FORMULATIONS AND ALGORITHMS TO DESIGN COMMUNICATION NETWORKS

FERNANDA SUMIKA HOJO DE SOUZA

FORMULATIONS AND ALGORITHMS TO DESIGN COMMUNICATION NETWORKS

Tese apresentada ao Programa de Pós-Graduação em Ciência da Computação do Instituto de Ciências Exatas da Universidade Federal de Minas Gerais como requisito parcial para a obtenção do grau de Doutor em Ciência da Computação.

Orientador: Geraldo Robson Mateus

Belo Horizonte

Maio de 2012

FERNANDA SUMIKA HOJO DE SOUZA

FORMULATIONS AND ALGORITHMS TO DESIGN COMMUNICATION NETWORKS

Thesis presented to the Graduate Program in Computer Science of the Federal University of Minas Gerais in partial fulfillment of the requirements for the degree of Doctor in Computer Science.

Advisor: Geraldo Robson Mateus

Belo Horizonte
May 2012

Souza, Fernanda Sumika Hojo de.

S729f Formulations and algorithms to design communication networks / Fernanda Sumika Hojo de Souza. — Belo Horizonte, 2012.
xxv, 104 f. : il. ; 29cm

Tese (doutorado) — Universidade Federal de Minas Gerais – Departamento de Ciência da Computação.

Orientadora: Geraldo Robson Mateus.

1. Computação - Teses. 2. Redes de computadores – Teses. 3. Programação Inteira – Teses. I. Orientador. II. Título

519.6*22(043)

Dedico esta tese a todos aqueles que me ajudaram durante esse percurso; em especial ao Daniel e aos meus pais Edina e Mauro.

Acknowledgments

Agradeço ao meu orientador, Prof. Geraldo Robson Mateus, pelos ensinamentos, paciência, compreensão e por acreditar na minha capacidade.

Ao Prof. Alexandre Salles da Cunha, agradeço pela dedicação a me ensinar, paciência e incentivo.

Ao Daniel, agradeço por estar sempre ao meu lado, apoiando e ajudando sempre que precisei.

Agradeço à minha família, Edina, Mauro e Natália, que sempre torceram pelo meu sucesso.

À tia Rô e tio It, que sempre foram presentes na minha vida, agradeço por todo apoio durante esse período da minha vida.

Agradeço aos meus amigos, que estiveram ao meu lado nos momentos de estudo e de lazer, compartilhando as dificuldades e as conquistas dessa caminhada.

Agradeço ainda ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) e à Coordenação de Aperfeiçoamento de pessoal de Nível Superior (CAPES) por proporcionar apoio financeiro durante meu curso.

"A educação é a arma mais poderosa que você pode usar para mudar o mundo."
(Nelson Mandela)

Resumo

O estudo de redes tem raízes na teoria dos grafos que remota a 1730. Desde então, redes tem sido utilizadas para modelar e simular interações entre elementos de sistemas complexos, tais como de transporte, de comunicação e de computadores. Redes de comunicação são amplamente utilizadas para trocar informações entre entidades de um sistema. A importância das redes de comunicação aumentou dramaticamente nos últimos anos, chamando atenção para o estágio de projeto de um sistema real, dando origem a diversos problemas de otimização. Técnicas e soluções de Pesquisa Operacional tem desempenhado papel fundamental sobre uma vasta gama de problemas de projeto de redes. Nesta tese, nós estudamos como aplicar técnicas de otimização no projeto de redes de comunicação. Primeiramente, nós abordamos o problema de projetar redes de telecomunicações hierárquicas assegurando resiliência contra falhas aleatórias e garantias de atraso na comunicação. Posteriormente, nós investigamos soluções para o problema de roteamento e alocação de comprimentos de onda com agregação de tráfego, proteção e qualidade de serviço em redes ópticas WDM. Finalmente, nós estudamos como projetar redes de comunicação eficientes com base em características de redes complexas. Um conjunto de métricas é usado como critério de otimização no projeto dessas redes. Diferentes formulações matemáticas para modelar os três problemas são propostas. Um algoritmo Branch-and-bound baseado nas formulações compactas é avaliado e comparado a uma abordagem Branch-and-price baseada nas formulações estendidas dos problemas. Uma análise comparativa é realizada, demonstrando que a abordagem Branch-and-price proposta é capaz de resolver problemas cujas dimensões estÑo fora do alcance de outras ferramentas tradicionais de otimização.

Palavras-chave: *Branch-and-Price*, Programação Inteira, Redes de Comunicação.

