Meta Heuristic Algorithms for Routing and Spectrum Assignment in Elastic Optical Networks

Michel Mbougni

A Thesis
in
The Department
of
Computer Science and Software Engineering

Presented in Partial Fulfillment of the Requirements
for the Degree of
Master of Science (Computer Science) at
Concordia University
Montréal, Québec, Canada

December 2018

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis prepared

By: **Michel Mbougni**

Entitled: **Meta Heuristic Algorithms for Routing and Spectrum Assignment in Elastic Optical Networks**

and submitted in partial fulfillment of the requirements for the degree of

Master of Science (Computer Science)

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

Signed by the Final Examining Committee:

______________________________ Chair
Dr.

______________________________ Examiner
Dr. Hovhannes Harutyunyan

______________________________ Examiner
Dr. Tristan Glatard

______________________________ Supervisor
Dr. Brigite Jaumard

Approved by ______________________________
Lata Narayanan, Chair
Department of Computer Science and Software Engineering

__________ 2019

____
Amir Asif, Dean
Faculty of Engineering and Computer Science

Abstract

Meta Heuristic Algorithms for Routing and Spectrum Assignment in Elastic Optical Networks

Michel Mbougni

The day to day improvement of bandwidth hungry applications such as video streaming, video gaming and many more has lead to more stringent requirements on networks. These requirements have generated a significant shift from the traditional fixed-grid Wavelength Division Multiplexing (WDM) networks to flexible-grid Elastic Optical networks (EONs). Flexible-grid EONs have brought in new ways for allocating spectrum in an efficient manner. Unfortunately, they have also brought in new challenges with respect to spectrum allocation. It is much more complex to grant demands in flexible-grid EONs, as they take into account more constraints than traditional fixed-grid WDM networks. Despite the effort that has been made by the research community to handle those constraints separately, very little has been done tackling them simultaneously, and under realistic scenarios. In this thesis, we propose two meta-heuristic algorithms for allocating demands in flexible-grid EONs while simultaneously taking into accounts all the constraints. Our algorithms are tested on small and large networks, with heavy load. Experimental results show that our algorithms perform quite well on all the instances that were selected.

Acknowledgments

My sincere thanks and gratitude goes to the Almighty God for everything life offered on my academic path right up to this moment and for all what life still has to offer. Thanks to God for His plan in my life.

I am deeply indebted to my supervisor, Professor B. Jaumard for her guidance, continued support, encouragement, analytical but explicitly meaningful comments and critiques to this thesis. I am humble and greatly inspired by her enthusiasm.

I would also like to extent my gratitude to my family. Thank you for your unfailing love, encouragement and support.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview

Optical networks are essential means of communication today due to the fact that they use the optical fiber for transmission. The use of optical fiber technology instead of copper cable provides abundant bandwidth for modern traffic provisioning. They also enable low loss transmission. The coming of the Wavelength Division Multiplexing (WDM) technology for optical networks has enabled multiple wavelengths (spaced to avoid interference) to be combined into a single fiber. This technology tremendously increases the network capacity. Nowadays, the bit rate per wavelength is in the order of 10, 40, 100 and 400 Gb/s but it is foreseen that we could have a rate of 1 Tb/s in the near future [38]. Optical networks enabled by the light technology also use the benefit of routing and switching in the optical domain. This leads to more flexible optical networks, and enable a single infrastructure to be used for the deployment of many services [35].

Due to their high traffic rates, optical networks are seen as a solution for the growth of traffic demands experienced nowadays. The growth in traffic demands emanate from a significant improvement in the development of bandwidth hungry applications; e.g., data, voice/video streaming, gaming. Traditional Wavelength Multiplexing optical networks with fixed grid leads to inefficient spectrum usage and spectrum management. Over the past few years a flexible and spectrum efficient elastic optical networks (EONs) have generated a lot of interest [5]. They are based on orthogonal frequency division multiplexing (OFDM) technology. EONs enable a flexible assignment of appropriate-sized bandwidth using contiguous concatenation of optical spectrum slots to given traffic demands [20].

1.2 Motivation

The advance of EONs that enables the spectrum to be divided into slots has brought new complexity to the provisioning of demands in such networks. In addition to constraints that are specific to provisioning demands in WDM networks, EONs have other constraints. Among them, there are the slot contiguity, modulation and regenerator constraints. Most of the algorithms or models that are proposed in literature to provision demands in EONs do not take into accounts all these constraints simultaneously. Furthermore, they are tested on small network instances and under low traffic. In this thesis, we propose metaheuristic algorithms to grant demands in EONs while taking into consideration all the above constraints simultaneously, and under realistic scenarios, i.e., on networks containing up to 100 nodes and traffic with an offered load up to 280 Tbps.

1.3 Thesis Contributions

In this thesis, we developed two metaheuristic algorithms for routing and assigning spectrum in EONs. Both algorithms allow the search of the solution to be done in the infeasible search space as a diversification technique. The first algorithm is inspired from the application of graph coloring to the routing and wavelength assignment problem in WDM networks as proposed in [27]. Our graph coloring algorithm first allows all the demands to be granted; allowing the algorithm to enter the infeasible state. It then repetitively alternate the path, modulation and slot allocations to reduce the number of conflicts, until the solution is back to the feasible state. The second metaheuristic we propose has two main phases, the optimization phase and the feasibility phase. In the optimization phase, we exchange the slots between demands while tentatively moving from one exit the local optimum to the next while still remaining in the feasible search space. The feasibility phase is used to diversify the search space by allowing more regenerators than the number required. One regenerator is then iteratively removed until the solution become feasible. To the best of our knowledge, there are no algorithms (heuristics, metaheuristics or exact methods) which were able to provision demands with an offered load of up to 280 Tbps on large networks (with up to 100 nodes).

1.4 Statement of the RSMA Problem

An optical network can be modeled as an undirected graph $G = (V, L)$ where V is the set of nodes (indexed by v) and L denotes the set of links (indexed by ℓ). Spectrum is made of a set S where the frequency of each slot s is 12.5 GHz. Let K be the set of requests (indexed by k), with D_k being the bandwidth requirement for k, expressed in Gigabits per second (Gbps). Let M be the set of modulations (indexed by m). The Routing Modulation and Spectrum Assignment (RMSA) problem can be formally stated as follows: choose a modulation for each request k in order to maximize the throughput, while using no more than n^{REGEN} regenerators. The output should provide a slot-path for each request, which satisfies both the continuity and the contiguity constraints. For each request, contiguous slots from a portion of the spectrum should be assigned in other to satisfy the spectrum contiguity constraint. The same contiguous slots assigned for a request should also be assigned on each link along the path used to grant the request in other to satisfy the spectrum continuity constraint.

1.5 Organization of Thesis

The organization of this thesis is as follows. Chapter 2 presents the general background and a literature review on related subjects. It first describes optical networks with a focus on WDM and EON. After a short description on the general operation of tabu search algorithms, the RSA problem and the RSMA problems are described.

Chapter 3 presents a greedy algorithm we developed for the RSMA problem.

In Chapter 4, we first state the classical and generalized graph coloring problems. We then propose a generalized graph coloring algorithm for the RMSA problem.

Chapter 5 presents a tabu search algorithm for the RMSA problem. We detail the main procedures used in the algorithm.

Chapter 6 conducts a numerical analysis on the performance of the designed algorithms in the previous chapters, as well as the characteristics of the solutions.

Chapter 7 presents the conclusions that were made in this thesis as well as some future lines of research.

Chapter 2

Background and Literature Review

In this chapter, we present some preliminaries about traditional WDM networks as well as EONs. We further define and describe some key concepts in the routing and spectrum assignment problem. We then present a literature review associated with the RSA problem as well as the RMSA problem.

2.1 Optical Networks

In this section, WDM networks and EON basic concepts are introduced and briefly discussed.

2.1.1 Wavelength Division Multiplexing Networks

The introduction of the WDM technologies enables today networks to transmit up to 160 wavelengths per fiber [10]. An example of an uni-directional WDM network is shown in Figure 2.1. Each transmitter uses a laser to send signals at different wavelengths. These signals then enter the multiplexer in different input ports, are combined and then output on a single port. From that port, a single fiber is connected. An optical amplifier can be used to boost the transmitted power. Depending on the type of fiber and the length of the

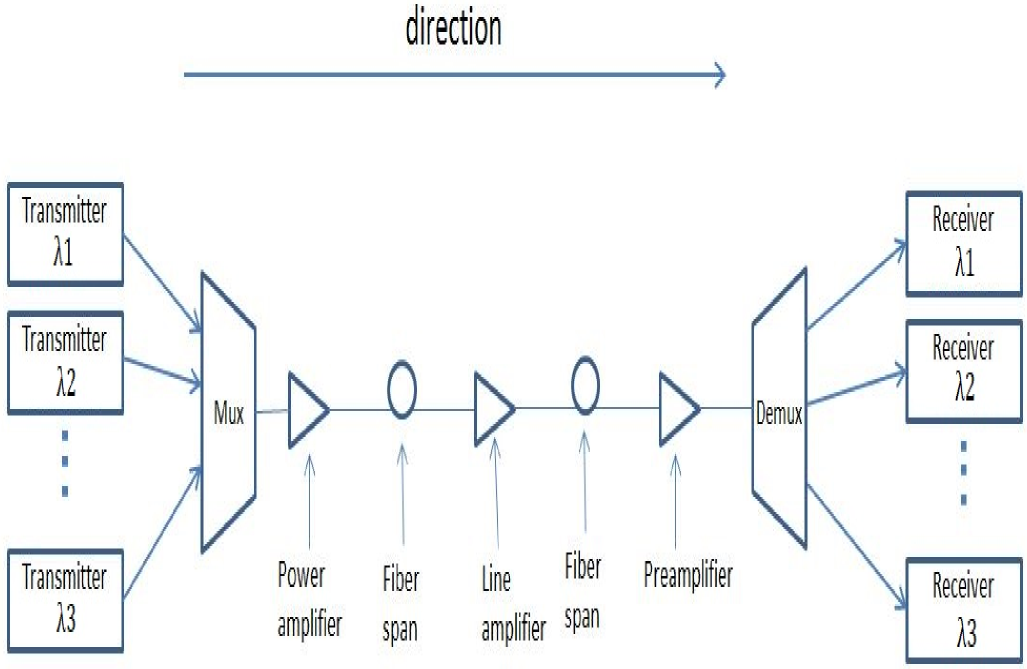


Figure 2.1: A Simple one Way Wavelength Division Multiplexing Network

fiber span, a line amplifier may be used to regenerate the signal. At the receiving side, the signal can be amplified before it goes into the demultiplexer on a single input port. The demultiplexer splits the signal into different signals at different wavelengths, with each wavelength going to a specific port. Each wavelength is then received by a photodetector of each receiver. In WDM networks, the optical spectrum is divided into fixed channels. As a result, spectrum is significantly wasted on demands with small granularity. Networks based on the WDM technology are therefore not seen as good candidates for current and next generation bandwidth hungry applications.

2.1.2 Elastic Optical Networks

EONs have been seen as an alternative of WDM networks for support of higher traffic demands. EONs are based on Orthogonal Frequency Division Multiplexing (OFDM)

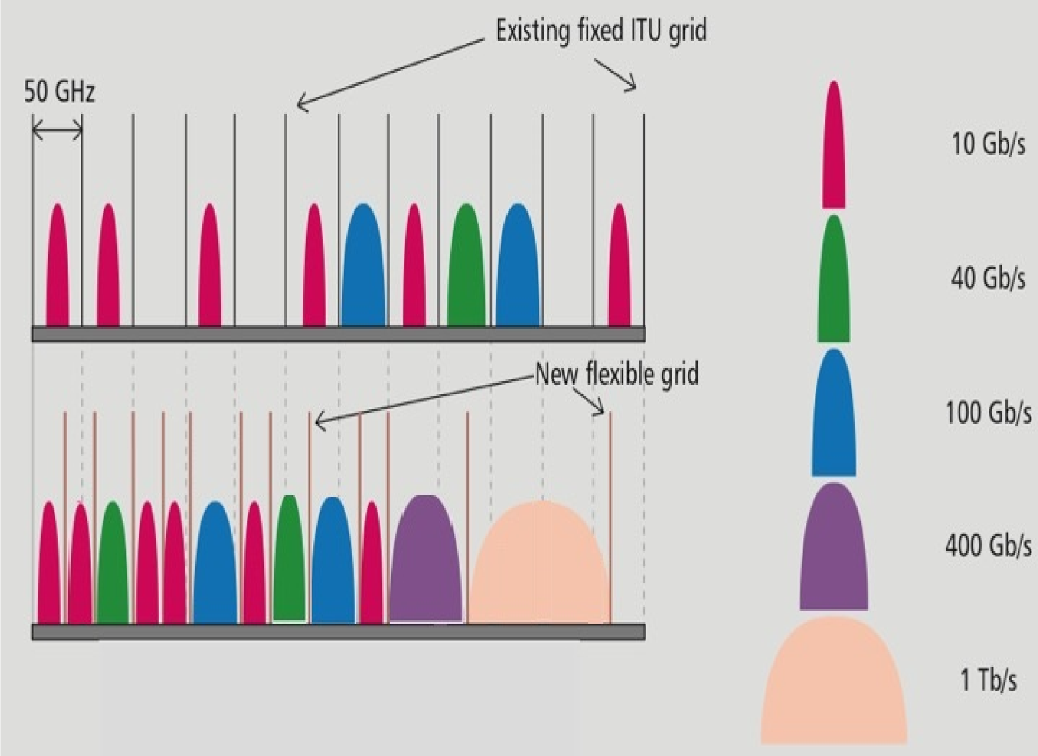


Figure 2.2: Fixed and Flexible grids [17]

technologies. Unlike WDM networks, they split optical spectrum into finer ”chunk” called slots spaced at 6.25 GHz or 12.5 GHZ [31]. Adjacent slots can be combined to form ”super channels” that can support traffic demand of up to 1 Tbs. An illustration of a fixed grid used by WDM networks versus flexible grids used by EONs is shown in Figure 2.2.

