scalability. However, due to the reduction in profit margins, another parameter taken into account is the network cost, either investment or operational. Therefore, planning and optimization tools arise as valuable instruments to exploit the features of next generation optical networks, and quantify their cost-efficiency impact.
The aim of this dissertation is to study the problem of optical network design. For this purpose, an optimization algorithm was developed where the objective function minimizes the network deployment cost. This algorithm allows the use of different terminal equipments with different line rates and comprises features such as regenerator placement, multi-hop grooming, inverse-multiplexing and spectrum assignment with low granularity. Moreover, in order to improve the algorithm efficiency, a multi-thread approach was used to fully take advantage of computational resources of current platforms.
The algorithm was applied and examined under different traffic and network topology sets. The design outcomes allow us to study the trade-off between equipment prices. Also, allow to understand the importance of each of the previously mentioned features in each of the proposed sets. The computational outcomes obtained show gains in multi-thread approach, particularly when a mechanism that lets sharing information between all threads is implemented.

Contents

List of Acronyms

3R Reamplificatiom, Reshaping and Retiming iii, 6, 27, 39, 42, 45, 46

DWDM Dense Wavelength Division Multiplexing iii, 5, 6

EDFA Erbium-Doped Fiber Amplifier iii, 3

EON Elastic Optical Network iii, 1, 2, 4, 47, 48

FEC Forward Error Correction iii, 6, 8

ILP Integer Linear Programming iii, 11

ITU-T International Telecommunications Union - Telecommunication Standardization Sector iii, 1, 7, 8

MEMS Micro-Electro-Mechanical Systems iii, 10

MLR Mixed Line Rate iii

MSLS Multi-Start Local Search iii, 11, 17, 21, 24, 26, 45, 47

OADM Optical Add-Drop Multiplexer iii, 6

OAM Operations, Administration and Management iii

OCh Optical Channel iii, 7, 14, 19–22, 27–29, 31, 42

OCX Optical Cross Connect iii, 6

ODU Optical Data Unit iii, 7, 8

OMS Optical Multiplex Section iii, 7

OSI Open Systems Interconnect iii, 7

OSNR Optical Signal-to-Noise Ratio iii, 8

OTN Optical Transport Network iii, 3, 4, 7, 8

OTS Optical Transmission Section iii, 7

OTU Optical Transport Unit iii, 7, 8, 13, 14, 17, 26–31, 42

PLC Planar Lightwave Circuit iii, 10

PM Phase Modulation iii, 8

QAM Quadrature Amplitude Modulation iii, 8

ROADM Reconfigurable Optical Add-Drop Multiplexer iii, 6, 7, 9

RSA Routing and Spectrum Allocation iii

SDH Synchronous Digital Hierarchy iii, 3, 8

SLR Single Line Rate iii

WCX Wavelength Cross Connect iii, 10

WDM Wavelength Division Multiplexing iii, 3, 5–10, 18

WSS Wavelength Selective Switch iii, 6, 9, 10, 13

List of Algorithms

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Framework and Motivation

Nowadays, there is an increasing demand for bandwidth due to the appearance of new services such as ultra high-definition television and more recently cloud computing applications. Network traffic is estimated to rise with an annual growth rate between 35% and 50% [1]. Considering a scenario where the initial traffic volume is 12 Tbps for the year of 2010 [2], the more reasonable rate will lead to a volume of approximately 240 Tbps in 2020, while the 50% growth rate means a volume of around 690 Tbps (figure 1.1). Together with the notorious traffic growth, network services have drastically changed throughout the years. Future optical transport networks are facing the challenge to cope with this traffic growth and dynamism. One can start consider scaling the interface rates to meet these challenges. Thus, migrate from the current 10 and 40 Gbps systems towards 100 Gbps systems and above is an expected reality [3]. However, it is not likely that these data rates will be feasible while keeping the 50 GHZ International Telecommunications Union - Telecommunication Standardization Sector (ITU-T) grid. Fitting a 400 Gbps signal into a 50 GHz width would sharply restrict the transmission distance [4, 3]. Therefore, the concept of Elastic Optical Network (EON), or flexgrid is proposed. This concept of EON arose as a way to offer higher efficient usage of optical resources while accommodating higher data-rates [4, 5]. This concept clearly requires changes and further developments on current optical network elements [4].

The term elastic generically refers to the ability of the network to adjust its resources, namely optical spectrum, transmission format or data rate [5]. Hence, two key properties of EONs are:

- flexible access to optical spectrum;
- flexible and dynamic tunable transponders.

Besides the technology advances required on transmission systems, EONs imposes new challenges on network design level. For instance, new routing and spectrum assignment algorithms need to be developed [5, 6].

All this had a great impact on the network evolution scenario: from static networks that handle rather homogeneous traffic to more configurable networks that carries heterogeneous services [7]. Moreover, fixed spectrum grid is becoming outdated and the evolution path is towards the flexible use of network resources to allow accommodating enhanced and higher data rates.

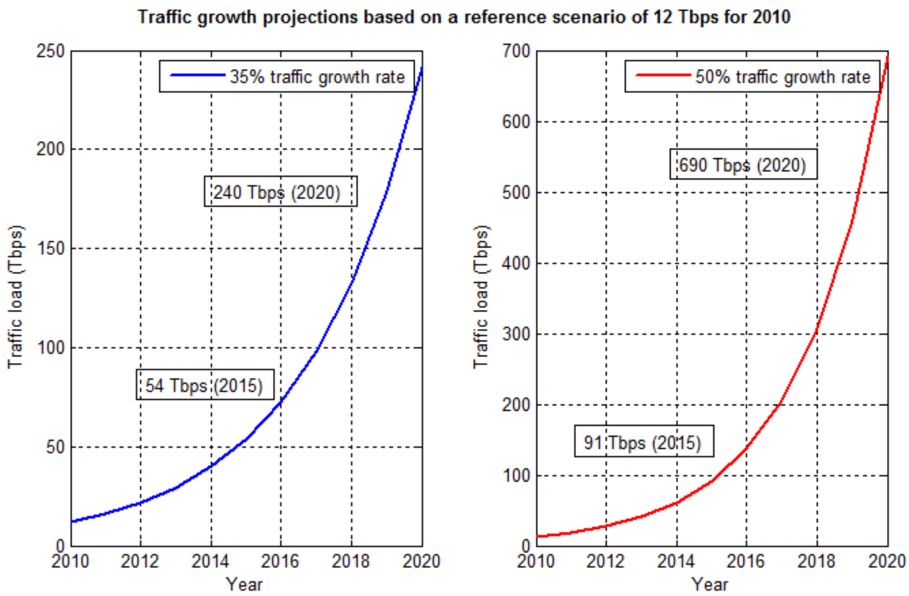


Figure 1.1: Traffic growth projections based on a reference scenario of 12 Tbps in 2010, with 35% and 50% annual increases. Adapted from [1]

1.2 Evolution of Optical Communication Systems and Networks

In the early years, copper and coaxial cables were the transmission mediums employed in the transmission systems. When semiconductor lasers and optical fibers become available, a new era that revolutionize the communication systems arise [8].

Optical fibers are less prone to electromagnetic interferences, have lower losses and provide higher capacities compared to copper or coaxial cable [9]. As a result, optical fiber is a more appropriate medium for transmission over longer distances and with higher speeds.

Evolution of optical transmission systems and networks were shaped and relied on several technology developments, particularly in key components such as the optical fiber, light sources, photodetectors and optical amplifiers [10].

Initially, transmission systems operate at 850 nm, but several impairments limited their capacity [10]. Additional development in both optical sources and photodetectors allowed a deviation in the transmission wavelength from 850 nm to 1300 nm. This change made possible transmission over longer distances without signal regeneration and larger capacity. Finally, 1550 nm systems become common. These systems are characterized by providing the lowest attenuation while having a higher chromatic dispersion than the previous transmission window. The known dispersion problem in this wavelength was solved by using dispersion-shifted fibers and or optical dispersion compensation modules. Thus, it was possible to reach near 10 Gbps rates using this window along with high quality lasers and receivers [10]. Simultaneously, single-mode fibers became the fibers choice for long haul transmisstion since they present higher bandwidth over longer distances compared with multi-mode fibers [8, 10].

Further, the development of Erbium-Doped Fiber Amplifiers (EDFAs) in the late 1980s and early 1990s [9] enabled the emergence of a new generation of optical transmission systems. A major advantage brought by these amplifiers is the ability to amplify several wavelengths simultaneously thus giving a new emphasis to the concept of wavelength transmission. The principle behind Wavelength Division Multiplexing (WDM) is to use several sources operating at different wavelengths to transmit different information streams over the same fiber. The use of WDM technology offers an extra increase in transmission capacity and this way, system capacity may be increased without necessarily increasing the bit rate. The use of both WDM and EDFAs not only offered higher capacity but also cost effectiveness [11]. Deployment of WDM systems with EDFAs started in the mid-1990s [9, 10] .

At the same time, transport protocols and standards keep up with the innovation advances. The so-called first-generation optical networks combined Synchronous Digital Hierarchy (SDH) with optical fiber. The main purpose was to provide capacity enhancement by using fiber as the transmission medium. Then, it was realized that fiber should be used for more than transmission. Hence, the second-generation optical networks introduces the optical layer to the protocol hierarchy. This layer is responsible for switching, routing and other functionalities [9]. Moreover, Optical Transport Network (OTN) arises due to the advent of WDM technology and the need for this layer to become an independent networking layer and have its own management capability. Note that originally SDH networks used a unique wavelength per fiber and were not prepared to offer wavelength support. OTN was built upon the concepts of SDH and is enhanced to operate with higher transmission rates. It provides the required management ground for WDM by, for instance, incorporating overhead to efficiently perform network monitoring and management and by providing powerful forward error correction [7].

It is expected that fiber designs along with technology will keep progressing to meet further demands in optical transport networks [8, 9].

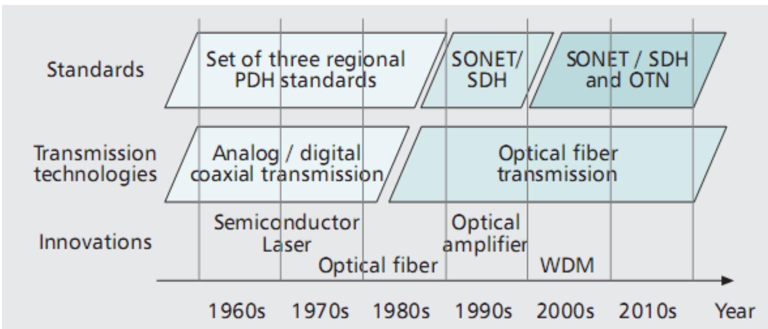


Figure 1.2: Evolution of transport network technologies and standards [11]

1.3 Structure

This dissertation is organized into five chapters as follows:

- Introduction;
- Background and Related Work;
- Optimization Algorithm;
- Case Studies and Computational Results;

- Conclusion.

In the current chapter the motivation behind the study of this topic is presented. An overview on the evolution of transport networks is also given, providing a context for this work. Lastly, the structure of the dissertation and the main contributions yielded are presented.

In the second chapter relevant technological aspects are covered, namely optical network elements, OTN technology and the so-called EONs paradigm. Also, relevant work on EONs is summarized.

In the third chapter the optimization algorithm proposed is explained. Firstly, the heuristic used is described. Then, the problem definition is clearly stated, identifying the input parameters, the optimization goal, the cost model used and the relevant outputs. Then, the algorithm is globally presented, showing how the heuristic was applied to the proposed problem. In particular, the strategy for the routing problem, spectrum assignment and each step of the algorithm are addressed and thoroughly described in the following sections. At last, a small example is given to illustrate the algorithm operations.

In the forth chapter the case studies together with the computational and design results are presented. The traffic profiles and network topologies chosen to test the algorithm are depicted. Separate analyses and conclusions for each OTN functionality are made.

In the final chapter the overall conclusions are presented along with suggestions to further work.

Finally, an appendix is included containing the detailed traffic matrices that were used to achieve the computational results previously showed.

1.4 Contributions

The main contributions of this dissertation are the following:

- Development of an optimization algorithm for OTN design;
- New strategies to perform routing and spectrum assignment;
- Computational evaluation of the optimization algorithm;
- Study the cost-efficiency of flex-grid over fixed grid solutions in different scenarios;

As a result of this work, a conference publication was made:

- P. Monteiro, A. de Sousa, M. Ribeiro, T. Trota, G. Sahin; "Algorithms in the deployment of optical transport networks" (Invited); International Conference on Transparent Optical Networks (ICTON), ICTON, 23-25 June 2013, pp. Mo.B4.3.

Chapter 2

Background and State of the Art

2.1 Network Elements

The nodal architecture and switching technology has evolved from the conventional optical-electrical-optical technologies towards the pure optical systems. The former architecture is characterized by low scalability and increasing cost with traffic growth [7]. The latter brought benefits specially in terms of capital and operational costs. However, all-optical architecture may not be feasible, since traffic may require electrical processing at nodes for regeneration, grooming or wavelength conversion.

2.1.1 Transponders and Muxponders

The purpose of transponders and muxponders is to provide an appropriate optical signal to be transmitted over a WDM system. Typically, these devices perform wavelength conversion and electrical-optical conversion. Thus, these elements are required to enable the transmission of client traffic in WDM networks.

Although similar, a muxponder generally handles multiple client signals. It aggregates lower rate client signals together into a higher line rate. Likewise, it demultiplexes optical signals and forwards them to client ports. A WDM system with muxponders is represented in figure 2.1 [12]. Each port or interface may accept signals from distinct laser sources, traffic types or physical media. Next, each interface wavelength is mapped to another one within the Dense Wavelength Division Multiplexing (DWDM) range and aggregated into a single optical signal. Finally, the signal is launched into the fiber.

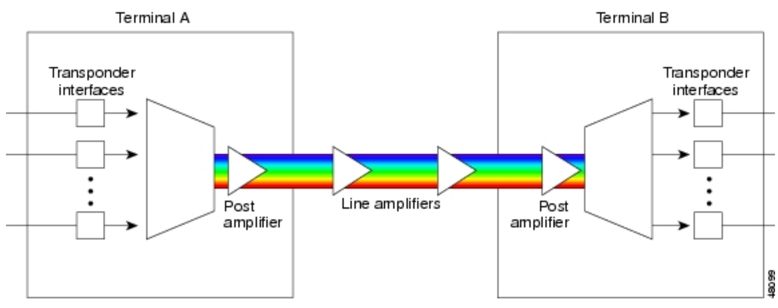


Figure 2.1: WDM system with muxponders [12].

Furthermore, we can characterize a given transponder or muxponder by its data rate and transparent optical reach [13].

Examples of such commercial devices are the Cisco *TXP_MR_10E* and *MXP_2.5G_10E_C* cards [14]. The former processes one 10 Gbps signal into one 10 Gbps DWDM while having the ability to be tuned over different channels. The latter multiplexes four 2.5 Gbps client signals into a 10 Gbps DWDM signal. It includes high level feature such as onboard Forward Error Correction (FEC) or monitoring and management of the optical interfaces.

2.1.2 Regenerators

The standard regeneration technique is known as Reamplificatiom, Reshaping and Retiming (3R). 3R regeneration is needed after an optical signal transmits for a long distance and becomes deteriorated or due to waveform distortion as the signal transverse optical nodes [15]. Optical transmission is affected by many factors such as dispersion, non-linear effects, and crosstalk [9, 10], which may eventually degrade an optical signal to be unrecognisable at the receiver.

Generally, 3R regeneration is based on optical-electronic-optical conversion. However, it is also possible to carry out 3R regeneration in all-optical domain without converting optical signal into electronic signal. The advantage of all-optical 3R regeneration is its bit-rate and framing protocol transparency [9].

2.1.3 Optical Add-Drop Multiplexers

The concept of wavelength add and drop in the nodes arise with the emergence of WDM, and were first implemented by the so-called Optical Add-Drop Multiplexers (OADMs). OADMs are network elements that enable optical-bypass: when a certain wavelength reaches a node, if it does not require processing at that node it is passed through in the optical domain [9, 7, 16]. Additionally, wavelengths could be added and dropped into the WDM stream. This differs from the previous architectures where every wavelength needed to be converted to electrical domain, even the ones that were not locally terminated.

OADMs may be seen as static network elements since manual reconfiguration of the add-drop structure is needed, this is, the wavelengths to be added or dropped need to be previously configured [7]. When fully configurable OADMs arised, also called Reconfigurable Optical Add-Drop Multiplexers (ROADMs), any wavelength could be added or dropped dynamically through software control. The major benefits of this outcome comprise huge flexibility to network design and operational cost advantages [16, 17]. Through the years, ROADMs suffered evolutions in its architecture starting from the traditional 2-degree fixed wavelengths add-drop up to colorless, directionless and contentionless add-drop architectures [16, 18]. These last functionalities improve wavelength agility by implementing add-drop structures that are not color or direction specific and allow multiple copies of the same wavelength. Furthermore, different implementations of ROADMs are possible by using different building blocks, such as Wavelength Selective Switchs (WSSs) or Optical Cross Connects (OCXs) [19]

The above scheme represents a possible implementation for a colorless, directionless and contentionless ROADM based on WSSs and optical splitters. As can be seen, only one add-drop structure is needed and each port supports any wavelength and may connects to any degree.

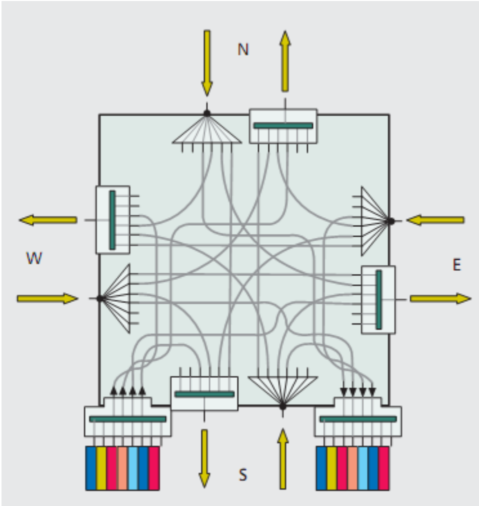


Figure 2.2: CDC ROADM design [16].

Besides the added flexibility, the ROADMs changed the switching from electrical into optical domain. Consequently, we progressively shifted from an architecture where all node ingress traffic is electronically terminated to one where a higher traffic volume remains in the optical domain through the network.

2.2 Optical Transport Network

OTN is a layer 1, in the Open Systems Interconnect (OSI) model, network technology that enables accommodation of several client signals at wavelength rates and sub-wavelength rates [13].

ITU-T defines an optical network based on domains and layers. Two domains are defined: the Inter-domain interface (IrDI), which represents the physical interface between different operators and the Intra-domain interface (IaDI) that is also a physcial interface but within the domain of a single operator. The optical layers are the Optical Channel (OCh), Optical Multiplex Section (OMS) and Optical Transmission Section (OTS) [20]. Each OCh transports a digital Optical Transport Unit (OTU) signal using one wavelength. The OMS layer is responsible for WDM multiplexing and to include further appropriate overhead. This layer refers to the section between an optical multiplexer and demultiplexer. Finally, the OTS is generally the fiber section between two optical amplifiers.

Concerning the interfaces, OTN specify client signal mapping into Optical Data Unit (ODU)k (k = 1,2,3 and 4), which have bit-rates of approximately 2.5 Gbps, 10 Gbps, 40 Gbps and 100 Gbps. The multiplexed ODUk is then transported as an OTUk signal. The main difference between ODU and OTU signals is the FEC overhead contained in the OTU frame (figure 2.3). The FEC functionality specified is added by OTN equipment on the transmitter side and is decoded on the receiver. Currently, OTN is offering the following line rates: OTU1, OTU2, OTU3 and OTU4 [21] supporting from SDH to 100G Ethernet signals.

Due to the emergence of new client signals and the need to be prepared for future evolutions, a new ODU was recently specified, named ODUflex. Differently from the traditional ODUs, this one provides a wide variety of payload sizes, since it can be properly sized to fit

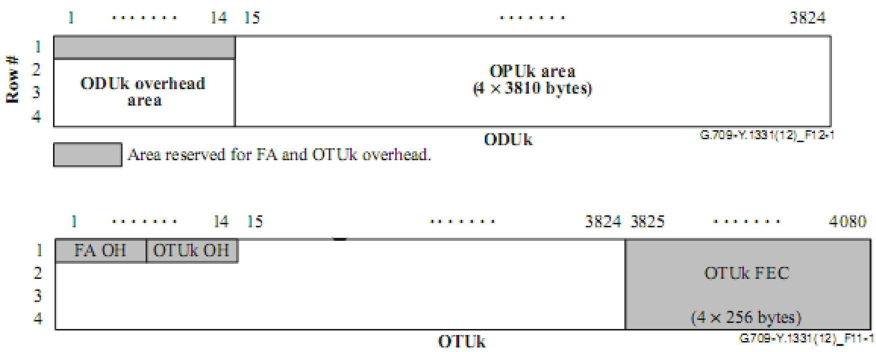


Figure 2.3: ODUk and OTUk frame structure [21].

any client rate, thus avoiding wastage of bandwidth. Consequently, ODUflex offers higher efficiency in signal transmission [13, 21].

Moreover, OTN assures highly efficient administration and management functionalities, while targeting the needs of WDM technology. Overhead bytes are, for example, used to provide monitoring, alarm indication and protection switching [11].

It is believed that OTN will continue to be the preferred technology to support future network services. Hereafter, enhancements on the standard will be required, such as new OTUs definitions (OTU5).

2.3 Elastic Optical Networks

2.3.1 Flexible Grid

The ITU-T defines a set of standards and recommendations for network operations. The standard named ITU-T G.694.1, characterizes spectral grids defining the center frequency along with the channel spacing, which can range from 12.5 GHz to 100 GHz and wider. The most common nowadays are the 50 and 100 GHz spacing. The optical components present on an optical network, such as transmitters, are designed to have frequency locking mechanisms that match with the grid granularity. While the introduction of coherent techniques and efficient modulation schemes enables 100 Gbps transmission within a 50 GHz channel, studies suggest that transmission of 400 Gbps and beyond is harder to implement in such a rigid grid. One can keep increasing the spectral efficiency of the modulation format in order to squeeze a 400 Gbps signal to fit in a 50 GHz channel spacing. However, adopting higher spectral efficiency modulation formats would only be suitable for short transmission distances due to the consequent need for higher Optical Signal-to-Noise Ratio (OSNR), representing a drawback for optical transport networks. Thus, alternatives to the ITU-T 50 GHz frequency grid for transmission of WDM channels in an optical network are of great interest. Flex-grid appears as a valid paradigm that meets the needs of future bandwidth demands. In general, a flexible frequency grid offers the possibility to allocate resources with granularities finer than a channel capacity. Figure 2.4 shows the allocation difference between the two approaches. The values of bandwidth for the 400 Gb/s and 1 Tb/s are obtained assuming Phase Modulation (PM) 16-Quadrature Amplitude Modulation (QAM) and PM 32-QAM modulation format, respectively.

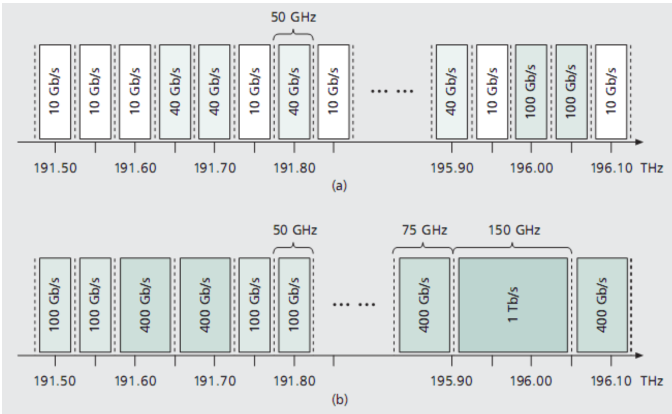


Figure 2.4: C-band 50 GHz a) fixed grid; b) flexible grid [16]

For the flexible WDM grid, the ITU-T defines the nominal central frequency, in THz, for the available frequency slots as follows,

$$193.1 + n \times 0.00625, \tag{2.1}$$

Where n is a positive or negative integer including 0 and 0,00625 is the nominal center frequency granularity in THz. The slot width is defined by,

$$12.5 \times m, \tag{2.2}$$

Where m is a positive integer and 12.5 is the slot width granularity in GHz. Any combination of slots is allowed as long as there is no overlap between two of them. An example of using this grid is shown in figure 2.5. It is represented two 50 GHz slots and two 75 GHz slots, and the values of m and n used to define the center frequency and the slot width. For instance, the unallocated space can be further used as an additional 50 GHz slot ($n = 8$, $m = 4$) or two 25 GHz slots ($n = 10$, $m = 2$).

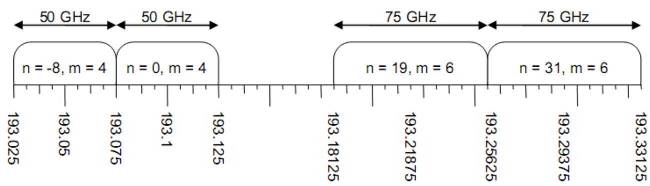


Figure 2.5: Flexible Grid example

The granularity of the center frequency and the channel width defined by this ITU-standard were chosen in order to make this grid compatible with the fixed one. This means that a 50 GHz fixed grid can also be described via suitable choices of parameters n and m.

2.3.2 Enabling Technologies

To proper address this flexgrid paradigm, one needs flexible and adaptive network elements that can adapt to the traffic needs. Generally, implementing a flexible optical network depends on two critical technologies – bandwidth-variable WSS, key element on a flexible bandwidth ROADM, and bandwidth-variable optical transponders (figure 2.6).

In traditional WDM networks, ROADMs are prepared to use frequency grids of 50 or 100 GHz channel spacing. A system that supports mixed channels will require flexible add/drop bandwidths as different channels may have different bandwidths. Bandwidth-variable WSS provides add and drop functionalities for local bandwidth variable signals as well as routing capabilities for transit signals. Incoming optical signals, with different optical bandwidths and center frequencies, can be routed to any of the output direction, as a regular WSS as shown in figure 2.6. These devices may be based on Micro-Electro-Mechanical Systems (MEMS), liquid crystals or Planar Lightwave Circuits (PLCs) technologies [16, 22].

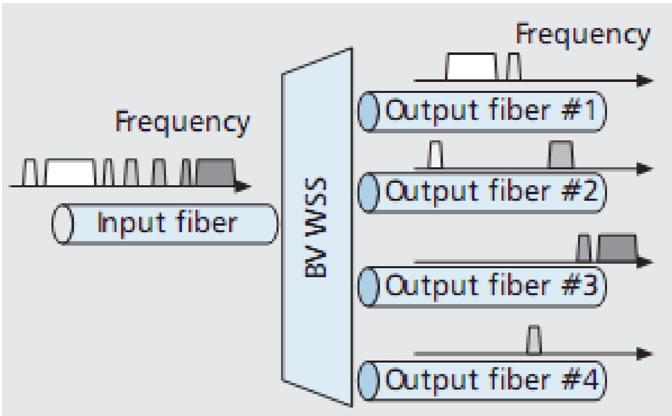


Figure 2.6: Bandwidth-variable WSS [21]

Opposite to the fixed WDM networks, where there is only one way to implement a given demand because the bit-rate, the spectrum and the optical reach are fixed, it is expected a flexible network to assign a given modulation format according to the optical reach and bit-rate. So, bandwidth-variable transponders are necessary to adjust modulation format according to the bit-rate, dynamically, in order to adjust for the better bandwidth allocation, generating an optical signal using only enough spectral resource to transmit it. At the same time, the existing bandwidth-variable Wavelength Cross Connects (WCXs) on the route of the optical path, allocates a cross-connection with the agreeing spectrum bandwidth. A few transponders implementations are suggested in, all provided of the flexibility specified previously.

When implementing a flexible grid based network, issues such as:

- nonlinearities from mixed-signal formats;
- nonlinearities from bit rates;
- optical power control.

must be considered, especially because the number of channels varies dynamically. Also, management issues must be addressed such as bandwidth assignment. The latter is particularly important since the spectrum of a flexible grid based network will vary dynamically as demands are implemented and dropped out, which creates fragmentation. This implies the need for reallocating mechanisms for spectrum to be efficiently used. Besides the optical components, control plane and network management need to be readdressed. Different routing and wavelength assignment strategies must incorporate the feature of channels with different spectral widths [23].

Chapter 3

Optimization Algorithm

3.1 Multi-Start Local Search Meta-Heuristic

In order to address and solve the challenging optimization problem of optical networks design, which is going to be described further in section 3.2, one can use mathematical methods, such as Integer Linear Programming (ILP), or heuristic approaches. The latter are preferable when the computational complexity becomes a limitation. This happens for larger network topologies, heavy traffic loads and when several constraints need to be met. In those cases one can trade optimality with computational time [3]. Moreover, according to the objective function defined, different design strategies may arise. For instance, focusing on network cost, power consumption [24] or spectrum usage will lead to different design solutions. Here, network cost minimization is the primary target.

The algorithm proposed in this work is based on a Multi-Start Local Search (MSLS) meta-heuristic. In algorithm 1 below, the framework of this heuristic is presented.

We can divide the MSLS into two steps. In the first one, an initial *Solution*, named greedy solution, is built (line 4). The purpose of the greedy solution is to generate a rather simple, non-extensive solution. Also, it feeds the local search, which is the second step of the MSLS meta-heuristic. The local search purpose is to improve the greedy solution. This is done by going individually through each of the neighbours of *Solution* (line 7), remove it from the current solution and reinsert it again (line 8). Afterwards, $Neighbour$ has the best neighbour solution from *Solution*. Every time a local search cycle finds a better $Neighbour$ than the current *Solution*, it repeats with this new solution, otherwise it ends. When local search ends, *Solution* is the local minimum solution. In the end, the $Incumbent$, which had an infinity cost and was initialized empty in the beginning, will carry the best among all local minimum solutions. The heuristic repeats itself as long as the initially defined execution time is not reached. Note that there is a randomness nature while building greedy solutions.

3.2 Problem Definition

Prior to the definition of our problem and its aim, it is relevant to clearly introduce the algorithm input requirements.

Firstly, the user must provide the network topology in which the algorithm will be applied. This means specifying both the nodes and physical links. The network is represented by a graph, $G = (N, E)$, where N is the set of nodes and E is the set of physical links. Each link

Algorithm 1: Multi-Start Local Search Meta-Heuristic

```
c_Incumbent ← +∞
Incumbent ← {}
while time do
    Solution ← greedy()
    while !finish do
        Neighbour ← Solution
        for each neighbour of Solution set do
            Compute Sol
            if c_Sol < c_Neighbour then
                Neighbour ← Sol
            end
        end
        if c_Neighbour < c_Solution then
            Solution ← Neighbour
        end
        else
            finish
        end
    end
    if c_Solution < c_Incumbent then
        Incumbent ← Solution
    end
end
```

is represented by a pair of nodes, $\{i, j\} \in E$, that are directly connected through a fiber, and $l_{\{i,j\}}$ is the length of the respective fiber. Furthermore, the capacity of each fiber, w, given in the number of frequency slots, is also an input parameter. We assume bidirectional fibers with 4 THz of available spectrum, corresponding to 160 frequency slots of 25 GHz width each. Despite this adopted value, other higher and practical capacity values may be considered and applied to the problem [13].

Secondly, the client demands to be supported by the network should be listed. Every single demand d is represented by its source node s_d, target node t_d, bandwidth unit b_d and number of bandwidth units n_d. We consider bandwidth units of 10 Gbps, 40 Gbps and 100 Gbps.

Finally, the algorithm needs the transmission options together with the respective cost model of the network equipments. For this study we select eleven possible transmission alternatives, which are shown in Table 3.1. Their properties and cost values were taken from [13]. Only the first nine alternatives were used in all case studies, while the last two were only applied to a particular case study. One of the reasons not to use those alternatives is due to the fact that the 400G equipments are only expected to be available in 2018 and consequently their costs are future estimations, while the rest of the cost values reflect current estimations. These estimations, according to the author, reflect either average values of actual prices or estimations based on technology evolutions. All the values are normalized to the cost of a 10 Gbps transponder with a transparent reach of 750 km. Regarding the optical transparent reach, the author assumes the use of standard single-mode fiber as well as coherent transmission systems.

Transmission Options							
o_t	w_t	n_t	b_t	c_t	r_t	l_t	Obs.
1	50	1	40	6.0	9.6	2500	1 OTU3 for 1x40G
1	50	4	10	5.0	9.6	2500	1 OTU3 for 4x10G
1	50	1	100	15.0	24.0	2000	1 OTU4 for 1x100G
1	50	2	40	16.0	24.0	2000	1 OTU4 for 2x40G
1	50	10	10	13.0	24.0	2000	1 OTU4 for 10x10G
3	50	1	100	20.0	9.6	2500	3 OTU3 for 1x100G
1	75	1	100	22.0	35.2	2500	1 OTU4 for 1x100G .
1	75	2	40	19.0	35.2	2500	1 OTU4 for 2x40G
1	75	10	10	16.0	35.2	2500	1 OTU4 for 10x10G
1	100	4	100	16.0	24	1000	1 OTU5 for 4x100G
1	100	10	40	18	24	1000	1 OTU5 for 10x40G

Table 3.1: Transmission options and their properties.

Each alternative t of Table 3.1 is defined by the number of light paths required o_t, frequency width w_t, number of input ports n_t, port bandwidth unit b_t (in Gbps), cost of end node equipment c_t, cost of each 3R regenerator r_t and maximum transparent reach l_t (in Km). Analysing the table above more carefully one can see that there are three line-rates available, namely 40 Gbps, 100 Gbps and 400 Gbps. Transmission is made possible using transponders of such line rates or using muxponders instead. Muxponders allow traffic aggregation. For example, it is possible to aggregate ten units of 10 Gbps to be transmitted at 100 Gbps. In the particular case of 100 Gbps, multiple distinct options are provided. They differ in the spectral width, enabling fixed-grid and flex-grid solutions, equipment costs and optical reach. Also, inverse-multiplexing is considered in our problem (sixth entry in Table 3.1). Thus, the corresponding c_t represents the cost of inverse-multiplexing hardware in the end-nodes, whereas r_t and l_t are the values of 3R regenerator and optical reach of a 40 Gbps OTU, since the transmission is made with 40 Gbps light paths. In addition, we consider that 3R regenerators can be placed only at the nodes. The optical degradation suffered by a light path traversing a node without regeneration is equivalent to the impairment incurred due to the transmission over a 160 Km fiber link, assuming that an optical channel traverse two WSS for each bypass node [25].