As shown in Figure 2.2, a connection that needs 10 Gbps, and another that needs 100 Gbps will use the same channel space using a fixed grid. On the other hand, channel space is assigned based on the connection data rate while using flexible grid. It can be seen that EONs use the optical spectrum in a more efficient and flexible way as compared to WDM networks. It makes them therefore more suitable to support current and future bandwidth hungry applications.

The key advantages of EONs include but are not limited to [1, 5]:

- Bandwidth segmentation: EONs allocate only the bandwidth needed from a traffic demand. This characteristic reduces the amount of spectrum that is wasted or unused.

- Bandwidth aggregation: EONs have the capability of combining optical bandwidth; hence creating a ”super channel optical path”.
- Energy saving: EONs provide energy efficient mechanisms that reduce the power consumption by turning off some of the OFDM subcarriers that are not utilized.

The provisioning problem of EONs, where a set of requests are provisioned using a suitable routing path and the number of OFDM subcarriers is known as the RSA problem.

2.2 Tabu Search

The RMSA problem is a combinatorial optimization problem that is known to be NP-hard [20]. For this reason, the problem can only be solved using exact algorithms on small instances. In practice, most combinatorial optimization problems are on large instances. They can therefore not be solved using exact algorithms; that is the reason why heuristics and meta-heuristics are usually used to solve large scaled combinatorial optimization problems. Genetic algorithms, simulated annealing and tabu search are the best known meta heuristic algorithms [22]. In theory, meta heuristic algorithms can be used to solve any combinatorial optimization problem; however, finding the right values of their tuning parameters is not trivial [22]. Genetic algorithms and simulated annealing both use randomization to approximate diversification. Genetic algorithm achieve that by joining population elements and applying crossover or mutation [13]. Simulated annealing makes random move to diversify a function of a temperature whose progressive decrease also reduces the variation away from global optimum with a given probability [13].

In this thesis, the tabu search algorithms we proposed are inspired by the original tabu search procedure as described in [18]. The approach attempts to leave the local optimal solution space using a strategy that forbids certain moves. The moves that are forbidden are called ”tabu” and they are used to avoid cycling. The tabu search approach might at an early state lead to a local optimum; nevertheless, that does not interfere or disrupt the search process since it can still select the best available move; or it might permit non-improving solution to be allowed in order to escape from a local optimum. To avoid taking a path that was previously taken, tabu search keeps track of moves recently made by storing them in a tabu list. The tabu list prevents some moves to be used for a certain number of iterations and therefore preventing cycling. Tabu search main steps as explained by [19] are shown in Figure 2.3. The steps usually start with an initial solution obtained by greedy algorithms or other heuristics. The next step is to create a list of moves where, each move creates a new solution from the current solution. The best admissible candidate is then chosen based on the tabu restrictions such as the tabu list size. The solution obtained from the best admissible candidate is stored as the new best solution if it improves the previous best solution; it is also label as the new current solution. The stopping criterion steps is basically checking if a number of iterations initially chosen have be reached. If that is the case, then based on the application, it either stops and stores the best current solution as the final solution, or tune some parameters (i.e. tabu list size) before repeating the same process. A more detail explanation of tabu search steps can be found in [19]. Tabu search main advantages are the following:

- it speeds up the search process by discouraging the use of random selection
- it allows both downhill moves and uphill moves
- it select best admissible candidates
- it can applied to both discrete and continuous solution spaces.

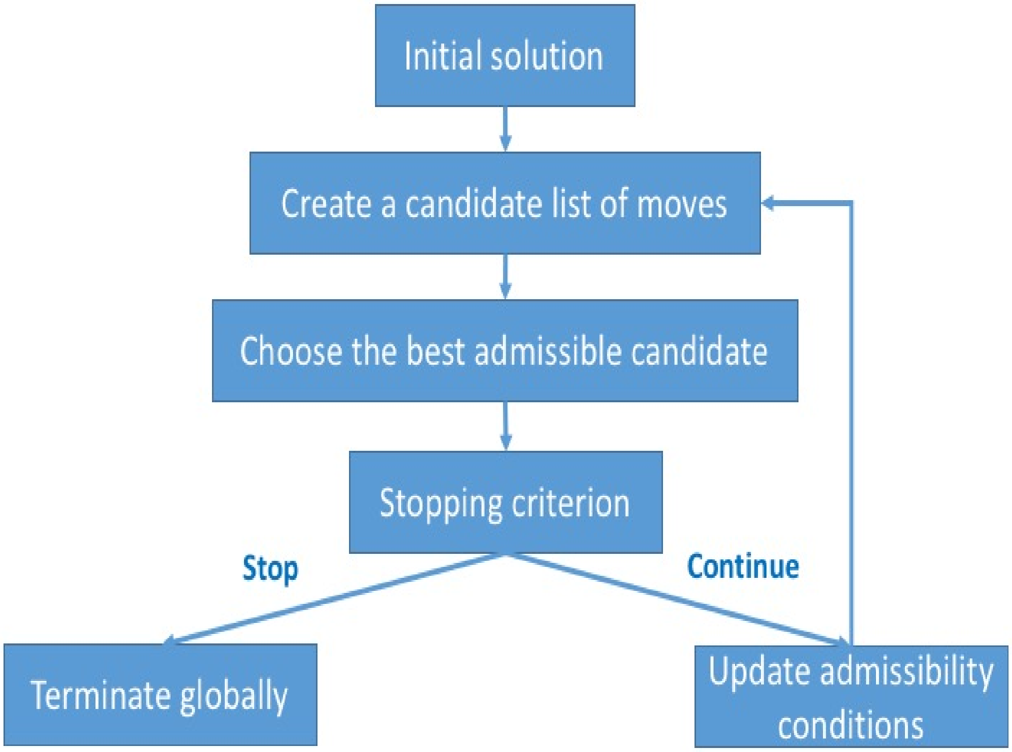


Figure 2.3: Tabu Search Steps

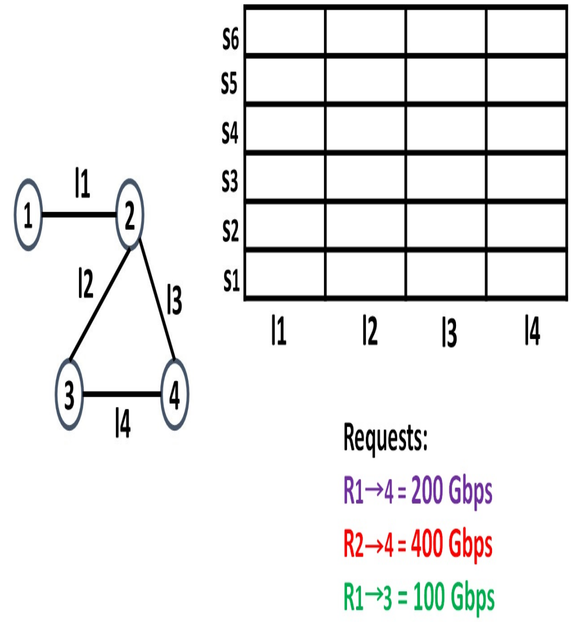
2.3 Routing and Spectrum Assignment Problem

2.3.1 Generalities

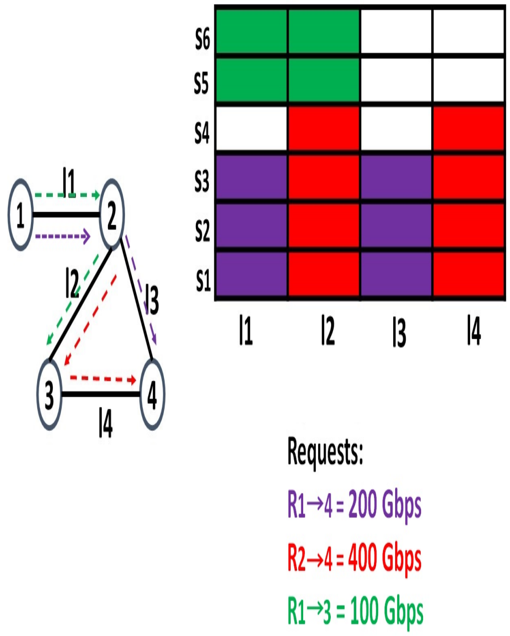
The RSA problem formulated for EONs is quite similar to the RWA problem formulated for WDM networks [40]. The RWA problem consists of choosing a routing path, and a wavelength for each given request while satisfying the continuity constraint. The continuity constraint is satisfied by ensuring that the same wavelength is used on all the links in the routing path for a given request. In addition to the continuity constraint, the RSA problem must also satisfy a contiguity constraint. The contiguity constraint is satisfied by guaranteeing that slots are contiguous on each link in the routing path for a request that need to be provisioned. An example of RSA is illustrated in Figure 2.4. Figure 2.4(a) shows a network, a grid associated with it, and the traffic demand in terms of requests that need to be provisioned. The RSA is usually visualized as grid where the horizontal axis represents the links and the vertical axis represents the slots or vice versa. We assume in our example that nothing has been provisioned initially; so the grid is empty. We assume for simplicity that the request $R_{1\to4}$ needs 3 slots to be provisioned; the requests $R_{2\to4}$ and $R_{1\to4}$ respectively need 4 and 2 slots to be provisioned. A possible grid allocation after provisioning of the given requests using RSA is shown in Figure 2.4(b). In this allocation, the request $R_{1\to4}$ uses links l_1 and l_3; the request $R_{2\to4}$ uses links l_2 and l_4 and the request $R_{1\to3}$ uses links l_1 and l_2. It is shown in Figure 2.4(b) that the continuity and contiguity constraints have been satisfied in this simple example.

2.3.2 Routing Modulation and Spectrum Assignment

Modulations have an important role in optical networks as modulated signals are the signals that hold information to be transmitted from the transmitter to the receiver. Modulation formats are used as parameters to establish the spectral resources to be allocated for an optical path [28]. There is a direct relationship between the data rates, the modulations and the reach [33, 41]. In WDM networks, the transmission is done in such a way that for a given data rate, the modulation format is chosen so that the signal can be transmitted with adequate quality along the worst path; that is the path with the most hops or the longest length. So all the requests for that given rate are assigned the same modulation format regardless of their path length. In EONs on the other hand, the modulation is based on the distance between the source and destination of each request. In other words, requests with the same data rate can be granted using different modulation formats depending on their path lengths [39]. The RSA that takes into account modulation is called the routing modulation and spectrum assignment (RMSA). It uses modulation formats to give a trade off between the transmission reach and the spectrum efficiency. A high level modulation format has a shorter reach but uses fewer slots as compared to low level modulation format



(a) Before Provisioning



(b) After Provisioning

Figure 2.4: RSA Example

that uses more slots but has a longer transmission reach [39, 40].

2.4 Literature Review

In this section, we first present some of the work that has been done to solve the RSA problem. Secondly, we discuss some of the research that has been done on a variation of the RSA problem called the routing modulation and spectrum assignment (RMSA) problem. Finally, we do a critical analysis on the literature review.

2.4.1 RSA Problem

Exact Algorithms

Jaumard and Daryalal [25] proposed an optimization model based on the column generation decomposition method to solve RSA problem. The objective of their model is to maximize the number of granted requests. Their decomposition scheme uses link and path formulations to guarantee that an optimal solution is obtained. The authors evaluated their model on a Spain network (21 nodes) against the model proposed by Ruiz *et al.*[36]. The results showed that Jaumard and Daryalal model outperforms the model proposed by Ruiz *et al.* [36].

Enoch and Jaumard [15] developed another decomposition model for solving the RSA problem. The authors used a column generation model with lightpaths. They used a shortest path algorithm to efficiently solve the lightpaths generator and to significantly improve the quality of the solution. The authors used the Spain (21 nodes) and the USA (24 nodes) networks to evaluate their model. The results showed that Enoch and Jaumard model is faster than the model proposed by Jaumard and Daryalal [25].

Heuristic Algorithms

A lot of work has been done in context of RSA for EONs. In this section, we present some of the work that has been done on RSA problem for EONs.