The problem can be defined as follows, given:

1. the fiber network which is represented by a graph $G = (N, E)$;
2. a set of demands to be transported;
3. a set of transmission formats;

the objective is to find the network configuration:

1. that minimizes the deployment cost while guaranteeing that all client demands are routed;
2. in case of multiple minimal costs solutions, that minimizes the highest assigned slot number;

imposed to the ensuing constraints:

1. fibers with a limited frequency spectrum of 160 slots;
2. ensure spectrum continuity, slot contiguity and non-overlapping spectrum assignment;

In the end, the algorithm will have computed a solution that determines the network design for the given input parameters. That solution comprises a full list of the OTUs/OChs established and the corresponding spectrum slots assigned individually to each. Furthermore, beyond computational information, valuable information regarding fiber fragmentation, weight of each network equipment in the total cost and multi-hop grooming percentages among others are outputted, allowing us to perform an analysis in different perspectives and draw several conclusions.

3.3 Candidate Paths Strategy

An important requirement of our problem is the need to assign paths to client demands, so they can be routed through the network. This issue was addressed by adopting a candidate paths strategy. We consider two alternatives to route client traffic. The first one is by establishing new optical connections, which means assigning a path in the fiber network, choosing an appropriate transmission format and deploying network equipment. The second option uses spare capacity of pre-established optical connections, thus routing the demands over different light paths. As a consequence, we need two sets of candidate paths to be built:

1. a set of candidate paths are computed on the fiber graph;
2. a set of candidate paths are computed on the logical graph;

The fiber graph was already introduced, and it is an input parameter of the problem. To obtain the first set of candidate paths, the K-shortest paths on that graph are initially computed based on the lengths of the fibers. Then, for each transmission format/OTU alternative, the associated costs are computed applied to each of the previous paths calculated. The costs comprise the already mentioned end node equipments plus the 3R regenerators. Finally, the paths are ranked by cost, forming in this way the set of candidate paths of the fiber graph.

All the OTUs/OChs established among source-destination nodes constitute the logical graph. However, at the beginning of the algorithm the network is empty since there are no OTUs/OChs created. So, this graph is formed by one logical link between each of the source-destination nodes enabling the computation of a set of K-shortest paths. The costs of the logical links between each pair of nodes is given by their shortest path in the fiber graph. Lastly, those paths are ranked by number of hops forming the second set of candidate paths.

Now, the algorithms necessary to implement this approach are described. Firstly, the MPS algorithm [26], here named K-shortest paths algorithm, was implemented (algorithm 2). This algorithm is used to generate and rank k loop-less paths. Generically, consider the graph $G = (N, A)$, where N is a set of nodes and A is a set of arcs defined as a pair of nodes, $a_l = (i, j)$. Each arc a_l has associated a positive cost w_l. A path from node i to node j, represented by p_{ij}, is a sequence of nodes and arcs such that all the nodes belong to S and arcs belong to A. Moreover, a path is said to be loop-less if all of its nodes are different. The cost of the path p is denoted by $w(p)$, while the minimum cost path from node i towards destination node t is denoted by c_i. It is intended to determine a set of K loop-less paths, p_k $(1 < k < K)$, such as $w(p_k) \leq w(p_{k+1})$ for $1 < k < K - 1$.

The first for loop (line 3) determines the minimum cost paths from every network node i towards the destination nodes t, p_{it} and their respective costs c_i. These paths are obtained using Dijkstra algorithm (algorithm 3). The next for loop employs an operation that consists of changing the initial arc costs. Here, the arc costs are redefined and the new ones are stated in line 7. This operation has the following property that simplifies the number of operations made by the algorithm: the minimum cost path between any node i and destination node t has null cost. In line 9, set A is sorted in such a way that, for any pair of arcs $(k, j), (i, l) \in A$, $(k, j) < (i, l)$ if $k < i$ or if $k = i$ and $w'_{kj} \leq w'_{il}$. This arrangement is named sorted forward star form [26] and it allows determining easily the next proper path to be inserted in set X, if it exists.

Another important operation used in this algorithm is the paths concatenation. As the name suggests, when the terminal node of a path p is the initial node of a path q, then both can be concatenate originating $p \diamond q$. Also, concatenation of a path p with an arc a_l can be made and is denoted as $p \diamond a_l$. Note that concatenating loop-less paths does not result necessarily in a loop-less path. In fact, paths with loops are allowed to belong to X and yet are used to generate new paths. It is also important to highlight that each path belonging to X has associated a node called deviation node, that is a reference node and it is recursively determined.

The main cycle (line 14) is executed as long as there are still paths to determine or set X is not empty. In each iteration, the path with lower cost within X is picked and also removed from X. If p is loop-less, then it is considered the kth shortest path. Then, regardless p is loop-less or not, the algorithm uses it to build new paths starting from its deviation node d until the last but one or while the subpath p_{sd} is loop-less (line 23). At this point, an arc a_l is chosen to be concatenated to p_{sd}. This arc must be obviously different from the one in p so a distinct path is generated. To determine a_l one must recall how the arcs are organized in set A. As set A is described in the sorted forward star form, it is straightforward to conclude that one must decide upon the arc positioned after the arc of p at d (line 25). This way, we guarantee the shortest path between d and t and that the new generated path never belonged to X.

Although set X allows paths with loops, in order to decrease the number of non feasible paths in X, a_l is chosen provided that it does not form a loop with p (line 26). Whenever determining a_l, it is possible that such arc does not exist and in those cases no path is joined to X (line 29).

Finally, if a_l does exists, the new path is built (line 31) using the shortest path from j, head node of a_l, to t. Since we are using reduced costs, this path has a associated null cost and the total cost of this path is given by the cost of p plus the cost of inserting arc a_l (line 33). This new path is inserted inserted in X and the corresponding deviation node is set to

Algorithm 2: K Shortest Paths

1 $c_t \leftarrow 0$
2 $p_{tt} \leftarrow \{\}$
3 **for** $i \in N \setminus \{t\}$ **do**
4   Compute p_{it} and c_i
5 **end**
6 **for** $(i, j) \in A$ **do**
7     $w'_{ij} = w_{ij} + c_j - c_i$
8 **end**
9 Rank A: $i_l < i_{l+1}$ or $[i_l = i_{l+1}$ and $w'_l \leq w'_{l+1}]$
10 $X \leftarrow \{p_{st}\}$
11 $d(p_{st}) \leftarrow s$
12 $w'(p_{st}) \leftarrow 0$
13 $k \leftarrow 0$
14 **while** $k < K$ *and* $X \neq \{\}$ **do**
15     $p \leftarrow$ minimum cost path of set X
16     $X \leftarrow X - \{p\}$
17   **if** p *is loop-less* **then**
18         $k \leftarrow k + 1$
19         $p_k \leftarrow p$
20   **end**
21     $d \leftarrow d(p)$
22     $p_{sd} \leftarrow$ partial path of p from node s to deviation node d
23   **while** p_{sd} *is loop-less and* $d \neq t$ **do**
24         $l \leftarrow$ position in A of the output arc of p at d
25         $l \leftarrow l + 1$
26   **while** $i_l = d$ *and* $p_{sd} \diamond a_l$ *forms a loop* **do**
27             $l \leftarrow l + 1$
28   **end**
29   **if** $i_l = d$ **then**
30             $j \leftarrow j_l$
31             $p_{st} \leftarrow p_{sd} \diamond a_l \diamond p_{jt}$
32             $d(p_{st}) \leftarrow d$
33             $w'(p_{st}) \leftarrow w'(p) + w'(a_l)$
34             $X \leftarrow X \cup \{p_{st}\}$
35   **end**
36         $d \leftarrow$ next node in p
37   **end**
38 **end**

d. Before repeating a new iteration in cycle 23, deviation node d is updated to the following node of path p.

The Dijkstra algorithm is important right in the beginning of the K-shortest paths algorithm as stated before. It provides the cost of the paths (line 13) and those paths are built

Algorithm 3: Dijkstra

```
1  S ← {}
2  p_t ← {}
3  c_t = 0
4  for i ∈ N \ {t} do
5  |   c_i = +∞
6  |   p_i ← {}
7  end
8  while |S| < |N| do
9  |   choose i ∉ S such that c_i = min_{j∉S} c_j
10 |   S = S ∪ {i}
11 |   for (j, i) ∈ S such that j ∉ S do
12 |   |   if c_j > c_i + w_ji then
13 |   |   |   c_j = ci + w_ji
14 |   |   |   p_j ← i
15 |   |   end
16 |   end
17 end
```

based on the successors structure (line 14).

Note that these algorithms apply equally to both the fiber and logical graphs, despite their differences. Also, the K-shortest paths in the logical graph can only be run after the fiber graph paths are available, since the latter provides the costs to the links of the former. The resulting paths are organized in different data structures. In order to obtain the candidate paths sets initially introduced, the K-shortest paths generated need a final step that is different among the graphs. In the logical graph the paths are sorted by number of hops thereby defining the logical paths set. On the other hand, the K-shortest paths in the fiber graph need a final step that consists of performing network equipment placement and accounting their respective costs. This operation is presented in algorithm 4.

As can be seen, the K-shortest paths are organized according the client nodes. Individually, each transmission format is applied to every paths (line 2 and 3). At once, the cost, location of 3R regenerators and cumulative distance are initialized. Next, the number and location of the 3R regenerators is evaluated. A 3R point is needed when the length of a fiber or the length of several fibers exceed the reach defined in the transmission format properties (line 14). In this case, the cost and 3R positioning array are updated (line 17). We are already considering the 160 Km optical degradation suffered by a light path when travelling through a node without a regenerator deployed. Whenever a fiber length itself exceeds the reach (line 9), then the path is considered unfeasible for the respective transmission format, and is dropped from the final candidate paths set (line 20). Ultimately, the subset of paths for each transmission format/OTU alternative is sorted by increasing cost. This improves the decision process executed by the MSLS algorithm since it does not require to thoroughly test every single candidate path, as soon as one feasible option is found.

Both the candidate path sets are built prior to the beginning of the MSLS. Those data structures are available during the algorithm execution.

Algorithm 4: Fiber Candidate Paths

for *each pair source-destination client nodes* d **do**
    for *each transmission format/OTU type* i **do**
        for *each path* k, p_{di}^{k} **do**
            $c(p_{di}^{k}) \leftarrow$ cost of deploying 2 end node equipments of type i
            $3R \leftarrow 0$
            $dist \leftarrow 0$
            for *each link* l *of* p_{di}^{k} **do**
                $leng_l \leftarrow$ distance in Km of link l of p_{di}^{k}
                if $leng_l > reach_i$ **then**
                    p_{di}^{k} not feasible
                    break
                end
                $dist = dist + len_l$
                if $dist > reach_i$ **then**
                    $3R \leftarrow$ insert 3R in the initial node of link l
                    $dist \leftarrow leng_l$
                    $c(p_{di}^{k}) = c(p_{di}^{k}) + c(3R_i)$
                end
            end
            if p_{di}^{k} *is feasible* **then**
                save p_{di}^{k}, $c(p_{di}^{k})$ and $3R$
            end
        end
    end
end
Sort each subset p by increasing cost

3.4 Spectrum Assignment

The spectrum assignment algorithm aims to assign proper amount of spectral resources for the optical connections routed through the network. Following the assumptions of a elastic optical network, it is intended to apply an algorithm that allocates spectral resources with finer granularity as opposed to the traditional WDM networks. Thus, instead of assigning a given wavelength with a fixed bandwidth to an optical connection, a set of contiguous spectrum slots are assigned to exploit the flexible access to the spectrum. Therefore, the spectrum of each fiber is organized into 25 GHz slots and an optical connection uses multiples of those slots according to the width required. For instance, using a transmission format with one light path that requires 75 GHz spectral width, will make use of three contiguous slots along its path. This allows accommodating different modulation formats and bit rates with higher efficiency and with more appropriate features to long-haul transmission [22].

The spectrum slot assignment (algorithm 5) begins by splitting the OCHs into light path segments without regeneration. This actually creates a list of all spectrum-continuous light path segments that need to be assigned slot numbers. Then, these light path segments are sorted in descending order by the number of fibers, and for the ones with same number of fibers the criteria is the most overloaded fiber. Finally, sequentially going through this list, we assign the first lower number of contiguous slots available in each fibers associated with the given light path.

When assigning spectrum slots a couple of constraints need to be verified. Firstly, the physical restriction of fibers is present. Naturally, the highest possible slot assigned is slot 160. Secondly, light paths must use the same spectrum slots in every fiber of its routing path. We assume nodes with 3R regenerators are capable of doing wavelength conversion. Hence, the continuity constraint only applies to light path segments. Thirdly, spectrum slots of a given light path must be contiguous. Finally, spectrum of different light paths can not overlap.

Typically, due to the constraints presented, the implementation of such spectrum assignment results in fragmentation of spectral resources. Fragmentation can be viewed as slot gaps around the spectrum. More precisely, those gaps consist of small non-contiguous groups of slots that, due to its sizes, can not be used to settle larger bandwidth demands, and so reduces the spectrum efficiency and makes the network less scalable. Thus, it is desired to have all free spectrum space in a single contiguous block, leading to a null fragmentation level. In order to measure the fragmentation in our case studies, we use the following expressions:

$$F^l_{fiber} = \text{Highest slot assigned - Number of slots needed}$$

$$F_{avg} = \frac{\sum_{l=1}^{L} F^l_{fiber}}{L}$$

The first one indicates the number of fragmented slots in a specific fiber l and the second estimates the average fragmentation over all fibers. Particularly, suppose the following example: two fibers L^1 and $^L2$ with spectrum grid of 20 slots. The value 1 represents a assigned slot while 0 a free slot.

$$L^1 = 11011001101100110000$$

$$L^2 = 11110011101110000000$$

As can be seen, L^1 is more fragmented than L^2 even though the number of assigned slots is the same. Considering the measure presented, L^1 would have a fragmentation of 6 and L^2 a fragmentation of 3.

The following picture (figure 3.1) represents a network with three established OChs. In this case, OCh λ_1 exists between end nodes 1 and 5, λ_2 between end nodes 4 and 3 and λ_3 exists also between end nodes 1 and 5 but with a different physical path. The first OCh requires three frequency slots while the other two require two frequency slots each. Lets assume there are no 3R regenerators so wavelength conversion is not an alternative here. Also, according to the algorithm, the lack of 3R regenerators makes it impossible to split the existing optical connections into light paths segments. The first slots assigned are for λ_3 since this is the longest physical path. The corresponding fibers ($1 \leftrightarrow 2$, $2 \leftrightarrow 4$ and $4 \leftrightarrow 5$) are assigned with the two consecutive free lowest slot numbers. In order not to violate the continuity constraint, these slot numbers must be the same in every fiber of the path. Next, the frequency slots assigned for λ_2 are computed. The lowest-index slot numbers equally available in both $4 \leftrightarrow 5$ and $5 \leftrightarrow 3$ fibers are indexes 3 and 4, since fiber $4 \leftrightarrow 5$ is already using indexes 1 and 2 from λ_3. These slots avoid spectrum overlapping and respect the continuity and contiguous constraints. Finally, 3 frequency slots need to be assigned to λ_1. The choice falls in indexes

Algorithm 5: Spectrum Assignment

```
Split_Light_Paths ← Structure with every light paths split by 3R points
sort Split_Light_Paths elements by:
(1) number of fibers of the routing path
(2) the one that used the most overloaded fiber
for each element of Split_Light_Pathsi do
    n ← number of slots to assign to i
    ind ← first available and consecutive n slots in the first fiber of i
    while finish do
        finish ← true
        for each remaining fiber l of i do
            if ind are available then
                continue to next fiber
            end
            else
                finish ← false
                ind ← following n available and consecutive slots in the first
                fiber of i break
            end
        end
    end
    if finish = true then
        assign the slots ind to light path i
        update state of slots: ind are now assigned
    end
    else
        invalid spectrum assignment
        break
    end
end
```

5, 6 and 7. Although indexes 1, 2 and 3 are available on fiber 1 ↔ 3, fiber 3 ↔ 5 is already using slot index 3, so the next three slots available on both fibers are 5, 6 and 7. The slots usage for each fiber is displayed in figure 3.2(a).

Now, lets assume node 5 has a 3R regenerator deployed. This means that light paths traversing intermediate node 5 can be converted from a given wavelength to another one. Similarly, the direct light path λ_3 is assigned with slot numbers 1 and 2. Then, the OCh λ_2 is split into two light paths due to the presence of a regenerator at node 5. The resulting light paths have less physical links than λ_1, so this one has higher priority to the algorithm. Therefore, λ_1 is assigned with slot numbers 1, 2 and 3 on both fibers. Finally, concerning λ_2, the light path between nodes 4 and 5 is given the slot numbers 3 and 4 while the one between nodes 5 and 3 is assigned with slot numbers 4 and 5, since these are the lowest slot numbers available in each fiber and the continuity constraint does not apply because of the existence of a 3R regenerator separating the two light paths. By inspection, allowing wavelength conversion

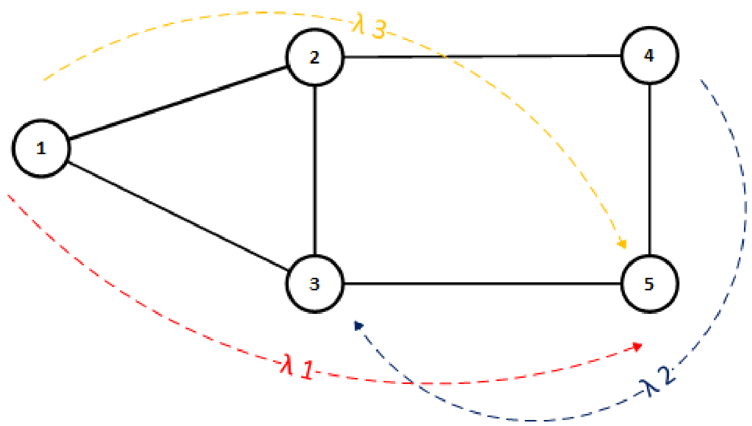


Figure 3.1: Spectrum allocation for a network with three established OTUs/OChs

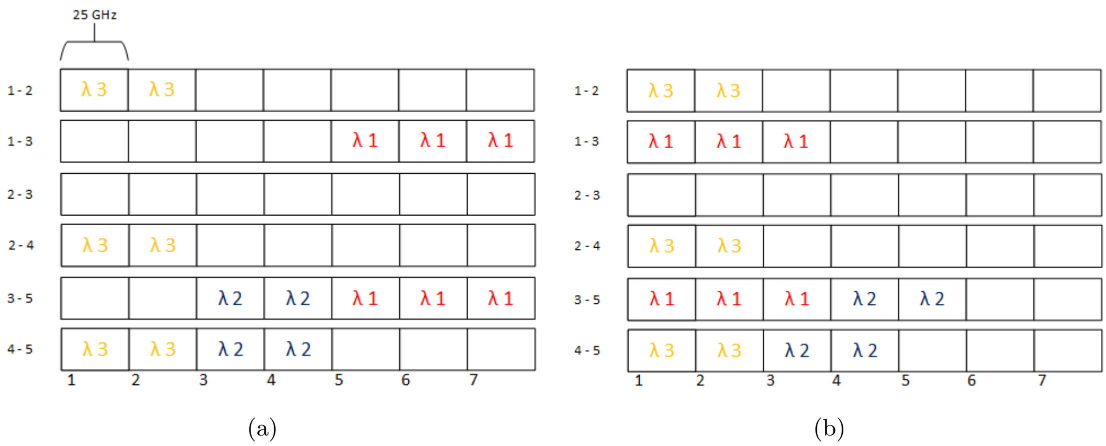


Figure 3.2: Slots assigned to each fiber.

may easily lead to more compact slot assignment solutions and consequently lower indexes of the highest slot assigned. Likewise, figure 3.2(b) sums up the slots used in each fiber and the matching OCh.

3.5 Multi-Thread Multi-Start Local Search Algorithm

3.5.1 Global Algorithm

The core of the overall algorithm is the MSLS algorithm shown in 6, based on the MSLS meta-heuristic before presented. This meta-heuristic was adapted in order to solve the proposed problem of optical network design. Similarly, for a given running time, the algorithm executes a cycle composed of a greedy solution plus a local search. Additionally, a spectrum assignment algorithm is introduced to fulfil our problem requirements. The MSLS algorithm solves the problem of client demands routing that consists of assigning a suitable path and transmission alternative for each client demand. The spectrum assignment problem is not included in the MSLS and is performed after. Thus, these two problems are solved sequentially, which means the slot assignment will be performed after all client demands are provisioned. Rather, the spectrum assignment is only done once a potentially new and feasible incumbent solution is found. By this we mean that the current local minimum solution must have a cost that is no higher than that of the incumbent solution (line 8). After the spectrum assignment, and assuming it ends successfully, the current local minimum solution is defined as the incumbent solution if it has a lower cost or if it has an equal cost and lower maximum frequency slot number assigned than the current incumbent solution (line 11). Recall that, as stated in the problem definition, if multiple solutions with the same cost are found, the one with lowest slot number assigned is selected as the incumbent.

Initially, we have an empty network, this is, with no installed OChs. Then, client demands are established, one at a time, in a random sequence starting with the ones with higher bit-rates. For each demand we first try to route the maximum number of bandwidth units through spare inputs of already existing OChs, using the logical graph. Already installed equipment is used to supply the given units of bandwidth, and so the solution cost remains the same. This strategy comprises the so called multi-hop grooming since client demands can use more than one light path to be delivered. More specifically, using the logical graph means to use one or more suitable paths from the corresponding K-shortest paths set, that enable routing those bandwidth units. A logical path is composed of several OChs between each segment of the path. Thus, it is required to verify if each of these segments have a OCh with spare inputs and with the port bandwidth unit that matches the bandwidth unit type of the demand. For instance, assume that we have 4 bandwidth units of 10G to route from node 1 towards 7. We will examine each of the logical K-shortest paths to check if it possible to route those units without establishing a new optical connection. One hypothetical path is represented in figure 3.3. Then, for each link of this path, the algorithm tries to find a suitable OCh that supports all or some units of the demand. Here, from node 1 to 5, the middle OCh can carry 3 bandwidth units, but since from node 5 to 7 the suitable OCh can only carry up to 2 units, it determines the maximum number of units supported. Finally, the number of spare inputs of the OChs used are updated. In this case, the first OCh still have one free input while the second one has none. Note that to be able to use multi-hop grooming in our problem in a valid manner, each link of the logical path need to have at least one appropriated OCh with spare inputs. In the same example situation, it is not possible to route demands of 40G between the given origin-destination pair even though the first link has one suitable OCh with one free input.

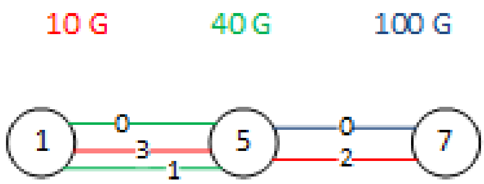


Figure 3.3: Logical Graph example

If, after draw upon multi-hop grooming, there are still bandwidth units to route, then a new OCh must be installed. In these situations, both greedy and local search steps require a decision mechanism to correctly establish new OChs according to cost optimization aim. This decision mechanism starts by identifying all transmission options that enable routing the desired bandwidth units, considering the available possibilities given in the input parameters.

Then, the selection involves choosing the cheapest path and format combination available. Figure 3.4 exemplifies the just described mechanism. It shows every single alternative, according to the input parameters, evaluated by the algorithm to support 9 units of 40 Gbps. As can be seen, there is always present a trade-off between cost and spectrum usage. The lower bit-rate interface requires more light paths to support the demand, even though, it might be chosen if it is cheaper to implement. To evaluate the cost, besides the transmission format, one needs to consider the path to choose due to the variable number of 3R regenerators from path to path. At the end, the cheapest path and format combination is selected as the one to deploy. Note that only valid transmission alternatives and paths exist within this set, since the candidate paths executed before already excludes impracticable alternatives.

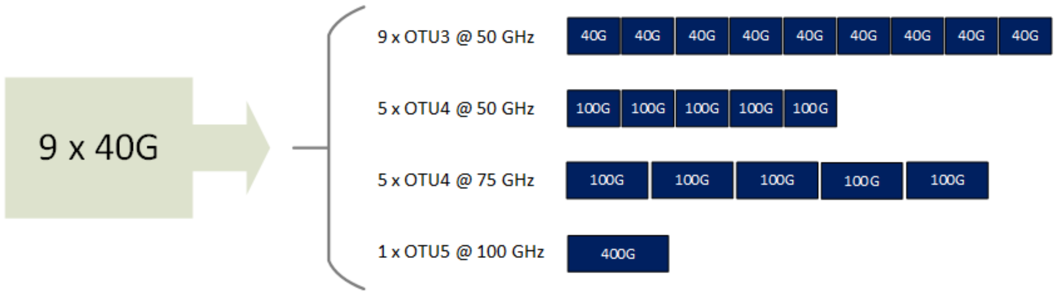


Figure 3.4: Transmission Alternatives for a traffic pattern of 9 x 40G

However, while building a solution, it is possible that the existing fiber network can not support all demands while meeting the fiber capacity constraint. This typically happens when a stressful set of demands, that is, with a high value of total traffic, needs to be supported by the network. In order to solve this problem a new functionality was implemented in the algorithm. On any occasion during the greedy construction, if there is none feasible alternative available (path and format combination that satisfies the fiber capacity limitation), the algorithm instead of minimizing the cost, minimizes the unfeasibility degree. In other words, the algorithm decides by the option that less exceeds the number of frequency slots above the 160 slots boundary. So, the greedy step always generates a solution, even if a feasible one was not found. If unfeasible, the local search shifts the objective function in order to try to achieve a feasible solution again. On the other hand, if the solution is feasible then the local search applies the usual cost optimization. Hence, during the local search, cost optimization is not considered until the current solution is feasible again.

Returning to the previous example, it is visible that different transmission formats imply different spectrum usage. Assume a situation where every single path of each transmission format have at least a fiber above or on the edge of the capacity limit. Here, the algorithm will not choose the lowest cost alternative but will choose the one that less increases the capacity of the used fibers.

3.5.2 Multi-Thread Approach

Higher levels of computational performance were achieved with the appearance of multi-core platforms and by increasing the instruction per cycle rate using thread parallelism. Since we are focused on building an efficient algorithm, implementing a multi-thread approach arises as a natural option. The key challenge in designing such multi-thread environment is guaranteeing the proper access to shared memory spaces and avoiding deadlock situations.

Algorithm 6: Global Algorithm

$buildCandidatePaths()$
$c_{Incumbent} \leftarrow +\infty$
$Incumbent \leftarrow \{\}$
while *time* **do**
  $Solution \leftarrow greedy()$
  $localSearch(Solution)$
  if *Solution is feasible* **then**
    if $c_{Solution} \leq c_{Incumbent}$ **then**
      $spectrumAssignment(Solution)$
    end
    if *spectrumAssignment was valid and Solution is better than Incumbent* **then**
      $Incumbent \leftarrow Solution$
    end
  end
end

Thus, a synchronization policy must be implemented so the communication between threads is held correctly.

Threads are independent executable entities, but within the context of the same process. Each thread is typically associated with a function that implements a particular and specific activity. However, in our situation, each thread has associated the same procedure, which is the MSLS algorithm. The main process, the one that launches the threads, writes all input parameters into a shared memory that is used by all threads. This memory space is read-only. Each thread runs the MSLS algorithm during the given running time. At the end, all threads return their own incumbent solution and the main process computes the global best solution, also designated as global incumbent solution. Besides this first strategy, a different one was also implemented. The main process creates another shared memory, this one with exclusive permission access, to write a global incumbent solution. The size of this structure is enough to register one solution (light paths created, slot assignment and routing paths information), its cost value and its maximum frequency slot value. When a thread finds a new minimum cost incumbent, it checks the cost value of the current global incumbent solution (stored in the suitable shared memory). If its own incumbent is worse, it saves the cost value of the global incumbent as its own incumbent cost value. Otherwise, it performs frequency slot assignment to its incumbent solution and it saves its incumbent solution in the shared memory if it is better than the current global incumbent solution. Unlike the first strategy, the information of the best incumbent solution found by any thread is quickly exchanged among all threads and the main process does not require to perform any computations to determine the best incumbent solution. Also, less computation time is wasted doing slot assignments because it is already known when a better solution exists.

In order to manage the access to the shared resource of the second strategy, a semaphore object was used. The semaphore ensures exclusive access to this critical piece of memory. Whenever a thread wants to access it, a wait function is used. This wait function determines whether it is safe or not to the thread performs its task, based on the semaphore value.

Consequently, threads with no permission will be blocked until access is allowed. When a thread finishes its task the shared resource must be freed, so it releases the semaphore allowing other threads waiting on the semaphore to unblock. Note that this semaphore implementation only applies to the resource that requires to be read and written, which is the one that contains the global incumbent solution.

The scheme 3.5 below represents, in a simple manner, the two strategies of the multi-thread environment previously explained. Figure 3.5(b) refers to to the situation where threads have separate lines of execution without communication amongst them and where each have a distinct memory space to return their own incumbent solution. Figure 3.5(a) represents the strategy that enables communication between threads and make use of a single memory space to write the best incumbent solution amongst all.

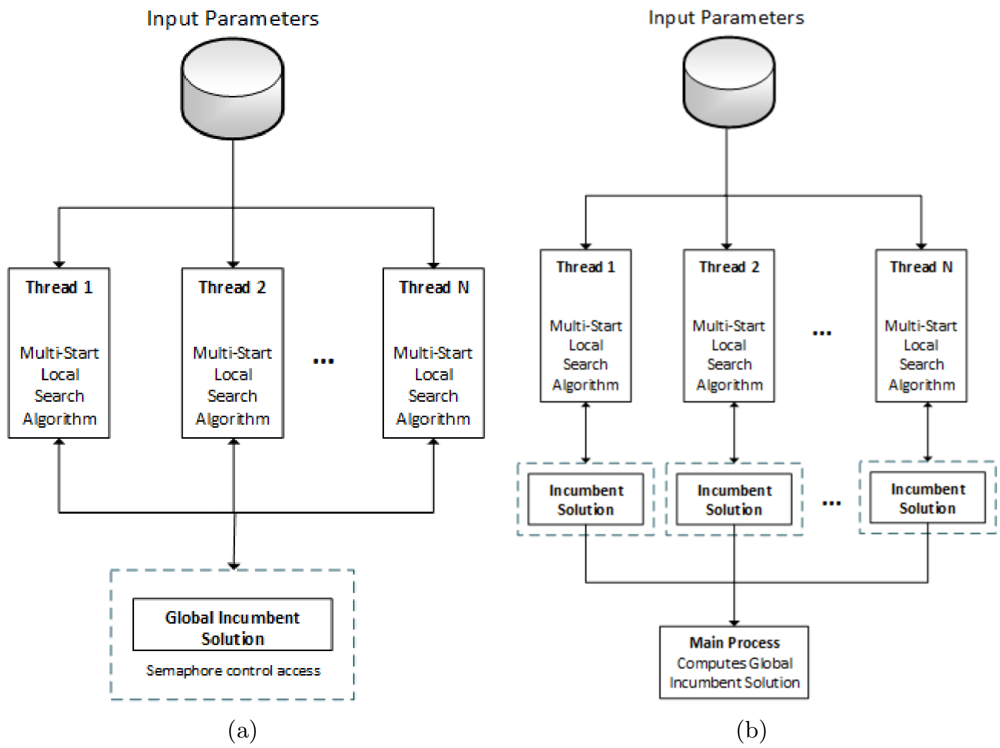


Figure 3.5: Multi-thread environment: a) with thread communication and b) without thread communication.

3.5.3 Greedy Solution

The first step of the global algorithm is the greedy (algorithm 7), where an initial solution is generated. To build an initial solution, we first choose a random order for the group of demands. Then, for each demand, by the chosen order, we first use the logical graph to route the maximum number of bandwidth units through muxponder spare inputs, as described before. Then, if necessary, new optical connections and equipment deployment is made so every bandwidth unit of the demand is provisioned, according to the decision mechanism implemented. More detail on both procedures is available in subsection 3.5.1.

The decision mechanism is based on the cost, this is, we want the cheapest alternative possible. However, during the decision process of establishing new connections, it might happen that multiple minimum cost OTUs alternatives exist. For instance, for a given OTU, different paths may lead to the same cost, due to existence of the same number of 3R regenerators or distinct OTUs alternatives, even with their corresponding network equipment costs, may also lead to the same deployment cost. In these situations, we evaluate other indicator to enable a decision that benefits the algorithm: the OTU chosen, from the multiple minimum cost OTUs alternatives, is the one that leads to the least overloaded fiber, already considering the frequency slots introduced by the given alternative. Thus, while building solutions, we guarantee the minimization of the highest slot required on a fiber. Note that this highest slot minimization is done in the context of a certain MSLS cycle, this is, during the computation of a given local minimum solution. However, several minimum cost solutions may still exist during the algorithm execution, and that is why the final condition that defines the incumbent solution compares not only the cost but also the highest slot assigned.

Also during the process of choosing the transmission alternative to establish, it might happen that no feasible options are available. This means that every single alternative exceeds, in at least a fiber, the limited slot capacity. Then, the alternative that minimizes the unfeasibility degree is preferred. We define unfeasibility degree as the number of slots above the limited slot capacity. As soon as one unfeasible alternative is chosen, we define the current greedy solution as unfeasible (line 16). This variable is extremely important to define the behaviour of the local search.

Algorithm 7: Greedy Algorithm

1 $shuffle(Demands)$
2 $unfeas_{greedy} \leftarrow 0$
3 $Solution \leftarrow \{\}$
4 $c_{Solution} \leftarrow 0$
5 **for** *each demand d* **do**
6     $b_d \leftarrow$ bandwidth unit
7     $n_d \leftarrow$ number of bandwidth units
8     $routingLogicalGraph(b_d, n_d)$
9   **if** $n_d > 0$ **then**
10   attempt to choose the minimum cost and feasible transmission alternative/OTU
11   **if** *multiple minimum cost and feasible alternatives exist* **then**
12   choose the transmission alternative/OTU that minimizes highest slot required
13   **end**
14   **else if** *no feasible alternative exists* **then**
15   choose the transmission alternative/OTU that minimizes unfeas. degree
16             $unfeas_{greedy} \leftarrow 1$
17   **end**
18         $Solution = Solution \cup \{newOTU\}$
19         $c_{Solution} = c_{Solution} + c_{OTUdeployed}$
20   **end**
21 **end**

3.5.4 Local Search

In the local search step, we start by initializing the current solution with the initial solution which is the greedy solution. If the current solution is feasible, we compute the minimum neighbour solution. A neighbour solution is computed as follows: we remove one optical connection of the current solution and we process the bandwidth units supported by the eliminated OCh in the same way as we described for the construction of the greedy solution. In this context, we only remove optical connections that are established with muxponders, thus, the number of neighbour solutions is equal to the number of optical connection using muxponders in the current solution. If the minimum neighbour solution is a better solution than the current solution, we set it as the current solution and repeat the local search cycle. If not, we stop and the current solution becomes the local minimum solution. A better solution has either a lower cost or, if both have an equal cost, has a better load balancing.

On the other hand, when the current solution is unfeasible, a sightly different local search is executed. We compute the minimum neighbour solution as similarly described above. However, the set of neighbours is different. Here, we remove every optical connection that uses at least one fiber that is above its slot limitation, and reinsert the supported bandwidth units. The idea here is to force the reduction of the unfeasibility degree. If the minimum neighbour solution less unfeasible than the current solution, we set it as the current solution and repeat the local search cycle If not, we stop the local search cycle and the solution, which is still unfeasible, is dropped, since it is no longer possible to become feasible. Relating with the global algorithm 6, one can see that in these situations, when local search finishes with an unfeasible solution, there is no need to perform slot assignment and other relevant operations. A new global cycle is initiated, if there is still running time left, starting by the known greedy step.

Note that, it is a possible and desirable behaviour that the local search use both the unfeasible and feasible contexts. This means that a current solution which was initially unfeasible, was able to become feasible, and consequently the context changed so the local search could optimize the cost of the newly feasible solution.

3.5.5 Illustrative Example

After a fully description of the algorithm developed, and before presenting and discussing the results obtained with several case studies, a brief example is explored in order to give an overview on the operation of the algorithm and to understand some of the decision makings performed during the construction of the solutions. Consider the network topology and the corresponding fiber lengths and capacity in figure 3.6 and table 3.2 respectively, along with the transmission formats available and client demands to be supported, depicted in table 3.3.

The fiber graph is defined by the nodes and fibers of the network topology. On the other hand, the logical graph is created by the client nodes and the logical link between each pair of them. Specifically, nodes 1, 2 and 4 and links $1 \leftrightarrow 2$ and $1 \leftrightarrow 4$ define the logical graph here.

First of all, the candidate paths for both pairs of end nodes is computed. We assume $k_{fib} = 5$ and $K_{log} = 1$, this is, 5 paths in the fiber graph and 1 path in the logical graph. The deployment cost and the 3R regenerators placement is attached to each routing path (table 3.4). Some OTUs are not feasible due to distant limitations. In this case, this occurs for OTUs with transparent reach higher than 1100 km, the longest fiber in the network. The

Algorithm 8: Local Search

$Solution \leftarrow$ greedy solution
while $!finish$ **do**
    $Neighbour \leftarrow Solution$
    if *Solution is feasible* **then**
        for *each neighbour of Solution* **do**
            Compute Sol
            if $c_{Sol} < c_{Neighbour}$ *or* $c_{Sol} = c_{Neighbour}$ *and* $hslot_{Slot} < hslot_{Neighbour}$ **then**
                $Neighbour \leftarrow Sol$
            end
        end
        if $c_{Neighbour} < c_{Solution}$ *or* $c_{Neighbour} = c_{Solution} and hslot_{Neighbour} < hslot_{Solution}$ **then**
            $Solution \leftarrow Neighbour$
        end
        else
            $finish$
        end
    end
    else
        for *each neighbour of Solution* **do**
            Compute Sol
            if $unfDeg_{Sol} < unfDeg_{Neighbour}$ **then**
                $Neighbour \leftarrow Sol$
            end
        end
        if $unfDeg_{Neighbour} < unfDeg_{Solution}$ *or* $unfDeg_{Neighbour} = unfDeg_{Solution}$ *and* $c_{Neighbour} < c_{Solution}$ **then**
            $Solution \leftarrow Neighbour$
        end
        else
            $finish$
        end
    end
end

only output fibers of node 1, $1 \leftrightarrow 2$ and $1 \leftrightarrow 3$, have lengths higher than the corresponding reach of those OTUs.

The greedy solution steps are going to be described first. The fiber graph is defined by the available fibers while the logical graph is initially empty, since it is defined by the already established OTUs/OChs. The demands are randomly sorted, but the algorithm starts by provisioning higher bit-rate demands. So, the 100 Gbps demands are the first to be inserted. Consider the order of demands presented in table 3.5. Firstly, the algorithm checks the logical graph paths and tries to find spare inputs to route demand units. In this case, the only logical

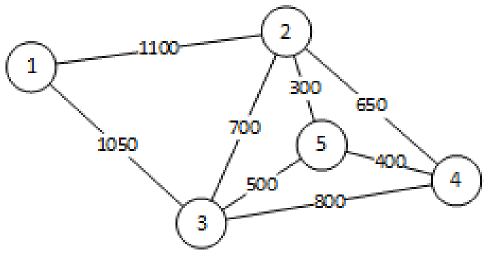


Figure 3.6: Network topology used in this example.

Fiber link	Length	Capacity (slots)
$1 \to 2$	1100	160
$1 \to 3$	1050	160
$2 \to 3$	700	160
$2 \to 4$	650	160
$2 \to 5$	300	160
$3 \to 4$	800	160
$3 \to 5$	500	160
$4 \to 5$	400	160

Table 3.2: Fiber lengths and capacities.

Transmission Options						
o_t	w_t	n_t	b_t	c_t	r_t	l_t
1	50	1	40	6	12	1500
1	50	4	10	8	12	1500
1	50	1	100	8	16	1000
1	50	2	40	12	16	1000
1	50	10	10	10	16	1000
3	50	1	100	20	12	1500

Client Demands			
Source-Destination	10G	40G	100G
$1 \to 2$	7	1	1
$1 \to 4$	5	2	1

Table 3.3: Transmission formats and client demands.

path available is $1 \to 4$, and none OTUs/OChs exist. The only option is to establish a new OTU. Two possibilities to accommodate 1 x 100 Gbps between nodes 1 and 4 exist: the first one is to use the native rate, this is, use an OTU4, and the other one consists of using inverse-multiplexing. However, only the latter applies here since all paths have at least one link that exceeds the OTU4 optical reach. So, the algorithm tests every path and decides towards the one that leads to the lower cost solution, or if multiple lower cost solutions exist it decides towards the one that is more balanced in terms of spectrum usage. More precisely, analysing table 3.4 of inverse-multiplexing, one conclude that the path chosen is the first one. The current solution cost is 760 and no 3R regenerators are needed. Furthermore, issues such as fiber physical limitation does not apply here since every fiber is far from the 160 slots limit. Before moving towards the following demand, updates are needed in the solution structures. The fibers of the path chosen, namely $1 \to 2$ and $2 \to 4$, are now using 6 spectrum slots. On the other side, the logical graph does not require updates since no spare inputs arise from the newly deployed inverse-multiplexing optical channels.

Next, the same procedure is made for the following demand, 1 x 100 Gbps between nodes 1 and 2. Again, inverse-multiplexing is the only alternative. The path chosen is the first one since is the cheapest one. The fiber $1 \leftrightarrow 2$ has now a used capacity of 12 slots. The cumulative solution cost is 1160.

Candidate Paths List		
i	1 → 2	1 → 4
1	1 2	1 2 4
2	1 3 2	1 2 5 4
3	1 3 5 2	1 3 4
4	1 3 4 2	1 3 5 4
5	1 3 4 5 2	1 3 2 4

Deployment cost and 3R placement for 1x40G				
	1 → 2		1 → 4	
i	Cost	3R location	Cost	3R location
1	120	-	240	2
2	240	3	240	2
3	240	3	240	3
4	360	3,4	240	3
5	360	3,5	360	3,2

Deployment cost and 3R placement for 4x10G				
	1 → 2		1 → 4	
i	Cost	3R location	Cost	3R location
1	160	-	280	2
2	280	3	280	2
3	280	3	280	3
4	400	3,4	280	3
5	400	3,5	400	3,2

Deployment cost and 3R placement for 1x100G as inverse-multiplexing				
	1 → 2		1 → 4	
i	Cost	3R location	Cost	3R location
1	400	-	760	2
2	760	3	760	2
3	760	3	760	3
4	1120	3,4	760	3
5	1120	3,5	1120	3,2

Table 3.4: Candidate paths list together with deployment cost and 3R node placement.

Now, 1 x 40 Gbps is inserted in the network. The lowest cost and feasible path from the candidate paths set is the first one, with a cost of 120. Again, the fiber of the path chosen requires plus 2 slots. The current solution cost is now 1280. The following 2 x 40 Gbps will require two OTU3. Since 4 paths have the same cost (2×240), the algorithm will choose the path with the lowest loaded fiber. In this case, the path chosen is the third as the first and the second make use of fiber $1 \leftrightarrow 2$, the current most loaded fiber. The solution cost grows to 1760. So far, the logical graph is still empty, since we only make use of transponders. The next demands of 10 Gbps will be supported by muxponders and thus giving rise to spare inputs that allows for modifications on the logical graph.

The first 5 x 10 Gbps need two OTU3 to be supported, but four 10 Gbps interfaces are left free. Hence, these four interfaces are reflected in the logical graph. Again, multiple cost solutions are available to deploy the mentioned OTUs (2×280). The choice falls for the third path according to the decision process already explained. Finally, 7 x 10 Gbps also need two OTU3 but only one interface is left free, which is also reflected in the logical graph. The lower cost path is the first one. In the end of the greedy, the solution has a cost value of 2640. Figure 3.7 represents the solution on both fiber and logical graph.

Order of demands
1 x 100 Gbps (1-4)
1 x 100 Gbps (1-2)
1 x 40 Gbps (1-2)
2 x 40 Gbps (1-4)
5 x 10 Gbps (1-4)
7 x 10 Gbps (1-2)

Table 3.5: Order of demands considered by the algorithm.

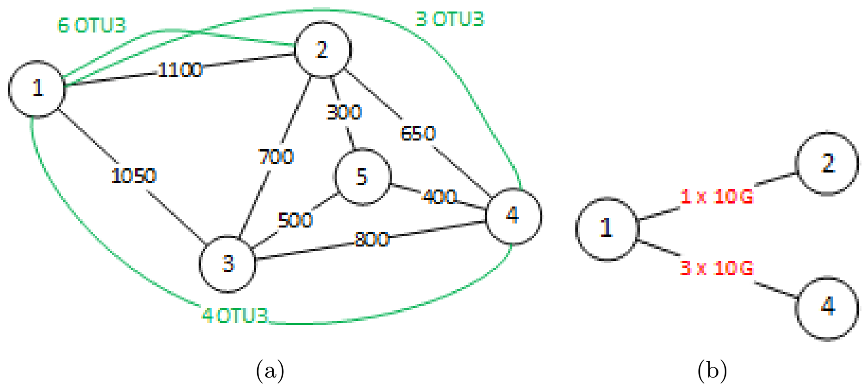


Figure 3.7: Greddy solution: a) fiber graph and b) logical graph (spare inputs).

Secondly, local search is performed, which is applied upon the greedy solution. Since the greedy is a feasible solution, only muxponders are removed and re-inserted again. This means that the only optical connections to be removed are the 10 Gbps carried with OTU3s. Moreover, therea are no cost benefits from re-routing those demands, as the only transmission alternative is multiplexing the signals into OTU3s.

One can also notice that limiting fiber $1 \leftrightarrow 2$ capacity will increase the solution cost. From table 3.4 we see that the lower cost transmission options use the first path of the candidate path, which contains the mentioned fiber. So, by limiting its capacity the algorithm is obliged to use other higher cost paths. For reference, using a capacity of 5 slots for fiber $1 \leftrightarrow 2$ will lead to a solution cost of 3360, that contrasts with 2640 obtained with the initial scenario.

Finally, multi-hop grooming would be possible in this example if we insert a new client demand with different end nodes, namely between 2 and 4. This demand will allow for spare inputs of established OTUs/OChs to be exploited, since two paths would exist in the logical graph.

Chapter 4

Case Studies and Computational Results

4.1 Case Studies

With the purpose of validating the implemented optimization algorithm and analyse its different aspects, we built several case studies. These case studies are presented hereafter.

In first place, four optical reference networks were chosen. Specifically, NSF network (NSFNET) (4.1), European Optical Network (EON) (4.3), German Backbone Network (GBN) (4.2) and GEANT2 (4.4) are the networks whose topology was used to test our algorithm [27]. EON has 19 nodes and 37 bidirectional links. It is the network with the highest mean nodal degree. Since it connects several country sites, the average link length is 753.8 km, also one of the highest values. GBN is a national network with 17 nodes with a mean nodal degree of approximately 3. It is also characterized by 26 links, typically with lengths below 200 km, resulting in the lowest average link length of our four networks. The GEANT2 network connects 34 nodes through a total of 52 links. Similarly to EON, this is an European scale network and despite the topological differences, it presents similar values of mean nodal degree and link length. Finally, NSFNET interconnects 14 nodes through 21 links. In fact, one of the links has a length of 2838 km and since this value is higher than the maximum reach of any of the interfaces available in our problem, we did no consider it. Also, it has the highest mean link length of all networks considered. Table 4.1 summarizes the parameters of the previous networks. The substantial differences between these four optical network topologies, observable in relevant network parameters such as the mean link length and the mean nodal degree, ensures that the algorithm is tested on uneven real networks. Furthermore, this diversity is important to identify solution patterns, quantify the usefulness of the different features present in the algorithm and related them with the corresponding network parameters.

Besides the reference networks chosen, we were concerned about providing traffic profiles as realistic as possible to apply to our case studies. In order to accomplish that, we generate nine traffic matrices for each of the four network topologies. This way, we manage to cover different traffic scenarios with different traffic scaling factors. These matrices were generated with the following process: firstly, pairs of end nodes are randomly chosen, forming the source-destination nodes. Then, 10, 40 and 100 Gbps bandwidth units are randomly distributed among all the end nodes, E, previously picked, up to the predefined limits N_{10G}, N_{40G},

Network parameters				
	Nodes	Links	Mean Nodal Degree	Mean Link Length (km)
EON	19	37	3.89	753.8
GBN	17	26	3.06	143.1
GEANT2	32	52	3.25	677
NSFNET	14	20	3	991.7

Table 4.1: Reference network parameters.

N_{100G}. Consequently, each combination of these three parameters along with the number of pairs of end nodes, gives rise to distinct traffic scenarios. Each scenario is defined by the total traffic load in Tbps, B_{total}, and by the total bandwidth of 10, 40 and 100 Gbps, which is also represented in percentage of the total traffic, P_{10G}, P_{40G} and P_{100G}. More specifically, each scenario, T_i, built has the following properties. The matrices $T_{1,4,7}$ are generated so 10G bandwidth units constitute the dominant traffic. Then, the weight of 40G bandwidth units is made the highest in matrices $T_{2,5,8}$. The final subset $T_{3,6,9}$ employs a growth in the 100G bandwidth units while lowering the 10G weight. As stated, each subset has similar profiles concerning the bandwidth weights. However, they differ in the number of source-destination pairs. One third of the total existing pairs of end nodes is considered in the first matrix of each subset, while half of those end node pairs is applied to the second matrix and two thirds are considered in the third matrix. Within each subset a growth factor of between 15% and 30% is applied individually to the number of bandwidth units, which reflects an increase in the total traffic load. The reason behind the growth factors chosen and applied is related with the maximum traffic volume that the corresponding network can support. With those values we can guarantee that the traffic matrices are closer but not exceed the network capacity. Due to the scaling factors applied, the matrices obtained have total traffic volumes that ranges from 10 Tbps to 33.66 Tbps. Table 4.2 shows the resulting matrix parameters obtained applying the rules before presented. More details on traffic matrices can be found on appendix A at the end of this document.

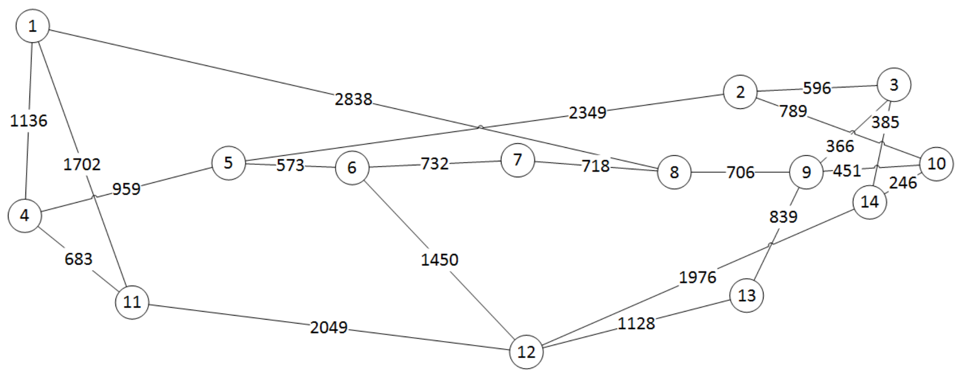


Figure 4.1: NSFNET network topology.

	EON					GBN				
	E	B_{total}	P_{10G}	P_{40G}	P_{100G}	E	B_{total}	P_{10G}	P_{40G}	P_{100G}
T_1	57	18.8 T	53.2%	25.5%	21.3%	45	17.0 T	52.9%	23.5%	23.5%
T_2	57	19.6 T	25.5%	49.0%	25.5%	45	17.8 T	25.3%	49.4%	25.3%
T_3	57	21.5 T	16.3%	46.5%	37.2%	45	18.8 T	16.0%	46.8%	37.2%
T_4	85	23.5 T	53.2%	25.5%	21.3%	68	20.4 T	52.9%	23.5%	23.5%
T_5	85	24.6 T	25.5%	48.9%	25.7%	68	21.4 T	25.3%	49.4%	25.3%
T_6	85	26.9 T	16.3%	46.5%	37.2%	68	22.6 T	16.0%	46.8%	37.2%
T_7	114	29.5 T	53.1%	25.5%	21.4%	90	23.5 T	52.8%	23.4%	23.8%
T_8	114	30.7 T	25.5%	48.8%	25.7%	90	24.7 T	25.2%	49.3%	25.5%
T_9	114	33.7 T	16.3%	46.6%	37.1%	90	26.0 T	15.9%	46.8%	37.3%
	GEANT2					NSFNET				
	E	B_{total}	P_{10G}	P_{40G}	P_{100G}	E	B_{total}	P_{10G}	P_{40G}	P_{100G}
T_1	165	18.5 T	54.1%	25.9%	20.0%	30	10 T	60.0%	20.0%	20.0%
T_2	165	18.7 T	26.7%	53.5%	19.8%	30	11.5 T	26.1%	52.2%	21.7%
T_3	165	20.7 T	19.3%	48.3%	32.4%	30	12.0 T	16.7%	50.0%	33.3%
T_4	248	21.3 T	53.9%	25.9%	20.2%	45	13.0 T	60.0%	20.0%	20.0 %
T_5	248	21.6 T	26.7%	53.4%	19.9%	45	15.0 T	26.0%	52.0%	22.0%
T_6	248	23.9 T	19.2%	48.2%	32.6%	45	15.6 T	16.7%	50.0%	33.3%
T_7	330	24.5 T	54.0%	25.6%	20.4%	60	16.9 T	59.9%	20.1%	20.1%
T_8	330	24.9 T	26.6%	53.3%	20.1%	60	19.5 T	26.0%	52.0%	22.0%
T_9	330	27.6 T	19.2%	48.2%	32.6%	60	20.3 T	16.6%	50.0%	33.4%

Table 4.2: Traffic Matrices.

4.2 Analysis of Computational Efficiency

The several tests were ran on a computational platform with two Intel Xeon processors at 2.30 GHz, 64 GB of RAM and Windows 7 operating system. First of all, an analysis concerning the effectiveness on number of threads to be launched was made. Using NSFNET with traffic matrix T_1^{NSFNET}, ten runs with 60 seconds were made for 1, 12 and 24 threads. Both with and without shared memory approaches were also considered for each case. The average number of iterations performed by the algorithm is depicted against the corresponding number of threads and approach (figure 4.5). Comparing with one thread, the results show that 12 threads obtain iteration gains around 2.5 times. This means that the algorithm is able to run 2.5 times more cycles within the same time period. With 24 threads, further iteration gains are no longer visible, hence, 12 threads is the most desirable alternative. The slightly lower gain values when shared memory is used can be explained by the threads blocking time, that is, the time when a thread is blocked, waiting to access the critical piece of memory. This time is more relevant when the algorithm is ran with higher thread values since is more likely to have threads to waste time waiting to execute instructions. Even without shared memory,

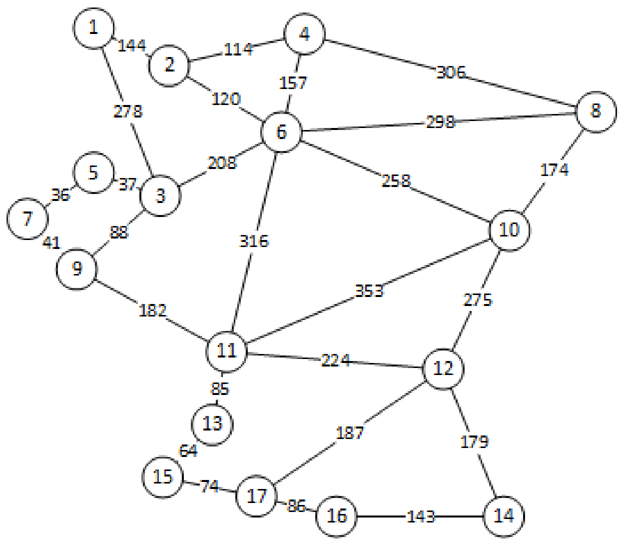


Figure 4.2: GBN network topology.

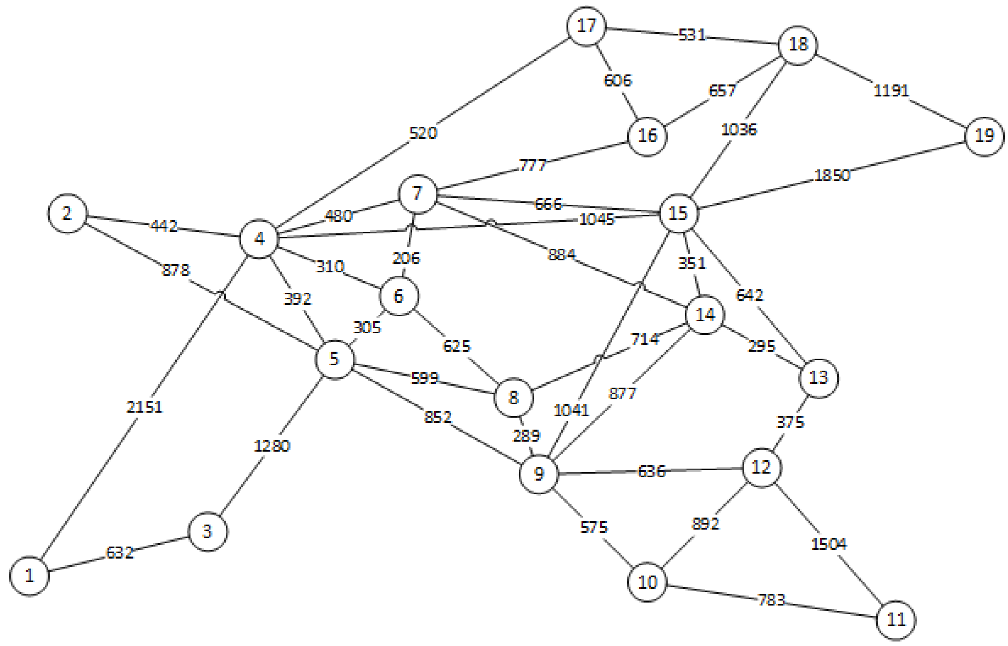


Figure 4.3: EON network topology.

there is an optimal thread value. Although the blocking problem does not apply here, a complex read-only data structure is given to each thread of the algorithm. These structures are independently accessed several times during the execution time and the corresponding access delays are more observable than when a single data structure is given to every thread.

Still within the computational efficiency scope, the best cost solution found along with the corresponding elapsed time must be taken into consideration. The full set of results (see appendix B.1 for more detail) allow us to immediately verify that algorithm productivity is drastically improved by multi-thread approach. This is noticeable in the higher number of iterations performed and in the lower cost solutions found when comparing 1 thread with

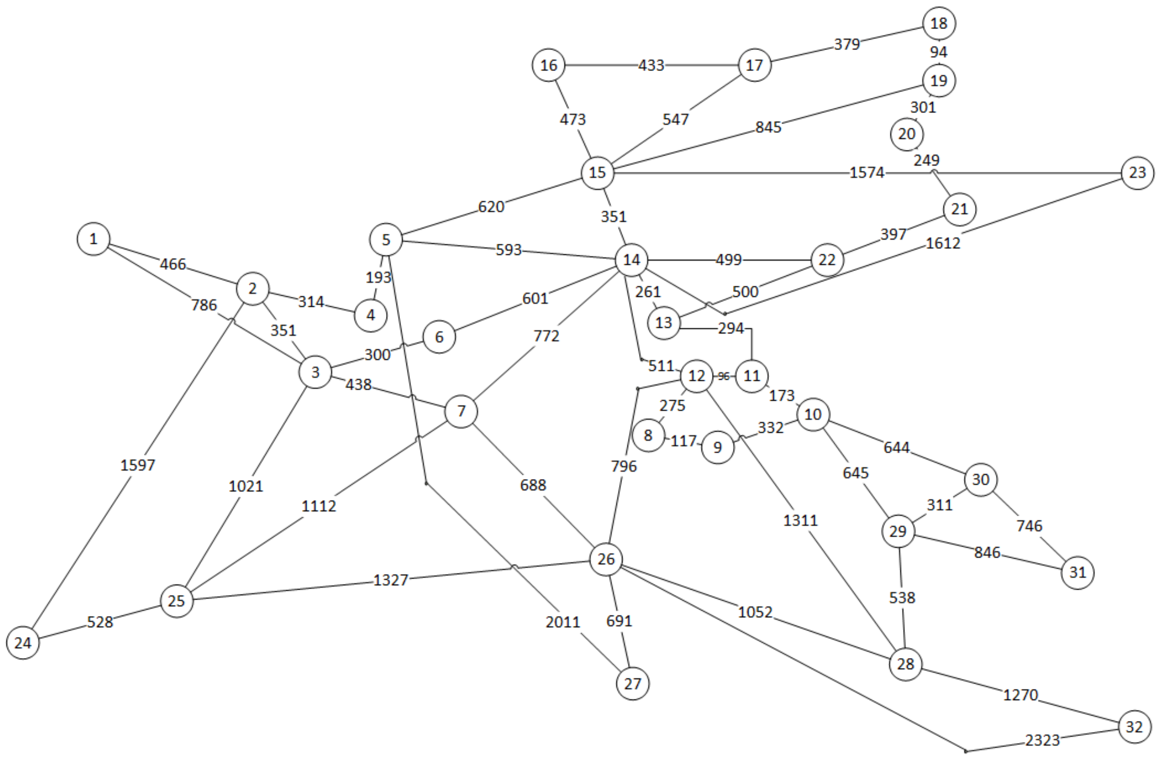


Figure 4.4: GEANT2 network topology.

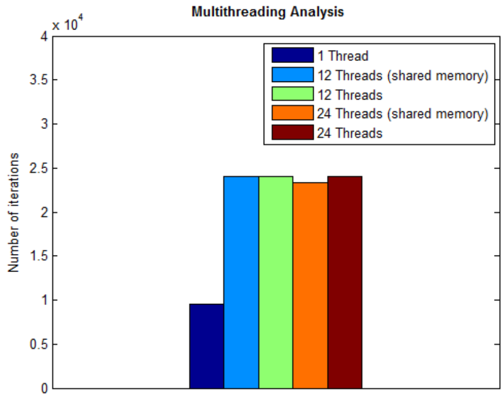


Figure 4.5: Multithread Analysis.

12 and 24 threads. Note that the execution time given to each run of the algorithm was always the same. Moreover, this set of results allow us to observe that the shared memory approach, either with 12 or 24 threads enables to find, on average, better cost solutions. For instance, T_3^{EON}, T_5^{EON}, T_6^{EON}, T_7^{EON}, T_8^{EON}, T_1^{GBN}, T_4^{GBN}, T_5^{GBN}, T_8^{GBN}, T_1^{GEANT2}, T_5^{GEANT2}, T_6^{GEANT2}, T_5^{NSFNET}, T_7^{NSFNET} and T_8^{NSFNET} represent situations where the average best cost solution was obtained using 12 threads with shared memory. On the other hand, 24 threads with shared memory gathered better results in T_1^{EON}, T_4^{EON}, T_2^{GBN}, T_3^{GBN}, T_1^{NSFNET}, T_2^{NSFNET} and T_9^{NSFNET}. The GEANT2 case studies seem not to take advantage

of this approach. One explanation for this behaviour is the really low iteration values obtained here when compared to the remaining case studies. The number of iterations is reduced because the number of end node pairs defined in this network is higher than the others. Consequently, each global algorithm cycle has to process more routing paths and deal with larger data structures. Therefore, it would be necessary to give more execution time to evaluate the benefits of shared memory approach on this specific network. Finally, before moving to a different computational metric, we notice that the thread gains in the results set discussed here (appendix B.1) are different from the unitary test firstly performed and shown in figure 4.5. This is most likely related to performance configurations of the computational platform as well as existing background processes that influence and originate fluctuations on the gains obtained.

Another relevant parameter that plays an important role on computational efficiency is the algorithm ability to handle unfeasible solutions and retrieve them from that state. Also, it is relevant to validate the local search block that is responsible for minimizing spectrum usage. At the end, it is desirable to build as many valid solutions as possible so it is more likely to find better cost solutions. For example, looking closer to T_7^{NSFNET}, we conclude that every single greedy solution generated was unfeasible. However, instead of discarding those solutions, the local search was able to turn feasible around 94% of them. Similarly, initially approximately 8% of the entire greedy solutions are unfeasible in T_8^{NSFNET}. After the local search step, no unfeasible solutions were accounted. In contrast, T_7^{EON} is only capable of retrieve around 12% of the unfeasible greedy solution, which implies that the traffic load and its profile is reaching the network capacity. Similar behaviours can be found in more cases.

We not only account the unfeasible solutions, but also the number of slot assignment failures. This situation is more critical in GBN, where a large number of solutions are rejected due to spectrum assignments that fail. For example, in T_2^{GBN} for 12 threads with shared memory, an average number of 247.6 solutions, in a total of nearly 39609.7 generated solutions, are discard during spectrum assignment due to physical constraints. Analysing the EON, one can see that solutions are rarely lost due to slot assignment failures. The same does not apply to GEANT2. For instance, observing T_6^{GEANT2} with 12 threads and shared memory, an average value of 143.3 iterations of a total of 9954.4 are invalid at the spectrum assignment. The NSFNET is the only network where spectrum assignment is easily accomplished. Consistently, the average number of invalid spectrum assignments with shared memory is less than the corresponding ones without shared memory. Remember that slot assignment, for the shared memory situation, is only performed when a best global incumbent solution is found, consequently the number of slot assignments performed is less than without shared memory, thus the number of failures will also be less. This constitutes an advantage in terms of efficiency, since each thread only runs the spectrum assignment on potential better solutions, handling and managing better the execution time.

Finally, a specific study concerning the number of paths computed in both fiber and logical candidate paths is conducted (see appendix B.2). The time required to calculate and organize these sets according to the description in the previous chapter, is added to the time of the best found solution. Here, we want to understand how different ranges of candidate paths influence solutions. Clearly, the number of paths considered may influence the time required to find the best solution and the cost value itself, since a very large number of paths may take longer to compute and a small set of paths may withdraw paths required to find lower cost solutions. This last behaviour is clearly seen when we use lower K values in both physical and logical graphs. More precisely, for the pair (10,5) the cost values achieved were always higher than with different K pairs, which proves the set to be too narrow to include the best cost solution. Each case study reveals consistently benefits when using a set bigger than the first one considered, (10,5). These benefits are reflected into better cost solutions. The major gains arise when we jump from (20,5) to a (10,20) scenario. This implies that the logical graph have an important role on the algorithm and allows improving the cost of the solutions.

Through the next sections, if nothing otherwise stated, the results presented were obtained using 12 threads employing the shared memory approach since, from the discussion above, this is generally the optimal approach.

4.3 Design Analysis for Fixed-Grid versus Flexible-Grid

It is expected that each case study will impact network cost differently. Moreover, the cost weight of each network equipment along with the weight of each feature available also depend on the case study. Thus, a first analysis considering the weight of each network equipment over the total network cost is presented in 4.6. This set of results provide a valuable first insight, especially on the algorithm validation.

In general, with the cost model used, the flexible grid options are rarely deployed. Instead, fixed grid alternatives and inverse-multiplexing are the dominant technologies chosen. The cost differences between fixed and flexible grid end node equipments are determinant and explain these results. Even with higher transparent reach, the flexible grid options are not capable of exploiting the cost efficiency expected by using less 3R points, again mainly due to the cost differences between network equipments In fact, there is a clear situation where flexible grid seems favourable, which is networks with short link lengths, such as GBN. Since optical reach is not a critical issue in these kind of network topologies, higher bit-rate connections, like 400 Gbps, are easily deployed. In networks with longer link lengths, 400 Gbps connections may not be feasible and even if feasible they might no be cost effective due to the cost rise introduced by regeneration. For instance, introducing 400 Gbps connections in the transmission alternatives set of NSFNET, EON or GEANT2 will bring no advantages since their average link lengths are closer to the optical transparent reach of the line-rate.

Among the different networks, the cost percentage of 3R regenerators is rather considerable in NSFNET, EON and GEANT2 while being inexistent in GBN. Also, muxponders and transponders are widely deployed but its cost percentage is more dependent on traffic pattern.

In addition, figures 4.7 and 4.8 illustrate the deployment cost value achieved and highest slot assigned respectively, to each case study. As can be seen, the cost values obtained increase progressively with the traffic matrix due to the increase of the traffic load. This does not apply for the last three traffic matrices of GBN. Remember that in those cases, 400 Gbps line rates were used. The cost of this transmission alternative together with the increase ability to groom client demands, explains the decrease in the deployment cost. Although the current low reach for 400 Gbps transmission, this alternative allows accommodating more demands and is more prepared to deal the future amount of traffic due to higher degree of grooming. Furthermore, EON and GEANT2 cases generate solutions with higher costs since they require more network equipment. NSFNET also requires a high number of network equipment. However, the cost values obtained are lower in comparison to EON and GEANT2, since the traffic load only ranges from 10 Tbps to 20.3 Tbps.