Alaskar *et al.* [1] proposed four offline RSA algorithms for EONs. The authors modeled the offline RSA problem as a multiple processor scheduling problem. In other words, they mapped the multiprocessor scheduling problem where the objective is to minimize the total processing time used by all the tasks into an offline RSA problem where the objective is to minimize the spectrum usage for a traffic demand. The algorithms proposed by Alaskar *et al.* are based on two well known algorithms namely the compact scheduling algorithm and the left edge algorithm. The first algorithm they proposed is called the longest then widest compact algorithm (LWC). It uses two sorting levels for requests for a given traffic demand. The first sorting level consists of sorting requests in a decreasing order based on the spectrum required for each request. The second level of sorting break the ties from the first sorting level by sorting in a decreasing order requests based on the number of links used by each request. The second algorithm proposed area compact algorithm (AC) by author first compute the area obtained by multiplying the spectrum required by each request as well as its corresponding number of links required. The requests are then sorted in an decreasing order based on the area of each request. The third algorithm namely the longest then left edge compact (LLEC) algorithm proposed by Alaskar *et. al* consisted of two sorting level as well where the first sorting level is identical to the longest then widest compact algorithm (LWC). However the second sorting level break the ties from the first level by sorting using the left edge algorithm. The author finally proposed the area then left edge compact (ALEC) algorithm by using the AC algorithm as the first sorting level and the left edge algorithm to break the ties (second sorting level) on request that have the same area. The authors evaluated their algorithms using simulation on two network topologies: a chained network and the NSFNET network. They compared their algorithms against two algorithms namely the longest first compact (LFC) algorithm and widest first compact (WFC) algorithm. The results showed that LWC, AC, LLEC, and ALEC algorithms all outperformed LFC and WFC algorithms in terms of average ratio; where the average ratio is ratio of the maximum number of subcarriers used in any link by the number of subcarriers used by the most loaded link.

Shirazipourazad *et al.* [37] proposed algorithms for the RSA problem. The authors developed approximation algorithms for binary tree and ring network topologies. They also proposed an heuristic for an arbitrary network topology. The main idea of their heuristic is that it tries to find disjoint paths for routing the requests, to optimize the reuse of slots in spectrum allocation. The algorithms they proposed were tested using simulation on the NSFnet network under varying load ranging from 5 to 25 requests. The authors had better results in terms of average spectrum usage than the Shortest Path with Maximum Spectrum Reuse and Balanced Load Spectrum Allocation heuristics.

Cai *et al.* [4] proposed a greedy heuristic and a multi iteration based algorithm for the RSA problem. The multi-iteration algorithm was mainly based on shuffling sequences in which the requests were allocated. The authors tested their algorithms using different network topologies with nodes ranging from 6 to 14 and links ranging from 9 to 21. The authors compared their algorithms against an ILP model on networks of small size having $(6, 10, 11)$ nodes respectively; and the results showed that their multi-iteration algorithms has a quality close to optimal. Cai *et. al* also compared their results against two heuristics namely: Maximum Spectrum Reuse and Balanced Load Spectrum Allocation on a NSFNET network. The results showed that their greedy heuristic and multi-iteration heuristic outperform in terms of spectrum usage, the Maximum Spectrum Reuse and Balanced Load Spectrum Allocation heuristics.

Duran *et al.* [14] proposed two heuristics for solving the RSA problem namely: the least unusable spectrum first (LUSF) and the adaptive unconstrained routing exhaustive Spectrum Search (AUR-ESS). The LUSF algorithm computes for each candidate solution, the total number of unused slots per link used for routing path of that solution. The solutions are then sorted increasingly based on the total number of unused slots. The first fit heuristic is used to break the ties. The candidate solutions are then evaluated in order until a feasible one can be applied. The AUR-ESS algorithm they proposed is based on the AUR algorithm used to solve the RWA problem. It first checks at each of the steps, the number of slots that are available. The next step consists of building an auxiliary graph; The nodes in the auxiliary graph corresponds to nodes in the physical network. There is a link between two nodes of the auxiliary graph if all the slots in the fiber connecting those two nodes in the physical network are not currently used by any other request. The authors then use the Dijkstra algorithm to find a shortest path. They finally grant the request on the shortest path found in the auxiliary graph. The authors evaluated their algorithm on NSFNet network (14 nodes). The results shows that their algorithm has a lower blocking probability than the traditional RWA heuristics such as first fit, most used, and least used.

Fadini [16] proposed a subcarrier-slot partition scheme with first-last fit spectrum allocation for EONs. The objective of their scheme was to increase the number of free contiguous slots; therefore reducing the blocking probability in the network. Their partition scheme makes room for more aligned available slots by splitting the allocation of the connections that are not disjoint. They evaluated their scheme on a network of 28 nodes and the results showed that their scheme has a lower blocking probabilities as compared to other spectrum allocation algorithms.

Meta Heuristics Algorithms

Hai [21] developed a multi-objective genetic algorithm for solving the RSA problem. The algorithm he proposed found the pareto optimal solutions (set of solutions that are equally optimal) for the RSA with multiple objectives. The objectives that were taken into account where: to minimize the spectrum width requirement to grant all requests and to minimize the overall spectrum link usage. The author evaluated his algorithm on a network of 11 nodes. He argued that his algorithm can converge to optimal solution after 200 iterations.

2.4.2 RMSA Problem

Exact Algorithms

Dharmaweera *et al.* [12] proposed an ILP model and a heuristic algorithm for solving the RMSA problem. They used a physical layer impairment model to capture EON impairments. The objective of their ILP model was to minimize the bandwidth, which the authors quantify as the highest subcarrier index assigned to a request. The heuristic algorithm they proposed has two steps. The first step consists of choosing the sites where regenerators are placed. The second step consists of assigning spectrum and modulation format for each request on each segment of the candidate routing path selected. They evaluated their algorithms on networks of different node sizes; that is networks with 6, 14 and 28 nodes. The authors showed that their algorithm increased spectral efficiency.

Heuristic Algorithms

Kadu *et al.* [29] developed a Modulation-aware Multipath Routing and Spectrum Allocation (MMRSA) algorithm to solve the RMSA problem in EONs. Their algorithm grants requests using a minimum number of paths with distance adaptive modulation methods. Their algorithm starts by selecting multiple potential routing paths starting with the shortest. It then tries to find the minimum number of routing paths that can be used to grant the traffic. The requests are then granted using the first fit method. They evaluated their algorithm on a network with 26 nodes and 41 links. The results from simulation showed that the algorithm proposed by Kadu *et al.* has a higher spectrum usage and a lower blocking rate as compared to algorithms proposed by Chen *et al.* [6].

Meta Heuristics Algorithms

Goscien *et al.* [20] proposed a tabu search algorithm for solving the RMSA (Routing, Modulation and Spectrum Assignment) problem in EONs with unicast and anycast traffic demands. They firstly presented ILP formulation for RMSAJAU (Routing, Modulation and Spectrum Assignment for Joint Unicast and Anycast). The objective functions related to the authors ILP formulations took into account: minimizing the network deployment cost, minimizing the energy consumption, minimizing the maximum spectrum usage, and minimizing the average spectrum usage. The authors solve each of the ILP formulations using a tabu search algorithm. The initial solution to the tabu search algorithm is computed using a greedy heuristic, namely the Adaptive Routing, Modulation and Spectrum Assignment (ARMSA). The author evaluated their algorithm using simulation on three different networks namely: NFS15, Euro16 and DT14. For each network topology and each objective function, the authors compared their algorithms to three other heuristics. The results showed that their algorithm outperforms the three other heuristics for each network topology and each objective function from the ILP formulations.

Christodoulopoulos *et al.* [9] proposed a greedy heuristic and simulated annealing meta heuristics for Routing Modulation Level and Spectrum Assignment. They used simulated annealing to obtain a good ordering of demands and then used the heuristics to grant all the demand based on the pre-ordering done by simulated annealing. The simulated annealing they proposed used the most subcarriers first (MFS) ordering of demands as its initial solution. The authors tested their algorithms using simulations under two network topologies of 6 and 14 nodes with 9 and 23 links respectively. The authors compared their proposed algorithms against an ILP model on the network with 6 nodes and reported that their solution is near optimal in terms of spectrum usage. They furthered compared their algorithms on the network with 14 nodes against heuristics such as MSF and longest path first (LPF) and showed that their algorithms provide a better solution.

Zhou *et al.* [43] developed a genetic algorithm to solve the dynamic RMSA problem. Their algorithm is designed for multi objectives optimization where each objective is based on the current state of the network. Under light traffic with no blocking, the objective of their algorithm is to minimize the maximum number of slots used by any fiber. Under heavy traffic, their algorithm objective is to minimize the blocking probability. They compared the performance of their algorithm against existing algorithms such as Shortest Path and First Fit Spectrum Assignment, and the K-Shortest Paths and Balanced Load Spectrum Assignment on the NSFNET (14 nodes) and US Backbone (28 nodes) networks. The results show that algorithm proposed by Zhou *et al.* [43] has a lower blocking probability than the existing algorithms against which they compared under the same load condition. The results also show that the genetic algorithm the authors proposed has a lower number of slots used per fiber under light load conditions as compared to the algorithms that were used for comparisons.

Klinkowski and Walkowiak [30] developed an ILP formulation to solve the routing, spectrum, transceiver, and regeneration allocation (RSTRA) problem which is a problem that has traditional RSA problem characteristics, with the addition that regenerators placement optimization are taken into account. The authors developed an algorithm (greedy lightpath allocation (GLA) and a standard simulated annealing algorithm) to generate solutions to their IPL formulation. Their GLA algorithm assigns requests one at a time based on a given order and assign them to routing paths in such a way that each assignment minimizes the cost function of ILP formulation the authors proposed. The order of the requests are then optimized using simulated annealing. The authors evaluated their algorithm on the German (12 nodes), European (28 nodes), and US (26 nodes) networks. The results from experiments showed that adaptable regeneration and modulation conversion can improve spectrum and transceivers usage in long-distance networks.

2.4.3 Critical Analysis

As discussed earlier, a lot of work has been done to solve the RSA problem. The authors in [25, 15] proposed exact algorithms; however, their algorithms are not practical for real networks with large instances. The heuristics proposed by the authors in [1, 4, 37, 14, 16] and the meta-heuristic proposed by [21] do not take into account modulation. Modulation has been taken into account by the authors in [29, 20, 9, 43, 30]. Nevertheless, the algorithms they proposed are applied on networks of small size (maximum of 28 nodes). In this thesis, we proposed meta-heuristics for solving the RMSA problem on large networks (up to 100 nodes) and under heavy traffic conditions (up to 280 Tbps).

Chapter 3

Greedy Algorithm

In this chapter, we present a greedy heuristic for the RSMA problem. This algorithm has been used as an initial solution for the meta heuristic algorithms we developed.

3.1 Greedy Heuristic for Routing Modulation and Spectrum Assignment

3.1.1 Notations

Let us first define some notations that will be used in this chapter and subsequent chapters.

G: graph such that $G = (V, L)$.

V: set of nodes indexed by v

L: set of links indexed by l

P^{sd}: set of paths with source-destination pair $\{v_s, v_d\}$

S^{AVAIL}: set of available slots.

S^{USED}: set of used slots.

K: set of requests indexed by k.

M: set of modulations indexed by m.

S: maximum allowed number of slots.

n^{REGEN}: maximum allowed number of regenerators.

w: integer that represents the number of shortest paths to be considered between a given source-destination pair $\{v_s, v_d\}$.

3.1.2 Greedy Algorithm

The input of our greedy algorithm consists of: G, M, n^{REGEN} and S. The greedy algorithm we propose in this thesis uses the following steps:

Step 1: Compute all the w-shortest paths for each $\{v_s, v_d\} \in \mathcal{SD}$. In this thesis, the w-shortest paths were computed using the Yen's algorithm[42] . It is important to note that a different w is chosen for each source-destination pair. The value of w is selected based on the fraffic between each source-destination pair. A heuristic proposed by Jaumard and Daryalal [26] is used to compute the value of w for each source-destination pair.

Step 2: Sort the requests $k \in K$ in the increasing order of their demands. Break the ties using the length of the shortest path for $\{v_s, v_d\}$, considering first the requests with the shortest path.

Step 3: Initialize S^{AVAIL} and S^{USED} with S and $\emptyset$ respectively; then grant the requests with the largest demands first using the algorithm GRANT_REQUEST(k).

Algorithm 1 works as follows: for each request k passed to the algorithm, we consider all the candidate paths that could be used to grant k. For each path selected, we compute the number of slots that are available. We then select the minimum feasible modulation. If the number of slots used by the modulation selected as well as their sequence is available in the grid, we grant k and update the set of slots used for each link that is used by k.

Algorithm 1 GRANT_REQUEST(k)

```
ISGRANTED ← .false
NO_PATH ← .false
Assume $k$ is between node pair $\{v_s, v_d\}$
while ISGRANTED = .false. or NO_PATH = .false. do
    Consider the next path $p \in P^{sd}$
    if no more path then NO_PATH ← .true.
    end if
    Select the modulation $m$ that uses the minimum number of slots, i.e., $n_k^{\text{SLOTS}}$, while not exceeding $n^{\text{REGEN}}$
    if there is a sequence of $n_k^{\text{SLOTS}}$ in $S^{\text{AVAIL}}$ then
        Grant request $k$ on $p$ using that sequence and modulation $m$
        Update $S^{\text{USED}}$
        $S^{\text{AVAIL}} \leftarrow S^{\text{AVAIL}} \setminus S^{\text{USED}}$
    end if
    ISGRANTED ← .true
end while
```

Chapter 4

Tabu Search: RMSA as a Generalized Graph Coloring Problem

It has been seen in Chapter 2 that the RSA problem for EON has some similarities with the RWA problem for WDM networks. Graph coloring has been successfully applied in the past by Jaumard *et al.* [27] to solve the RWA problem. In this chapter, we propose a generalized graph coloring algorithm for the RMSA problem.