The next subsections provide alternative analysis, focusing on individual aspects and

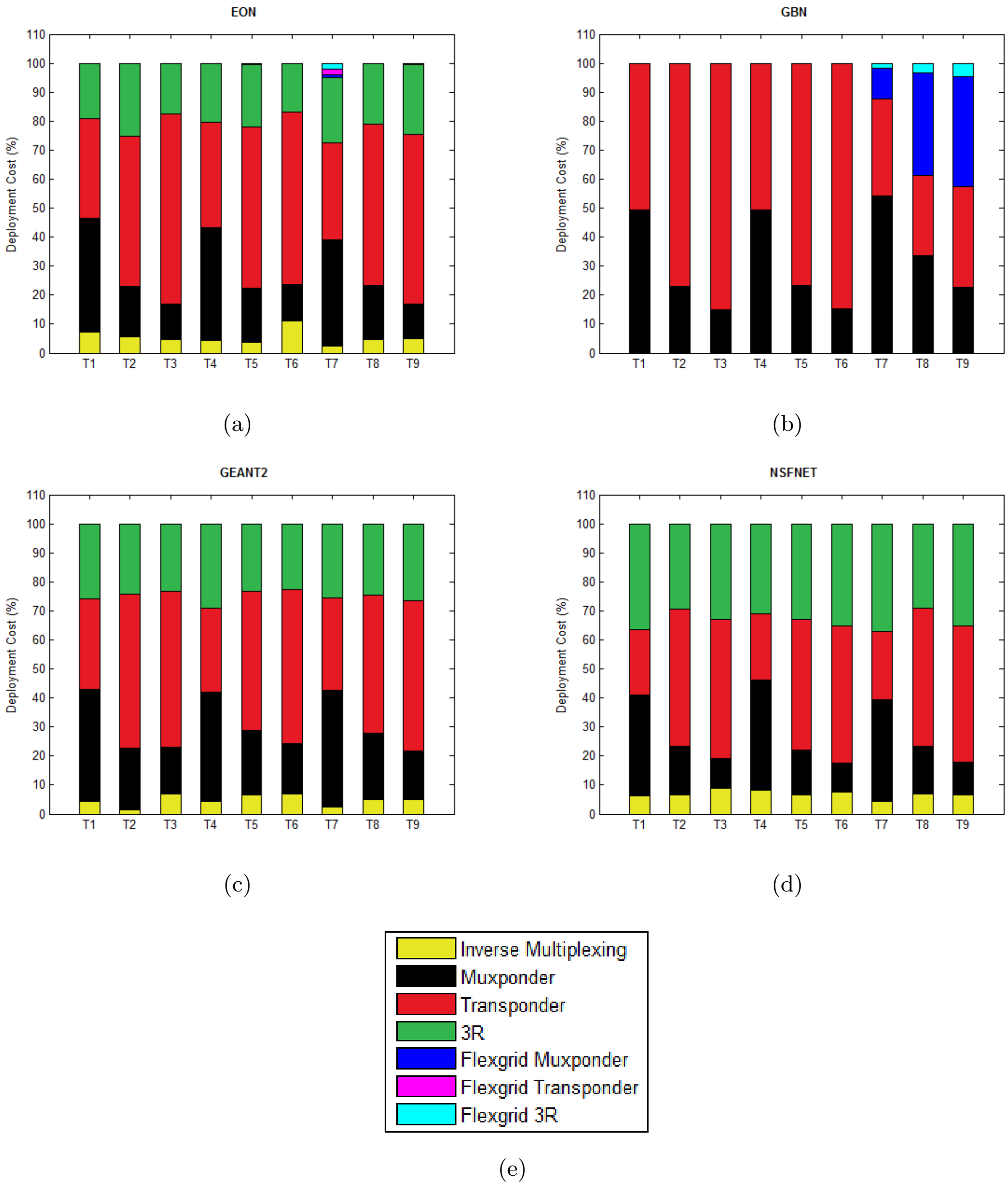


Figure 4.6: Network deployment cost for each case scenario. a) EON, b) GBN, c) GEANT2 and d) NSFNET. Legend of the network cost components e).

features compared against each network and traffic matrix, allowing a different analysis perspective.

4.3.1 3R Placement

The number of 3R regenerators needed in each case (figure 4.9) is barely dependent on traffic profile. Instead, it is heavily dependent on network topology and, more precisely, on the link lengths. The values shown comprise 3R regenerators of both fixed and, when applicable, flexible grid. However, since lower rate interfaces have longer transparent reaches,

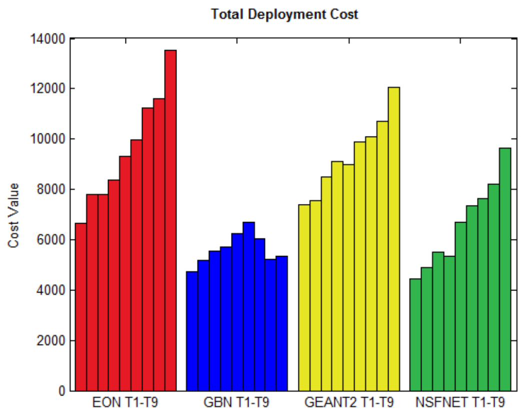


Figure 4.7: Total deployment cost in each solution shown in 4.6.

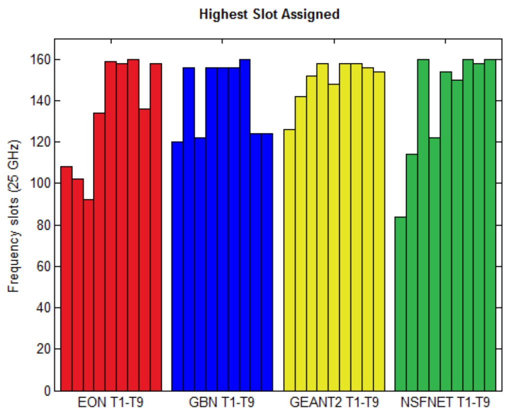


Figure 4.8: Highest slot assigned in each solution shown in 4.6.

when traffic is dominated by 10G demands less 3R regenerators are required. On the opposite, when traffic is dominated by 100G, the number of 3R elements should increase.

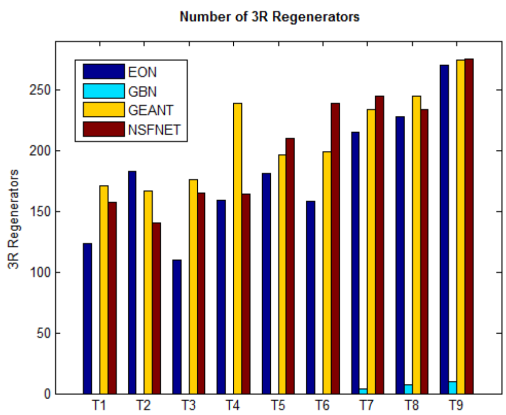
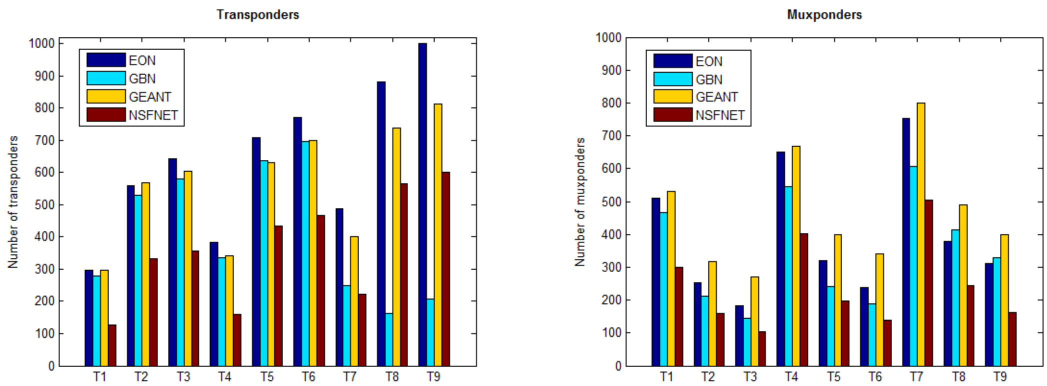


Figure 4.9: Number of 3R regenerators deployed.

4.3.2 Transponders and Muxponders

Opposite to 3R regenerators deployment, transponder and muxponders are dependent on traffic pattern. Muxponders are widely deployed when 10 Gbps demands are the dominant traffic elements. So, $T_{1,4,7}$ of every network have more muxponders than the other cases, as can be seen in figure 4.10(b). Transponders are then less used in $T_{1,4,7}$, but are more necessary in the remaining situations (figure 4.10(a)). Also, it is visible that more transponders are required as we move from T_1 to T_9. This is motivated by the traffic load increase.



(a) (b)

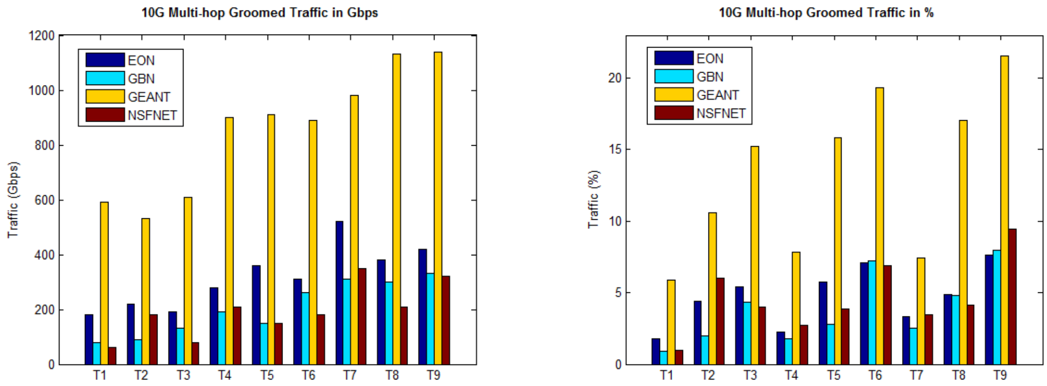
Figure 4.10: End node equipments: a) number of transponder and b) number of muxponders

4.3.3 Multi-Hop Grooming

The total traffic routed by multi-hop grooming is shown in figure 4.11(a) while the percentage view is illustrated in figure 4.11(b). Only 10 Gbps client demands are routed through multi-hop grooming in most of the case studies. Analysing the transmission options of our problem, one can see that the only way to have 40 Gbps spare inputs is by using the 100 Gbps OTU (2x40 Gbps). This will originate at most 1 spare input every time this OTU is used. Then, it is difficult to concatenate already installed OTUs to route 40 Gbps demands. Particularly, multi-hop grooming seems to be extremely useful in GEANT2 as it is capable of routing near 20% of its 10 Gbps traffic in a couple of situations. This is related to the high range of origin-destination pairs, E, generated in contrast to the other networks 4.2. This heavily affects the logical graph construction, resulting in more connections and a more mesh graph, and therefore the ability to perform multi-hop grooming. Moreover, $T_{7,8,9}$, of GBN are capable of using the logical graph to route 40 Gbps, as seen in figures 4.12(a) and 4.12(b). This happens as by introducing a 400 Gbps muxponder (10x40 Gbps), a lot more spare inputs will be available in the logical graph. Also, it is important to note that this feature is highly dependent on the logical graph and the respective current installed OChs. So, the order of provisioning the demands affect the possibilities to perform, or not, multi-hop grooming.

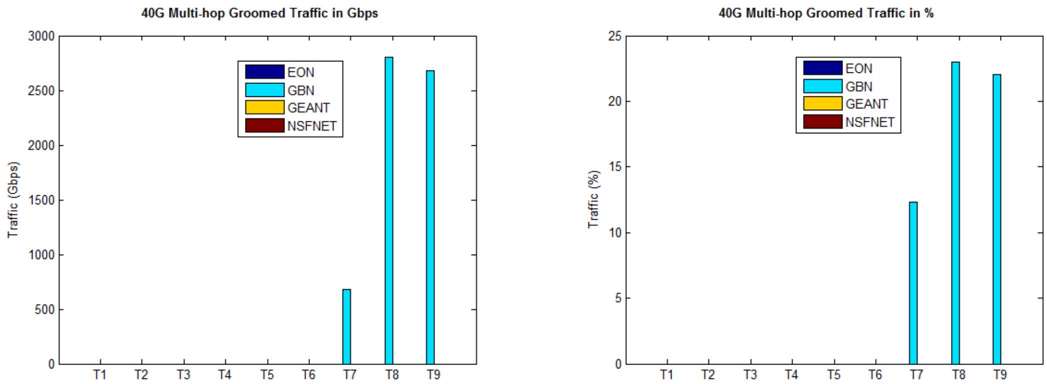
4.3.4 Inverse Multiplexing

This functionality is rarely used and its application is restricted to situations where transmitting 100 Gbps client signals at its native rate is not possible due to distance limitations



(a) (b)

Figure 4.11: Multi-hop grooming: a) total traffic routed in Gbps and b) in percentage.



(a) (b)

Figure 4.12: Multi-hop grooming: a) total traffic routed in Gbps and b) in percentage.

(figure 4.13). More precisely, the results show that inverse multiplexing is only used in long distance networks, especially in traffic matrices with higher number of demands of 100 Gbps. However, it also depends on the origin-destinations pairs randomly generated and it is possible to identify particular situations where is more cost efficient to use inverse-multiplexing instead of a transponder line rate.

Another issue related with inverse-multiplexing has to do with the spectrum usage. Splitting a 100 Gbps signal into three light paths results in extra spectrum usage. In case studies where unfeasibility is a reality, alternatives to this transmission option are well seen.

4.3.5 Spectrum Fragmentation

The first figure, 4.14(a), indicates the average fragmentation values of every network topology for each traffic matrix, while the second one indicates the highest fragmentation value of each instance, this is, the value of the most fragmented fiber in each case scenario. It is visible that GEANT2 network experiences higher values of fragmentation over all the

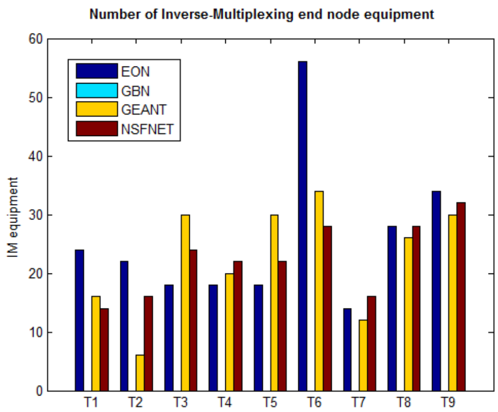


Figure 4.13: Number of Inverse-Multiplexing end node equipment deployed.

studied cases. On the opposite, NSFNET have the lower values of fragmentation. EON and GBN have approximately similar tendencies in most of the case studies. More precisely, EON suffers an average fragmentation increase after the third traffic matrix, while GBN shows a more constant and traffic independent fragmentation profile. Figure 4.14(b) identifies the highest slot gap of the solution and, similarly, GEANT2 owns the fibers with larger gaps, that can go up to around 120 slots in both T_3 and T_8. This means there is at least a fiber whose spectrum usage can be highly compacted.

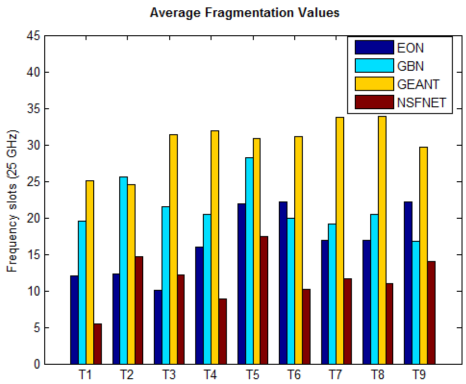
According to the problem definition made, 3R placement is tight with spectrum assignment and consequently with spectrum fragmentation. Remember that we only allow wavelength conversion of a given light path at nodes with a 3R regenerator deployed. So, 3R regenerators ease the spectrum assignment constraints thus leading, more likely, to situations with less fragmentation since a higher number of path segments are only required to choose the first n free and contiguous slots. Considering this, NSFNET is the network that presents lower fragmentation values and is, at the same time, the one with more number of regenerators per traffic volume. However, GEANT2 seems not to capable of benefiting from the 3R regenerators to obtain lower values of fragmentation.

The inclusion of a defragmentation algorithm is an option that would enable reducing the values before presented by compacting spectrum. However, performing defragmentation will impact the overall algorithm performance due to its complexity. So, considering the values of fragmentation obtained together with the desirable algorithm efficiency we can conclude that the spectrum assignment policy developed is suitable and fulfils the needs of our problem.

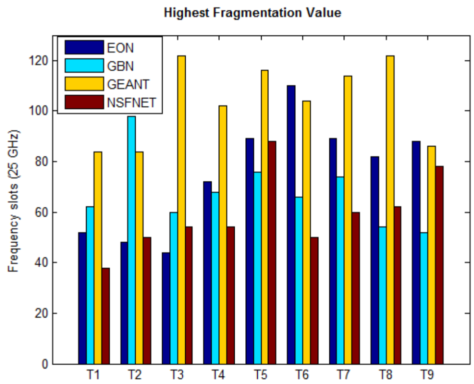
4.4 Conclusions

We show that the multi-thread approach developed is highly beneficial for algorithm efficiency. This is visible in the iterations gains and since it allows obtaining better solutions than 1 thread approach for the same execution time. Also, allowing communication between threads reveals, generally, computational benefits. It allows finding better solutions and reduces the time needed to obtain them.

The local search block that optimizes spectrum occupation instead of cost when a solution is beyond the defined physical limitations, was also analysed here. We can conclude that the



(a)



(b)

Figure 4.14: Fragmentation: a) average values and b) highest value

development of such functionality proved to be important, because in some case studies the algorithm could not be able to find valid solutions. Furthermore, measuring the number of unfeasible solutions allow us to see if the traffic load is near the network capacity, this is, when a large portion of the greedy unfeasible solutions are still unfeasible after the local search one can say that we are approaching the network capacity limit.

For traffic loads far from the physical limitation of the network, which can be seen in the highest slot number assigned, the algorithm easily builds feasible solutions and therefore it mainly chooses the cheaper transmission alternatives. For higher traffic loads, and particularly when unfeasible solutions appear, the algorithm is required to choose transmission alternatives that allow higher spectrum savings instead of lower cost alternatives.

Some networks are harder to perform spectrum assignment than others. This happens for short-range networks that have a low number of 3R, hardening the spectrum assignment constraints. GBN is the most critical network because the best cost solutions found do not require any 3R regenerator. Consequently, we see a high number of MSLS solutions discarded because of failures at this part of the algorithm. On the other hand, NSFNET do not have issues regarding sepctrum assignment failures. Mostly because of the high number of 3R regenerators deployed. In this point of view, deployment of 3R regenerators constitute an advantage.

Concerning the number of candidate paths provided to the algorithm, we observe benefits when considering more than 5 paths on the logical paths. It allows improving considerably the cost of the solutions. Furthermore, using more than 10 paths on the fiber graph proves to be profitable only in a few and specific case studies. However, the number of iterations performed is similar in both (10,20) and (20,20) scenarios which indicates no loss of performance by using a bigger set of candidate paths. Thus, in terms of the number of candidate paths, we may conclude that, despite the variations seen from case to case, the option that lead to the best solutions is the one with $K_{fis} = 10$ and $K_{log} = 20$.

Referring to the last three case scenarios of GBN, where 400G transmission is allowed, the results suggest the network capacity is increased due to the use of this line rate. Moreover, this line rate enables 40 Gbps demands to be routed using the multi-hop grooming technique. However, applying this line rate to long-haul networks such as EON, GEANT2 or NSFNET would not be so advantageous, in both cost and feasibility, due to the limited optical reach of this line rate.

More specifically on the design results, we conclude that 3R deployment is only dependent on the network topology and not on the traffic profile. This is more visible for long-haul networks since the 500 Km difference between the optical reaches of the transmissions alternatives hardly change the number of regenerators. On the opposite, the number of transponders and muxponders required vary according to the traffic weights. Multi-hop grooming reveals to be an excellent feature as long as the logical graph, dependent on the end node pairs generated, provides enough routing options. This is clearly the case for the GEANT2 network, as the high number of end node pairs result in a logical graph with more connections. Traffic percentage routed using this technique can be further increase by using muxponders with higher number of interfaces, that will likely result in more spare inputs. Finally, we seen that inverse-multiplexing is not a relevant feature in terms of usage percentage. Additionally, it wastes more spectrum than transmitting at the native line rate. However, we can not say how much the solution cost will fluctuate by disabling inverse-multiplexing.

Chapter 5

Conclusions

5.1 Summary

In this dissertation the problem of optical transport networks design was addressed, considering a wide set of functionalities that arise with the development of future optical network. We focused on static network planning and the main goal was to build a versatile optimization tool that could deal with the heterogeneity of client demands and apply current and future line-rates in a feasible and practical manner, considering, for instance the paradigm of flexible grid. So, this tool can provide important outcomes for network operators. Also, it was possible to apply and evaluate the benefits of a multi-thread strategy that improves the algorithm efficiency.

In the second chapter we briefly describe and explain some important concepts used during the development of the optimization algorithm. Also, concepts on EONs were presented and relevant issues regarding its evolution path were stated. Then, in chapter 3 the building steps of the algorithm were explained and finally in chapter 4 the case studies and results generated were reported.

We have shown that the multi-thread approach enables effective use of computational resources of the current multi-core platforms, and it effects positively the algorithm ability to find good solutions. Also, two approaches related with threads communication were validated. The shared memory approach is preferred over the other one although those benefits were not always seen. To dissipate doubts one should perform a similar test battery but with runs of 5 minutes instead.

The candidate paths strategy was used to route the client demands. We conclude that a certain number of paths on both graphs need to be considered so the algorithm is not conditioned. In fact, the highest K values used, $K_{fis} = 10$ and $K_{log} = 20$ is the most desired option since the computation time of these paths is almost negligible and due to the cost ordering made, the algorithm does not have to test every single path if it does not need to, thus, not losing performance.

A set of case studies were built to validate and perform the design analysis. In order to do this, four reference optical networks were chosen along with traffic matrices generated accordingly our input requirements. Concerning the design analyses itself, the outcomes of the algorithm allow to evaluate the cost percentage of each network element deployed over the total cost solution. Further information regarding multi-hop grooming, fiber fragmentation and inverse-multiplexing is also comprised. Multi-hop grooming is a feature very dependent on the end node pairs generated and order of demands provisioning but can be highly valuable as it enables routing considerable amount of traffic in some case studies. Furthermore, introducing muxponders with more interfaces, such as 10 x 40 Gbps, will likely bring more paths on the logical graph made by spare inputs. The spectrum assignment policy used, first-fit, is appropriate according to the fragmentation levels obtained and since we do no want to increase complexity and consequently lower the performance.

5.2 Future Work

As suggestions to further future work the following topics are proposed:

- Combine the spectrum assignment with the MSLS. This will allow having a different control over 3R node deployment. Even when a 3R regenerator is not needed, it may still be deployed to ease the slot assignment problem and reduce the number of invalid solutions resulting from this part of the algorithm;
- Evaluate the algorithm with different cost values for flexible grid equipments and evaluate at which point it becomes more cost effective than fixed grid options;
- Evaluate the importance of inverse-multiplexing more closely, by removing it from the input alternatives and inspecting how much the solution costs increase;
- In order to exploit other EONs advantages, one can consider introducing rate-flexible transponders and muxponders. This would allow comparing, for instance, the number of network equipments needed to design a fixed-grid solution over a flex-grid solution. This would require a dynamic network planning instead of a static one.

Bibliography

[1] A. A. M. Saleh and J. M. Simmons, "Technology and Architecture to Enable the Explosive Growth of the Internet," vol. 49, no. 1, pp. 126–132, 2013.

[2] A. Klekamp, U. Gebhard, A.-l. B. Labs, and D. Stuttgart, "Benefits for Mixed-Line-Rate (MLR) and Elastic Networks Using Flexible Frequency Grids," pp. 1–3, 2012.

[3] A. Eira, J. Pedro, and J. Pires, "Cost-optimized dimensioning of translucent WDM networks with Mixed-Line-Rate spectrum-flexible channels," *2012 IEEE 13th International Conference on High Performance Switching and Routing*, pp. 185–190, Jun. 2012. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6260848

[4] O. Gerstel, "Elastic Optical Networking : A New Dawn for the Optical Layer ?" no. February, pp. 12–20, 2012.

[5] I. Tomkos, E. Palkopoulou, and M. Angelou, "A survey of recent developments on flexible/elastic optical networking," *2012 14th International Conference on Transparent Optical Networks (ICTON)*, pp. 1–6, Jul. 2012. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6254409

[6] A. Rosa, C. Cavdar, S. Carvalho, J. Costa, and L. Wosinska, "Spectrum allocation policy modeling for elastic optical networks," *High Capacity Optical Networks and Emerging/Enabling Technologies*, pp. 242–246, Dec. 2012. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6421472

[7] J. Berthold, S. Member, A. A. M. Saleh, L. Blair, J. M. Simmons, S. Member, and A. N. S. Evolution, "Optical Networking : Past , Present , and Future," vol. 26, no. 9, pp. 1104–1118, 2008.

[8] M.-J. Li and D. a. Nolan, "Optical Transmission Fiber Design Evolution," *Journal of Lightwave Technology*, vol. 26, no. 9, pp. 1079–1092, May 2008. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4542888

[9] R. Ramaswami, K. Sivarajan, and G. Sasaki, *Optical Networks - A Practical Perspective*, 3rd ed., 2010.

[10] G. Keiser, *Optical Fiber Communications*, 3rd ed., 2000.

[11] M. Carroll, "The Operator ' s View of OTN Evolution," no. September, pp. 46–52, 2010.

[12] "DWDM Overview." [Online]. Available: http://www.cisco.com/en/US/docs/optical/15000r7_0/dwdm/planning/guide/d7ovw.html

[13] F. Rambach, B. Konrad, L. Dembeck, U. Gebhard, M. Gunkel, M. Quagliotti, L. Serra, and V. López, "A Multilayer Cost Model for Metro/Core Networks," *Journal of Optical Communications and Networking*, vol. 5, no. 3, p. 210, Feb. 2013. [Online]. Available: http://www.opticsinfobase.org/abstract.cfm?URI=jocn-5-3-210

[14] Cisco, "Cisco ONS 15454 DWDM Reference Manual," 2012.

[15] M. Jinno, B. Kozicki, H. Takara, A. Watanabe, and Y. Sone, "Distance-Adaptive Spectrum Resource Allocation in Spectrum-Sliced Elastic Optical Path Network," no. August, pp. 138–145, 2010.

[16] S. Gringeri, B. Basch, V. Shukla, R. Egorov, T. J. Xia, and V. Laboratories, "Flexible Architectures for Optical Transport Nodes and Networks," no. July, pp. 40–50, 2010.

[17] A. N. Pinto, R. M. Morais, J. Pedro, and P. Monteiro, "Cost evaluation in optical networks: Node architecture and energy consumption," *2012 14th International Conference on Transparent Optical Networks (ICTON)*, pp. 1–4, Jul. 2012. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6253772

[18] W. I. Way, "Next Generation ROADM Architectures," *OSA*, pp. 8–10, 2012.

[19] R. M. Morais, J. a. Pedro, P. Monteiro, and A. N. Pinto, "Impact of Node Architecture in the Power Consumption and Footprint Requirements of Optical Transport Networks," *Journal of Optical Communications and Networking*, vol. 5, no. 5, p. 421, Apr. 2013. [Online]. Available: http://www.opticsinfobase.org/abstract.cfm?URI=jocn-5-5-421

[20] ITU-T, "ITU-T recommendation G.872. Architecture of Optical Transport Networks," 2012.

[21] ——, "ITU-T recommendation G.709. Interfaces for the Optical Transport Network," 2012.

[22] M. Jinno, H. Takara, B. Kozicki, Y. Tsukishima, and Y. Sone, "Spectrum-Efficient and Scalable Elastic Optical Path Network : Architecture , Benefits , and Enabling Technologies," no. November, pp. 66–73, 2009.

[23] P. Monteiro, A. D. Sousa, M. Ribeiro, T. Trota, and G. Sahin, "Algorithms in the Deployment of Optical Transport Networks," *International Conference on Transparent Optical Networks (ICTON)*, pp. 2–5, 2013.

[24] N. Yoshikane, K. Igarashi, and T. Tsuritani, "Comparison of network cost and power consumption between a single-line-rate network and a flexible-bitrate network in the beyond-100-Gb/s networks era," *Optical Fiber Communication Conference/National Fiber Optic Engineers Conference 2013*, no. i, p. NW3E.4, 2013. [Online]. Available: http://www.opticsinfobase.org/abstract.cfm?URI=NFOEC-2013-NW3E.4

[25] A. Eira, J. a. Pedro, and J. a. Pires, "On the Impact of Optimized Guard-Band Assignment for Superchannels in Flexible-Grid Optical Networks," pp. 5–7, 2013.

[26] E. Martins, M. Pascoal, and J. Santos, "An Algorithm For Ranking Loopless Paths," pp. 1–12, 1999.

[27] "Reference Transport Netwoks." [Online]. Available: http://www.av.it.pt/anp/on/refnet2.html

Appendix A

Traffic Matrices

The following tables provide, in detail, the traffic matrices used. For each randomly generated source-destination pair, the number of, also randomly generated, bandwidth units of 10, 40 and 100 Gbps is given. These values vary from matrix to matrix so its consistent with the profiles built (table 4.2). The numbers used to identify each source-destination pair comply with the respective figures 4.1, 4.2, 4.3 and 4.4. There are 9 tables for each reference network, $T_i^{network}$.

T_1^{EON}					T_2^{EON}					T_3^{EON}				
s_d	t_d	n_{10G}	n_{40G}	n_{100G}	s_d	t_d	n_{10G}	n_{40G}	n_{100G}	s_d	t_d	n_{10G}	n_{40G}	n_{100G}
1	3	11	0	0	1	4	12	2	0	1	2	8	5	0
1	4	16	2	1	1	5	7	3	0	1	9	7	4	0
1	5	17	4	1	1	7	6	6	3	1	19	5	3	1
1	11	18	1	1	1	9	10	1	3	2	3	5	8	1
1	13	19	1	1	1	12	7	1	0	2	7	9	4	3
1	15	21	4	0	1	17	9	4	0	2	10	6	2	1
1	16	26	3	0	1	19	10	1	3	2	14	5	6	1
2	3	21	4	2	2	3	7	4	0	2	15	10	5	0
2	5	17	1	0	2	4	9	5	0	2	18	5	4	2
2	8	16	0	0	2	8	7	4	0	2	19	3	4	1
2	16	29	4	0	2	9	13	1	0	3	4	5	1	2
2	19	16	2	1	2	10	10	5	1	3	7	3	4	0
3	4	18	0	0	2	11	8	4	1	3	11	5	7	0
3	6	19	4	0	2	12	11	7	3	3	12	8	3	2
3	9	16	1	0	2	15	7	5	0	3	14	9	1	6
3	19	13	1	1	2	16	10	1	1	4	5	5	6	0
4	5	19	2	1	2	17	5	4	1	4	7	7	2	1
4	8	11	1	3	2	19	5	3	1	4	8	8	3	1
4	11	18	4	0	3	6	10	5	2	4	15	4	0	1
4	13	19	5	0	3	14	6	6	1	4	17	6	6	1
4	18	13	5	1	3	16	7	4	0	4	18	3	2	2
4	19	15	0	0	4	5	9	2	0	4	19	6	6	4
5	6	21	5	0	4	9	14	9	0	5	6	5	5	1
5	7	17	1	0	4	16	8	5	1	5	7	7	7	0

5	8	24	1	1	4	19	18	3	0	5	8	11	4	1
5	14	21	2	0	5	10	14	2	4	5	9	4	3	0
5	15	16	0	1	5	11	5	3	3	5	14	7	7	2
5	17	19	3	1	5	12	5	3	0	5	15	5	8	1
5	18	11	3	1	5	19	11	3	1	5	17	6	7	0
6	9	16	1	0	6	7	8	3	2	5	19	10	5	1
6	14	23	0	1	6	15	14	5	0	6	8	5	5	3
6	18	20	3	3	7	8	11	4	1	6	12	6	8	0
7	10	11	0	0	7	9	9	4	0	6	13	3	4	4
7	11	17	3	1	7	11	14	5	1	6	15	9	2	1
7	13	18	0	0	7	12	8	6	3	7	14	8	5	1
7	16	20	6	0	7	14	7	7	1	7	15	3	4	2
7	19	22	1	1	8	10	7	3	0	7	16	5	1	2
8	9	20	3	0	8	15	5	3	1	7	18	7	5	1
8	15	26	2	1	9	10	9	3	0	8	9	4	6	2
8	17	17	1	1	9	11	7	5	1	8	19	7	3	2
9	12	13	3	1	9	13	10	1	0	9	10	5	4	1
9	14	22	2	3	9	16	11	2	0	9	15	7	3	1
10	12	20	2	1	9	17	4	10	1	9	18	8	4	3
10	18	19	2	0	9	19	9	7	1	9	19	6	1	1
11	14	21	1	2	10	11	8	2	1	10	11	3	4	1
11	19	17	1	2	10	13	6	6	2	10	12	8	0	3
12	13	18	2	0	10	16	6	5	0	10	19	5	3	3
12	14	10	5	1	10	19	13	10	0	11	14	7	11	0
12	15	14	1	0	11	14	10	5	1	12	15	12	2	1
12	16	14	1	0	11	15	5	2	1	12	16	5	10	0
12	17	14	2	2	12	13	5	5	2	12	18	9	3	3
12	19	16	2	0	13	19	14	5	0	13	15	8	7	2
14	17	14	2	1	14	15	3	2	0	14	16	7	3	3
16	19	18	3	1	15	16	10	5	0	16	17	4	3	1
17	18	12	2	0	15	17	12	7	1	16	18	5	5	0
17	19	9	4	0	15	19	8	4	1	17	19	6	5	2
18	19	22	1	1	16	17	7	8	0	18	19	1	7	1

T_4^{EON}					T_5^{EON}					T_6^{EON}				
s_d	t_d	n_{10G}	n_{40G}	n_{100G}	s_d	t_d	n_{10G}	n_{40G}	n_{100G}	s_d	t_d	n_{10G}	n_{40G}	n_{100G}
1	2	16	4	0	1	2	12	3	0	1	3	4	5	2
1	3	10	2	1	1	3	4	4	2	1	5	3	4	2
1	5	13	2	3	1	5	9	5	0	1	8	7	3	3
1	14	8	0	1	1	8	3	2	0	1	9	6	5	0
1	18	17	3	0	1	9	5	5	1	1	14	6	5	1
2	4	14	2	0	1	11	5	4	2	1	16	8	3	1
2	5	12	2	0	1	18	8	2	1	1	18	7	4	3
2	6	13	3	0	1	19	14	0	0	2	5	4	1	1

2	8	15	3	0	2	4	4	2	1	2	6	7	6	1
2	9	15	2	1	2	6	9	6	2	2	11	1	4	1
2	11	13	1	1	2	7	6	3	0	2	13	6	1	1
2	12	15	1	0	2	8	13	5	0	2	14	3	4	0
2	14	15	1	1	2	15	4	5	1	3	5	6	3	2
2	15	12	0	0	2	16	11	5	0	3	7	4	4	3
2	16	21	3	0	2	17	6	5	2	3	8	5	4	3
2	17	13	3	0	2	19	6	6	0	3	9	3	2	1
2	18	14	1	0	3	4	6	3	1	3	12	2	4	0
3	6	14	2	0	3	5	4	2	0	3	14	9	4	1
3	8	21	4	1	3	6	6	3	0	3	16	1	4	0
3	9	14	0	0	3	7	6	2	0	3	17	14	3	1
3	10	15	3	1	3	9	6	5	1	3	18	4	3	0
3	12	14	1	0	3	10	15	1	0	4	5	6	3	1
3	13	12	3	2	3	12	11	5	1	4	7	3	3	1
3	17	9	0	1	3	13	7	3	0	4	14	8	4	2
3	19	17	3	0	3	16	9	4	0	4	17	6	3	4
4	5	16	0	0	3	17	12	8	1	5	6	9	3	0
4	8	6	2	1	3	18	3	1	3	5	7	4	5	1
4	9	14	4	2	3	19	8	6	1	5	9	8	4	2
4	10	16	1	0	4	7	6	1	0	5	10	4	7	2
4	11	14	2	0	4	8	5	2	1	5	11	5	2	0
4	13	21	0	1	4	10	6	4	0	5	15	5	2	4
4	18	13	1	0	4	12	4	2	2	5	16	9	4	2
4	19	24	2	0	4	17	9	6	0	5	17	7	6	4
5	6	14	2	2	5	6	7	4	0	5	18	3	3	1
5	8	13	3	2	5	9	5	2	2	6	7	4	5	0
5	11	6	3	1	5	13	6	4	0	6	10	6	2	0
5	12	11	2	0	5	14	11	2	0	6	13	8	3	0
5	13	11	2	1	5	15	7	5	1	6	15	4	2	1
5	14	13	0	0	5	16	6	7	0	6	16	6	1	2
5	18	18	1	0	5	17	6	6	1	6	17	5	3	4
6	8	19	3	0	6	10	5	3	2	6	18	3	1	0
6	11	19	1	1	6	14	5	0	0	7	8	3	2	0
6	14	20	1	1	6	15	9	2	0	7	12	2	3	0
6	17	16	4	0	6	17	7	5	1	7	14	3	5	0
6	19	8	2	0	6	19	7	3	1	7	15	10	6	0
7	8	18	3	0	7	8	9	3	0	7	17	8	4	2
7	9	12	3	1	7	9	7	9	1	7	18	5	4	0
7	12	9	0	1	7	10	8	5	0	7	19	8	4	2
7	14	23	1	1	7	11	6	4	1	8	10	7	4	1
7	15	14	2	0	7	14	5	4	0	8	13	6	7	0
7	17	11	0	0	7	16	6	4	2	8	14	11	3	0
8	12	18	4	0	7	17	8	2	0	8	15	5	5	2
8	13	16	0	0	7	19	2	2	1	8	18	9	3	0
8	14	14	6	0	8	10	8	5	4	8	19	6	5	2
8	15	18	1	0	8	11	11	3	2	9	10	4	4	2

8	17	18	1	1	8	13	10	2	0	9	12	6	3	1
9	10	10	2	2	8	15	9	5	1	9	16	7	2	1
9	11	11	2	2	8	17	7	4	1	9	18	0	4	3
9	12	12	4	1	8	19	3	3	0	9	19	3	2	1
9	16	12	1	0	9	11	9	3	0	10	12	1	5	0
9	17	17	2	1	9	14	4	3	4	10	14	5	1	2
9	19	16	1	1	9	16	4	4	0	10	16	7	5	1
10	11	8	0	0	9	17	10	4	0	10	17	3	6	0
10	13	22	1	0	9	19	10	5	0	11	12	2	3	0
10	14	11	1	0	10	13	7	1	1	11	13	3	7	1
10	18	18	0	1	10	19	8	4	2	11	14	2	4	1
10	19	9	1	0	11	17	14	3	1	11	16	5	6	1
11	13	17	2	0	11	18	9	6	0	11	17	5	5	0
11	16	20	1	2	12	15	5	6	0	11	18	5	1	0
11	17	11	0	0	12	16	11	3	1	11	19	2	2	2
11	18	23	0	2	12	19	13	2	0	12	13	2	4	1
11	19	13	4	0	13	16	9	3	0	12	15	4	3	1
12	13	13	1	0	13	17	2	3	1	12	16	4	4	0
12	14	22	0	1	13	18	3	3	1	13	15	6	3	3
12	17	12	2	0	14	15	6	3	4	13	16	3	1	2
12	19	16	3	1	14	16	7	2	0	13	17	6	7	1
13	14	14	4	2	14	17	12	4	2	13	18	8	2	3
13	17	14	2	0	14	18	9	2	0	14	17	6	3	0
13	18	22	0	2	14	19	8	2	1	15	16	4	5	1
14	15	7	2	1	15	16	8	4	0	15	17	3	4	1
14	16	19	4	0	15	17	11	4	0	15	18	8	2	1
14	19	18	2	0	15	18	5	3	1	15	19	5	5	1
15	18	17	0	0	16	17	11	4	0	16	17	3	6	0
15	19	15	2	2	17	19	3	0	0	17	18	3	3	0
16	18	16	0	0	18	19	5	1	0	17	19	7	4	2

T_7^{EON}					T_8^{EON}					T_9^{EON}				
s_d	t_d	n_{10G}	n_{40G}	n_{100G}	s_d	t_d	n_{10G}	n_{40G}	n_{100G}	s_d	t_d	n_{10G}	n_{40G}	n_{100G}
1	2	16	2	0	1	4	10	3	0	1	2	5	4	2
1	3	13	3	0	1	6	7	3	0	1	3	4	1	1
1	5	14	4	0	1	7	8	2	0	1	4	6	3	0
1	6	15	3	1	1	8	9	3	4	1	5	4	6	0
1	7	20	1	0	1	9	5	3	0	1	6	3	4	2
1	8	13	3	0	1	10	3	3	0	1	7	5	2	0
1	9	13	3	2	1	11	7	1	2	1	9	4	1	2
1	10	6	3	1	1	14	11	7	1	1	10	2	4	2
1	11	12	3	1	1	18	10	3	0	1	11	7	1	1
1	12	15	3	2	1	19	4	5	0	1	13	5	4	0
1	13	14	3	0	2	3	13	3	0	1	14	5	3	0

1	14	12	1	0	2	4	4	3	0	1	15	9	5	0
1	15	7	1	1	2	6	9	5	1	1	17	6	2	0
1	17	10	2	1	2	7	14	4	2	1	18	2	1	1
1	19	13	2	1	2	9	12	0	1	1	19	6	6	0
2	4	13	1	0	2	10	8	2	0	2	6	0	4	0
2	6	11	1	0	2	12	7	5	0	2	7	4	2	0
2	7	13	1	1	2	13	8	3	0	2	9	5	4	1
2	10	8	3	0	2	14	6	5	1	2	10	3	3	1
2	12	17	1	1	2	15	7	5	1	2	11	4	6	1
2	14	10	1	0	2	16	4	3	1	2	12	4	2	1
2	15	19	2	0	2	17	9	6	0	2	14	6	0	1
2	18	13	1	0	2	19	9	5	0	2	15	5	3	1
3	4	16	0	0	3	4	4	3	0	2	16	6	1	2
3	5	13	1	1	3	6	9	3	1	2	17	5	4	0
3	6	14	3	1	3	11	6	3	0	2	18	6	5	2
3	7	21	1	0	3	12	3	1	0	3	4	2	1	0
3	8	11	3	2	3	13	10	6	1	3	6	9	2	1
3	9	17	0	0	3	14	6	1	0	3	7	6	6	1
3	10	9	3	1	3	15	6	1	1	3	9	6	4	0
3	11	15	4	1	3	18	9	1	0	3	10	4	5	2
3	12	16	2	1	3	19	8	5	0	3	11	4	3	1
3	14	13	2	0	4	5	8	2	1	3	12	7	1	1
3	15	18	0	0	4	6	3	3	0	3	13	6	3	1
3	16	14	3	0	4	7	6	0	0	3	15	9	6	0
3	17	18	4	1	4	9	14	4	0	3	16	3	4	3
3	19	10	2	0	4	10	7	3	0	3	18	4	4	1
4	5	18	1	0	4	12	3	3	3	3	19	3	1	0
4	6	15	7	0	4	16	3	3	0	4	5	6	3	0
4	7	16	2	2	4	17	6	1	0	4	6	8	3	0
4	11	9	3	0	4	18	3	2	1	4	7	4	5	2
4	13	12	0	2	5	8	3	2	0	4	8	2	5	3
4	14	13	2	0	5	9	5	3	1	4	10	2	3	0
4	15	8	2	1	5	10	10	3	0	4	11	7	2	2
4	16	16	2	1	5	11	5	2	1	4	13	6	3	0
4	17	12	1	0	5	12	7	5	0	4	14	8	3	0
4	18	5	1	0	5	13	5	2	1	4	15	4	4	1
5	7	10	1	0	5	14	8	2	0	4	16	11	2	1
5	9	18	1	1	5	15	9	4	1	4	19	3	3	1
5	10	17	2	1	5	16	7	4	0	5	7	5	7	2
5	11	14	1	0	5	17	13	6	4	5	8	6	4	0
5	12	15	0	1	5	18	2	3	1	5	9	2	3	1
5	13	13	2	0	5	19	7	3	0	5	10	4	6	2
5	16	13	0	0	6	7	6	1	0	5	11	5	1	2
5	17	12	0	0	6	11	6	7	0	5	12	3	4	1
5	18	10	0	0	6	12	6	5	0	5	14	9	5	2
5	19	18	0	0	6	14	12	3	1	5	15	2	1	1
6	7	11	3	2	6	15	5	5	1	5	16	2	0	0

6	10	17	1	1	6	16	7	6	0	5	17	5	1	2
6	12	12	2	1	6	17	5	4	1	5	18	6	4	1
6	13	13	1	0	6	18	9	3	1	5	19	6	2	0
6	15	6	2	0	7	9	9	1	2	6	7	4	2	1
6	18	14	2	1	7	11	7	5	0	6	9	4	2	1
7	9	14	1	1	7	13	6	5	0	6	10	6	7	2
7	10	14	0	0	7	14	4	4	2	6	15	1	1	1
7	11	12	0	1	7	16	8	2	0	6	16	4	4	2
7	13	16	3	0	7	17	4	6	0	6	17	5	1	0
7	15	21	1	1	7	18	10	4	0	6	18	7	8	0
7	17	13	0	1	7	19	10	3	1	7	8	4	7	1
8	9	12	3	1	8	9	10	5	0	7	9	7	2	0
8	11	13	1	2	8	11	10	3	1	7	10	4	4	2
8	12	16	3	0	8	12	11	4	0	7	11	2	4	1
8	13	17	2	1	8	13	4	2	0	7	14	4	4	1
8	14	11	1	1	8	14	7	6	0	7	15	6	3	1
8	17	21	1	0	8	15	5	4	0	7	18	5	1	1
8	18	19	1	0	8	16	8	3	1	7	19	6	1	2
8	19	21	0	0	8	18	2	4	1	8	9	3	7	0
9	11	18	2	1	8	19	4	2	1	8	10	5	3	4
9	12	22	0	1	9	10	1	4	0	8	12	6	2	1
9	13	12	2	1	9	12	5	1	2	8	14	6	4	2
9	14	16	0	0	9	14	8	2	2	8	15	6	2	2
9	15	13	1	0	9	15	7	5	0	8	16	5	4	0
9	16	13	2	0	9	17	7	4	0	8	18	3	7	1
9	17	10	0	0	9	19	4	2	1	9	11	4	6	1
9	18	13	1	1	10	11	9	4	0	9	12	6	1	0
9	19	16	2	0	10	13	4	2	1	9	13	7	2	1
10	11	13	3	2	10	14	5	5	2	9	14	1	6	0
10	12	10	2	0	10	15	11	2	1	9	17	2	3	1
10	16	10	2	0	10	16	8	5	0	9	18	4	3	1
10	17	15	3	0	10	17	5	4	1	9	19	2	6	1
10	19	18	2	1	10	18	4	4	0	10	13	9	1	1
11	12	16	3	0	11	13	7	4	0	10	16	7	1	4
11	13	10	1	1	11	15	3	1	1	10	17	3	4	2
11	14	16	1	0	11	16	3	2	1	10	18	7	7	1
11	17	17	1	1	11	17	4	0	3	11	12	4	3	0
12	13	19	2	1	11	19	5	2	1	11	16	3	3	2
12	15	9	2	0	12	13	4	5	0	11	17	3	5	2
12	16	11	1	2	12	15	9	0	1	11	18	3	3	2
12	18	11	2	0	12	16	5	3	0	11	19	4	4	1
13	14	13	2	1	12	18	4	4	1	12	13	7	3	1
13	15	7	2	1	12	19	13	6	2	12	15	10	1	1
13	16	11	1	2	13	14	10	7	1	12	16	4	4	2
13	19	13	2	0	13	15	5	2	1	12	17	5	11	3
14	17	14	0	1	13	16	8	2	0	12	19	9	0	1
14	18	11	1	0	13	17	8	4	1	14	15	3	2	3

14	19	12	1	0	14	16	7	0	0	14	16	4	0	0
15	16	23	1	0	14	18	5	1	1	14	17	3	7	1
15	18	14	1	1	15	16	10	5	4	14	19	3	4	1
15	19	16	2	1	15	17	5	4	1	15	16	5	7	0
16	17	11	0	0	15	18	11	3	0	15	17	4	6	1
16	18	15	1	0	16	18	8	3	3	15	19	2	1	3
17	18	9	1	0	17	18	9	5	1	16	18	5	4	2
17	19	12	4	1	17	19	5	1	2	17	18	4	3	0
18	19	15	2	0	18	19	2	4	0	17	19	8	6	4

T_1^{GBN}					T_2^{GBN}					T_3^{GBN}				
s_d	t_d	n_{10G}	n_{40G}	n_{100G}	s_d	t_d	n_{10G}	n_{40G}	n_{100G}	s_d	t_d	n_{10G}	n_{40G}	n_{100G}
1	2	20	3	1	1	8	13	5	2	1	2	13	6	0
1	5	21	1	1	1	9	13	6	1	1	3	9	2	0
1	8	19	2	0	1	12	9	10	1	1	11	5	5	1
1	10	19	1	0	1	14	5	14	1	1	13	9	4	2
1	11	22	2	0	2	3	7	5	1	2	9	7	2	5
1	12	17	3	0	2	6	10	5	1	2	10	12	8	3
1	13	16	1	1	2	7	11	4	1	2	15	10	5	1
1	16	22	3	2	2	12	4	3	0	3	4	8	4	0
2	4	20	4	0	2	13	13	3	0	3	5	6	5	2
2	9	19	1	2	2	14	10	7	1	3	14	7	6	2
2	17	21	4	1	2	16	11	7	1	4	8	2	6	3
3	7	18	4	1	3	6	8	3	2	4	9	3	5	2
3	9	22	3	2	3	11	10	5	2	4	11	11	5	1
3	14	24	1	3	3	14	11	4	2	4	12	5	8	0
3	15	27	1	1	3	15	9	4	0	4	17	7	4	2
3	16	24	1	1	3	16	12	12	1	5	6	4	3	3
3	17	19	2	1	4	6	8	4	0	5	7	5	4	0
4	9	21	1	1	4	8	8	6	0	5	8	7	5	2
4	10	11	1	1	4	9	7	4	0	5	12	6	3	4
4	11	26	2	2	4	10	7	2	1	5	13	11	5	2
5	7	22	0	0	5	6	13	1	0	5	15	7	8	0
5	10	17	2	1	5	10	7	1	2	5	16	5	6	0
5	11	19	2	1	5	12	11	7	1	6	9	3	9	4
5	12	18	1	0	6	7	11	4	2	6	10	8	6	1
5	14	13	2	0	6	10	11	7	1	6	11	5	7	1
6	8	21	4	0	6	14	10	4	1	6	13	4	4	2
6	12	24	2	0	6	15	15	4	1	6	17	6	7	2
7	8	21	2	1	6	16	7	6	2	7	10	9	5	2
7	16	20	1	0	6	17	19	5	1	7	15	7	5	3
8	10	26	1	2	7	9	11	2	0	8	9	7	4	2
8	12	21	2	1	7	17	23	2	1	8	10	1	2	0
8	13	24	1	1	9	10	10	3	1	8	14	5	6	1

8	14	20	1	1	9	13	7	5	0	8	16	5	6	0
8	15	16	1	0	9	14	15	3	1	8	17	7	6	0
8	16	17	2	0	9	17	11	6	0	9	11	10	3	1
9	16	20	4	1	10	12	10	8	1	9	13	6	2	2
10	11	17	4	2	10	15	9	3	0	10	11	3	3	2
10	15	25	5	1	10	16	15	5	1	10	15	6	4	0
11	15	12	5	1	10	17	5	5	2	11	12	5	4	1
12	13	11	3	1	11	14	8	6	1	11	16	8	8	2
12	15	20	3	1	12	17	4	4	2	12	14	7	3	1
13	15	22	3	0	13	14	11	2	4	12	15	7	4	3
14	16	28	3	1	13	16	8	4	1	12	16	11	5	0
14	17	18	1	2	14	17	7	4	1	14	16	5	6	3
15	17	20	4	1	15	16	6	6	0	15	17	6	2	2

T_4^{GBN}					T_5^{GBN}					T_6^{GBN}				
s_d	t_d	n_{10G}	n_{40G}	n_{100G}	s_d	t_d	n_{10G}	n_{40G}	n_{100G}	s_d	t_d	n_{10G}	n_{40G}	n_{100G}
1	3	12	0	0	1	6	12	2	1	1	2	7	3	1
1	4	13	4	1	1	8	5	3	0	1	5	6	9	4
1	6	15	5	1	1	9	5	5	0	1	7	4	6	0
1	8	16	0	1	1	10	7	5	0	1	8	5	8	1
1	9	14	0	1	1	11	13	2	0	1	9	5	2	2
1	15	18	0	1	1	12	7	4	0	1	10	3	2	1
1	16	21	1	0	1	15	4	5	1	1	13	8	4	0
1	17	20	2	1	1	16	5	8	0	1	14	7	5	0
2	5	14	0	1	1	17	8	4	0	1	15	6	3	1
2	6	10	3	1	2	5	4	4	0	1	16	6	1	2
2	8	14	2	0	2	6	7	1	1	2	3	5	5	0
2	10	18	5	1	2	7	4	3	2	2	6	9	3	3
2	11	17	1	0	2	9	5	4	2	2	7	4	2	1
2	13	14	1	1	2	10	8	6	0	2	9	3	7	2
2	15	18	1	0	2	11	9	4	0	2	12	7	6	2
3	5	11	4	1	2	12	4	1	0	2	13	6	5	0
3	6	24	1	0	2	13	11	3	0	2	14	7	4	0
3	7	12	5	1	2	14	8	2	1	2	17	8	2	1
3	10	10	4	1	2	17	13	5	0	3	4	3	3	0
3	11	18	1	2	3	4	4	4	4	3	5	7	6	1
3	12	12	2	0	3	5	9	4	1	3	7	9	6	2
3	13	13	2	2	3	7	11	1	0	3	14	1	3	1
3	14	15	2	1	3	8	9	3	2	3	16	8	6	1
3	16	16	1	1	3	9	6	6	2	4	5	4	5	2
3	17	9	2	2	3	10	10	5	1	4	7	8	3	1
4	7	17	1	0	3	12	8	9	0	4	9	9	5	0
4	8	15	3	0	3	15	4	7	1	4	10	4	4	1
4	10	16	2	1	3	16	7	2	0	4	11	5	8	0

T_7^{GBN}					T_8^{GBN}					T_9^{GBN}				
s_d	t_d	n_{10G}	n_{40G}	n_{100G}	s_d	t_d	n_{10G}	n_{40G}	n_{100G}	s_d	t_d	n_{10G}	n_{40G}	n_{100G}
4	13	22	0	0	3	17	9	4	0	4	13	6	2	0
4	14	16	3	1	4	5	15	3	0	4	14	6	5	2
4	16	18	2	1	4	6	6	5	1	4	15	5	3	0
5	7	14	1	2	4	12	6	4	1	4	17	5	3	5
5	8	11	1	0	4	13	14	0	1	5	6	5	1	2
5	10	20	1	0	4	16	10	3	1	5	8	5	6	0
5	12	18	0	0	4	17	4	7	0	5	12	3	5	1
5	14	16	0	0	5	7	8	7	2	5	14	2	3	2
5	15	16	2	1	5	9	4	7	2	5	16	7	4	3
6	8	21	2	1	5	12	12	4	1	6	7	3	1	1
6	9	10	0	0	5	13	8	5	1	6	10	5	3	2
6	10	16	1	1	5	17	3	3	0	6	11	4	9	1
6	13	18	5	2	6	9	12	4	2	6	12	5	2	1
6	15	18	2	0	6	10	13	3	0	6	13	3	3	2
7	9	14	1	0	6	11	7	2	1	6	17	4	1	1
7	12	20	1	1	6	13	6	1	0	7	8	10	2	1
7	16	17	0	0	6	14	11	4	0	7	11	6	2	0
7	17	15	3	2	6	15	4	4	4	7	15	7	3	2
8	11	19	2	0	6	16	11	2	1	8	10	9	3	2
8	13	14	1	0	6	17	13	6	0	8	12	6	3	1
8	14	17	0	2	7	8	7	0	1	8	14	1	7	0
8	16	16	3	1	7	9	9	3	0	8	17	4	4	1
8	17	15	1	0	7	10	8	4	1	9	11	6	4	0
9	14	17	5	0	7	13	5	10	0	9	13	6	3	2
9	16	12	1	0	7	14	7	2	0	9	14	4	6	3
10	11	18	0	1	7	16	10	2	1	9	17	5	4	2
10	14	19	1	0	7	17	12	3	0	10	11	4	2	0
10	15	16	1	1	8	11	5	3	0	10	12	4	7	1
10	16	12	1	2	8	15	9	1	4	11	13	6	3	1
10	17	12	1	1	9	17	9	3	1	11	15	1	1	4
11	12	10	4	0	10	12	7	7	1	11	16	3	2	1
11	13	24	3	0	10	13	5	3	0	11	17	7	2	2
11	15	22	2	0	10	15	9	3	0	12	14	2	5	1
11	16	24	2	1	11	12	9	2	0	12	15	5	4	1
13	14	13	0	0	11	13	11	4	0	12	17	7	5	3
13	15	12	4	0	11	14	7	7	2	13	15	8	3	1
13	17	18	4	2	12	13	4	5	0	13	17	6	3	1
14	15	13	1	1	13	14	7	3	5	14	15	4	3	1
14	17	18	1	1	13	15	8	7	0	14	17	2	2	0
16	17	17	2	1	14	17	9	2	1	16	17	5	4	0

1	3	12	2	1	1	3	12	2	2	1	2	4	8	1
1	4	12	2	1	1	5	2	5	1	1	4	6	6	1
1	5	16	0	1	1	6	9	6	0	1	5	8	6	1
1	8	20	3	0	1	7	8	5	0	1	6	3	2	0
1	9	10	1	3	1	10	4	4	1	1	7	3	0	0
1	11	16	1	1	1	11	4	3	0	1	10	10	6	1
1	12	14	1	2	1	12	5	1	0	1	11	9	5	1
1	14	10	2	1	1	15	9	2	1	1	12	9	4	1
2	3	15	3	0	1	17	9	1	2	1	13	5	4	0
2	4	14	0	0	2	3	3	3	0	1	15	9	2	1
2	6	8	2	1	2	4	5	4	0	1	16	6	4	3
2	7	13	3	0	2	5	6	4	1	1	17	4	8	1
2	8	8	1	0	2	8	9	4	0	2	4	3	6	1
2	12	13	1	1	2	9	6	3	1	2	5	6	4	0
2	13	16	1	0	2	10	7	0	0	2	7	2	4	3
2	14	11	2	2	2	11	11	1	1	2	10	4	4	1
2	15	11	2	1	2	14	5	7	2	2	11	2	3	1
2	16	11	1	1	2	15	8	2	1	2	13	3	5	1
2	17	18	0	1	2	17	7	3	2	2	14	4	2	3
3	5	8	4	0	3	4	8	3	0	2	16	7	5	1
3	6	20	2	0	3	6	7	2	3	3	4	3	2	1
3	7	14	1	0	3	7	8	3	0	3	5	7	3	0
3	8	16	3	0	3	8	10	3	1	3	6	4	3	1
3	9	10	1	2	3	9	8	1	0	3	7	6	1	1
3	10	16	1	0	3	13	11	1	0	3	12	9	8	2
3	11	19	2	0	3	14	6	3	1	3	13	3	4	0
3	14	14	1	2	3	16	7	6	0	3	15	2	3	2
3	17	13	0	1	3	17	10	4	0	3	16	8	6	0
4	7	10	2	2	4	6	6	4	1	4	5	3	6	1
4	10	22	1	1	4	8	11	1	0	4	8	3	2	0
4	11	20	1	0	4	9	5	4	0	4	9	10	1	2
4	12	16	3	2	4	10	3	3	2	4	12	6	1	1
4	13	14	1	0	4	11	6	4	0	4	16	4	2	0
4	15	18	2	1	4	14	4	2	1	4	17	3	1	1
4	16	19	1	0	4	15	5	3	0	5	6	5	2	1
5	6	13	1	0	4	16	10	3	0	5	7	5	5	0
5	10	9	0	0	5	9	4	2	0	5	8	3	1	2
5	12	10	0	0	5	10	6	4	0	5	9	4	4	0
5	13	13	3	0	5	11	8	3	0	5	13	2	5	3
5	14	12	1	1	5	12	12	1	0	5	14	4	6	1
5	15	13	3	0	5	14	8	6	0	5	15	1	5	0
5	16	12	2	1	5	15	7	4	0	5	16	0	3	3
5	17	10	2	0	5	16	9	3	0	5	17	2	2	2
6	8	4	1	1	5	17	8	3	0	6	7	4	3	1
6	9	12	3	1	6	7	5	4	1	6	8	3	4	2
6	10	14	0	0	6	8	5	6	0	6	10	7	0	2

6	13	18	0	0	6	9	7	3	2	6	11	5	3	1
6	14	14	1	0	6	10	4	2	3	6	12	3	6	2
6	15	18	1	0	6	12	4	4	0	6	13	4	3	2
6	16	13	2	0	6	13	8	2	2	6	14	8	1	1
7	8	22	1	1	6	15	7	4	1	6	16	4	1	0
7	10	11	2	1	6	16	7	3	1	6	17	5	4	0
7	11	17	4	0	7	8	3	6	1	7	8	3	2	0
7	14	19	3	1	7	9	8	4	0	7	9	5	5	2
7	15	17	0	2	7	10	10	2	0	7	12	6	4	2
7	16	18	1	0	7	11	5	3	0	7	14	8	4	1
7	17	20	3	0	7	12	5	6	0	7	15	4	5	2
8	9	11	2	0	7	13	4	10	1	7	16	5	2	0
8	11	17	0	1	7	14	7	3	2	7	17	4	2	2
8	12	21	0	1	7	15	12	6	0	8	9	3	4	1
8	13	11	1	0	8	9	10	6	3	8	10	3	0	1
8	15	10	3	0	8	10	6	9	1	8	12	3	1	0
8	16	13	2	0	8	11	6	4	1	8	13	3	4	0
8	17	14	1	0	8	12	11	3	1	8	14	2	3	1
9	10	11	1	1	8	14	6	1	2	8	15	5	4	3
9	11	13	0	1	8	15	8	2	1	8	17	3	0	2
9	12	9	0	2	8	16	3	7	1	9	13	3	7	2
9	13	11	2	1	8	17	7	5	1	9	14	2	0	1
9	14	14	2	1	9	10	6	4	0	9	15	5	5	1
9	16	14	1	0	9	11	2	2	1	9	16	5	5	1
9	17	18	3	0	9	13	6	2	1	10	11	6	5	1
10	12	14	0	1	9	14	8	5	0	10	12	7	1	1
10	13	11	2	0	9	17	11	7	1	10	13	3	2	1
10	14	9	1	0	10	11	4	0	0	10	14	1	1	0
10	15	11	1	1	10	12	10	6	3	11	13	4	4	0
10	17	18	2	0	10	15	7	3	1	11	14	6	1	3
11	12	9	0	0	10	17	6	2	0	11	15	4	4	0
11	13	15	1	0	11	15	7	2	0	11	16	6	3	1
11	15	10	2	0	11	16	8	4	0	12	13	5	1	0
11	16	8	1	0	11	17	6	2	0	12	14	6	6	5
11	17	21	2	1	12	13	5	1	1	12	15	3	4	0
12	14	16	2	1	12	15	9	2	0	12	16	5	2	0
12	16	16	1	0	12	16	5	0	2	12	17	5	4	1
13	14	16	1	0	12	17	6	5	3	13	14	5	2	1
13	15	12	4	1	13	14	7	3	0	13	15	3	6	1
13	16	21	2	1	14	15	5	4	0	13	17	3	5	1
13	17	9	0	1	14	17	6	4	0	14	15	8	2	1
14	15	11	4	3	15	16	4	3	0	14	16	4	2	0
14	17	9	3	0	15	17	7	2	1	14	17	6	2	1
15	17	13	3	1	16	17	12	0	0	15	16	5	1	0

T_1^{GEANT2}					T_2^{GEANT2}					T_3^{GEANT2}				
s_d	t_d	n_{10G}	n_{40G}	n_{100G}	s_d	t_d	n_{10G}	n_{40G}	n_{100G}	s_d	t_d	n_{10G}	n_{40G}	n_{100G}
1	2	7	3	0	1	3	4	1	0	1	3	2	2	1
1	7	4	2	0	1	4	7	3	1	1	5	3	2	1
1	15	7	0	0	1	10	6	1	0	1	7	3	2	0
1	16	6	2	0	1	17	1	3	0	1	8	6	2	0
1	17	5	0	1	1	19	4	0	0	1	9	2	0	0
1	19	8	1	1	1	21	1	3	0	1	10	4	1	0
1	22	8	2	0	1	25	2	0	0	1	13	1	0	1
1	24	2	0	1	1	29	3	1	1	1	14	2	2	1
1	25	4	0	0	1	30	3	3	0	1	15	2	2	1
1	31	6	0	1	1	31	4	0	0	1	18	0	1	0
1	32	10	0	0	2	12	4	0	0	1	19	6	3	1
2	3	5	1	1	2	14	4	2	1	1	22	1	1	0
2	7	11	0	0	2	15	3	1	1	1	25	3	1	0
2	12	6	0	0	2	16	4	0	0	1	27	1	0	0
2	17	9	2	0	2	18	3	1	0	1	31	1	1	0
2	18	9	1	0	2	21	3	2	0	2	4	0	0	1
2	22	7	0	0	2	24	1	2	0	2	5	3	1	1
2	23	3	0	1	2	25	3	1	0	2	6	2	1	1
2	28	2	1	0	2	26	3	1	0	2	7	2	4	1
2	32	9	1	0	2	29	2	1	0	2	13	3	1	0
3	5	3	1	0	2	30	1	4	0	2	19	2	0	0
3	8	3	0	1	3	4	4	2	1	2	22	3	2	0
3	11	6	0	0	3	6	2	2	0	2	25	6	2	0
3	13	8	0	1	3	10	5	1	0	2	30	3	3	0
3	14	10	0	0	3	11	1	2	0	2	31	0	1	0
3	15	6	1	0	3	17	4	0	0	3	5	1	0	2
3	17	9	0	0	3	19	2	3	0	3	6	3	4	0
3	20	4	1	1	3	21	0	0	0	3	7	5	3	0
3	21	2	1	0	3	22	1	3	1	3	12	1	1	0
3	22	12	1	0	3	23	2	2	1	3	13	3	2	1
3	23	5	0	1	3	26	5	1	0	3	15	0	1	0
3	27	6	2	0	3	27	4	1	1	3	17	2	0	0
3	30	6	0	0	4	9	4	2	0	3	18	2	1	0
4	5	5	0	1	4	11	5	1	1	3	20	4	3	1
4	7	12	0	0	4	13	7	2	0	3	22	2	0	0
4	8	5	0	0	4	17	4	0	0	3	26	1	1	0
4	16	4	2	0	4	25	5	2	0	3	31	3	2	2
4	24	7	1	0	4	26	3	2	0	3	32	2	4	0
4	28	7	4	0	4	28	3	0	0	4	5	4	2	0
4	29	4	1	0	4	31	1	1	1	4	10	2	1	0
4	32	5	0	0	5	12	3	4	0	4	15	4	1	0
5	12	6	0	0	5	13	5	1	0	4	16	3	3	1
5	19	14	2	0	5	16	3	1	0	4	21	2	3	0
5	21	5	0	0	5	18	5	2	1	4	24	4	2	1

5	27	1	0	0	5	20	3	3	0	4	27	2	1	0
6	7	7	1	0	5	21	3	1	0	4	28	1	3	1
6	9	2	0	0	5	27	5	2	0	4	31	1	1	1
6	13	6	4	0	5	28	6	3	0	4	32	2	3	0
6	14	7	3	0	5	31	1	2	0	5	8	2	2	0
6	16	5	0	0	6	8	4	0	0	5	10	3	2	2
6	17	8	0	0	6	10	3	2	0	5	11	3	0	0
6	23	8	1	0	6	14	7	1	0	5	23	4	4	0
6	24	7	2	0	6	15	4	1	0	5	24	3	1	0
6	25	11	0	1	6	17	4	2	1	5	25	2	1	0
6	28	6	0	0	6	18	2	0	0	5	26	1	1	0
6	29	8	1	1	6	25	0	2	0	5	27	2	1	0
6	31	10	0	0	6	26	4	2	0	5	28	1	1	1
7	8	7	1	0	6	28	2	3	0	5	29	2	3	0
7	9	5	0	0	7	8	5	1	0	5	30	6	1	0
7	10	2	1	0	7	9	2	1	0	5	32	1	5	0
7	11	10	0	0	7	12	3	0	0	6	10	3	0	1
7	14	3	1	0	7	13	2	2	0	6	11	3	1	1
7	17	5	0	0	7	15	1	0	1	6	13	4	0	1
7	22	9	2	1	7	20	2	2	0	6	14	1	1	0
7	24	8	1	1	7	22	3	2	0	6	15	1	1	0
7	28	6	0	0	7	26	2	3	0	6	21	3	2	0
7	29	7	0	0	7	27	3	2	0	6	22	3	0	0
7	30	2	1	0	7	28	4	2	0	6	23	3	4	0
8	12	6	0	0	7	29	2	3	1	6	25	5	1	0
8	14	3	0	0	8	12	3	0	1	6	29	3	1	0
8	18	6	2	0	8	14	3	1	0	6	31	3	0	0
8	20	8	0	0	8	17	2	2	0	6	32	2	0	1
8	32	5	0	0	8	24	4	1	0	7	9	2	0	0
9	10	4	2	0	8	25	4	0	0	7	14	3	3	0
9	11	6	0	0	8	26	7	3	0	7	24	3	1	0
9	13	2	1	1	8	28	2	1	0	8	12	4	1	1
9	14	7	2	0	8	29	2	1	0	8	15	5	0	0
9	15	10	0	0	8	31	0	1	0	8	16	5	0	0
9	17	5	0	0	8	32	8	0	2	8	21	3	3	1
9	19	3	0	0	9	10	1	1	1	8	28	3	2	0
9	20	1	2	0	9	15	3	0	0	8	31	4	5	1
9	21	6	0	1	9	20	4	3	0	9	12	1	2	0
9	29	8	0	0	9	28	2	1	0	9	14	5	1	0
10	14	4	1	2	9	29	2	2	0	9	15	2	2	0
10	17	7	0	0	10	12	3	2	0	9	16	1	1	0
10	19	8	1	0	10	15	5	1	1	9	17	2	0	1
10	25	4	1	0	10	17	2	2	0	9	19	1	1	0
10	28	6	2	1	10	18	2	2	0	9	22	2	1	0
10	29	4	1	1	10	27	4	3	0	9	23	5	1	3
11	13	9	1	0	10	32	4	1	0	9	26	2	2	1
11	24	3	1	0	11	13	6	2	0	10	19	0	2	1