4.1 Classical Graph Coloring Problems

4.1.1 Graph Coloring Problem

The classical graph coloring problem can be defined on an undirected graph G as follows. Let $G = (V, L)$ where V is the set of nodes (indexed by v) and L is the set of edges (indexed by ℓ). The graph coloring problem consists of finding a node coloring with the minimum number of colors such that connected nodes are assigned different colors [38].

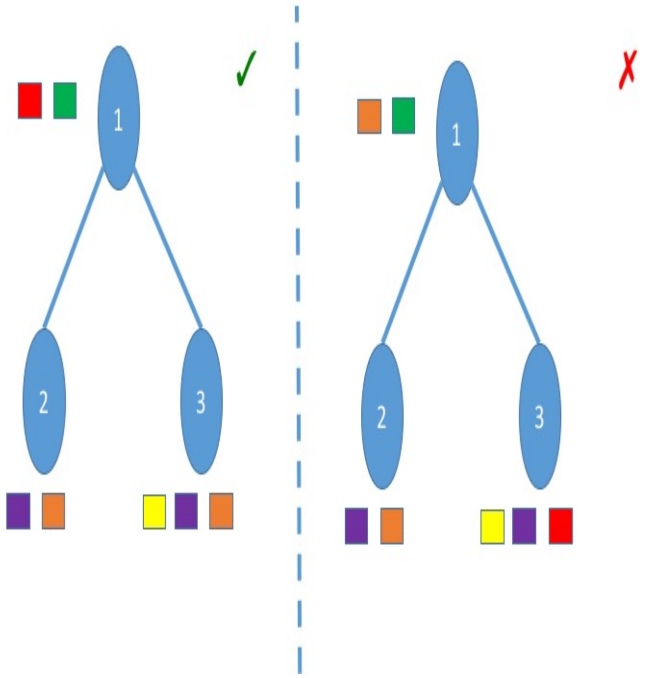


Figure 4.1: GGC Example

4.1.2 Generalized Graph Coloring Problem

We now define the Generalized Graph Coloring (GGC) problem [27] on an undirected graph $G = (V, E)$ as follows. Let C be the set of all possible colors. In the Generalized Graph Coloring Problem, each node v must be assigned a predefined number of colors, defined by $C_v \subseteq C$. Not all nodes are necessarily assigned the same number of colors. A node coloring consists in assigning a subset C_v of colors to each node v such that, if v and v' are connected, $C_v \cap C_{v'} = \emptyset$. The GGC problem amounts then to find a node coloring that minimizes the number of colors, i.e., the size of C. An illustration of a generalized graph coloring is shown in Figure 4.1

4.2 Generalized Graph Coloring for the RMSA problem

In this section we propose a Generalized Graph Coloring algorithm to the RMSA problem.

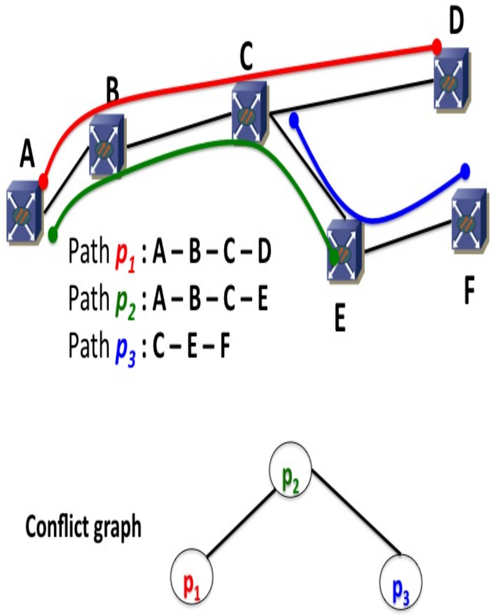


Figure 4.2: Conflict Graph Construction

4.2.1 Definition of the Conflict Graph

We define the conflict graph G as follows. Let $G = (P, E)$, where

$P = \bigcup_{\{v_s, v_d\} \in L} P^{sd}$, and each node $p \in P^{sd}$ is associated with a potential path for a given request k between v_s and v_d in the optical network $\mathcal{N}$, and

E is the set of edges such that, there is an edge between nodes p and p' if they share at least one (optical) link. Each node p is assigned a set of frequency slots, denoted by S_p. An illustration of a conflict graph is constructed is shown in Figure 4.2.

In order to derive a feasible RMSA provisioning, it is required to find a feasible generalized graph coloring for G, taking into account that each color set is defined by a modulation and a set of contiguous frequency slots. Modulation should be selected while taking into account the length of the path, and the presence of regenerators along that path. Number of frequency slots depend on demand D_k of k and on the selected modulation.

Number of nodes in P depends on the number of pre-selecting w-shortest paths, which may vary from one node pair to the next in the optical network.

4.2.2 TS_GGC_RMSA: A Generalized Graph Coloring Algorithm for the Routing, Modulation and Spectrum Assignment Problem

In this section, a generalized coloring algorithm called TS_GGC_RMSA is proposed for the RMSA problem. The initial solution is the solution obtained from the greedy algorithm presented in Chapter 3. Before describing the generalized coloring algorithm, let us first briefly provide some definitions. Let:

ITER$^{\max}$: maximum number of iterations during which we tolerate no improvement.

RMSA$^{\text{best}}$: current best RMSA provisioning with respect to the throughput.

TABU$^{\text{best}}$: best feasible RMSA solution.

$\mathcal{L}^{\text{TABU}}$: forbidden set of moves.

R_p^{con}: set of requests with conflicts on a path p.

RMSA$^{\text{init}}$: initial solution obtained from the greedy algorithm.

K^G: set of granted requests from RMSA$^{\text{init}}$.

K^D: set of denied requests from RMSA$^{\text{init}}$.

ITER: current number of iteration(s).

In the TS_GGC_RMSA algorithm we proposed, we first try to grant all requests by allowing slot-conflicts. We then change the path, modulation and slots selection in order to resolve the conflicts. Finally, if some conflicts still exist, we sequentially remove all the requests that lead to conflicts on the graph in order to get a feasible solution. We repeat those three steps several times, and keep track of the best solution. The pseudo code for the TS_GGC_RMSA algorithm is presented in **Algorithm** 2. The TS_GGC_RMSA algorithm has two main procedures namely COMPUTE-CONFLICTS and REDUCE_CONFLICTS. A more detailed description of those procedures is presented next.

Algorithm 2 TS_GGC_RMSA

```
Input: RMSA$^{\text{init}}$, $L$, $K^D$, $K^G$, $M$, $S$, $n^{\text{REGEN}}$ and $G$
Output: RMSA provisioning and its throughput
ITER $\leftarrow 0$
RMSA$^{\text{best}}$ $\leftarrow$ RMSA$^{\text{init}}$
$\#$REGEN $\leftarrow$ number of regenerators in RMSA$^{\text{init}}$
isConflicted $\leftarrow$ .false.
while some requests are not granted and ITER $<$ ITER$^{\max}$ and $\#$REGEN $< n^{\text{REGEN}}$ do
    for each $\{v_s, v_d\}$ do
        Select the request $k$ between $v_s$ and $v_d$ with the largest demand $D_k$ and that
                has not been granted
        Grant $k$ using $m$ and $n_k^{\text{SLOTS}}$.
    end for
    isConflicted $\leftarrow$ .true.
    Call COMPUTE-CONFLICTS procedure
    Call REDUCE_CONFLICTS procedure
    while isConflicted do
        Select the request $k$ and its associated $p \in P^{sd}$ that has the largest number
                of conflicts;
        Remove traffic associated with request $k$;
        Update CONFLICTS
        if CONFLICTS does not hold any conflicts then
            isConflicted $\leftarrow$ .false.
        end if
    end while
    ITER $\leftarrow$ ITER $+ 1$
    if the throughput of the current solution is better than RMSA$^{\text{best}}$ then
        ITER $\leftarrow 0$
        Update RMSA$^{\text{best}}$ and its related $\#$REGEN
    end if
end while
Update the conflict graph $G$ using RMSA$^{\text{best}}$
```

COMPUTE-CONFLICTS Procedure

The pseudocode of the COMPUTE-CONFLICTS procedure is presented in **Algorithm** 3. This procedure computes for each path p (node in the conflict graph) the number of conflicts with each path p' adjacent to p. The conflicts computed are stored using a double linked lists called CONFLICTS, which is updated easily once a conflict is removed. It is important to notice that conflicts are computed and stored once and they are used to get all the conflicts for each denied request.

Algorithm 3 COMPUTE-CONFLICTS Procedure

```
Input: $G$, $K^D$
Output: CONFLICTS
CONFLICTS $\leftarrow \emptyset$
for each $\{v_s, v_d\}$ do
    for each $p \in P^{sd}$ do
        Compute the number of conflicts with each node $p'$ adjacent to $p$ and
                store it into CONFLICTS
    end for
end for
```

REDUCE_CONFLICTS Procedure

In this subsection, the pseudocode of the REDUCE_CONFLICTS Procedure is presented. The pseudocode as described in **Algorithm** 4 reduces the number of conflicts for each request that has been granted by finding an alternate path that could reduce the largest number of conflicts. Using that alternate path, the denied request is granted using the required modulation and number of slots. We then check if the total number of conflicts has been reduced; if that is the case, we update the current solution and repeat the process, keeping track of the current solution.

Algorithm 4 REDUCE_CONFLICTS Procedure

```
ITER $\leftarrow 0$
TABU$^{\text{best}}$ $\leftarrow$ Current Solution
while ITER $<$ ITER$^{\max}$ do
    for each $\{v_s, v_d\}$ do
        Select a path that has a conflict
        Select an alternate path and a modulation that uses the minimum number $\#$
            slots that could reduce the largest number of conflicts
            denoted by $\#Con_r^{\text{reduce}}$
    end for
    Choose the request $r$ with largest $\#Con_r^{\text{reduce}}$, and grant it on $p_r^{\text{select}}$ with
            $m_r^{\text{select}}$ and $s_r^{\text{select}}$
    Put $p$ into $\mathcal{L}^{\text{TABU}}$, and update $\mathcal{L}^{\text{TABU}}$
    ITER $\leftarrow$ ITER $+ 1$
    if the global number of conflicts is reduced then
        ITER $\leftarrow 0$
        Update TABU$^{\text{best}}$ with information of the granted request $r$
    end if
end while
Update current solution with TABU$^{\text{best}}$
```

Chapter 5

Tabu Search : Best Insertion

In this chapter, we propose an algorithm called best insertion algorithm (BI_RMSA) for RMSA problem.

5.1 BI_RMSA: Best Insertion Algorithm

In the BI_RMSA algorithm proposed, the current solution is not always feasible. In fact, the algorithm alternates between feasible and infeasible phases. The idea is to diversify the solution space by searching both the feasible and infeasible search spaces. The goal is, after obtaining an infeasible solution, the algorithm gets back to a feasible solution space by removing some regenerators based on some given choice(s). The algorithm iteratively (at each step) removes a regenerator associated with a request that uses the minimum number of slots and that uses at least one regenerator. If after removing a regenerator there is still room for insertion, the request is inserted into the grid. If, on the other hand, there is a conflict with another request that uses the minimum number of slices and at least one regenerator, then the request that reduces the least the throughput is removed from the granted list. The process is repeated until the algorithm gets back into the feasible solution space. For each switch, only the feasible solution is updated. Before going into the details of our algorithm, let us first define some more notations and recall some previously defined. Let:

K^{TABU}: set of forbidden requests.

INFREG: maximum number of regenerators used for the infeasible solution spaces.

$\text{ITER}^{\max}$: maximum number of iterations during which we tolerate no improvement

$\text{RMSA}^{\text{init}}$: initial solution obtained from the greedy algorithm.

K^G: set of granted requests from $\text{RMSA}^{\text{init}}$.

K^D: set of denied requests from $\text{RMSA}^{\text{init}}$.

TEMPREG: overall number of regenerator used by BI_RMSA algorithm

The BI_RMSA pseudocode is described in **Algorithm** 5. It is also important to recall that a conflict exist between two requests if their slots usage overlap on at least one link. As mentioned earlier, it consists of two main phases namely: the optimization and feasibility phases. We shall discuss each phase in the following subsections.

Algorithm 5 BI_RMSA

```
Input: initial solution, $G$, $M$, $K^D$, $K^G$, $S$ and $n^{\text{REGEN}}$
Output: RMSA provisioning and its throughput
TEMPREG $\leftarrow n^{\text{REGEN}}$ + INFREG
$\text{RMSA\_sol}^{\text{BEST}}_{\text{FINAL}} \leftarrow 0$
while $n^{\text{REGEN}} \leq$ TEMPREG do
    Call TABU_SEARCH_RMSA procedure
    Call FEASIBILITY procedure
    if $\text{RMSA\_sol}^{\text{BEST}}_{\text{FINAL}} < \text{RMSA\_sol}^{\text{BEST}}$ then
        $\text{RMSA\_sol}^{\text{BEST}}_{\text{FINAL}} \leftarrow \text{RMSA\_sol}^{\text{BEST}}$;
    end if
    Update TEMPREG
end while
```

5.1.1 Optimization Phase

The basic idea in the optimization phase is that, for each request that has not been granted, we try to find its best insertion into the grid. The variation of the throughput is then estimated. The variation of the throughput could either increase or decrease. We keep track of the best increase in throughput. In the case where there is no increase, we select the position that leads to the least decrease in throughput. Even if there is no increase in throughput, the denied request is still inserted into grid. One of the reasons that it helps in exiting a local optimum region of the solution space with a goal that at a later stage, the throughput could be increased by adding one or more requests into the grid. The TABU_SEARCH_RMSA procedure actually exchanges the order in which requests are granted. It allows the solution to enter the infeasible space by increasing the maximum number of regenerators that are allowed to be used in the network. The pseudocode of TABU_SEARCH_RMSA is shown in **Algorithm** 6.

Algorithm 6 TABU_SEARCH_RSA **Procedure**

```
Input: initial solution, $G$, $M$, $K^D$, $K^G$, $S$ and TEMPREG.
Output: RMSA_sol^BEST
ITER $\leftarrow 0$ ;
RMSA_sol $\leftarrow$ initial solution
REG $\leftarrow$ Number of regenerators used by the initial solution.
while ITER $\leq$ ITER$^{\max}$ & $K^D \neq \emptyset$ & REG $\leq$ TEMPREG do
    for all $k \in K^D \setminus K^{\text{TABU}}$ do
        Assume $k$ is a request from $v_s$ to $v_d$
        for all $p \in P^{sd}$ do
            Assuming $k$ is routed on $p$, find best modulation and slot assignment to
                grant $k$, i.e., the one that gives the best value for RMSA_sol($p$)
        end for
        $p^\star \leftarrow \operatorname*{argmin}_{p \in P^{sd}}$ RMSA_sol($p$)
        Route $k$ on $p^\star$
        Compute the new current solution: RMSA_sol
        Update $K^D$ and $K^G$
        Update REG with the number of regenerators used by $k$;
    end for
    ITER $\leftarrow$ ITER $+ 1$
    if If RMSA_sol $\geq$ RMSA_sol^BEST then
        Update RMSA_sol^BEST with RMSA_sol
        ITER $\leftarrow 0$
    end if
end while
```

5.1.2 Feasibility Phase

As mentioned earlier, the optimization phase is not always feasible as it allows the maximum number of regenerators that can be used in the network to exceed; the feasibility phase consists of getting back to the feasible solution space. It does that by removing some regenerators based on a given choice(s). The feasibility procedure iteratively removes a regenerator associated with a request that uses the minimum number of slots and that uses at least one regenerator. If after removing a regenerator, there is enough space in the grid to grant the request, insert into the grid. If on the other hand there is a conflict with another request that uses the minimum number of slices and at least one regenerator, then the request that reduces the less throughput is removed from the granted list. The process is repeated until the solution gets back to the feasible space. For each switch between the feasible and infeasible search spaces, the incumbent solution is updated using the current solution if and only if the current solution is feasible.

Algorithm 7 FEASIBILITY Procedure

```
Input: RMSA_sol$^{\text{BEST}}$, $G$, $M$, $K^D$, $K^G$, $S$ and TEMPREG
Output: RMSA_sol$^{\text{BEST}}$.
Let $count$ be a parameter that keeps track of the number of regenerators removed
$count \leftarrow 0$
while $(\text{TEMPREG} - count) \geq n^{\text{REGEN}}$ do
    Select $k \in K^G$ such that $k$ uses the minimum number of slots and $k$ uses at least one regenerator
    Let $\text{REG}^k$ be the number of regenerator(s) used by $k$
    while $\text{REG}^k > 0$ do
        $\text{REG}^k \leftarrow \text{REG}^k - 1$
        Readjust the number of slots used by $k$ accordingly;
        if $k$ conflict with any other request $k\prime \in K^G$ then
            Let $r_k$ and $r_{k\prime}$ be the traffic demand of requests $k$ and $k\prime$ respectively
            if $r_k < r_{k\prime}$ then
                $K^G \leftarrow K^G \setminus \{k\}$
                RMSA_sol$^{\text{BEST}} \leftarrow$ RMSA_sol$^{\text{BEST}} - r_k$
                $count \leftarrow count + \text{REG}^k + 1;$
                $\text{REG}^k \leftarrow 0;$
            else
                $K^G \leftarrow K^G \setminus \{k\prime\}$
                RMSA_sol$^{\text{BEST}} \leftarrow$ RMSA_sol$^{\text{BEST}} - r_{k\prime}$
                Update $count$ with the number of regenerators used by $k\prime$
            end if
        else
            $count \leftarrow count + 1;$
        end if
    end while
end while
```

Chapter 6

Numerical Results

6.1 Data Sets Characteristics

The experiments were run on different networks, namely: Spain [11], USA [3], Coronet75 [2], Coronet100 [2], Euro28 [8], and Germany50 [8]. The topologies of these networks are displayed in Figures 6.1 to 6.5. For all traffic instances, the number of available slots were set to 400 as in [25]. Requests with data rates of 100Gbps, 200Gbps, and 400Gbps were considered with different proportions. Modulation formats QPSK, 8QAM and 16QAM were also taken into account. A summary of each data rate, corresponding modulation, slots and reach is shown in Table 6.1.

Our experiments were done only with static traffic; that is, all the traffic demands are known in advance. The offered load was uniformly distributed among node pairs, except for a case on the USA network (see Table 6.2 non uniform node pairs), where the offered load is distributed following a traffic model [24]. In other words, the node pair selections, for the offered load distribution were based on population of each node (city). A summary of the data sets we used as well as their characteristics is reported in Table 6.2.

The $w-$shortest paths between source-destination pair nodes were generated using a library [34] that implements an extension of Yen's algorithm [32]. In our experiments, the

Figure 6.1: Spain Network

Data rates	Modulation	# Slots (12.5 GHz)	Reach (Km)
100Gbps	QPSK	4	4000
200Gbps	QPSK	6	2500
	8QAM	4	1200
	16QAM	3	600
400Gbps	16QAM	6	300

Table 6.1: Data rate, Modulation and Reach Relationship [23]

value of w depends on the traffic demand. For a given (v_s, v_d), $w_{sd} \leftarrow D_{sd}/\rho$, where ρ is a constant. The value of ρ was chosen to be 0.5, as proposed in [26]. In the case where we have one hop request, only one possible path exist.

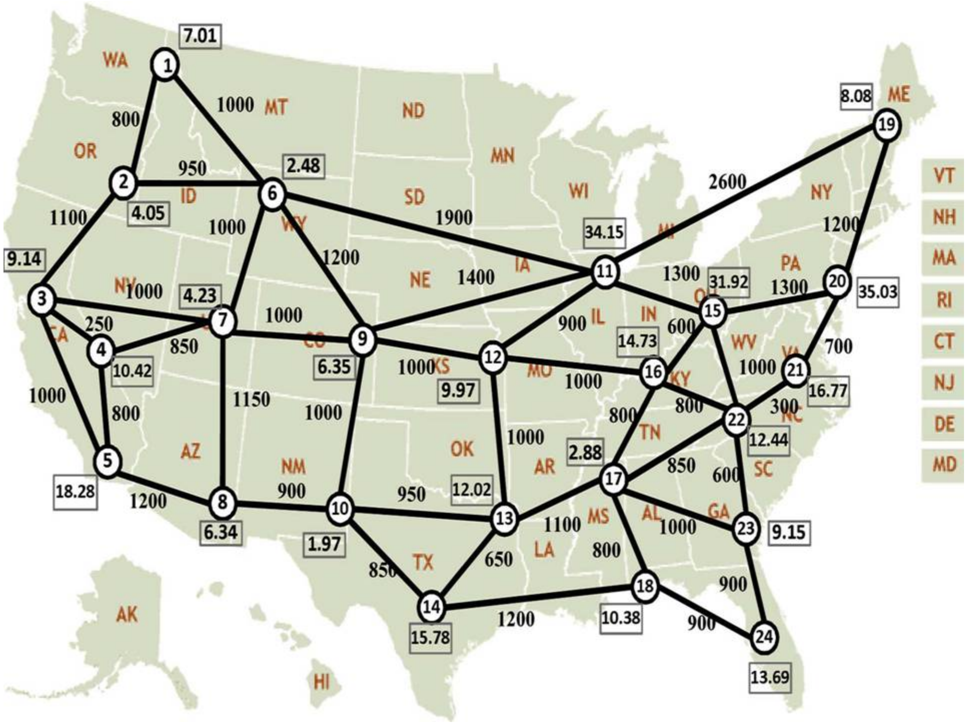


Figure 6.2: USA Network

data sets	Ref.	# nodes	# links	node pairs distribution	Requests Granularity (%)			Offered Load (Gbps)
					100Gbps	200Gbps	400 Gbps	
Spain_1	[11]	19	32	Uniform	70	20	10	280000
Spain_2				Uniform	60	25	15	
Spain_3				Uniform	50	30	20	
USA_1	[3]	24	43	Uniform	70	20	10	215000
USA_2				Uniform	60	25	15	
USA_3				Uniform	50	30	20	
USA_4				Non Uniform	70	20	10	215000
USA_5				Non Uniform	60	25	15	
USA_6				Non Uniform	50	30	20	
Euro28_1	[8]	28	41	Uniform	70	20	10	215000
Euro28_2				Uniform	60	25	15	
Euro28_3				Uniform	50	30	20	
Germany50_1	[8]	50	88	Uniform	70	20	10	220000
Germany50_2				Uniform	60	25	15	
Germany50_3				Uniform	50	30	20	
Coronet75_1	[2]	75	99	Uniform	70	20	10	100000
Coronet75_2				Uniform	60	25	15	
Coronet75_3 3				Uniform	50	30	20	
Coronet100_1	[2]	100	136	Uniform	70	20	10	100000
Coronet100_2				Uniform	60	25	15	
Coronet100_3 3				Uniform	50	30	20	

Table 6.2: Characteristics of the Data Instances

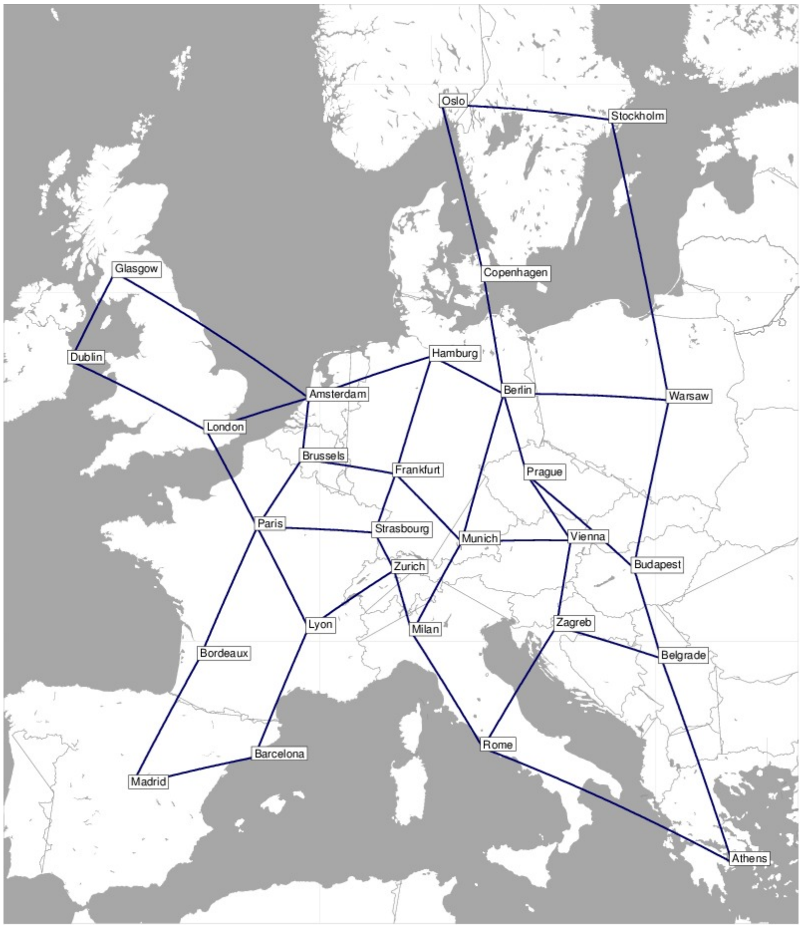


Figure 6.3: Euro28 Network

6.2 Optimization Parameters

In this subsection, we report the optimization parameters that were selected for our tabu search algorithms. The optimization parameters that we selected are: tabu list size and number of iterations. The values that were tested and those that were selected are shown in Table 6.3. The values that were selected are the values from which we could find the best trade off between the throughput and the performance for our algorithms.

Parameter	Values analyzed	Value Selected
Tabu list size	10,25,50,75,100	75
Iterations	30,50,70,100	30

Table 6.3: Optimization Parameter

6.3 Performance Analysis Comparisons

In this section, the performance of the algorithms developed are analyzed. It is important to mention that GC (Graph Coloring) and BI (Best Insertion) are shorter version name of TS_GGC_RMSA and BI_RMSA respectively. The performance metrics we used are:

- Throughput: total traffic that is granted from the offered load.
- CPU time: computational time (in seconds) used to solve the RMSA problem.