11	25	6	2	0	11	15	1	3	0	10	22	1	2	0
11	29	4	2	0	11	18	2	1	1	10	24	3	1	0
11	30	8	1	0	11	21	1	1	0	10	26	4	1	1
11	31	8	1	0	11	24	2	1	3	11	13	1	3	3
12	15	10	1	0	11	25	5	2	2	11	15	0	1	1
12	16	5	0	0	11	29	3	5	0	11	19	2	2	1
12	18	4	0	0	12	13	2	2	0	11	20	3	1	0
12	22	5	1	1	12	16	3	2	0	11	21	1	0	1
12	24	6	0	1	12	20	7	0	0	11	23	4	4	0
12	26	8	0	0	12	21	3	1	0	11	24	3	1	0
13	18	6	0	0	12	28	2	2	0	11	25	1	4	0
13	27	5	0	0	12	30	3	0	0	12	14	0	3	0
14	18	2	1	1	12	31	2	1	0	12	17	3	1	0
14	19	4	1	0	13	14	0	0	0	12	20	1	0	0
14	20	6	1	1	13	18	3	0	0	12	23	0	2	1
14	21	8	1	0	13	23	1	3	0	13	15	1	0	1
14	22	11	1	1	13	27	6	2	0	13	16	4	0	1
14	29	5	0	0	14	17	4	0	0	13	21	0	1	0
14	30	5	1	0	14	21	1	1	0	13	31	4	0	1
15	22	5	1	0	14	23	0	0	1	13	32	6	0	0
15	24	7	2	0	14	26	2	3	2	14	16	2	1	0
15	28	5	2	0	14	29	2	4	2	14	17	2	1	0
15	29	6	0	0	14	30	0	3	0	14	27	2	0	0
15	30	6	0	0	14	32	3	0	0	14	28	0	1	0
16	17	6	1	0	15	17	3	1	0	14	31	1	7	0
16	18	1	0	0	15	27	2	2	0	15	20	3	0	0
16	20	5	1	1	15	28	2	2	0	15	22	1	1	1
16	27	6	0	0	15	32	3	2	0	15	23	6	0	0
16	29	7	0	0	16	19	3	1	0	15	25	2	0	1
17	20	6	1	0	16	20	3	2	0	15	29	2	0	0
17	28	6	0	0	16	26	3	1	1	16	17	0	6	0
17	29	10	0	0	16	27	3	2	0	16	19	3	1	1
18	22	8	0	0	16	28	0	2	0	16	20	2	0	0
18	24	7	0	0	16	30	4	0	0	16	21	2	0	0
18	26	9	0	0	17	18	2	2	2	16	22	4	1	0
18	27	5	2	1	17	27	1	2	0	16	25	2	1	0
18	31	4	0	1	17	30	2	3	0	16	26	2	2	1
19	20	1	2	0	17	31	0	1	0	16	27	4	1	2
19	24	3	1	0	18	20	3	1	0	17	24	2	1	1
19	27	7	0	0	18	26	3	1	1	17	26	1	1	0
19	29	6	3	1	18	30	1	4	0	17	30	2	0	0
20	22	9	0	0	19	27	4	1	0	17	32	2	2	0
20	23	6	0	0	19	30	3	4	0	18	19	3	4	0
20	26	8	1	0	19	31	5	1	0	18	22	3	1	1
20	27	5	0	0	20	23	2	2	0	18	26	3	2	1
20	30	6	1	0	20	26	4	0	0	18	31	3	0	2
20	32	6	1	0	20	28	2	1	0	18	32	5	1	0

21	26	5	1	1	20	29	1	1	0	19	21	3	2	0
21	27	6	1	0	20	32	4	1	0	19	22	6	1	0
21	29	3	3	0	21	23	1	1	0	19	23	4	1	0
21	30	4	1	0	21	24	7	3	0	19	24	1	2	0
22	23	4	0	0	21	25	2	2	0	20	22	4	3	0
22	24	6	0	0	21	31	7	2	0	20	31	1	3	1
22	25	8	0	2	22	23	2	1	0	20	32	1	4	0
22	26	10	1	0	22	25	3	0	0	21	22	1	2	0
22	27	6	0	0	22	26	4	1	0	21	23	2	2	0
22	31	7	0	2	22	28	2	0	1	21	30	1	0	0
23	25	7	1	0	22	30	4	3	0	21	31	0	1	1
23	29	9	0	0	23	25	5	1	0	22	23	1	2	0
23	31	4	0	1	23	26	5	1	0	22	27	1	2	0
23	32	5	2	0	24	25	2	1	0	22	29	1	2	2
24	25	9	1	0	24	26	4	1	0	23	24	3	1	0
24	26	6	0	0	24	28	3	1	0	23	27	2	2	0
24	30	9	0	1	24	31	6	0	1	23	30	3	2	1
24	32	3	0	0	25	28	2	2	1	23	32	4	2	0
25	26	7	0	0	25	29	0	2	0	24	29	3	3	0
25	30	5	0	0	25	30	2	1	0	24	32	2	0	0
26	28	5	1	0	26	27	2	2	0	25	32	3	1	0
26	31	4	1	0	26	29	4	4	0	26	28	1	2	0
28	31	2	1	0	26	30	3	2	0	28	29	2	2	2
28	32	8	1	0	27	31	1	0	0	28	30	3	1	1
29	31	7	1	0	29	32	6	3	0	29	30	7	1	1
29	32	9	1	0	30	31	3	2	0	30	31	1	3	0
30	32	3	1	0	30	32	5	0	0	31	32	1	3	0

T_4^{GEANT2}					T_5^{GEANT2}					T_6^{GEANT2}				
s_d	t_d	n_{10G}	n_{40G}	n_{100G}	s_d	t_d	n_{10G}	n_{40G}	n_{100G}	s_d	t_d	n_{10G}	n_{40G}	n_{100G}
1	3	6	1	0	1	2	2	3	0	1	2	4	2	0
1	9	5	1	0	1	3	3	2	0	1	3	2	2	1
1	12	5	1	0	1	4	4	3	2	1	4	3	1	1
1	13	4	0	0	1	5	0	3	1	1	7	3	1	1
1	14	8	0	0	1	6	0	0	0	1	8	1	0	1
1	15	6	0	0	1	7	4	1	0	1	12	0	0	0
1	18	6	1	0	1	8	3	2	0	1	13	3	3	0
1	19	3	0	0	1	9	6	0	0	1	14	4	1	1
1	22	3	1	0	1	10	4	0	0	1	17	1	1	0
1	23	2	1	1	1	11	3	2	0	1	19	4	1	0
1	24	3	0	0	1	12	3	0	0	1	21	1	0	1
1	25	7	0	0	1	13	2	2	0	1	22	2	1	0
1	27	8	1	0	1	14	4	2	1	1	24	3	2	0
1	29	4	1	0	1	17	2	1	0	1	25	5	0	1

1	31	5	1	0	1	18	2	1	0	1	26	3	2	0
2	4	4	1	0	1	19	3	1	1	1	31	0	1	1
2	6	2	0	0	1	20	1	0	0	2	3	1	1	0
2	8	9	0	0	1	21	4	1	1	2	6	0	2	0
2	9	7	1	1	1	24	1	0	0	2	7	3	1	0
2	10	3	1	0	1	25	3	0	0	2	8	2	2	0
2	11	3	1	0	1	28	4	1	0	2	12	0	0	0
2	12	6	2	0	1	30	1	2	0	2	15	0	2	1
2	14	5	2	0	2	3	2	1	0	2	16	2	3	0
2	16	4	0	1	2	4	1	0	0	2	17	4	0	0
2	18	3	1	0	2	7	2	0	0	2	19	2	1	0
2	19	5	0	1	2	9	2	1	1	2	21	2	1	0
2	24	4	1	1	2	12	5	1	1	2	23	2	2	0
2	27	8	1	1	2	13	3	0	0	2	24	3	1	0
2	28	8	0	0	2	14	2	2	0	2	28	0	0	0
2	29	2	1	0	2	16	2	1	0	2	29	1	0	0
2	30	5	1	0	2	17	3	0	1	2	32	0	0	0
3	5	2	0	1	2	19	0	1	0	3	4	3	4	0
3	8	7	1	2	2	21	2	2	0	3	5	4	0	2
3	9	5	0	0	2	22	3	5	0	3	7	2	1	0
3	14	5	0	0	2	23	5	1	0	3	8	2	1	0
3	16	6	1	0	2	24	4	2	0	3	9	1	2	0
3	19	5	0	0	2	25	3	2	0	3	11	3	0	0
3	21	5	1	0	2	26	1	1	0	3	12	5	2	0
3	24	3	0	0	2	27	6	0	0	3	13	1	0	0
3	26	7	1	0	2	28	2	3	0	3	18	3	3	1
3	27	2	0	1	3	4	5	0	0	3	19	1	2	1
3	28	5	1	0	3	6	4	0	0	3	21	0	1	0
3	29	6	0	2	3	8	3	1	0	3	24	1	3	0
3	31	5	0	0	3	9	3	3	0	3	25	2	1	1
4	5	6	0	0	3	12	1	0	0	3	27	1	0	0
4	6	5	0	0	3	14	2	1	1	3	28	0	0	0
4	7	2	0	0	3	15	1	1	1	3	30	0	0	0
4	8	8	1	0	3	16	2	1	0	3	31	2	2	0
4	10	4	0	1	3	20	1	2	0	3	32	1	0	1
4	12	4	0	0	3	23	4	0	0	4	5	2	0	0
4	13	3	3	0	3	26	2	1	0	4	8	1	1	0
4	14	5	1	0	3	28	2	0	0	4	9	5	3	1
4	22	7	0	1	3	29	2	3	0	4	11	1	0	1
4	23	5	0	0	3	31	3	1	0	4	13	3	0	0
4	24	4	1	0	3	32	3	0	0	4	14	1	1	0
4	25	5	0	0	4	6	3	1	0	4	15	2	1	1
4	26	2	0	0	4	8	3	3	0	4	17	2	2	1
4	31	6	1	0	4	9	3	1	0	4	21	1	2	1
4	32	6	0	0	4	13	1	1	0	4	23	1	0	0
5	7	2	0	0	4	15	2	1	0	4	25	1	3	0
5	8	5	0	0	4	18	0	0	0	4	26	2	0	1

5	9	7	0	0	4	20	3	3	0	4	28	8	2	1
5	11	7	0	0	4	23	3	3	1	5	6	1	1	0
5	15	5	1	0	4	24	5	3	0	5	8	1	1	1
5	18	3	0	0	4	26	2	0	0	5	12	1	3	1
5	20	3	1	0	4	28	0	1	0	5	13	1	5	0
5	22	4	1	1	4	32	1	0	0	5	14	2	1	0
5	23	2	2	0	5	8	2	0	0	5	17	2	3	0
5	24	5	1	0	5	9	2	1	0	5	18	2	0	1
5	25	5	1	0	5	13	2	2	0	5	20	3	1	1
5	27	1	0	0	5	14	1	0	0	5	21	1	0	1
5	28	4	0	0	5	19	4	1	0	5	23	3	1	0
5	29	3	2	0	5	20	3	0	0	5	24	1	0	0
5	30	3	0	1	5	22	1	1	0	5	27	0	1	0
5	32	3	0	0	5	24	5	3	1	5	30	0	0	1
6	11	4	2	0	5	25	1	1	0	5	32	2	0	1
6	12	8	0	0	5	30	2	1	0	6	8	2	2	1
6	13	7	0	0	5	31	0	1	0	6	10	3	2	0
6	14	2	3	0	5	32	1	3	0	6	11	0	2	0
6	15	6	1	0	6	8	2	0	0	6	13	1	1	1
6	17	6	1	0	6	9	6	2	0	6	14	0	0	0
6	18	6	0	0	6	11	1	1	0	6	15	0	1	0
6	21	9	0	0	6	12	3	0	0	6	18	2	0	0
6	22	5	1	0	6	17	2	0	0	6	20	0	1	0
6	23	8	0	0	6	19	3	3	0	6	21	2	1	1
6	25	4	0	0	6	22	1	0	0	6	23	1	2	0
6	28	8	2	0	6	26	2	0	0	6	25	1	1	0
6	29	3	0	0	6	28	3	0	0	6	27	2	1	0
6	30	1	0	0	6	29	1	0	0	6	29	1	0	0
6	32	4	0	0	7	12	1	1	0	6	31	4	0	0
7	9	5	0	0	7	17	1	2	0	7	8	2	2	0
7	10	7	1	0	7	18	0	0	0	7	10	1	0	1
7	11	3	2	0	7	21	1	1	0	7	11	2	1	0
7	12	10	0	1	7	22	1	2	0	7	17	2	0	0
7	14	2	1	0	7	24	1	1	1	7	18	1	1	0
7	21	3	0	1	7	25	6	3	0	7	21	2	2	0
7	22	2	1	0	7	26	3	1	0	7	22	3	0	1
7	23	5	0	0	7	27	1	2	0	7	27	2	1	3
7	26	6	0	0	7	28	2	4	0	7	28	2	1	0
7	27	4	0	0	7	29	4	4	0	7	30	4	2	0
7	30	2	1	0	7	31	3	1	0	7	31	3	1	0
8	10	3	0	0	8	10	3	1	0	8	12	1	1	0
8	12	6	0	0	8	13	2	3	0	8	13	1	1	0
8	13	1	0	0	8	15	1	2	0	8	15	3	0	0
8	15	4	0	0	8	16	4	1	0	8	17	5	0	0
8	16	5	1	0	8	17	4	1	1	8	18	1	1	0
8	17	5	0	0	8	22	0	2	0	8	22	0	1	0
8	18	1	1	0	8	30	2	1	0	8	23	1	2	1

8	23	6	1	0	8	32	2	1	0	8	24	1	2	0
8	24	3	0	0	9	11	2	0	0	8	25	0	2	0
8	25	5	1	0	9	12	3	1	0	8	26	3	2	0
8	29	2	0	0	9	14	2	3	0	8	31	2	1	0
8	31	5	0	0	9	19	0	0	0	8	32	3	1	1
8	32	6	0	0	9	20	4	0	1	9	10	0	2	0
9	12	3	0	1	9	25	3	0	0	9	11	2	2	0
9	16	1	0	0	9	26	3	1	0	9	12	1	0	2
9	17	9	0	0	9	27	3	1	0	9	14	0	0	0
9	19	4	0	0	9	30	2	1	0	9	15	0	1	0
9	20	5	0	0	10	11	1	2	0	9	17	2	1	2
9	23	6	0	0	10	12	0	1	2	9	20	2	2	1
9	24	6	2	1	10	14	2	1	0	9	22	3	1	1
9	25	7	0	0	10	15	4	2	0	9	23	3	0	0
9	31	7	1	0	10	17	1	0	0	9	24	2	0	0
10	11	3	0	1	10	20	2	0	0	9	25	3	3	0
10	12	7	0	0	10	21	2	3	0	9	27	2	0	0
10	14	8	1	0	10	26	3	1	0	9	28	4	2	0
10	15	5	0	1	10	29	5	0	0	9	31	1	2	0
10	17	5	2	0	10	32	1	3	2	9	32	2	0	0
10	18	3	1	0	11	12	3	1	0	10	11	1	1	0
10	19	4	0	0	11	13	4	0	0	10	12	1	4	0
10	20	7	0	0	11	16	1	0	0	10	13	0	1	0
10	21	8	0	0	11	17	0	2	0	10	15	4	1	0
10	23	1	1	0	11	18	4	0	0	10	16	4	1	0
10	26	2	1	0	11	20	4	0	0	10	18	3	0	1
11	13	3	2	0	11	21	4	1	0	10	19	1	2	0
11	14	5	0	0	11	22	5	2	0	10	20	1	1	0
11	18	4	0	1	11	23	1	0	1	10	24	2	0	0
11	20	1	1	1	11	25	2	2	1	10	26	2	0	1
11	21	1	0	0	11	27	2	3	0	10	27	2	2	0
11	25	10	1	0	11	28	5	1	0	10	29	5	1	0
11	26	7	0	0	11	29	2	1	0	10	30	2	0	0
11	29	4	1	0	11	31	2	1	0	10	31	3	0	0
11	30	5	0	0	12	14	8	2	0	11	15	1	1	0
11	31	4	0	0	12	16	0	0	0	11	18	2	0	1
12	14	5	0	1	12	17	2	3	0	11	21	1	3	1
12	17	7	0	1	12	18	3	1	0	11	22	4	0	1
12	19	3	0	0	12	19	2	0	0	11	24	1	2	0
12	20	7	1	0	12	20	3	1	0	11	26	2	1	0
12	27	5	1	0	12	21	2	1	0	11	27	4	0	0
12	28	6	2	0	12	22	1	1	1	11	28	0	1	1
12	29	2	0	0	12	24	1	0	0	11	31	1	2	1
12	32	5	0	1	12	25	3	1	0	11	32	2	1	0
13	15	2	0	0	12	29	1	1	0	12	13	3	1	0
13	22	7	1	0	12	31	2	1	0	12	14	2	2	0
13	25	6	0	0	13	14	1	1	1	12	19	3	2	0

13	27	6	0	0	13	15	5	0	0	12	20	1	1	0
13	28	5	0	1	13	16	2	2	0	12	21	1	1	0
13	31	1	1	0	13	17	5	1	0	12	22	1	3	1
13	32	6	0	0	13	19	1	1	0	12	24	0	0	0
14	15	4	0	0	13	20	2	0	1	12	26	0	4	0
14	16	6	1	0	13	22	1	0	0	13	14	1	1	0
14	20	4	0	0	13	25	0	0	0	13	18	3	2	0
14	21	5	0	0	13	27	3	1	0	13	19	4	3	0
14	27	8	0	1	14	15	2	1	0	13	20	0	0	0
14	29	5	2	0	14	16	3	2	1	13	21	2	0	0
14	32	6	2	0	14	19	3	1	0	13	23	2	1	0
15	16	5	0	0	14	22	1	1	0	13	32	2	3	0
15	18	2	1	0	14	23	1	1	0	14	15	0	0	0
15	20	2	1	0	14	24	3	1	0	14	17	3	1	0
15	21	7	0	0	14	27	3	0	1	14	18	3	2	0
15	22	6	2	0	14	29	4	1	0	14	19	2	1	0
15	23	9	0	1	14	32	3	1	0	14	20	3	4	0
15	24	4	1	0	15	16	3	2	0	14	25	2	0	1
15	25	5	3	0	15	17	1	5	0	14	28	3	0	0
15	27	2	0	1	15	20	2	1	0	14	30	3	1	0
15	31	4	1	0	15	22	2	4	1	15	18	0	1	0
16	17	9	0	0	15	25	2	1	0	15	21	5	2	0
16	18	1	1	0	15	29	3	0	0	15	22	1	0	0
16	20	4	1	0	15	30	2	1	0	15	23	1	0	0
16	21	3	0	0	15	32	1	0	0	15	26	1	2	1
16	23	5	0	0	16	19	3	1	0	15	27	0	2	0
16	26	5	2	0	16	20	1	0	0	15	29	0	0	0
16	28	1	1	0	16	21	2	1	0	15	32	1	1	0
16	30	2	0	0	16	23	0	3	0	16	17	2	1	2
16	31	6	0	0	16	27	1	1	0	16	18	1	0	0
16	32	2	1	1	16	29	3	0	1	16	20	3	1	0
17	18	6	1	0	16	30	4	2	0	16	26	1	1	0
17	21	4	1	0	16	31	2	3	0	16	27	3	1	0
17	27	8	1	0	16	32	5	1	0	16	31	4	3	0
17	28	6	0	0	17	19	1	0	0	17	18	2	1	0
17	29	4	3	0	17	20	2	1	0	17	20	1	2	0
17	30	2	0	0	17	21	5	1	0	17	21	2	2	1
18	20	3	1	0	17	23	5	0	1	17	22	1	0	1
18	21	5	1	0	17	24	4	0	0	17	23	4	0	0
18	23	5	0	1	17	25	1	0	0	17	24	2	4	0
18	24	1	0	0	17	32	3	1	0	17	25	3	2	1
18	25	7	2	1	18	20	6	1	1	17	29	0	3	0
18	26	2	2	1	18	22	1	0	0	17	30	3	0	0
19	21	3	0	0	18	24	0	1	0	18	21	1	1	1
19	23	5	0	0	18	26	3	1	1	18	25	2	2	0
19	25	6	0	0	18	27	2	2	1	18	27	3	1	0
19	26	3	1	0	18	30	4	2	1	18	29	1	1	0

19	28	6	1	0	18	32	2	0	0	18	30	3	0	0
19	29	7	2	0	19	20	0	1	0	19	20	0	0	1
19	30	6	0	0	19	22	0	2	1	19	22	2	1	0
19	31	2	2	1	19	24	4	1	0	19	23	1	2	1
19	32	9	0	1	19	25	2	2	0	19	24	2	3	0
20	21	4	0	0	19	26	1	0	0	19	25	1	1	0
20	23	2	1	0	19	27	3	3	0	19	27	1	1	0
20	24	4	0	0	19	29	2	0	0	19	28	2	1	0
20	25	4	2	0	19	31	1	0	1	19	31	1	0	0
20	26	1	0	0	19	32	1	2	1	19	32	2	3	0
20	29	2	1	1	20	22	3	0	0	20	22	1	1	0
20	31	2	0	0	20	23	0	1	0	20	23	1	1	0
20	32	5	0	0	20	24	1	2	1	20	24	2	1	0
21	22	7	1	1	20	25	2	0	0	20	26	2	0	0
21	23	3	1	0	20	26	0	1	0	20	27	1	0	0
21	24	5	0	0	20	29	2	0	0	20	28	2	1	0
21	27	8	1	1	20	30	1	3	0	20	31	1	2	0
21	28	7	0	0	21	25	7	2	0	21	22	4	0	0
22	25	4	0	0	21	27	3	0	0	21	23	1	0	1
22	26	2	0	0	21	29	2	2	0	21	25	2	0	1
22	27	5	1	0	21	30	1	2	1	21	26	3	1	1
22	28	1	0	0	21	32	0	1	0	21	28	2	2	0
22	29	4	1	0	22	23	2	2	0	21	29	3	1	1
22	30	6	1	0	22	26	3	1	0	21	30	2	3	1
22	32	3	0	0	22	27	1	5	0	21	31	4	2	1
23	24	2	0	1	22	31	3	1	0	22	24	2	1	1
23	27	5	1	0	23	24	1	2	0	22	26	2	3	2
23	29	8	0	0	23	25	4	1	1	22	27	1	2	0
23	30	5	0	0	23	26	1	0	0	22	28	0	4	0
23	31	6	1	0	23	29	1	1	0	22	29	3	1	0
23	32	3	0	0	24	25	3	3	0	23	24	0	1	1
24	29	4	1	0	24	26	3	1	0	23	26	1	0	0
24	30	4	0	0	24	27	2	0	0	23	28	3	1	0
24	31	1	1	0	24	30	3	0	0	23	32	6	1	1
25	26	5	0	0	24	31	2	0	0	24	26	3	1	1
25	29	8	1	0	24	32	0	1	0	24	27	2	2	1
25	31	8	0	1	25	26	1	2	0	24	31	3	1	0
26	28	3	0	0	25	30	1	3	2	25	26	0	2	1
26	29	4	0	0	25	32	1	1	1	25	30	1	0	1
26	30	4	0	0	26	27	4	3	0	25	31	1	1	0
27	29	6	0	0	26	29	1	0	0	26	27	1	0	0
27	30	3	1	1	27	28	3	1	0	27	30	3	0	0
27	31	4	0	0	27	30	3	0	0	28	30	2	2	0
28	29	6	1	0	28	30	2	1	0	28	31	3	1	0
28	32	4	0	0	28	31	2	1	0	29	32	0	1	0
29	31	6	0	0	31	32	2	1	0	30	31	2	1	0

T_7^{GEANT2}					T_8^{GEANT2}					T_9^{GEANT2}				
s_d	t_d	n_{10G}	n_{40G}	n_{100G}	s_d	t_d	n_{10G}	n_{40G}	n_{100G}	s_d	t_d	n_{10G}	n_{40G}	n_{100G}
1	2	4	3	1	1	3	2	2	1	1	2	2	0	0
1	3	2	3	0	1	5	1	1	0	1	4	0	0	0
1	6	5	0	0	1	9	2	0	0	1	5	1	1	0
1	7	2	0	1	1	10	4	0	0	1	7	0	1	1
1	8	4	1	0	1	11	2	0	0	1	8	0	1	0
1	9	4	0	0	1	12	0	0	0	1	9	3	0	1
1	12	6	0	0	1	13	0	0	0	1	10	1	1	0
1	13	3	0	0	1	14	3	0	0	1	11	1	0	1
1	14	4	1	0	1	15	1	2	0	1	12	0	3	0
1	15	4	0	0	1	18	1	2	0	1	13	2	0	1
1	17	1	0	0	1	19	1	1	0	1	14	4	0	0
1	18	3	0	0	1	20	2	2	0	1	15	2	0	0
1	19	3	1	0	1	21	1	1	0	1	17	1	2	0
1	22	3	1	3	1	22	4	2	0	1	20	0	0	1
1	24	6	2	0	1	25	1	2	0	1	21	4	1	1
1	25	5	0	0	1	27	1	3	1	1	22	0	2	0
1	27	5	0	0	1	28	3	2	0	1	23	1	1	0
1	28	3	1	0	1	30	1	2	0	1	26	2	2	0
1	29	8	1	1	2	3	2	3	0	1	28	1	1	0
2	7	4	1	1	2	4	3	0	0	1	29	3	2	0
2	8	4	0	1	2	6	1	0	0	1	30	2	2	0
2	9	3	0	0	2	7	2	2	0	1	31	0	1	0
2	11	4	1	0	2	9	1	0	1	1	32	2	0	1
2	16	2	0	0	2	12	3	0	1	2	3	1	1	0
2	19	5	0	0	2	13	1	2	0	2	4	0	2	0
2	20	3	1	0	2	14	2	0	0	2	7	0	1	0
2	21	2	2	0	2	15	3	2	0	2	9	1	1	0
2	23	5	1	0	2	17	6	0	0	2	10	3	0	1
2	24	7	0	0	2	19	2	0	0	2	12	1	3	1
2	25	9	0	0	2	21	2	2	0	2	14	2	2	0
2	27	6	0	0	2	23	1	0	0	2	15	2	3	0
2	29	2	1	0	2	26	3	2	0	2	16	2	0	0
2	32	2	1	1	2	29	2	0	0	2	18	1	2	0
3	4	1	0	0	2	32	1	2	1	2	21	0	3	0
3	6	4	0	0	3	4	2	0	0	2	22	3	4	0
3	7	1	1	0	3	5	3	0	0	2	24	3	1	1
3	8	3	0	0	3	6	5	2	0	2	26	3	3	0
3	10	7	2	0	3	8	0	0	0	2	27	0	1	0
3	12	4	0	0	3	9	5	1	0	2	28	1	2	0
3	13	4	1	0	3	11	2	0	0	2	29	4	1	0
3	14	3	0	0	3	12	2	3	0	2	32	2	0	0
3	16	3	0	0	3	13	3	1	0	3	6	0	3	1

3	17	2	1	0	3	14	3	3	0	3	7	1	2	0
3	18	4	1	0	3	16	1	1	1	3	11	1	1	0
3	19	5	0	0	3	17	3	1	0	3	12	2	2	0
3	20	5	0	0	3	18	2	1	0	3	18	2	0	1
3	22	6	2	0	3	20	2	0	0	3	19	1	2	0
3	24	1	0	0	3	21	2	0	0	3	20	0	0	0
3	25	7	0	1	3	24	1	1	0	3	22	2	0	0
3	28	3	0	0	3	25	0	2	0	3	26	2	0	0
3	30	2	1	0	3	26	2	0	0	3	28	2	3	0
3	31	7	0	0	3	27	3	1	0	3	29	2	0	0
3	32	1	0	0	3	28	2	3	0	3	30	0	0	0
4	5	3	0	0	3	29	2	2	0	3	31	1	3	0
4	6	3	0	0	3	30	0	0	1	3	32	1	1	0
4	7	8	0	0	3	31	0	3	0	4	5	4	1	0
4	8	1	0	0	4	7	3	1	0	4	6	2	1	0
4	9	2	0	0	4	9	1	1	0	4	8	1	2	0
4	10	0	0	0	4	11	3	1	0	4	10	4	3	1
4	11	6	0	0	4	14	2	2	0	4	13	0	2	0
4	13	7	1	0	4	15	1	1	0	4	14	3	3	1
4	14	2	0	0	4	20	1	2	0	4	15	0	1	0
4	15	2	1	0	4	22	2	0	0	4	16	5	0	0
4	17	5	0	0	4	23	3	1	0	4	17	4	0	0
4	18	5	1	0	4	25	0	0	1	4	18	1	2	0
4	19	1	1	0	4	26	4	1	0	4	19	4	0	0
4	21	6	0	0	4	27	1	0	1	4	20	0	0	0
4	22	6	1	0	4	28	0	1	0	4	21	2	2	0
4	23	8	1	0	4	29	3	2	1	4	22	3	1	0
4	25	3	0	1	4	30	3	0	0	4	23	1	1	0
4	30	2	0	0	4	32	5	2	0	4	24	0	2	0
4	31	2	2	0	5	7	4	2	0	4	27	1	1	2
5	6	2	1	0	5	8	6	0	0	4	28	1	0	0
5	9	4	0	0	5	9	3	0	0	4	29	2	2	0
5	10	4	0	0	5	11	4	1	0	4	30	2	0	1
5	11	4	0	0	5	14	3	1	0	4	31	2	2	1
5	13	2	0	0	5	16	1	1	0	4	32	2	0	0
5	14	3	1	0	5	17	1	2	0	5	8	2	1	0
5	16	4	0	0	5	18	0	2	0	5	15	1	1	1
5	17	6	1	0	5	19	1	1	0	5	16	2	1	0
5	18	6	0	0	5	20	0	2	0	5	17	2	1	1
5	19	2	0	0	5	22	4	2	0	5	20	1	0	0
5	20	3	0	0	5	24	8	0	0	5	24	0	1	1
5	21	3	0	1	5	25	0	3	0	5	25	1	0	0
5	22	1	0	0	5	26	3	0	0	5	27	2	1	0
5	23	9	1	0	5	27	0	2	0	5	28	1	2	0
5	24	4	1	0	5	29	5	2	0	5	29	5	0	0
5	25	2	0	0	5	30	3	0	0	5	31	3	2	0
5	26	5	2	0	6	9	1	1	0	5	32	2	0	1

5	27	7	1	0	6	10	1	2	0	6	7	2	0	0
5	28	5	1	0	6	12	0	0	0	6	8	0	1	1
5	31	1	0	0	6	13	1	1	0	6	9	0	0	0
5	32	4	0	0	6	15	2	1	0	6	10	1	0	0
6	7	5	0	1	6	16	2	2	0	6	11	1	0	0
6	8	3	0	1	6	17	2	0	0	6	12	1	1	0
6	9	3	2	0	6	20	2	2	0	6	13	2	2	0
6	11	3	2	0	6	21	3	1	1	6	14	1	1	0
6	12	4	1	0	6	23	3	1	0	6	16	3	3	1
6	13	3	1	0	6	24	3	0	0	6	17	0	4	0
6	14	8	1	0	6	27	3	1	0	6	19	3	0	0
6	15	3	1	0	6	28	1	2	0	6	20	0	0	0
6	16	2	0	0	6	32	5	0	0	6	21	1	1	1
6	17	2	1	1	7	9	2	3	1	6	22	1	0	0
6	19	4	0	0	7	10	6	0	0	6	23	1	1	0
6	21	3	0	0	7	13	2	1	0	6	24	1	5	0
6	22	3	1	0	7	14	1	0	0	6	25	3	0	0
6	24	6	0	0	7	15	3	2	0	6	27	1	0	0
6	27	2	1	1	7	16	2	2	0	6	29	1	1	0
6	29	3	0	0	7	17	1	0	0	6	30	1	1	0
6	30	5	0	0	7	18	1	2	0	6	31	1	3	0
6	31	0	0	1	7	19	1	0	0	7	8	2	2	0
7	8	4	0	0	7	20	3	1	0	7	10	0	2	1
7	11	4	0	0	7	25	0	0	1	7	12	2	0	0
7	12	4	1	0	7	26	0	0	0	7	14	2	0	0
7	14	2	2	0	7	28	4	1	0	7	15	1	1	2
7	15	4	1	0	7	30	1	1	0	7	16	2	1	0
7	16	5	0	0	7	31	1	2	0	7	17	1	0	2
7	17	3	0	0	7	32	1	0	0	7	18	0	1	0
7	18	5	1	0	8	9	3	0	0	7	19	1	2	0
7	19	1	1	0	8	10	5	0	0	7	20	1	0	0
7	20	1	1	0	8	11	2	2	0	7	24	1	1	0
7	21	8	0	0	8	12	4	2	1	7	25	3	2	0
7	22	4	0	0	8	13	4	0	0	7	27	1	1	0
7	26	6	0	0	8	14	2	0	0	7	28	0	1	0
7	27	4	0	0	8	15	1	0	0	7	29	3	0	1
7	28	3	1	0	8	16	8	1	0	7	30	0	1	0
7	30	3	0	0	8	17	1	0	0	7	31	1	0	2
7	32	3	0	0	8	18	2	0	0	8	10	1	1	0
8	10	4	0	0	8	19	2	0	0	8	12	0	2	0
8	11	5	0	0	8	21	3	0	0	8	13	3	0	0
8	12	4	0	1	8	22	2	0	0	8	14	3	3	0
8	15	4	0	0	8	23	1	4	0	8	15	2	1	0
8	16	3	1	0	8	24	2	1	0	8	17	0	2	0
8	17	3	0	0	8	25	1	0	0	8	18	1	1	0
8	18	5	0	0	8	26	2	0	0	8	19	3	0	0
8	19	4	0	0	8	27	2	1	0	8	23	0	0	2