All the algorithms are analyzed in terms of throughput. Furthermore, the GC and the BI algorithms are also analyzed in terms of computational time (CPU time). The summary of the results in terms of throughput and CPU time of all the algorithms we proposed can be seen in Table 6.4.

In terms of throughput, it can be seen that the GC algorithm outperforms the greedy algorithm by about 8% on average on networks having less than 30 nodes and by about 16% on average on networks with larger nodes. We can also see that the BI algorithm outperforms the GC algorithms in terms of throughput by about 3% on average. This is due to the fact that BI algorithm uses a better strategy to exit local optimum than the GC. BI has two strategies to exit local optimum, firstly exchanging requests in the optimization phase. Secondly by alternating between feasible and infeasible solution spaces. On the other hand, GC only goes from the infeasible search space (requests with conflicts) to the feasible search space (after requests with conflicts have been removed). The strategy used by the BI therefore explore a wider solution space than GC. That could be the reason why it has a better throughput than the GC algorithm. In terms of computational time, the GC algorithm requires more computational time than BI algorithm to solve the RMSA problem. The GC algorithm takes approximately 1.12 more CPU time than the BI algorithm to solve the RMSA problem.This may due to the fact that, even though GC uses algorithm 3 procedure only in one pass, the procedure itself is quite expensive as it explore all possible nodes from the conflict graph. That affect the overall computational times of the GC algorithm.

It is also important to mention that the effect of the population on the traffic distribution has an impact on the throughput. The instances (USA_1 to USA_3) with uniform traffic distributions have a better throughput than instances (USA_4 to USA_6) with non uniform traffic distributions with requests that have the same data rate proportions. There are about 10%, 7% and 5% increases in throughput in the greedy, GC and BI algorithms respectively from non uniform to uniform instances with data rates of the same proportions. That could be explained by the fact nodes that have a significant population will likely have outgoing links that will be highly loaded; therefore, more requests going through those links will be denied, resulting lower throughput than the with instances that use uniform distributions with the same data rate proportions.

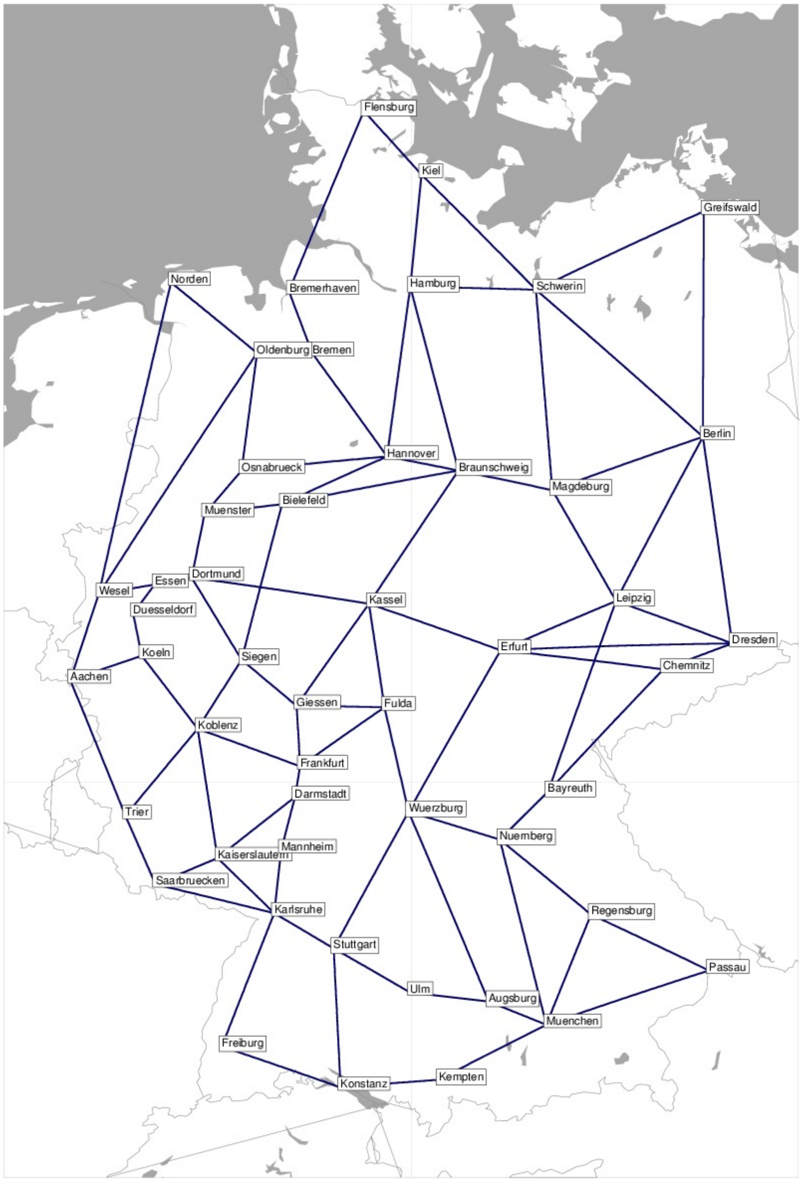


Figure 6.4: Germany Network

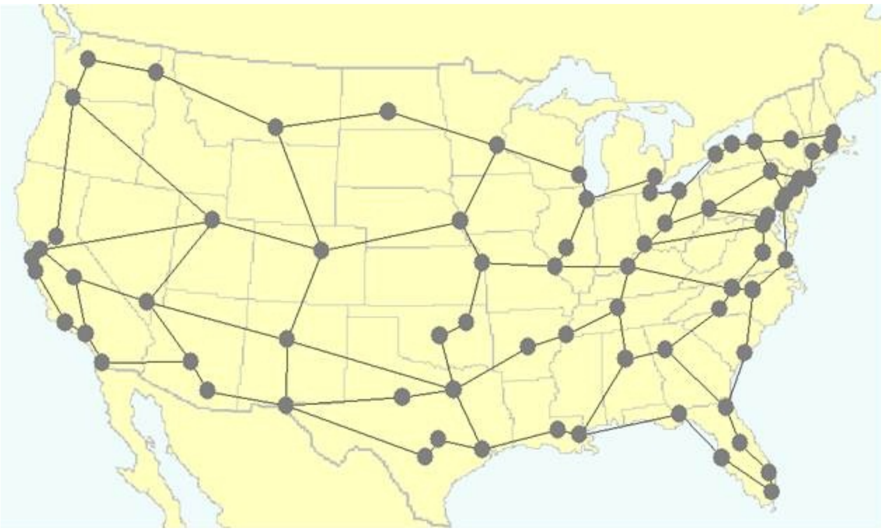


Figure 6.5: Coronet75 Network

Data sets	Offered load (Gbps)	Throughput greedy (Gbps)	Throughput greedy (%)	Throughput GC (Gbps)	Throughput GC (%)	Throughput BI (Gbps)	Throughput BI (%)	#Reg	CPU usage BI	CPU usage GC
Spain_1	280,000	192000	68.57	225,300	80.5	226,700	81.0	50	117	183
Spain_2		199000	71.07	226,600	80.9	227,400	81.2		126	191
Spain_3		198700	70.96	228,700	81.7	230,400	82.3		128	196
USA_1	215,000	164400	76.47	187,100	87.0	187,900	87.4	50	304	319
USA_2		164700	76.60	184,300	85.7	187,200	87.1		315	330
USA_3		163900	76.23	177,000	82.3	179,200	83.3		322	352
USA_4		141300	65.72	168,100	78.2	173,500	80.7		319	342
USA_5		142400	66.23	167,900	78.1	174,300	81.1		346	364
USA_6		144600	67.26	166,600	77.5	175,800	81.8		364	382
Euro28_1	220,000	162200	73.73	175,700	79.9	177,500	80.7	50	408	433
Euro28_2		161200	73.27	173,900	79.0	175,800	79.9		414	447
Euro28_3		160400	72.91	175,200	79.6	176,900	80.4		429	464
Germany50_1	150,000	124100	82.73	125,100	83.4	127,400	84.9	70	623	765
Germany50_2		124600	83.07	129,200	86.1	131,200	87.5		640	790
Germany50_3		125400	83.60	126,400	84.3	130,300	86.9		655	802
Coronet75_1	100,000	78200	78.20	80,200	80.2	82,500	82.5	70	801	918
Coronet75_2		81700	81.70	83,700	83.7	84,200	84.2		822	944
Coronet75_3		84600	84.60	85,800	85.8	87,200	87.2		835	963
Coronet100_1	100,000	65000	65.00	84,800	84.8	86,400	86.4	70	842	979
Coronet100_2		68300	68.30	86,100	86.1	86,700	86.7		857	1012
Coronet100_3		64400	64.40	87,600	87.6	87,900	87.9		877	1020

Table 6.4: Performance Comparison

6.4 Slots Usage

In this section, we present slots usage for some network instances. The slots usage of Euro28_1 and Germany50_1 are displayed in Figure 6.6 and Figure 6.7 respectively. The network details of those instances are shown in Appendix A. From Figure 6.6 and Figure 6.7, it can be seen that the spectrum usage on links of Euro28_1 fluctuates less than the spectrum usage on links of Germany50_1. This can be explained by the topology of Euro28 and Germany50 networks. The Euro28 network topology has a much more regular topology than the Germany50 network topology. This may means that almost all links participate with the same proportion in the computation of k-shortest paths and hence in the provisioning of requests. That is, there are less links that are bottlenecks or that are highly loaded on the Euro28_1 network than on the Germany50_1.

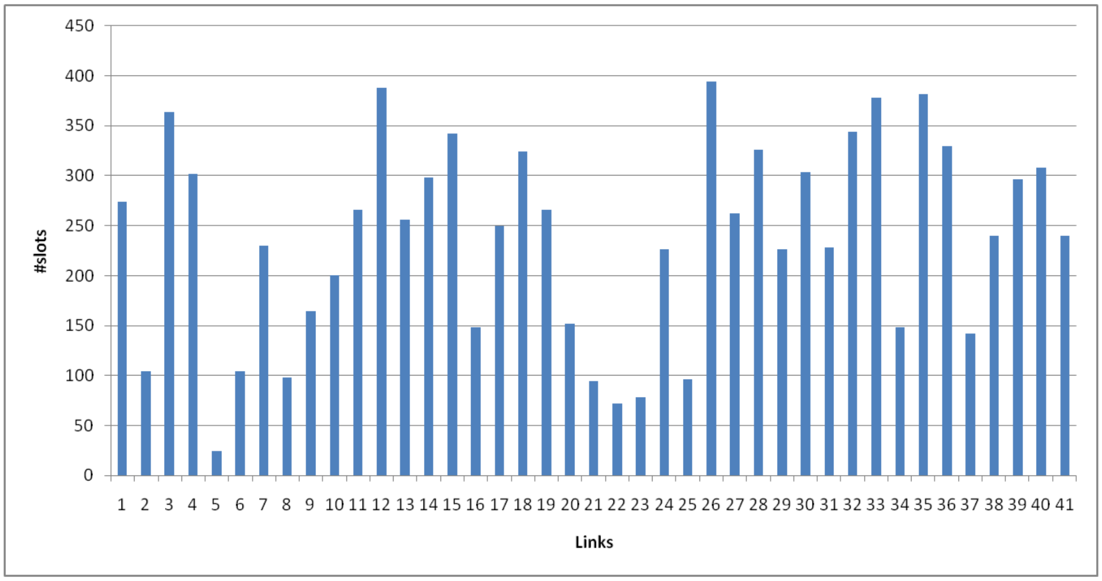


Figure 6.6: Slice Usage Euro28_1

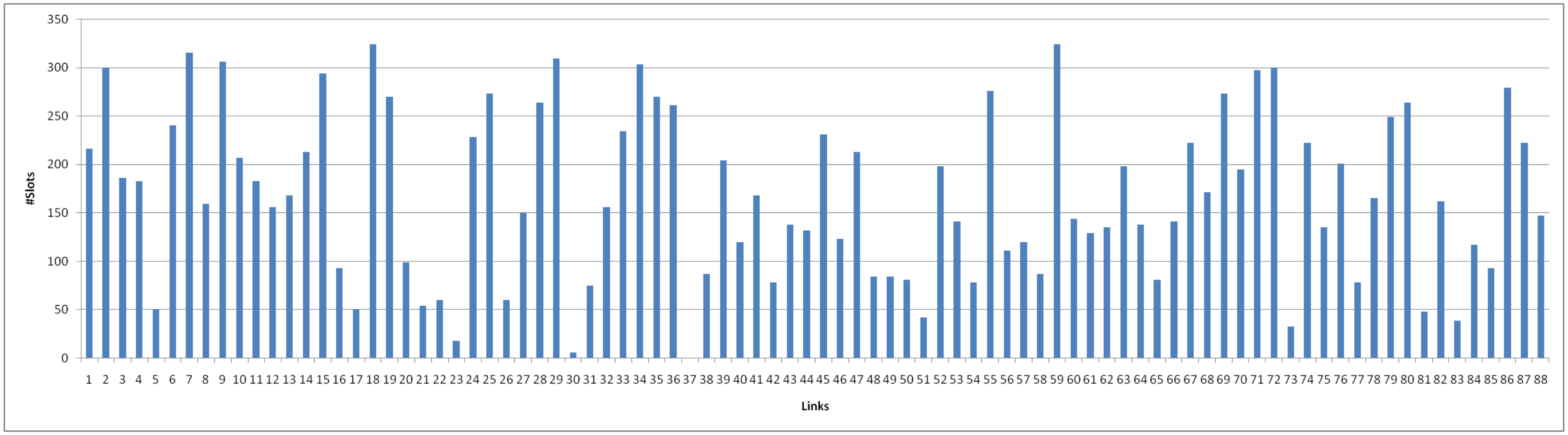


Figure 6.7: Slice Usage Germany50_1

6.5 Throughput versus Number of Regenerators

This section presents an evolution of the throughput with the number of regenerators for two network instances namely USA and Coronet100 and their traffic distributions. It can be seen that the throughput steadily increases with an increased in the number of regenerators, until about 80% and 85% of the total number of regenerators allowed have been reached for USA and Coronet100 respectively. We then observe that there is a sudden sharp increase in throughput. That is in fact the gain made by allowing the BI algorithm to enter in a infeasible search space to diversify the search space. It is important to mention that the states that are presented in the graphs plotted in Figure 6.8 and Figure 6.9 are the final states. That is the values of throughput are the last one for the given number of regenerators. A more detail view of the behavior of the BI algorithm with respect to the iterations, throughput and number of regenerators is displayed in Figure 6.10. It shows the repetitive switch from the feasible to infeasible states and vice versa. The initial feasible phase is the optimization phase with the maximum allowed regenerators. From Figure 6.10, it can be observed that alternating between the optimization phase and feasibility phase improves the throughput of the initial optimization phase by about 4%. The feasibility phase helps in fact to diversify the search space. The behavior of the BI algorithm with all its phases is illustrated in Figure 6.10

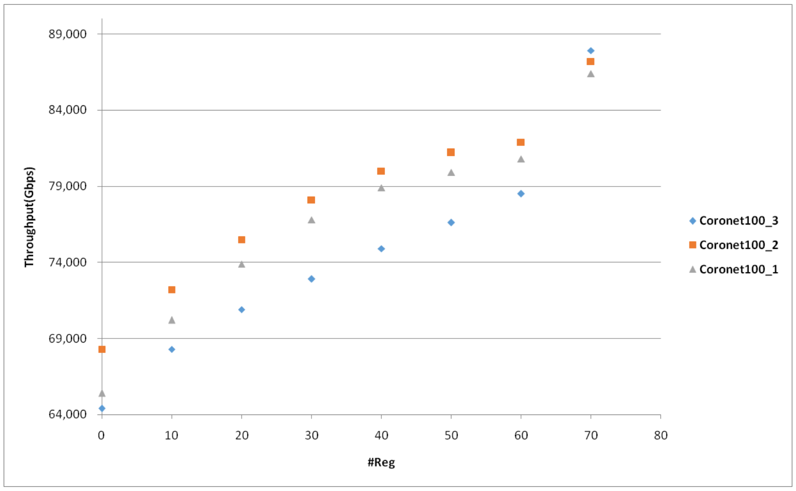


Figure 6.8: Regenerators vs Throughput Coronet100

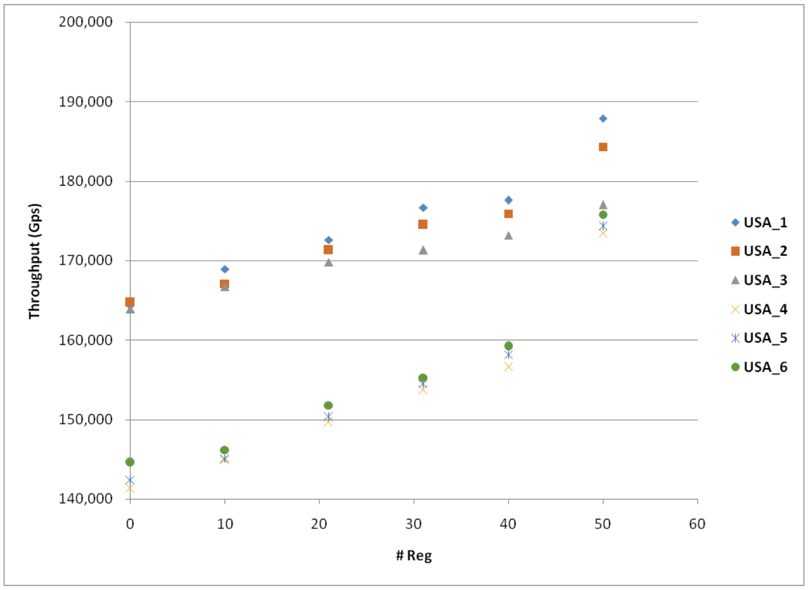


Figure 6.9: Regenerators vs Throughput USA

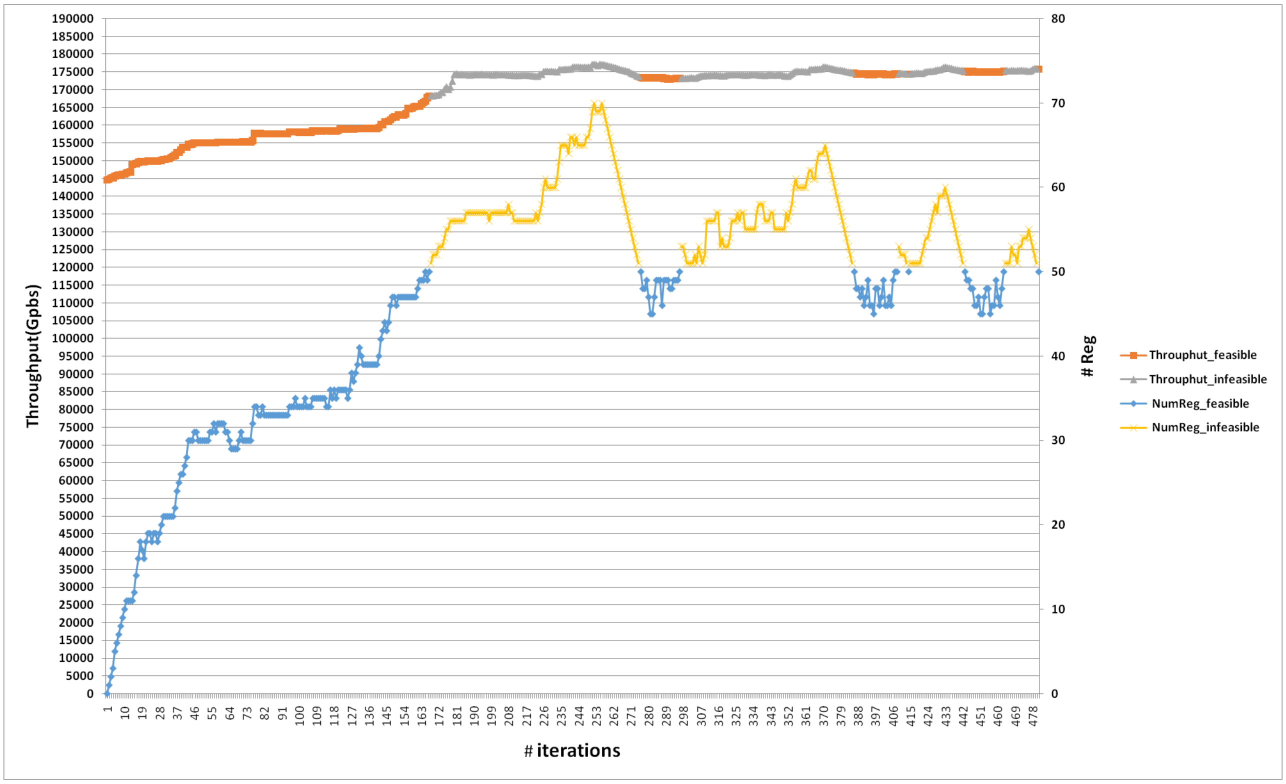


Figure 6.10: BI Behavior

Chapter 7

Conclusions and Future Work

7.1 Conclusions

In this thesis, we proposed two tabu search meta-heuristic algorithms for solving the RMSA problem in EONs. Our algorithms in addition simultaneously take into account the number of regenerators. Unlike previous works from which algorithms proposed solved the RSA or RMSA problem for small instances (maximum of 28 nodes), our algorithms solved the RMSA under realistic scenarios (heavy load) and on large networks. Even though each of our algorithms use a strategy fundamentally different, they lead to almost similar results in terms of throughput; with BI algorithm slightly outperforming GC algorithm (about 3%).

7.2 Future Work

As a future work, we plan to extent our algorithms from uni-objective to multi-objectives. In other words, we plan to do a joint optimization of both throughput and regenerator placements. We need to find the Pareto front to be able to get the best compromise between the use of RGs and the increased throughput when using different modulations.

Appendix A

Network Details

In this appendix, node and link details for two network instances namely germany50 and Euro28 are presented. It is important to note that the link distance between two nodes are computed using haversine formula as described in [7].

Node Id	Node Name	Node Id	Node Name
0	Aachen	26	Kassel
2	Augsburg	27	Kempten
3	Bayreuth	28	Kiel
4	Berlin	29	Koblenz
5	Bielefeld	30	Koeln
6	Braunschweig	31	Konstanz
7	Bremen	32	Leipzig
8	Bremerhaven	33	Magdeburg
9	Chemnitz	34	Mannheim
10	Darmstadt	35	Muenchen
11	Dortmund	36	Muenster
12	Dresden	37	Norden
13	Duesseldorf	38	Nuernberg
14	Erfurt	39	Oldenburg
15	Essen	40	Osnabrueck
16	Flensburg	41	Passau
17	Frankfurt	42	Regensburg
18	Freiburg	43	Saarbruecken
19	Fulda	44	Schwerin
20	Giessen	45	Siegen
21	Greifswald	46	Stuttgart
22	Hamburg	47	Trier
23	Hannover	48	Ulm
24	Kaiserslautern	49	Wesel
25	Karlsruhe	50	Wuerzburg

Table A.1: Germany50 Node Details

Links	L1 (Duesseldorf Essen)	L23 (Dresden Berlin)	L45 (Oldenburg Bremen)	L67 (Fulda Giessen)
	L2 (Dortmund Essen)	L24 (Schwerin Berlin)	L46 (Bremerhaven Bremen)	L68 (Saarbruecken Trier)
	L3 (Wesel Essen)	L25 (Magdeburg Berlin)	L47 (Hannover Bremen)	L69 (Wuerzburg Fulda)
	L4 (Koeln Duesseldorf)	L26 (Greifswald Berlin)	L48 (Flensburg Kiel)	L70 (Karlsruhe Saarbruecken)
	L5 (Aachen Koeln)	L27 (Dresden Leipzig)	L49 (Norden Oldenburg)	L71 (Karlsruhe Stuttgart)
	L6 (Koblenz Koeln)	L28 (Erfurt Leipzig)	L50 (Osnabrueck Oldenburg)	L72 (Ulm Stuttgart)
	L7 (Muenster Dortmund)	L29 (Magdeburg Leipzig)	L51 (Bremerhaven Flensburg)	L73 (Konstanz Stuttgart)
	L8 (Siegen Dortmund)	L30 (Bayreuth Leipzig)	L52 (Bielefeld Hannover)	L74 (Wuerzburg Stuttgart)
	L9 (Kassel Dortmund)	L31 (Erfurt Dresden)	L53 (Braunschweig Hannover)	L75 (Freiburg Karlsruhe)
	L10 (Wesel Aachen)	L32 (Chemnitz Dresden)	L54 (Osnabrueck Hannover)	L76 (Augsburg Ulm)
	L11 (Trier Aachen)	L33 (Chemnitz Erfurt)	L55 (Braunschweig Bielefeld)	L77 (Freiburg Konstanz)
	L12 (Bielefeld Muenster)	L34 (Kassel Erfurt)	L56 (Kassel Braunschweig)	L78 (Kempten Konstanz)
	L13 (Osnabrueck Muenster)	L35 (Wuerzburg Erfurt)	L57 (Giessen Kassel)	L79 (Augsburg Muenchen)
	L14 (Siegen Koblenz)	L36 (Bayreuth Chemnitz)	L58 (Fulda Kassel)	L80 (Kempten Muenchen)
	L15 (Frankfurt Koblenz)	L37 (Magdeburg Schwerin)	L59 (Darmstadt Frankfurt)	L81 (Passau Muenchen)
	L16 (Kaiserslautern Koblenz)	L38 (Greifswald Schwerin)	L60 (Giessen Frankfurt)	L82 (Nuernberg Muenchen)
	L17 (Trier Koblenz)	L39 (Hamburg Schwerin)	L61 (Fulda Frankfurt)	L83 (Regensburg Muenchen)
	L18 (Bielefeld Siegen)	L40 (Kiel Schwerin)	L62 (Mannheim Darmstadt)	L84 (Wuerzburg Augsburg)
	L19 (Giessen Siegen)	L41 (Braunschweig Magdeburg)	L63 (Kaiserslautern Darmstadt)	L85 (Regensburg Passau)
	L20 (Oldenburg Wesel)	L42 (Kiel Hamburg)	L64 (Karlsruhe Mannheim)	L86 (Bayreuth Nuernberg)
	L21 (Norden Wesel)	L43 (Hannover Hamburg)	L65 (Saarbruecken Kaiserslautern)	L87 (Wuerzburg Nuernberg)
	L22 (Leipzig Berlin)	L44 (Braunschweig Hamburg)	L66 (Karlsruhe Kaiserslautern)	L88 (Regensburg Nuernberg)