8	20	4	2	0	8	29	2	2	0	8	24	0	2	1
8	22	2	1	0	8	30	0	2	0	8	25	4	0	0
8	24	3	0	0	8	32	3	0	0	8	27	0	2	0
8	25	4	1	0	9	10	3	2	0	8	29	4	2	0
8	26	1	0	0	9	11	3	0	0	8	32	4	1	0
8	27	4	1	0	9	12	0	0	1	9	10	3	2	0
8	28	3	1	1	9	13	1	1	0	9	11	2	2	0
8	29	7	0	0	9	15	2	4	0	9	12	2	0	0
8	31	8	1	0	9	16	2	1	0	9	13	2	0	0
9	11	5	0	0	9	17	3	0	0	9	14	3	2	0
9	12	8	0	0	9	18	3	0	0	9	15	1	1	0
9	13	8	1	0	9	21	0	0	0	9	17	2	1	0
9	16	6	0	0	9	22	1	0	0	9	18	2	0	0
9	17	8	0	0	9	23	0	0	0	9	19	1	2	1
9	18	3	1	0	9	24	0	1	0	9	21	1	1	1
9	19	9	0	0	9	25	2	0	0	9	22	2	1	0
9	21	4	1	0	9	26	3	1	1	9	23	3	1	0
9	25	2	1	0	9	27	3	2	0	9	24	0	1	0
9	27	3	0	0	9	30	0	0	0	9	25	0	2	0
9	28	3	1	0	9	32	1	3	0	9	26	4	0	0
9	29	5	0	1	10	12	2	4	0	9	27	1	0	0
9	31	2	0	0	10	15	1	4	0	9	30	0	0	1
9	32	2	1	0	10	16	1	1	0	10	12	3	0	0
10	12	7	1	0	10	18	3	1	0	10	13	2	1	0
10	14	1	0	0	10	19	2	1	0	10	14	2	2	0
10	15	4	1	0	10	20	3	0	1	10	16	6	0	0
10	16	6	1	0	10	24	2	1	0	10	17	0	0	0
10	19	4	0	0	10	25	1	1	0	10	19	3	0	0
10	20	4	0	0	10	27	3	1	1	10	20	2	0	0
10	21	1	0	0	10	28	1	2	0	10	21	2	0	0
10	22	3	0	0	10	29	3	0	1	10	22	0	2	0
10	23	6	1	1	10	30	1	3	0	10	23	2	0	0
10	24	4	0	1	10	31	1	1	1	10	26	1	0	1
10	27	8	2	0	10	32	0	0	0	10	27	2	0	0
10	29	3	0	0	11	12	1	0	0	10	28	1	1	0
10	30	4	0	0	11	15	3	1	0	10	30	3	0	0
11	12	2	1	0	11	17	2	3	0	10	31	2	1	0
11	13	5	1	0	11	21	3	2	0	11	12	3	2	0
11	17	5	0	1	11	23	4	0	1	11	13	1	0	2
11	18	8	0	1	11	24	0	1	0	11	16	4	3	0
11	19	1	0	0	11	25	4	0	1	11	18	2	1	2
11	20	3	1	1	11	26	2	1	1	11	19	2	0	0
11	21	5	0	1	11	28	4	2	0	11	20	1	2	0
11	22	10	1	0	11	29	0	2	0	11	21	2	0	0
11	23	3	0	0	12	14	0	0	0	11	24	4	1	0
11	24	4	0	0	12	15	3	0	0	11	25	1	1	0
11	25	2	1	0	12	17	1	2	0	11	26	1	1	1

11	26	3	0	0	12	18	1	0	0	11	27	0	2	0
11	27	4	0	0	12	19	2	3	0	11	28	1	3	0
11	29	3	0	0	12	21	2	1	0	11	29	1	1	0
11	30	2	0	1	12	23	2	0	0	12	13	1	0	0
11	32	1	1	0	12	24	2	3	1	12	14	0	0	0
12	13	2	0	0	12	25	2	0	1	12	15	0	1	0
12	14	9	0	0	12	28	3	1	1	12	16	3	0	1
12	18	3	1	0	12	30	1	1	0	12	17	3	1	0
12	19	2	0	0	12	32	5	0	0	12	18	2	0	0
12	20	2	0	0	13	14	1	2	0	12	19	4	1	2
12	22	5	0	0	13	15	2	1	0	12	20	3	0	0
12	24	4	0	0	13	16	1	3	0	12	22	2	0	0
12	25	3	0	1	13	17	2	2	1	12	23	0	0	0
12	26	4	0	1	13	18	4	3	0	12	25	2	0	0
12	27	3	0	0	13	19	0	1	0	12	26	1	3	0
12	31	6	2	0	13	20	1	0	0	12	27	2	0	1
12	32	4	1	0	13	22	2	2	0	12	28	0	3	0
13	14	3	1	0	13	23	1	1	0	12	30	2	1	0
13	15	1	0	1	13	24	4	1	0	12	31	2	1	0
13	16	6	0	1	13	25	3	0	1	12	32	3	0	0
13	17	6	0	1	13	26	1	0	0	13	14	2	0	2
13	18	5	0	0	13	27	2	0	0	13	15	1	1	0
13	19	3	0	0	13	29	0	0	1	13	16	2	1	1
13	22	5	0	1	13	30	1	2	0	13	17	2	0	1
13	23	3	0	0	13	31	1	0	0	13	18	3	0	1
13	24	7	1	1	14	15	6	0	0	13	19	0	3	0
13	25	6	0	0	14	17	4	0	0	13	22	2	1	0
13	26	3	0	0	14	19	1	1	0	13	24	3	0	0
13	27	5	0	0	14	22	4	0	0	13	26	1	4	0
13	29	4	0	0	14	23	4	0	0	13	27	3	0	0
13	31	6	0	0	14	24	2	2	0	13	30	4	1	0
13	32	5	0	0	14	26	1	2	1	13	31	2	0	0
14	15	7	1	0	14	27	1	0	0	13	32	2	4	0
14	17	3	0	0	14	28	0	1	0	14	15	1	2	0
14	18	4	0	1	14	30	3	0	1	14	16	1	0	0
14	19	5	0	0	14	32	2	1	1	14	17	3	1	2
14	20	6	0	0	15	16	4	3	0	14	18	3	1	0
14	21	4	1	0	15	17	3	1	0	14	20	2	1	0
14	22	2	1	0	15	18	2	0	0	14	22	3	1	0
14	24	2	1	0	15	20	3	2	0	14	23	1	2	1
14	25	4	0	0	15	22	4	1	0	14	24	2	0	1
14	26	3	1	0	15	24	2	1	0	14	25	3	2	1
14	27	8	0	1	15	25	2	0	0	14	26	3	2	1
14	28	7	2	0	15	26	3	0	0	14	29	2	2	0
14	29	6	0	0	15	27	3	0	0	14	30	0	0	0
14	31	7	1	0	15	29	1	0	0	14	32	0	1	1
14	32	5	1	0	15	30	3	1	0	15	16	5	2	0

15	16	5	0	1	15	31	5	0	0	15	18	3	0	0
15	17	4	0	0	16	18	1	1	0	15	20	1	1	0
15	20	5	1	0	16	19	1	0	0	15	21	1	1	1
15	22	2	1	0	16	20	1	1	0	15	22	0	3	0
15	23	3	1	0	16	23	3	0	0	15	23	2	0	0
15	26	2	1	0	16	24	1	0	0	15	24	2	0	0
15	27	5	2	1	16	25	3	2	0	15	25	0	1	0
15	29	5	1	0	16	29	2	1	0	15	26	2	0	1
15	31	3	0	0	16	30	1	2	0	15	27	3	3	0
15	32	6	0	0	16	31	1	0	0	15	28	0	1	1
16	17	3	1	0	16	32	1	1	0	15	29	3	1	1
16	18	4	1	0	17	18	2	0	0	15	30	2	0	0
16	19	4	0	0	17	20	3	3	0	15	31	1	4	0
16	20	4	2	0	17	21	3	2	0	15	32	4	0	0
16	21	4	0	1	17	22	2	1	0	16	17	1	3	0
16	22	1	0	1	17	24	2	1	1	16	19	2	2	0
16	27	4	0	1	17	25	1	1	0	16	20	0	2	1
16	28	2	0	1	17	26	2	0	0	16	21	1	2	1
16	29	3	1	0	17	28	2	1	0	16	27	0	1	0
16	30	1	2	0	17	29	1	1	0	16	29	2	0	0
16	31	5	0	0	17	30	0	2	0	16	30	0	3	0
16	32	4	0	0	17	31	2	0	0	16	32	1	1	0
17	18	3	0	0	17	32	2	0	0	17	18	1	2	0
17	20	2	1	0	18	20	4	0	0	17	19	3	2	1
17	22	4	0	0	18	21	3	0	0	17	20	3	2	1
17	23	4	0	0	18	22	1	1	0	17	21	1	0	0
17	24	2	1	0	18	23	2	1	0	17	22	0	1	0
17	25	4	0	0	18	27	3	1	1	17	26	4	2	0
17	27	3	0	0	18	28	5	1	0	17	27	1	2	2
17	28	7	1	0	18	30	2	0	0	17	28	1	0	0
17	29	4	0	0	18	32	2	1	0	17	30	2	2	0
17	31	4	1	0	19	20	4	1	0	17	32	0	0	0
17	32	3	1	0	19	21	1	1	0	18	21	2	1	0
18	19	2	0	0	19	23	0	1	1	18	23	0	3	0
18	20	2	0	0	19	25	0	2	0	18	24	3	0	2
18	21	4	0	0	19	26	3	0	0	18	25	2	0	1
18	22	3	0	0	19	28	1	2	0	18	26	1	0	0
18	24	3	1	0	19	29	2	1	0	18	27	1	1	0
18	25	6	0	0	19	31	3	1	0	18	29	0	1	1
18	26	7	0	0	19	32	2	1	0	18	30	5	1	0
18	28	5	0	1	20	22	3	0	1	19	21	2	1	0
18	29	3	2	0	20	23	0	1	1	19	22	1	1	0
18	31	6	0	0	20	24	2	0	0	19	23	3	1	0
18	32	4	0	0	20	25	1	0	0	19	26	0	0	0
19	20	4	0	0	20	26	0	0	1	19	27	2	1	0
19	21	2	0	0	20	27	2	2	1	19	28	1	0	0
19	22	3	0	0	20	28	1	2	0	19	30	0	0	0

19	23	6	0	1	20	29	0	0	0	19	32	1	2	0
19	26	3	1	0	20	31	3	1	0	20	21	4	0	1
19	27	4	0	0	20	32	4	1	0	20	22	0	0	0
19	29	3	0	0	21	24	1	0	1	20	23	1	0	0
19	30	3	1	0	21	25	2	2	0	20	25	1	2	0
20	21	4	0	0	21	26	0	1	0	20	26	1	0	0
20	22	4	0	0	21	27	7	1	0	20	27	3	0	0
20	23	5	1	0	21	28	1	1	0	20	28	3	0	0
20	25	2	0	0	21	30	0	1	0	20	29	1	0	0
20	26	6	1	0	21	32	3	0	0	20	32	0	2	0
20	28	4	0	1	22	23	0	4	1	21	22	1	1	0
20	29	5	0	0	22	24	2	2	0	21	23	2	0	0
20	30	5	0	0	22	25	3	1	0	21	24	1	0	0
21	22	6	0	0	22	26	0	1	1	21	25	2	0	0
21	24	5	0	0	22	27	3	0	0	21	26	2	0	0
21	26	3	0	0	22	28	0	1	0	21	27	3	0	2
21	27	8	1	0	22	29	1	1	0	21	28	1	1	0
21	29	1	0	0	22	31	1	0	0	21	30	1	1	0
21	30	2	0	1	22	32	1	1	0	21	31	0	2	0
21	31	4	0	0	23	24	2	3	1	21	32	0	2	1
21	32	2	1	0	23	25	2	2	0	22	24	1	0	0
22	23	8	0	0	23	26	1	3	0	22	26	3	2	0
22	24	4	2	0	23	27	2	1	1	22	28	1	0	0
22	26	2	0	0	23	29	2	3	0	23	24	0	0	1
22	27	3	1	0	23	30	3	0	0	23	25	1	2	0
22	28	2	0	0	23	31	1	2	0	23	27	2	0	0
22	29	4	0	0	23	32	2	0	1	23	28	1	1	0
22	31	5	1	0	24	25	0	1	0	23	29	1	1	1
23	25	5	1	0	24	26	1	2	0	23	31	0	1	0
23	26	3	0	0	24	27	2	2	0	24	25	4	0	0
23	27	3	0	0	24	28	4	1	0	24	27	2	1	0
23	28	4	0	0	24	29	1	0	0	24	28	1	2	0
23	29	3	2	0	24	30	0	1	0	24	29	2	0	0
24	25	2	0	0	24	31	2	0	0	24	30	2	0	2
24	26	1	0	0	25	26	1	2	1	24	31	3	1	0
24	27	4	0	0	25	27	0	0	0	25	27	1	0	1
24	29	4	1	1	25	29	1	0	0	25	28	1	1	0
24	32	6	2	0	25	32	1	2	0	25	29	1	1	0
25	28	3	0	0	26	27	1	1	0	26	29	1	2	0
25	30	8	1	0	26	28	1	0	0	26	32	1	0	0
26	27	6	1	0	26	29	3	2	0	27	29	1	0	0
26	28	2	2	0	26	30	1	1	1	27	30	0	1	0
26	31	8	1	1	26	31	5	0	0	27	31	1	0	0
26	32	1	1	0	26	32	2	1	0	27	32	0	1	1
27	29	5	0	0	27	28	2	3	1	28	29	2	0	0
27	30	4	1	1	27	29	3	0	0	28	30	1	0	0
27	31	4	0	0	27	30	1	0	0	28	31	2	0	0

27	32	6	0	1	28	31	1	0	0	28	32	5	1	1
28	31	3	0	0	28	32	3	2	0	29	30	0	0	1
28	32	7	1	0	29	30	1	1	1	29	31	4	2	2
29	31	8	1	0	29	32	3	1	1	29	32	1	0	0
29	32	8	1	0	30	31	3	0	0	30	31	0	1	1
31	32	4	1	0	31	32	3	2	0	31	32	1	2	0

T_1^{NSFNET}					T_2^{NSFNET}					T_3^{NSFNET}				
s_d	t_d	n_{10G}	n_{40G}	n_{100G}	s_d	t_d	n_{10G}	n_{40G}	n_{100G}	s_d	t_d	n_{10G}	n_{40G}	n_{100G}
1	2	15	2	0	1	4	7	3	0	1	2	7	7	0
1	7	16	3	1	1	5	8	8	0	1	5	5	5	1
1	8	20	1	0	2	6	14	4	1	1	6	8	5	1
1	9	22	2	1	2	12	15	6	0	1	7	8	3	2
2	3	20	3	0	3	4	11	4	1	1	9	9	3	1
2	7	30	2	2	3	5	11	5	1	1	11	4	9	2
2	9	15	2	1	3	7	11	5	0	1	12	5	4	1
2	11	31	0	0	3	8	11	3	0	2	3	6	7	1
2	14	12	2	1	3	9	8	4	2	2	4	5	6	3
3	5	14	3	0	4	6	8	7	1	2	8	6	5	3
3	8	18	1	2	4	9	9	11	1	2	9	4	4	0
3	9	20	1	0	4	12	13	5	0	2	10	11	6	1
3	11	24	1	1	4	13	15	4	1	2	11	11	5	0
3	13	21	1	1	5	6	7	3	0	2	12	11	6	2
3	14	26	2	1	5	7	2	5	2	2	14	4	1	0
4	11	24	1	1	5	9	9	3	1	3	9	12	7	2
4	12	19	2	1	6	9	9	4	2	3	13	8	4	1
4	13	23	1	0	6	12	8	3	1	3	14	10	8	3
5	6	21	1	0	6	13	12	6	0	4	7	6	5	0
5	9	25	3	0	7	8	14	4	1	4	9	3	4	1
5	13	26	0	1	7	14	13	6	0	4	14	8	7	2
6	13	15	1	1	8	14	9	8	3	5	7	8	6	2
7	10	15	1	1	9	13	16	4	1	5	11	2	5	1
8	11	20	1	0	9	14	14	5	1	6	9	4	2	1
8	12	25	0	1	10	11	9	5	1	6	11	7	2	3
8	14	15	5	0	11	12	2	5	1	6	13	9	6	3
9	10	15	2	0	11	13	8	6	1	7	10	6	1	0
9	11	21	1	2	11	14	6	5	1	9	13	4	6	1
9	14	15	3	0	12	13	11	4	0	10	11	6	7	1
13	14	17	2	1	12	14	10	5	1	11	14	3	4	1

T_4^{NSFNET}	T_5^{NSFNET}	T_6^{NSFNET}

s_d	t_d	n_{10G}	n_{40G}	n_{100G}	s_d	t_d	n_{10G}	n_{40G}	n_{100G}	s_d	t_d	n_{10G}	n_{40G}	n_{100G}
1	2	15	3	0	1	2	6	7	0	1	2	3	0	1
1	5	17	2	2	1	5	6	4	0	1	3	1	6	0
1	6	16	2	0	1	7	11	5	0	1	4	6	2	1
1	10	17	3	0	1	8	15	0	2	1	5	6	4	2
1	11	14	2	1	1	10	8	1	0	1	8	7	2	0
1	12	19	0	1	1	11	12	5	0	1	9	4	5	0
1	13	15	3	0	1	12	7	2	1	1	11	8	2	0
1	14	14	0	0	1	13	4	4	1	1	13	5	9	2
2	8	17	0	1	1	14	4	6	0	1	14	3	5	2
2	9	15	0	0	2	6	9	6	1	2	3	9	6	3
2	10	20	5	0	2	10	8	4	1	2	4	8	6	0
2	11	15	1	1	2	11	10	4	0	2	5	6	7	0
2	13	16	0	1	2	12	6	5	1	2	6	7	1	1
2	14	19	0	1	2	14	4	7	1	2	7	5	3	1
3	5	18	1	0	3	8	4	7	0	2	9	4	5	0
3	6	18	1	1	3	11	7	6	0	2	11	6	4	2
3	9	13	0	0	3	14	7	6	0	3	4	5	6	1
3	10	14	1	0	4	8	12	7	0	3	8	8	7	1
3	12	23	0	0	4	10	10	2	1	3	11	7	3	0
3	14	21	3	0	4	11	8	4	0	3	12	2	3	3
4	8	19	1	1	5	7	10	3	0	3	13	5	3	2
4	10	12	2	0	5	9	11	5	1	3	14	8	4	2
4	14	16	1	1	5	10	12	8	1	4	7	9	3	2
5	6	19	1	1	5	11	14	4	2	4	9	4	6	1
5	8	22	1	0	5	12	10	4	0	4	11	4	7	0
5	11	15	0	1	5	14	10	5	0	4	14	6	4	2
5	12	21	3	0	6	7	6	4	1	5	6	9	3	0
6	7	20	3	0	6	10	8	3	0	5	7	11	6	2
6	8	17	1	1	6	11	9	3	0	5	9	1	7	2
6	9	13	3	1	6	14	8	0	1	6	7	6	4	2
6	10	11	1	0	7	9	12	2	2	6	9	5	4	1
6	12	23	0	1	7	13	3	4	1	6	12	6	6	1
6	13	9	2	0	7	14	9	3	2	6	13	7	2	3
6	14	27	1	1	8	11	12	6	2	6	14	8	3	0
7	12	13	0	1	8	13	4	4	2	7	8	3	3	3
7	13	17	2	0	9	10	4	4	1	7	9	6	2	0
8	14	18	2	1	9	11	9	5	1	7	11	6	4	1
9	10	23	2	1	9	12	13	3	2	7	14	4	4	0
9	11	23	2	1	9	13	9	6	1	9	11	10	6	0
9	12	16	4	1	9	14	11	6	0	9	12	4	7	2
9	14	21	1	2	10	11	13	4	0	9	14	4	4	3
10	11	17	0	0	10	12	8	4	1	10	11	4	7	0
10	14	14	2	0	10	13	8	5	1	10	14	7	0	1
11	13	21	2	2	10	14	5	4	1	11	14	7	5	1

11	14	17	1	0	12	14	14	4	1	12	13	6	5	1

T_7^{NSFNET}					T_8^{NSFNET}					T_9^{NSFNET}				
s_d	t_d	n_{10G}	n_{40G}	n_{100G}	s_d	t_d	n_{10G}	n_{40G}	n_{100G}	s_d	t_d	n_{10G}	n_{40G}	n_{100G}
1	3	27	2	0	1	3	10	3	0	1	2	4	4	4
1	4	17	0	1	1	4	10	3	0	1	5	6	6	1
1	5	17	1	0	1	5	3	5	1	1	6	9	3	1
1	6	19	2	0	1	6	10	5	0	1	7	8	6	1
1	7	15	3	2	1	7	9	7	0	1	8	8	4	2
1	9	18	4	1	1	9	5	5	1	1	9	8	3	1
1	10	21	5	0	1	11	7	1	0	1	12	3	5	1
1	11	18	1	0	1	13	8	4	1	1	13	4	4	0
1	13	17	0	0	2	5	14	5	1	1	14	7	4	2
1	14	18	2	1	2	7	11	6	0	2	3	3	4	4
2	7	14	2	1	2	9	10	4	0	2	4	4	5	1
2	8	14	1	1	2	10	7	4	1	2	5	1	5	0
2	9	16	2	0	2	12	5	5	1	2	9	5	3	1
2	10	11	0	0	2	13	8	3	1	2	12	7	5	0
2	11	15	5	0	3	5	8	2	1	2	14	9	2	0
2	12	18	1	1	3	6	12	4	1	3	5	5	3	2
2	13	13	1	2	3	7	8	4	1	3	8	7	4	2
2	14	20	1	2	3	9	10	6	3	3	9	4	5	2
3	4	13	1	2	3	10	8	8	0	3	10	8	5	1
3	6	13	0	1	3	13	9	8	1	3	11	7	5	1
3	7	15	1	0	4	5	9	4	1	3	12	7	4	2
3	8	12	1	1	4	6	4	2	2	3	13	5	8	1
3	11	19	1	0	4	7	10	6	1	3	14	6	5	2
3	12	20	1	1	4	8	9	7	1	4	5	3	5	2
3	13	16	0	0	4	10	8	5	0	4	6	8	3	0
3	14	16	2	0	4	11	7	1	2	4	7	7	1	0
4	5	17	1	0	4	12	11	6	1	4	8	4	3	1
4	6	12	1	0	4	13	15	3	0	4	9	8	4	4
4	7	16	0	3	5	6	8	2	0	4	10	2	2	0
4	10	26	2	0	5	7	2	3	0	4	12	4	1	0
4	11	16	0	0	5	8	11	3	1	4	13	5	4	0
4	12	17	0	0	5	9	6	6	1	5	6	7	5	1
5	6	18	2	0	5	13	14	5	0	5	9	11	3	2
5	7	12	0	0	5	14	12	5	1	5	10	9	5	2
5	9	15	0	1	6	7	8	4	1	5	12	6	3	0
5	13	10	2	1	6	8	8	4	1	5	14	7	6	0
6	7	13	2	0	6	12	6	6	0	6	7	9	3	2
6	8	21	1	0	6	13	9	5	2	6	9	7	7	2
6	11	21	4	0	6	14	9	5	0	6	11	5	4	2
6	12	25	1	1	7	8	10	5	2	6	12	6	5	0

6	13	19	0	1	7	9	7	3	0	6	13	1	6	2
7	8	11	0	0	7	11	5	2	0	6	14	2	3	1
7	9	14	3	0	7	13	9	7	0	7	8	4	4	0
7	10	13	2	0	8	9	10	2	1	7	11	6	4	1
7	13	15	2	1	8	10	8	0	0	7	12	4	5	0
8	9	18	1	1	8	11	7	5	0	8	9	3	8	1
8	10	15	0	1	8	12	8	6	0	8	10	8	7	1
8	11	21	2	2	8	14	18	5	1	8	11	3	3	0
9	10	15	3	0	9	10	9	2	0	8	13	6	1	2
9	11	17	5	1	9	12	9	1	0	8	14	3	3	3
9	12	27	1	0	9	13	6	4	0	9	12	3	3	1
9	14	10	0	0	9	14	6	8	0	9	13	8	5	0
10	11	15	0	1	10	11	5	6	0	9	14	6	2	0
10	12	25	0	0	10	12	5	2	4	10	11	7	6	3
10	14	18	3	0	10	13	11	5	1	10	12	8	4	0
11	12	24	1	2	10	14	8	2	2	10	13	6	5	0
11	14	17	1	0	11	12	10	2	1	10	14	3	3	2
12	13	18	2	1	11	14	6	5	0	11	13	4	6	0
12	14	18	1	0	12	13	9	3	1	12	14	5	3	1
13	14	13	2	0	12	14	3	5	1	13	14	5	7	0

Appendix B

Computational Results

B.1 Multi-Start Local Search Overall Results

The full results obtained with the optimization tool of this dissertation are presented next. For each traffic matrix of a given network, $T_i^{network}$, the algorithm was executed with 1 thread, 12 threads, 12 threads with shared memory, 24 threads and 24 threads with shared memory. The results represent average values obtained from 10 runs of 1 minute each. In this first set of results, the number of candidate paths in the fiber and logical networks was set to 20.

T_1^{EON}							
$Threads$	Sol_{Cost}	Sol_{Slot}	Sol_{Time}	$Iterations$	$InvalidSA$	$Greedy_{unf}$	LS_{unf}
1	6688.1	107.0	20.0	1049.7	0.0	0.0	0.0
12	6682.3	104.4	26.1	9815.6	0.0	0.0	0.0
12_{shared}	6682.1	107.4	29.3	9776.7	0.0	0.0	0.0
24	6680.9	106.8	36.6	9256.7	0.0	0.0	0.0
24_{shared}	6680.3	108.0	37.9	9057.6	0.0	0.0	0.0

T_2^{EON}							
$Threads$	Sol_{Cost}	Sol_{Slot}	Sol_{Time}	$Iterations$	$InvalidSA$	$Greedy_{unf}$	LS_{unf}
1	7824.9	100.6	32.9	1953.4	0.0	0.0	0.0
12	7817.3	101.0	25.6	18063.9	0.0	0.0	0.0
12_{shared}	7819.6	101.4	30.7	18446.7	0.0	0.0	0.0
24	7820.2	98.8	25.8	16303.9	0.0	0.0	0.0
24_{shared}	7817.7	100.6	30.8	16442.6	0.0	0.0	0.0

T_3^{EON}							
$Threads$	Sol_{Cost}	Sol_{Slot}	Sol_{Time}	$Iterations$	$InvalidSA$	$Greedy_{unf}$	LS_{unf}
1	7819.7	94.0	35.5	2930.6	0.0	0.0	0.0
12	7817.2	93.2	23.4	28026.3	0.0	0.0	0.0
12_{shared}	7816.7	93.8	28.8	28114.2	0.0	0.0	0.0
24	7816.7	93.8	25.3	26023.7	0.0	0.0	0.0
24_{shared}	7817.0	92.0	20.0	25687.5	0.0	0.0	0.0

T_4^{EON}							
$Threads$	Sol_{Cost}	Sol_{Slot}	Sol_{Time}	$Iterations$	$InvalidSA$	$Greedy_{unf}$	LS_{unf}
1	8385.5	136.6	36.3	635.8	0.0	0.0	0.0
12	8372.8	136.4	30.5	6487.8	0.0	0.0	0.0
12_{shared}	8371.2	136.6	28.5	6485.5	0.0	0.0	0.0
24	8370.5	135.2	26.6	6334.8	0.0	0.0	0.0
24_{shared}	8370.0	137.6	29.4	6316.7	0.0	0.0	0.0

T_5^{EON}							
$Threads$	Sol_{Cost}	Sol_{Slot}	Sol_{Time}	$Iterations$	$InvalidSA$	$Greedy_{unf}$	LS_{unf}
1	9360.6	159.9	42.0	853.9	0.1	801.7	0.0
12	9351.2	159.3	30.0	8674.4	0.7	8165.3	0.0
12_{shared}	9347.0	159.7	25.5	8695.5	0.7	8189.8	0.0
24	9350.2	159.8	32.7	8427.2	1.5	7940.5	0.0
24_{shared}	9352.2	159.9	35.9	8427.6	0.7	7946.4	0.0

T_6^{EON}							
$Threads$	Sol_{Cost}	Sol_{Slot}	Sol_{Time}	$Iterations$	$InvalidSA$	$Greedy_{unf}$	LS_{unf}
1	10022.7	159.8	21.6	1521.6	0.0	752.7	0.0
12	10005.6	159.2	32.7	15245.2	0.0	7596.2	0.0
12_{shared}	10001.9	159.2	24.5	15154.4	0.0	7565.9	0.0
24	10004.8	159.4	33.3	14882.1	0.0	7420.0	0.0
24_{shared}	10003.6	159.4	32.7	14941.8	0.0	7457.1	0.0

T_7^{EON}							
$Threads$	Sol_{Cost}	Sol_{Slot}	Sol_{Time}	$Iterations$	$InvalidSA$	$Greedy_{unf}$	LS_{unf}
1	11391.7	160.0	23.7	325.9	0.1	325.9	255.1
12	11321.1	160.0	38.8	3407.2	0.0	3407.2	2676.7
12_{shared}	11320.0	160.0	40.1	3385.7	0.0	3385.7	2657.4
24	11322.9	160.0	35.4	3310.8	0.0	3310.8	2582.5
24_{shared}	11323.6	160.0	31.6	3334.2	0.0	3334.2	2610.7

T_8^{EON}							
$Threads$	Sol_{Cost}	Sol_{Slot}	Sol_{Time}	$Iterations$	$InvalidSA$	$Greedy_{unf}$	LS_{unf}
1	11655.2	141.4	26.5	682.4	0.0	0.0	0.0
12	11637.1	140.6	28.9	6982.1	0.1	0.0	0.0
12_{shared}	11635.0	137.4	34.6	7081.0	0.0	0.0	0.0
24	11639.2	138.4	34.7	6855.2	0.1	0.0	0.0
24_{shared}	11635.2	140.2	30.8	6912.0	0.0	0.0	0.0

T_9^{EON}							
$Threads$	Sol_{Cost}	Sol_{Slot}	Sol_{Time}	$Iterations$	$InvalidSA$	$Greedy_{unf}$	LS_{unf}
1	13524.7	159.8	36.3	650.8	6.2	650.8	0.0
12s	13497.0	159.9	28.6	6508.7	75.9	6508.7	0.1
12_{shared}	13507.0	159.7	29.3	6653.2	25.5	6653.2	0.2
24	13502.8	159.7	24.8	6388.1	125.6	6388.1	0.2
24_{shared}	13501.1	159.8	16.8	6496.4	35.8	6496.4	0.1

T_1^{GBN}							
$Threads$	Sol_{Cost}	Sol_{Slot}	Sol_{Time}	$Iterations$	$InvalidSA$	$Greedy_{unf}$	LS_{unf}
1	4731.2	141.8	27.4	1803.4	0.5	0.0	0.0
12	4730.0	129.2	26.0	14724.5	8.1	0.0	0.0
12_{shared}	4730.0	127.4	30.7	14664.9	2.7	0.0	0.0
24	4730.0	129.6	24.9	12753.1	12.0	0.0	0.0
24_{shared}	4730.0	128.2	22.5	13042.0	5.1	0.0	0.0

T_2^{GBN}							
$Threads$	Sol_{Cost}	Sol_{Slot}	Sol_{Time}	$Iterations$	$InvalidSA$	$Greedy_{unf}$	LS_{unf}
1	5182.2	157.8	27.6	5218.1	117.4	126.7	7.9
12	5179.6	158.4	38.7	38088.6	1409.8	910.2	47.0
12_{shared}	5180.0	155.8	33.4	39609.7	247.6	975.6	54.9
24	5179.8	157.0	20.2	33497.0	2372.2	784.5	43.0
24_{shared}	5179.2	158.2	21.1	34906.8	283.0	814.2	46.4

T_3^{GBN}							
$Threads$	Sol_{Cost}	Sol_{Slot}	Sol_{Time}	$Iterations$	$InvalidSA$	$Greedy_{unf}$	LS_{unf}
1	5570.2	137.2	37.4	6211.7	0.8	0.0	0.0
12	5567.6	131.0	27.4	47699.3	5.9	0.0	0.0
12_{shared}	5566.8	135.2	27.1	48184.1	1.9	0.0	0.0
24	5568.0	132.0	22.5	42651.1	10.5	0.0	0.0
24_{shared}	5566.8	134.8	19.4	42732.1	3.8	0.0	0.0

T_4^{GBN}							
$Threads$	Sol_{Cost}	Sol_{Slot}	Sol_{Time}	$Iterations$	$InvalidSA$	$Greedy_{unf}$	LS_{unf}
1	5716.4	158.6	31.4	1104.3	281.1	516.7	40.5
12	5702.4	158.0	36.7	9831.6	3288.3	4624.4	370.3
12_{shared}	5701.6	157.2	26.5	10146.4	484.4	4756.7	377.4
24	5703.0	159.0	21.2	9300.9	4602.3	4385.1	348.9
24_{shared}	5703.8	156.6	24.6	9677.3	594.2	4551.3	358.4

T_5^{GBN}							
$Threads$	Sol_{Cost}	Sol_{Slot}	Sol_{Time}	$Iterations$	$InvalidSA$	$Greedy_{unf}$	LS_{unf}
1	6257.0	159.4	24.4	1573.9	124.3	1140.7	649.1
12	6247.0	158.8	24.3	14857.7	1417.8	10708.9	6027.3
12_{shared}	6246.0	158.6	34.3	15217.5	217.9	10961.4	6180.0
24	6246.8	158.6	34.5	14128.6	2390.6	10197.4	5743.3
24_{shared}	6246.6	158.4	36.0	14619.4	267.9	10534.0	5930.0