Table A.2: Germany50 Link Details

Links	Distances(Km)	Links	Distances(Km)	Links	Distances (Km)	Links	Distances(Km)
L1	15	L23	82	L45	88	L67	31
L2	32	L24	104	L46	55	L68	63
L3	31	L25	68	L47	28	L69	38
L4	22	L26	26	L48	50	L70	56
L5	13	L27	102	L49	28	L71	16
L6	38	L28	22	L50	32	L72	20
L7	32	L29	25	L51	56	L73	25
L8	42	L30	55	L52	81	L74	81
L9	13	L31	101	L53	23	L75	75
L10	44	L32	62	L54	131	L76	77
L11	73	L33	13	L55	138	L77	105
L12	63	L34	10	L56	16	L78	24
L13	21	L35	24	L57	67	L79	56
L14	36	L36	86	L58	53	L80	13
L15	20	L37	65	L59	54	L81	32
L16	70	L38	96	L60	16	L82	31
L17	47	L39	35	L61	32	L83	102
L18	47	L40	114	L62	31	L84	62
L19	77	L41	51	L63	29	L85	104
L20	24	L42	10	L64	33	L86	31
L21	115	L43	50	L65	35	L87	26
L22	146	L44	81	L66	15	L88	15

Table A.3: Germany50 Link Distances

Node id	Node Name	Node id	Node Name
0	Amsterdam	14	Lyon
1	Athens	15	Madrid
2	Barcelona	16	Milan
3	Belgrade	17	Munich
4	Berlin	18	Oslo
5	Bordeaux	19	Paris
6	Brussels	20	Prague
7	Budapest	21	Rome
8	Copenhagen	22	Stockholm
9	Dublin	23	Strasbourg
10	Frankfurt	24	Vienna
11	Glasgow	25	Warsaw
12	Hamburg	26	Zagreb
13	London	27	Zurich

Table A.4: Euro28 Node Details

Links	L1 (Amsterdam Brussels)	L22 (Copenhagen Oslo)
	L2 (Amsterdam Glasgow)	L23 (Dublin Glasgow)
	L3 (Amsterdam Hamburg)	L24 (Dublin London)
	L4 (Amsterdam London)	L25 (Frankfurt Hamburg)
	L5 (Athens Belgrade)	L26 (Frankfurt Munich)
	L6 (Athens Rome)	L27 (Frankfurt Strasbourg)
	L7 (Barcelona Lyon)	L28 (London Paris)
	L8 (Barcelona Madrid)	L29 (Lyon Paris)
	L9 (Belgrade Budapest)	L30 (Milan Munich)
	L10 (Belgrade Zagreb)	L31 (Milan Rome)
	L11 (Berlin Copenhagen)	L32 (Milan Zurich)
	L12 (Berlin Hamburg)	L33 (Munich Vienna)
	L13 (Berlin Munich)	L34 (Oslo Stockholm)
	L14 (Berlin Prague)	L35 (Paris Strasbourg)
	L15 (Berlin Warsaw)	L36 (Prague Vienna)
	L16 (Bordeaux Madrid)	L37 (Rome Zagreb)
	L17 (Bordeaux Paris)	L38 (Stockholm Warsaw)
	L18 (Brussels Frankfurt)	L39 (Zurich Lyon)
	L19 (Brussels Paris)	L40 (Strasbourg Zurich)
	L20 (Budapest Prague)	L41 (Vienna Zagreb)
	L21 (Budapest Warsaw)	

Table A.5: Euro28 Link Details

Links	Distances(Km)	Links	Distances(Km)
L1	116	L22	260
L2	225	L23	153
L3	77	L24	127
L4	61	L25	232
L5	607	L26	142
L6	348	L27	122
L7	352	L28	193
L8	81	L29	240
L9	223	L30	218
L10	80	L31	293
L11	210	L32	151
L12	69	L33	146
L13	288	L34	266
L14	154	L35	203
L15	273	L36	137
L16	337	L37	328
L17	317	L38	401
L18	110	L39	135
L19	140	L40	83
L20	207	L41	195
L21	365		

Table A.6: Euro28 Distances

Bibliography

[1] R. W. Alaskar, I. Ahmad, and A. Alyatama. Offline routing and spectrum allocation algorithms for elastic optical networks. *Optical Switching and Networking*, 21:79–92, 2016.

[2] M. N. Architects. Coronet continental united states (conus) topology. `http://www.monarchna.com/topology.html`. Accessed: 2018-05-20.

[3] M. Batayneh, D. Schupke, M. Hoffmann, A. Kirstadter, and B. Mukherjee. On routing and transmission-range determination of multi-bit-rate signals over mixed-line-rate WDM optical networks for carrier ethernet. *IEEE/ACM Transactions on Networking*, 19(5):1304–1316, October 2011.

[4] A. Cai, G. Shen, L. Peng, and M. Zukerman. Novel node-arc model and multiiteration heuristics for static routing and spectrum assignment in elastic optical networks. *Journal of Lightwave Technology*, 31(21):3402–3413, 2013.

[5] B. C. Chatterjee, N. Sarma, and E. Oki. Routing and spectrum allocation in elastic optical networks: a tutorial. *IEEE Communications Surveys & Tutorials*, 17(3):1776–1800, 2015.

[6] X. Chen, Y. Zhong, and A. Jukan. Multipath routing in elastic optical networks with distance-adaptive modulation formats. In *IEEE International Conference on Communications*, pages 3915–3920, 2013.

[7] N. R. Chopde and M. Nichat. Landmark based shortest path detection by using a* and haversine formula. *International Journal of Innovative Research in Computer and Communication Engineering*, 1(2):298–302, 2013.

[8] R. Christian, H. Kai, O. Sebastian, and K. Roman. Networks and problem instances of sndlib. `http://sndlib.zib.de`. Accessed: 2018-05-20.

[9] K. Christodoulopoulos, I. Tomkos, and E. Varvarigos. Elastic bandwidth allocation in flexible ofdm-based optical networks. *Journal of Lightwave Technology*, 29(9):1354–1366, 2011.

[10] Cisco. Introduction to DWDM technology. `https://www.cisco.com/c/dam/global/de_at/assets/docs/dwdm.pdf`. Accessed: 2018-08-18.

[11] M. Dallaglio, A. Giorgetti, N. Sambo, L. Velasco, and P. Castoldi. Routing, spectrum, and transponder assignment in elastic optical networks. *Journal of Lightwave Technology*, 33(22):4648–4658, 2015.

[12] N. Dharmaweera, L. Yan, J. Zhao, M. Karlsson, and E. Agrell. Regenerator site selection in impairment-aware elastic optical networks. In *Optical Fiber Communication Conference - OFC*, pages 1–3, 2016.

[13] D.-Z. Du and P. M. Pardalos. *Handbook of combinatorial optimization: supplement*, volume 1. Springer Science & Business Media, 2013.

[14] R. Durán, I. Rodriguez, N. Fernandez, I. de Miguel, N. Merayo, P. Fernandez, J. Aguado, T. Jimenez, R. Lorenzo, and E. Abril. Performance comparison of methods to solve the routing and spectrum allocation problem. In *IEEE International Conference of on Transparent Optical Networks*, pages 1–4, 2012.

[15] J. Enoch and B. Jaumard. Towards optimal and scalable solution for routing and spectrum allocation. *Electronic Notes in Discrete Mathematics*, 64:335–344, 2018.

[16] W. Fadini, B. C. Chatterjee, and E. Oki. A subcarrier-slot partition scheme with first-last fit spectrum allocation for elastic optical networks. *Computer Networks*, 91:700–711, 2015.

[17] O. Gerstel, M. Jinno, A. Lord, and S. B. Yoo. Elastic optical networking: A new dawn for the optical layer? *IEEE Communications Magazine*, 50(2), 2012.

[18] F. Glover. Future paths for integer programming and links to artificial intelligence. *Computers & Operations Research*, 13(5):533–549, 1986.

[19] F. Glover. Tabu search: A tutorial. *Interfaces*, 20(4):74–94, 1990.

[20] R. Goścień, K. Walkowiak, and M. Klinkowski. Tabu search algorithm for routing, modulation and spectrum allocation in elastic optical network with anycast and unicast traffic. *Computer Networks*, 79:148–165, 2015.

[21] D. T. Hai. Multi-objective genetic algorithm for solving routing and spectrum assignment problem. In *International Conference on Information Science and Technology*, pages 177–180, 2017.

[22] A. Hertz and M. Widmer. Guidelines for the use of meta-heuristics in combinatorial optimization. *European Journal of Operational Research*, 151(2):247–252, 2003.

[23] Huawei. White paper on technological developments of optical networks. `http://www-file.huawei.com/-/media/CORPORATE/PDF/white%20paper/White-Paper-on-Technological-Developments-of-Optical-Networks.pdf`. Accessed: 2018-08-18.

[24] R. Hulsermann, A. Betker, M. Jager, S. Bodamer, M. Barry, J. Spath, C. Gauger, and M. Kohn. A set of typical transport network scenarios for network modelling. *ITG FACHBERICHT*, 182:65–72, 2004.

[25] B. Jaumard and M. Daryalal. Scalable elastic optical path networking models. In *IEEE International Conference of on Transparent Optical Networks*, pages 1–4. IEEE, 2016.

[26] B. Jaumard and M. Daryalal. Efficient spectrum utilization in large scale rwa problems. *IEEE/ACM Transactions on Networking (TON)*, 25(2):1263–1278, 2017.

[27] B. Jaumard, C. Meyer, and X. Yu. Of how much wavelength conversion allows to reduce the blocking rate. *Journal of Optical Networking*, 5(22):881–900, 2005.

[28] M. Jinno, B. Kozicki, H. Takara, A. Watanabe, Y. Sone, T. Tanaka, and A. Hirano. Distance-adaptive spectrum resource allocation in spectrum-sliced elastic optical path network [topics in optical communications]. *IEEE Communications Magazine*, 48(8), 2010.

[29] N. Kadu, S. Shakya, and X. Cao. Modulation-aware multipath routing and spectrum allocation in elastic optical networks. In *IEEE International Conference on Advanced Networks and Telecommuncations Systems*, pages 1–6, 2014.

[30] M. Klinkowski and K. Walkowiak. On performance gains of flexible regeneration and modulation conversion in translucent elastic optical networks with superchannel transmission. *Journal of lightwave technology*, 34(23):5485–5495, 2016.

[31] V. López and L. Velasco. *Elastic Optical Networks: Architectures, Technologies, and Control*. Springer, 2016.

[32] E. Q. Martins and M. M. Pascoal. A new implementation of yens ranking loopless paths algorithm. *Quarterly Journal of the Belgian, French and Italian Operations Research Societies*, 1(2):121–133, 2003.

[33] C. T. Politi, V. Anagnostopoulos, C. Matrakidis, A. Stavdas, A. Lord, V. López, and J. P. Fernández-Palacios. Dynamic operation of flexi-grid ofdm-based networks. In

Optical Fiber Communication Conference, pages OTh3B–2. Optical Society of America, 2012.

[34] Y. Qi. K shortest path java implementation. `https://github.com/yan-qi/k-shortest-paths-java-version`. Accessed: 2018-05-20.

[35] R. Ramaswami, K. Sivarajan, and G. Sasaki. *Optical Networks: a Practical Perspective*. Morgan Kaufmann, 2009.

[36] M. Ruiz, M. Pióro, M. Żotkiewicz, M. Klinkowski, and L. Velasco. Column generation algorithm for rsa problems in flexgrid optical networks. *Photonic network communications*, 26(2-3):53–64, 2013.

[37] S. Shirazipourazad, C. Zhou, Z. Derakhshandeh, and A. Sen. On routing and spectrum allocation in spectrum-sliced optical networks. In *IEEE Annual Joint Conference of the IEEE Computer and Communications Societies*, pages 385–389, 2013.

[38] J. M. Simmons. *Optical network design and planning*. Springer, 2014.

[39] S. Talebi, E. Bampis, G. Lucarelli, I. Katib, and G. N. Rouskas. Spectrum assignment in optical networks: A multiprocessor scheduling perspective. *Journal of Optical Communications and Networking*, 6(8):754–763, 2014.

[40] K. Walkowiak. *Modeling and optimization of cloud-ready and content-oriented networks*, volume 56. Springer, 2016.

[41] K. Walkowiak and M. Klinkowski. On the impact of modulation format transmission reach on spectrum usage in elastic optical networks. In *Asia Communications and Photonics Conference*, pages AS2D–6. Optical Society of America, 2016.

[42] J. Y. Yen. Finding the k shortest loopless paths in a network. *management Science*, 17(11):712–716, 1971.

[43] X. Zhou, W. Lu, L. Gong, and Z. Zhu. Dynamic rmsa in elastic optical networks with an adaptive genetic algorithm. In *IEEE Global Telecommunications Conference - GLOBECOM*, pages 2912–2917, 2012.