T_6^{GBN}							
$Threads$	Sol_{Cost}	Sol_{Slot}	Sol_{Time}	$Iterations$	$InvalidSA$	$Greedy_{unf}$	LS_{unf}
1	6713.6	156.8	22.8	3112.4	19.1	1.4	0.0
12	6708.2	154.4	31.8	28569.0	217.3	14.2	0.0
12_{shared}	6706.8	157.2	27.9	28801.6	56.2	12.7	0.2
24	6706.4	155.6	27.8	27426.5	369.3	12.6	0.2
24_{shared}	6707.8	155.2	38.2	26698.9	87.0	13.3	0.0

T_7^{GBN}							
$Threads$	Sol_{Cost}	Sol_{Slot}	Sol_{Time}	$Iterations$	$InvalidSA$	$Greedy_{unf}$	LS_{unf}
1	6127.4	159.4	34.0	343.0	70.7	343.0	225.7
12	6081.6	159.4	30.9	3463.2	652.3	3463.2	2375.7
12_{shared}	6081.2	159.2	20.0	3469.9	164.9	3469.8	2363.5
24	6074.8	158.6	33.6	3424.1	821.1	3424.1	2333.0
24_{shared}	6076.8	159.2	32.6	3448.3	194.6	3448.3	2355.7

T_8^{GBN}							
$Threads$	Sol_{Cost}	Sol_{Slot}	Sol_{Time}	$Iterations$	$InvalidSA$	$Greedy_{unf}$	LS_{unf}
1	5277.6	127.8	25.5	940.2	0.0	0.0	0.0
12	5259.4	127.6	25.8	9000.2	0.2	0.0	0.0
12_{shared}	5251.8	127.8	38.2	9075.6	0.0	0.0	0.0
24	5256.0	127.8	35.6	8509.5	0.3	0.0	0.0
24_{shared}	5255.4	127.2	36.6	8373.4	0.0	0.0	0.0

T_9^{GBN}							
$Threads$	Sol_{Cost}	Sol_{Slot}	Sol_{Time}	$Iterations$	$InvalidSA$	$Greedy_{unf}$	LS_{unf}
1	5419.0	133.4	14.7	1067.9	0.2	0.0	0.0
12	5364.2	134.4	30.3	10164.9	0.4	0.0	0.0
12_{shared}	5369.8	137.0	32.2	9997.2	0.7	0.0	0.0
24	5368.8	134.0	34.0	9328.5	0.7	0.0	0.0
24_{shared}	5375.0	133.2	19.8	9262.4	0.4	0.0	0.0

T_1^{GEANT2}							
$Threads$	Sol_{Cost}	Sol_{Slot}	Sol_{Time}	$Iterations$	$InvalidSA$	$Greedy_{unf}$	LS_{unf}
1	7406.0	126.0	41.8	244.2	0.0	0.0	0.0
12	7383.2	126.2	28.9	2369.0	0.0	0.0	0.0
12_{shared}	7380.4	130.4	27.4	2432.0	0.0	0.0	0.0
24	7383.4	126.0	31.9	2321.9	0.0	0.0	0.0
24_{shared}	7381.6	123.0	29.1	2350.1	0.0	0.0	0.0

T_2^{GEANT2}							
$Threads$	Sol_{Cost}	Sol_{Slot}	Sol_{Time}	$Iterations$	$InvalidSA$	$Greedy_{unf}$	LS_{unf}
1	7597.0	147.6	34.6	435.9	0.8	0.0	0.0
12	7570.9	148.4	33.5	4125.3	11.0	0.0	0.0
12_{shared}	7579.6	145.6	29.7	4148.4	6.8	0.0	0.0
24	7578.4	148.0	29.9	4048.5	19.0	0.0	0.0
24_{shared}	7575.6	146.6	28.5	4119.0	9.8	0.0	0.0

T_3^{GEANT2}							
$Threads$	Sol_{Cost}	Sol_{Slot}	Sol_{Time}	$Iterations$	$InvalidSA$	$Greedy_{unf}$	LS_{unf}
1	8532.4	152.2	30.1	395.8	0.5	20.6	0.0
12	8511.8	150.2	29.8	3765.8	4.6	177.3	0.0
12_{shared}	8513.7	151.0	27.8	3869.6	3.8	175.3	0.0
24	8505.1	151.0	42.0	3722.1	11.3	176.3	0.0
24_{shared}	8507.7	151.8	23.3	3755.5	5.2	177.7	0.0

T_4^{GEANT2}							
$Threads$	Sol_{Cost}	Sol_{Slot}	Sol_{Time}	$Iterations$	$InvalidSA$	$Greedy_{unf}$	LS_{unf}
1	9217.9	160.0	34.7	66.5	3.1	66.5	0.0
12	9174.3	159.8	32.0	333.9	21.9	333.9	0.0
12_{shared}	9184.6	159.6	41.6	343.8	11.5	343.8	0.1
24	9189.3	159.2	34.8	322.2	32.8	322.2	0.1
24_{shared}	9183.5	159.4	36.2	338.7	17.3	338.7	0.0

T_5^{GEANT2}							
$Threads$	Sol_{Cost}	Sol_{Slot}	Sol_{Time}	$Iterations$	$InvalidSA$	$Greedy_{unf}$	LS_{unf}
1	9022.4	154.2	29.7	138.5	0.3	5.0	0.0
12	9015.7	151.6	39.3	578.4	3.4	19.4	0.0
12_{shared}	9008.9	152.4	33.3	592.2	2.7	20.0	0.0
24	9021.9	154.0	36.9	457.6	4.6	13.2	0.0
24_{shared}	9020.9	153.8	39.8	460.0	2.2	12.6	0.0

T_6^{GEANT2}							
$Threads$	Sol_{Cost}	Sol_{Slot}	Sol_{Time}	$Iterations$	$InvalidSA$	$Greedy_{unf}$	LS_{unf}
1	10025.8	158.3	34.5	124.8	92.7	68.5	0.0
12	9983.4	158.8	27.4	388.1	361.8	220.3	0.0
12_{shared}	9954.4	159.6	22.9	385.6	143.3	217.5	0.0
24	9982.9	159.4	35.3	336.3	319.5	189.8	0.0
24_{shared}	9991.5	159.8	39.3	330.0	203.5	184.9	0.0

T_7^{GEANT2}							
$Threads$	Sol_{Cost}	Sol_{Slot}	Sol_{Time}	$Iterations$	$InvalidSA$	$Greedy_{unf}$	LS_{unf}
1	10168.9	159.8	38.7	36.4	15.7	36.3	0.0
12	10109.0	159.2	47.4	61.5	43.6	61.4	0.0
12_{shared}	10137.0	159.2	45.5	60.9	30.5	60.6	0.0
24	10111.2	158.8	53.0	60.5	47.9	60.5	0.0
24_{shared}	10131.6	159.8	56.0	58.9	39.5	58.8	0.0

T_8^{GEANT2}							
$Threads$	Sol_{Cost}	Sol_{Slot}	Sol_{Time}	$Iterations$	$InvalidSA$	$Greedy_{unf}$	LS_{unf}
1	10809.1	159.7	30.5	30.9	9.9	30.9	0.3
12	10846.0	160.0	46.4	53.1	46.8	53.1	0.4
12_{shared}	10830.5	159.4	44.7	54.4	36.2	54.4	0.3
24	10860.0	159.3	68.2	54.0	49.8	54.0	0.4
24_{shared}	10836.3	159.0	65.9	54.8	47.7	54.8	0.3

T_9^{GEANT2}							
$Threads$	Sol_{Cost}	Sol_{Slot}	Sol_{Time}	$Iterations$	$InvalidSA$	$Greedy_{unf}$	LS_{unf}
1	12313.6	157.6	37.8	17.0	7.8	17.0	0.1
12	12225.9	158.6	46.6	34.9	24.9	34.9	0.9
12_{shared}	12272.4	158.7	60.2	33.8	19.8	33.8	0.2
24	12281.5	158.9	70.6	35.3	27.6	35.3	0.6
24_{shared}	12232.2	157.9	74.8	35.1	23.4	35.1	0.4

T_1^{NSFNET}							
$Threads$	Sol_{Cost}	Sol_{Slot}	Sol_{Time}	$Iterations$	$InvalidSA$	$Greedy_{unf}$	LS_{unf}
1	4440.0	84.0	12.7	5043.5	0.0	0.0	0.0
12	4440.0	84.0	30.4	15911.2	0.0	0.0	0.0
12_{shared}	4440.0	84.0	8.0	16149.4	0.0	0.0	0.0
24	4440.0	84.0	31.8	15531.4	0.0	0.0	0.0
24_{shared}	4440.0	84.0	7.3	15019.7	0.0	0.0	0.0

T_2^{NSFNET}							
$Threads$	Sol_{Cost}	Sol_{Slot}	Sol_{Time}	$Iterations$	$InvalidSA$	$Greedy_{unf}$	LS_{unf}
1	4882.6	118.4	14.4	11026.8	0.0	0.0	0.0
12	4882.2	117.0	31.2	36121.2	0.0	0.0	0.0
12_{shared}	4882.2	117.8	21.2	36223.5	0.0	0.0	0.0
24	4882.3	118.2	23.1	34440.9	0.0	0.0	0.0
24_{shared}	4882.2	117.0	17.6	32862.3	0.0	0.0	0.0

T_3^{NSFNET}							
$Threads$	Sol_{Cost}	Sol_{Slot}	Sol_{Time}	$Iterations$	$InvalidSA$	$Greedy_{unf}$	LS_{unf}
1	5494.5	160.0	13.1	21411.3	0.0	0.0	0.0
12	5489.7	160.0	16.1	75771.6	0.0	0.0	0.0
12_{shared}	5491.4	160.0	21.1	74426.1	0.0	0.0	0.0
24	5492.7	160.0	31.6	69767.5	0.0	0.0	0.0
24_{shared}	5491.3	160.0	20.8	73463.7	0.0	0.0	0.0

T_4^{NSFNET}							
$Threads$	Sol_{Cost}	Sol_{Slot}	Sol_{Time}	$Iterations$	$InvalidSA$	$Greedy_{unf}$	LS_{unf}
1	5369.2	116.0	21.8	2208.9	0.0	0.0	0.0
12	5363.2	116.2	32.5	10275.0	0.0	0.0	0.0
12_{shared}	5366.0	116.0	21.8	10474.5	0.0	0.0	0.0
24	5365.0	116.6	33.1	9969.3	0.0	0.0	0.0
24_{shared}	5365.0	116.0	34.7	9984.1	0.0	0.0	0.0

T_5^{NSFNET}							
$Threads$	Sol_{Cost}	Sol_{Slot}	Sol_{Time}	$Iterations$	$InvalidSA$	$Greedy_{unf}$	LS_{unf}
1	6691.1	153.8	26.4	5630.0	0.0	0.0	0.0
12	6691.1	152.8	34.5	29342.5	0.0	0.0	0.0
12_{shared}	6689.3	152.6	29.5	28942.5	0.0	0.0	0.0
24	6690.1	154.0	18.5	28025.0	0.0	0.0	0.0
24_{shared}	6689.7	154.2	21.3	27750.4	0.0	0.0	0.0

T_6^{NSFNET}							
$Threads$	Sol_{Cost}	Sol_{Slot}	Sol_{Time}	$Iterations$	$InvalidSA$	$Greedy_{unf}$	LS_{unf}
1	7339.8	154.2	34.6	6641.9	0.0	0.0	0.0
12	7333.0	153.4	34.3	35219.0	0.0	0.0	0.0
12_{shared}	7336.3	152.8	28.7	34891.0	0.0	0.0	0.0
24	7334.5	154.2	30.4	34091.2	0.0	0.0	0.0
24_{shared}	7334.8	152.2	24.4	33073.2	0.0	0.0	0.0

T_7^{NSFNET}							
$Threads$	Sol_{Cost}	Sol_{Slot}	Sol_{Time}	$Iterations$	$InvalidSA$	$Greedy_{unf}$	LS_{unf}
1	7674.5	159.7	33.1	797.1	0.0	797.1	44.8
12	7646.5	159.8	30.0	6011.6	0.0	6011.6	364.0
12_{shared}	7648.4	159.7	30.9	5812.2	0.0	5812.2	340.3
24	7652.6	159.5	30.2	5514.5	0.0	5514.5	318.8
24_{shared}	7651.0	159.7	13.8	5426.5	0.0	5426.5	313.2

T_8^{NSFNET}							
$Threads$	Sol_{Cost}	Sol_{Slot}	Sol_{Time}	$Iterations$	$InvalidSA$	$Greedy_{unf}$	LS_{unf}
1	8225.7	158.6	36.4	3062.0	0.0	245.6	0.0
12	8213.5	159.2	31.7	23313.8	0.0	1894.9	0.0
12_{shared}	8211.3	158.2	28.3	24598.6	0.0	2023.1	0.0
24	8212.9	158.2	32.9	20510.0	0.0	1660.3	0.0
24_{shared}	8214.7	158.8	32.4	20488.9	0.0	1680.8	0.0

T_9^{NSFNET}							
$Threads$	Sol_{Cost}	Sol_{Slot}	Sol_{Time}	$Iterations$	$InvalidSA$	$Greedy_{unf}$	LS_{unf}
1	9743.0	160.0	36.2	701.8	0.0	701.8	1.0
12	9721.1	160.0	32.9	5928.5	0.0	5928.5	4.0
12_{shared}	9713.4	160.0	29.3	5906.0	0.0	5906.0	5.3
24	9698.8	160.0	30.2	5559.9	0.0	5559.9	4.1
24_{shared}	9706.8	160.0	43.0	5502.0	0.0	5502.0	4.9

B.2 Impact of Candidate Paths in the Algorithm

The following results were used to measure the impact of the number of candidate paths initially chosen and used by the algorithm. Four couples of values were used:

- $K_{fis} = 10$ and $K_{log} = 5$;
- $K_{fis} = 20$ and $K_{log} = 5$;
- $K_{fis} = 10$ and $K_{log} = 20$.
- $K_{fis} = 20$ and $K_{log} = 20$;

Again, the values summarized next represent average values of 10 runs of 1 minute each, using 12 threads with shared memory.

T_1^{EON}							
(K_{fis}, K_{log})	Sol_{Cost}	Sol_{Slot}	Sol_{Time}	$Iterations$	$InvalidSA$	$Greedy_{unf}$	LS_{unf}
$(10, 5)$	6696.6	105.4	32.3	10032.5	0.0	0.0	0.0
$(20, 5)$	6696.9	104.8	24.5	9865.9	0.0	0.0	0.0
$(10, 20)$	6682.6	105.8	34.0	9970.0	0.0	0.0	0.0
$(20, 20)$	6682.1	107.4	29.3	9776.7	0.0	0.0	0.0

T_2^{EON}							
(K_{fis}, K_{log})	Sol_{Cost}	Sol_{Slot}	Sol_{Time}	$Iterations$	$InvalidSA$	$Greedy_{unf}$	LS_{unf}
$(10, 5)$	7835.9	102.2	29.9	21378.7	0.0	0.0	0.0
$(20, 5)$	7833.1	99.0	28.7	22458.1	0.0	0.0	0.0
$(10, 20)$	7816.5	105.4	31.0	17614.4	0.0	0.0	0.0
$(20, 20)$	7819.6	101.4	30.7	18446.7	0.0	0.0	0.0

T_3^{EON}							
(K_{fis}, K_{log})	Sol_{Cost}	Sol_{Slot}	Sol_{Time}	$Iterations$	$InvalidSA$	$Greedy_{unf}$	LS_{unf}
$(10, 5)$	7833.8	94.6	33.5	35467.4	0.0	0.0	0.0
$(20, 5)$	7831.0	93.0	29.9	35126.6	0.0	0.0	0.0
$(10, 20)$	7816.6	92.0	25.1	28455.8	0.0	0.0	0.0
$(20, 20)$	7816.7	93.8	28.8	28114.2	0.0	0.0	0.0

T_4^{EON}							
(K_{fis}, K_{log})	Sol_{Cost}	Sol_{Slot}	Sol_{Time}	$Iterations$	$InvalidSA$	$Greedy_{unf}$	LS_{unf}
$(10, 5)$	8398.9	136.0	28.5	7418.3	0.0	0.0	0.0
$(20, 5)$	8396.6	135.2	33.5	7394.4	0.0	0.0	0.0
$(10, 20)$	8372.4	136.8	33.2	6541.3	0.0	0.0	0.0
$(20, 20)$	8371.2	136.6	28.5	6485.5	0.0	0.0	0.0

T_5^{EON}							
(K_{fis}, K_{log})	Sol_{Cost}	Sol_{Slot}	Sol_{Time}	$Iterations$	$InvalidSA$	$Greedy_{unf}$	LS_{unf}
$(10, 5)$	9383.7	160.0	28.0	9701.4	0.2	9536.2	0.0
$(20, 5)$	9389.2	159.8	38.2	9705.3	0.3	9175.2	0.0
$(10, 20)$	9343.6	159.4	29.4	8086.5	0.6	7933.2	0.0
$(20, 20)$	9347.0	159.7	25.5	8695.5	0.7	8189.8	0.0

T_6^{EON}							
(K_{fis}, K_{log})	Sol_{Cost}	Sol_{Slot}	Sol_{Time}	$Iterations$	$InvalidSA$	$Greedy_{unf}$	LS_{unf}
$(10, 5)$	10055.1	158.8	41.4	20493.7	0.0	12479.2	0.0
$(20, 5)$	10053.2	158.6	27.7	20554.4	0.0	10348.1	0.0
$(10, 20)$	10003.2	159.6	34.3	15308.4	0.0	9390.2	0.0
$(20, 20)$	10001.9	159.2	24.5	15154.4	0.0	7565.9	0.0

T_7^{EON}							
(K_{fis}, K_{log})	Sol_{Cost}	Sol_{Slot}	Sol_{Time}	$Iterations$	$InvalidSA$	$Greedy_{unf}$	LS_{unf}
$(10, 5)$	11350.2	160.0	26.7	3591.1	0.0	3591.1	3111.4
$(20, 5)$	11361.4	160.0	39.6	3531.1	0.1	3531.1	3015.8
$(10, 20)$	11313.1	160.0	29.9	3437.3	0.0	3437.3	2738.8
$(20, 20)$	11320.0	160.0	40.1	3385.7	0.0	3385.7	2657.4

T_8^{EON}							
(K_{fis}, K_{log})	Sol_{Cost}	Sol_{Slot}	Sol_{Time}	$Iterations$	$InvalidSA$	$Greedy_{unf}$	LS_{unf}
$(10, 5)$	11700.8	139.4	30.6	8695.8	0.1	0.0	0.0
$(20, 5)$	11697.8	139.2	33.4	8930.2	0.1	0.0	0.0
$(10, 20)$	11637.5	138.4	26.6	7041.5	0.2	0.0	0.0
$(20, 20)$	11635.0	137.4	34.6	7081.0	0.0	0.0	0.0

T_9^{EON}							
(K_{fis}, K_{log})	Sol_{Cost}	Sol_{Slot}	Sol_{Time}	$Iterations$	$InvalidSA$	$Greedy_{unf}$	LS_{unf}
$(10, 5)$	13721.1	160.0	29.4	4861.2	4.4	4861.2	1.1
$(20, 5)$	13573.5	159.8	30.6	7657.0	26.1	7657.0	0.4
$(10, 20)$	13654.6	160.0	36.3	4285.0	6.3	4285.0	0.2
$(20, 20)$	13507.0	159.7	29.3	6653.2	25.5	6653.2	0.2

T_1^{GBN}							
(K_{fis}, K_{log})	Sol_{Cost}	Sol_{Slot}	Sol_{Time}	$Iterations$	$InvalidSA$	$Greedy_{unf}$	LS_{unf}
$(10, 5)$	4749.6	123.6	21.6	15082.7	4.2	0.0	0.0
$(20, 5)$	4748.8	128.0	23.0	16729.7	5.2	0.0	0.0
$(10, 20)$	4730.0	128.4	35.9	14257.6	2.8	0.0	0.0
$(20, 20)$	4730.0	127.4	30.7	14664.9	2.7	0.0	0.0

T_2^{GBN}							
(K_{fis}, K_{log})	Sol_{Cost}	Sol_{Slot}	Sol_{Time}	$Iterations$	$InvalidSA$	$Greedy_{unf}$	LS_{unf}
$(10, 5)$	5189.0	154.8	27.5	39343.9	287.7	676.9	30.6
$(20, 5)$	5189.2	153.8	16.9	39130.8	227.0	1058.6	67.1
$(10, 20)$	5179.2	156.4	27.4	39426.3	148.7	599.1	25.0
$(20, 20)$	5180.0	155.8	33.4	39609.7	247.6	975.6	54.9

T_3^{GBN}							
(K_{fis}, K_{log})	Sol_{Cost}	Sol_{Slot}	Sol_{Time}	$Iterations$	$InvalidSA$	$Greedy_{unf}$	LS_{unf}
$(10, 5)$	5580.0	123.6	32.2	46798.6	0.7	0.0	0.0
$(20, 5)$	5580.0	125.8	28.7	52375.9	2.3	0.0	0.0
$(10, 20)$	5566.8	129.6	22.7	47035.0	1.1	0.0	0.0
$(20, 20)$	5566.8	135.2	27.1	48184.1	1.9	0.0	0.0

T_4^{GBN}							
(K_{fis}, K_{log})	Sol_{Cost}	Sol_{Slot}	Sol_{Time}	$Iterations$	$InvalidSA$	$Greedy_{unf}$	LS_{unf}
$(10, 5)$	5715.4	158.0	23.7	10372.1	335.6	4621.1	347.7
$(20, 5)$	5718.2	159.0	25.6	10172.9	810.8	5153.8	648.5
$(10, 20)$	5701.0	158.6	28.8	10072.6	304.4	4098.9	201.8
$(20, 20)$	5701.6	157.2	26.5	10146.4	484.4	4756.7	377.4

T_5^{GBN}							
(K_{fis}, K_{log})	Sol_{Cost}	Sol_{Slot}	Sol_{Time}	$Iterations$	$InvalidSA$	$Greedy_{unf}$	LS_{unf}
$(10, 5)$	6253.2	158.4	30.1	17062.4	175.4	11916.5	4797.0
$(20, 5)$	6257.2	159.6	27.8	14533.7	240.3	10765.9	6167.2
$(10, 20)$	6244.4	157.8	31.2	17307.8	133.5	11718.0	4634.7
$(20, 20)$	6246.0	158.6	34.3	15217.5	217.9	10961.4	6180.0

T_6^{GBN}							
(K_{fis}, K_{log})	Sol_{Cost}	Sol_{Slot}	Sol_{Time}	$Iterations$	$InvalidSA$	$Greedy_{unf}$	LS_{unf}
$(10, 5)$	6716.8	153.4	22.6	27773.7	45.8	2.1	0.0
$(20, 5)$	6716.8	157.2	30.3	27400.3	63.4	17.7	0.3
$(10, 20)$	6706.8	152.8	24.9	28409.2	38.3	2.1	0.0
$(20, 20)$	6706.8	157.2	27.9	28801.6	56.2	12.7	0.2

T_7^{GBN}							
(K_{fis}, K_{log})	Sol_{Cost}	Sol_{Slot}	Sol_{Time}	$Iterations$	$InvalidSA$	$Greedy_{unf}$	LS_{unf}
$(10, 5)$	6173.0	159.6	21.1	3683.6	70.0	3683.6	3305.3
$(20, 5)$	6178.8	159.4	39.1	3204.9	104.4	3204.9	2846.8
$(10, 20)$	6074.4	159.0	32.8	3955.1	140.1	3955.1	2664.5
$(20, 20)$	6081.2	159.2	20.0	3469.9	164.9	3469.8	2363.5

T_8^{GBN}							
(K_{fis}, K_{log})	Sol_{Cost}	Sol_{Slot}	Sol_{Time}	$Iterations$	$InvalidSA$	$Greedy_{unf}$	LS_{unf}
$(10, 5)$	5428.0	126.2	18.2	9387.6	0.0	0.0	0.0
$(20, 5)$	5421.4	129.2	33.0	10434.5	0.0	0.0	0.0
$(10, 20)$	5254.8	125.2	31.0	9086.0	0.0	0.0	0.0
$(20, 20)$	5251.8	127.8	38.2	9075.6	0.0	0.0	0.0

T_9^{GBN}							
(K_{fis}, K_{log})	Sol_{Cost}	Sol_{Slot}	Sol_{Time}	$Iterations$	$InvalidSA$	$Greedy_{unf}$	LS_{unf}
$(10, 5)$	5531.2	135.0	17.4	11488.7	0.4	0.0	0.0
$(20, 5)$	5529.2	134.2	30.1	11526.4	1.0	0.0	0.0
$(10, 20)$	5369.0	128.6	29.5	10181.8	0.2	0.0	0.0
$(20, 20)$	5369.8	137.0	32.2	9997.2	0.7	0.0	0.0

T_1^{GEANT2}							
(K_{fis}, K_{log})	Sol_{Cost}	Sol_{Slot}	Sol_{Time}	$Iterations$	$InvalidSA$	$Greedy_{unf}$	LS_{unf}
$(10, 5)$	7500.9	130.4	34.8	3421.2	0.0	0.0	0.0
$(20, 5)$	7501.3	129.6	21.2	3374.0	0.0	0.0	0.0
$(10, 20)$	7378.5	129.6	37.0	2417.8	0.0	0.0	0.0
$(20, 20)$	7380.4	130.4	27.4	2432.0	0.0	0.0	0.0

T_2^{GEANT2}							
(K_{fis}, K_{log})	Sol_{Cost}	Sol_{Slot}	Sol_{Time}	$Iterations$	$InvalidSA$	$Greedy_{unf}$	LS_{unf}
$(10, 5)$	7665.4	154.6	28.2	5174.2	8.0	1707.4	0.0
$(20, 5)$	7666.0	148.2	25.9	5244.8	8.9	0.0	0.0
$(10, 20)$	7569.7	153.6	40.5	4110.2	6.1	1056.3	0.0
$(20, 20)$	7579.6	145.6	29.7	4148.4	6.8	0.0	0.0

T_3^{GEANT2}							
(K_{fis}, K_{log})	Sol_{Cost}	Sol_{Slot}	Sol_{Time}	$Iterations$	$InvalidSA$	$Greedy_{unf}$	LS_{unf}
$(10, 5)$	8803.6	158.4	30.6	2344.3	4.4	2344.3	0.1
$(20, 5)$	8616.5	152.0	32.4	5052.5	8.0	543.8	0.0
$(10, 20)$	8680.0	157.8	30.8	2033.0	4.8	2033.0	0.0
$(20, 20)$	8513.7	151.0	27.8	3869.6	3.8	175.3	0.0

T_4^{GEANT2}							
(K_{fis}, K_{log})	Sol_{Cost}	Sol_{Slot}	Sol_{Time}	$Iterations$	$InvalidSA$	$Greedy_{unf}$	LS_{unf}
$(10, 5)$	9631.8	160.0	29.8	498.4	8.0	498.4	134.3
$(20, 5)$	9405.1	160.0	31.0	454.8	13.4	454.8	2.2
$(10, 20)$	9369.2	160.0	36.3	366.7	7.5	366.7	11.9
$(20, 20)$	9184.6	159.6	41.6	343.8	11.5	343.8	0.1

T_5^{GEANT2}							
(K_{fis}, K_{log})	Sol_{Cost}	Sol_{Slot}	Sol_{Time}	$Iterations$	$InvalidSA$	$Greedy_{unf}$	LS_{unf}
$(10, 5)$	9263.9	159.0	27.7	785.1	13.9	785.1	0.0
$(20, 5)$	9172.3	151.0	34.3	1026.4	3.4	61.5	0.0
$(10, 20)$	9100.1	158.6	20.4	516.2	9.1	516.2	0.0
$(20, 20)$	9008.9	152.4	33.3	592.2	2.7	20.0	0.0

T_6^{GEANT2}							
(K_{fis}, K_{log})	Sol_{Cost}	Sol_{Slot}	Sol_{Time}	$Iterations$	$InvalidSA$	$Greedy_{unf}$	LS_{unf}
$(10, 5)$	10250.1	159.6	24.2	647.0	236.6	647.0	0.0
$(20, 5)$	10149.7	159.8	34.2	916.9	527.3	763.9	0.0
$(10, 20)$	10062.0	159.8	23.0	363.2	114.0	363.2	0.0
$(20, 20)$	9954.4	159.6	22.9	385.6	143.3	217.5	0.0

T_7^{GEANT2}							
(K_{fis}, K_{log})	Sol_{Cost}	Sol_{Slot}	Sol_{Time}	$Iterations$	$InvalidSA$	$Greedy_{unf}$	LS_{unf}
$(10, 5)$	10525.6	160.0	43.1	54.1	16.9	54.1	0.7
$(20, 5)$	10277.2	159.4	39.5	77.3	36.8	77.3	0.0
$(10, 20)$	10363.7	160.0	39.6	49.0	12.3	49.0	0.3
$(20, 20)$	10137.0	159.2	45.5	60.9	30.5	60.6	0.0

T_8^{GEANT2}							
(K_{fis}, K_{log})	Sol_{Cost}	Sol_{Slot}	Sol_{Time}	$Iterations$	$InvalidSA$	$Greedy_{unf}$	LS_{unf}
$(10, 5)$	11142.2	160.0	53.6	88.0	19.0	88.0	65.8
$(20, 5)$	11034.3	159.0	38.0	58.6	41.0	58.6	0.6
$(10, 20)$	10914.9	159.6	38.2	69.4	26.4	69.4	32.1
$(20, 20)$	10830.5	159.4	44.7	54.4	36.2	54.4	0.3

T_1^{NSFNET}							
(K_{fis}, K_{log})	Sol_{Cost}	Sol_{Slot}	Sol_{Time}	$Iterations$	$InvalidSA$	$Greedy_{unf}$	LS_{unf}
$(10, 5)$	4440.0	84.0	4.4	18350.3	0.0	0.0	0.0
$(20, 5)$	4440.0	84.0	3.5	18178.1	0.0	0.0	0.0
$(10, 20)$	4440.0	84.0	6.9	15748.1	0.0	0.0	0.0
$(20, 20)$	4440.0	84.0	8.0	16149.4	0.0	0.0	0.0

T_2^{NSFNET}							
(K_{fis}, K_{log})	Sol_{Cost}	Sol_{Slot}	Sol_{Time}	$Iterations$	$InvalidSA$	$Greedy_{unf}$	LS_{unf}
$(10, 5)$	4901.0	116.0	15.3	48467.1	0.0	0.0	0.0
$(20, 5)$	4901.0	116.0	11.1	52628.3	0.0	0.0	0.0
$(10, 20)$	4881.9	117.4	14.1	35599.1	0.0	0.0	0.0
$(20, 20)$	4882.2	117.8	21.2	36223.5	0.0	0.0	0.0

T_3^{NSFNET}							
(K_{fis}, K_{log})	Sol_{Cost}	Sol_{Slot}	Sol_{Time}	$Iterations$	$InvalidSA$	$Greedy_{unf}$	LS_{unf}
$(10, 5)$	5516.0	160.0	0.4	83041.8	0.0	0.0	0.0
$(20, 5)$	5516.0	160.0	0.2	83515.5	0.0	0.0	0.0
$(10, 20)$	5489.9	160.0	23.1	75634.6	0.0	0.0	0.0
$(20, 20)$	5491.4	160.0	21.1	74426.1	0.0	0.0	0.0

T_4^{NSFNET}							
(K_{fis}, K_{log})	Sol_{Cost}	Sol_{Slot}	Sol_{Time}	$Iterations$	$InvalidSA$	$Greedy_{unf}$	LS_{unf}
$(10, 5)$	5376.0	116.0	25.3	11474.2	0.0	0.0	0.0
$(20, 5)$	5377.3	116.0	32.3	11274.7	0.0	0.0	0.0
$(10, 20)$	5366.1	116.0	19.9	9838.5	0.0	0.0	0.0
$(20, 20)$	5366.0	116.0	21.8	10474.5	0.0	0.0	0.0

T_5^{NSFNET}							
(K_{fis}, K_{log})	Sol_{Cost}	Sol_{Slot}	Sol_{Time}	$Iterations$	$InvalidSA$	$Greedy_{unf}$	LS_{unf}
$(10, 5)$	6715.8	152.6	21.5	44893.1	0.0	0.0	0.0
$(20, 5)$	6715.0	153.0	31.0	43881.9	0.0	0.0	0.0
$(10, 20)$	6688.6	154.0	13.2	28712.9	0.0	0.0	0.0
$(20, 20)$	6689.3	152.6	29.5	28942.5	0.0	0.0	0.0

T_6^{NSFNET}							
(K_{fis}, K_{log})	Sol_{Cost}	Sol_{Slot}	Sol_{Time}	$Iterations$	$InvalidSA$	$Greedy_{unf}$	LS_{unf}
$(10, 5)$	7367.0	153.4	23.7	39359.8	0.0	0.0	0.0
$(20, 5)$	7366.8	155.2	11.1	36448.1	0.0	0.0	0.0
$(10, 20)$	7334.6	153.6	27.3	36443.6	0.0	0.0	0.0
$(20, 20)$	7336.3	152.8	28.7	34891.0	0.0	0.0	0.0

T_7^{NSFNET}							
(K_{fis}, K_{log})	Sol_{Cost}	Sol_{Slot}	Sol_{Time}	$Iterations$	$InvalidSA$	$Greedy_{unf}$	LS_{unf}
$(10, 5)$	7681.1	159.1	28.9	6840.3	0.0	6840.3	637.5
$(20, 5)$	7667.0	159.5	30.1	6828.8	0.0	6828.8	655.2
$(10, 20)$	7649.0	159.7	29.0	5983.7	0.0	5983.7	349.3
$(20, 20)$	7648.4	159.7	30.9	5812.2	0.0	5812.2	340.3

T_8^{NSFNET}							
(K_{fis}, K_{log})	Sol_{Cost}	Sol_{Slot}	Sol_{Time}	$Iterations$	$InvalidSA$	$Greedy_{unf}$	LS_{unf}
$(10, 5)$	8270.5	158.0	38.3	33381.2	0.1	3050.0	0.0
$(20, 5)$	8271.5	157.0	38.0	34551.3	0.0	3139.5	0.0
$(10, 20)$	8215.0	159.0	24.4	22897.5	0.0	1894.9	0.0
$(20, 20)$	8211.3	158.2	28.3	24598.6	0.0	2023.1	0.0

T_9^{NSFNET}							
(K_{fis}, K_{log})	Sol_{Cost}	Sol_{Slot}	Sol_{Time}	$Iterations$	$InvalidSA$	$Greedy_{unf}$	LS_{unf}
$(10, 5)$	9751.9	160.0	22.7	6304.1	0.0	6304.1	15.5
$(20, 5)$	9775.1	160.0	37.6	5929.7	0.0	5929.7	8.9
$(10, 20)$	9685.3	160.0	38.2	6111.7	0.0	6111.7	13.9
$(20, 20)$	9713.4	160.0	29.3	5906.0	0.0	5906.0	5.3