Abstract

The study of networks has roots in graph theory dating back to 1730s. From then on, networks have been used to model and simulate interactions among elements of intricate systems, such as transportation, communication and computer ones. Communication networks are widely used to exchange information among entities of a system. The importance of communication networks has dramatically increased over the past few years, drawing attention to the design stage of a real system, giving rise to many optimization problems. Operations Research techniques and solutions have been playing a fundamental role across a wide range of network design problems. In this thesis, we study how to apply optimization techniques in the design of communication networks. Firstly, we dedicate to the problem of designing hierarchical telecommunication networks ensuring resilience against random failures and maximum delay guarantees in the communication. Later, we investigate solutions to the routing and wavelength assignment problem with traffic grooming, protection and quality of service in WDM optical networks. Finally, we study how to design efficient communication networks based on complex networks features. A set of metrics is used as optimization criteria while designing such networks. Different mathematical formulations to model the three problems are proposed. A Branch-and-bound algorithm based on compact formulations is evaluated and compared to a Branch-and-price approach based on extended formulations of the problems. Our comparative analysis demonstrates that the proposed Branch-and-price approach is able to solve problems whose dimensions are out of reach for other traditional optimization tools.

Keywords: Branch-and-price, Integer Programming, Communication Networks.

List of Figures

List of Tables

Contents

UNIVERSIDADE FEDERAL DE MINAS GERAIS

PPGCC
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO
DCC - UFMG

ATA DA DEFESA DE TESE DA ALUNA **FERNANDA SUMIKA HOJO DE SOUZA**

Realizou-se, no dia 11 de maio de dois mil doze, às 10:00 horas, na Sala 2077 do Instituto de Ciências Exatas, da Universidade Federal de Minas Gerais, a 117ª defesa de tese de Doutorado em Ciência da Computação, intitulada FORMULAÇÕES E ALGORITMOS PARA PROJETAR REDES DE COMUNICAÇÃO, apresentada por **Fernanda Sumika Hojo de Souza**, graduada no curso de Bacharelado em Ciência da Computação, como requisito parcial para a obtenção do grau de Doutor em Ciência da Computação, à seguinte Comissão Examinadora: PROF. GERALDO ROBSON MATEUS - Orientador (Departamento de Ciência da Computação - UFMG), PROF. ALEXANDRE SALLES DA CUNHA (Departamento de Ciência da Computação - UFMG), PROF. MARCUS VINÍCIUS SOLEDADE POGGI DE ARAGÃO (Departamento de Informática - PUC/Rio), PROF. MAURÍCIO CARDOSO DE SOUZA (Departamento de Engenharia de Produção - UFMG), PROF. NELSON MACULAN FILHO (COPPE - UFRJ).

A Comissão considerou a tese:

(X) Aprovada
( ) Aprovada condicionalmente, sujeita a alterações, conforme folha de modificações, anexa
( ) Reprovada, conforme folha de modificações, anexa

Finalizados os trabalhos, lavrei a presente ata que, lida e aprovada, vai assinada por mim e pelos membros da Comissão.
Belo Horizonte, 11 de maio de 2012.

RENATA VIANA MORAES ROCHA - Secretária

PROF. GERALDO ROBSON MATEUS - ORIENTADOR
Doutor em Engenharia de Sistemas e Computação – Universidade Federal do Rio de Janeiro- Brasil

PROF. ALEXANDRE SALLES DA CUNHA
Doutor em Engenharia de Sistemas e Computação - Universidade Federal do Rio de Janeiro - Brasil

PROF. MARCUS VINICIUS SOLEDADE POGGI DE ARAGÃO
Doutor em Matemática Aplicada - Ecole Polytechnique de Montreal - Canadá

UNIVERSIDADE FEDERAL DE MINAS GERAIS

PPGCC
PROGRAMA DE PÓS-GRADUAÇÃO
EM CIÊNCIA DA COMPUTAÇÃO
DCC - UFMG

PROF. MAURÍCIO CARDOSO DE SOUZA
Doutor em Informática - Université Blaise Pascal - Clermont-Ferrand - França

PROF. NELSON MACULAN FILHO
Doutor em Engenharia de Produção – Universidade Federal do Rio de Janeiro - Brasil

Chapter 1

Introduction

In this Chapter, we provide a brief introduction to the main problems we are going to discuss along the present thesis. Section 1.1 is devoted to the motivation of this work while the main objectives are detailed in Section 1.2. In Section 1.3, we specify the main contributions of this work. Finally, in Section 1.4, the structure of the present thesis is described.

1.1 Motivation

The study of networks has roots in graph theory dating back to 1730s, when Leonard Euler formulated the Königsberg Bridge Problem as a graph optimization problem [Bondy and Murty, 1976; Diestel, 2005]. From then on, networks have been used to model and simulate interactions among elements of intricate systems (transportation, communication and computer, just to name a few ones). In the context of transportation, consider a road network as an illustrative example. Cities are crossing points among different roads enabling vehicles to create their own routes along the network. Rail networks and airline networks work in a similar fashion. One ancient well known network is the postal network, in which messages are exchanged among different people regardless how far they might be located. On the other hand, with the advent of technology, today we also have other huge networks such as data distribution, cellular and the Internet.

Communication networks [Frank et al., 1972] are widely used to exchange information (or a packet, a commodity) among entities of a system. We usually call these entities as "nodes" or "vertices"[1], which may be classified into *source*, *destination* (or even both) and *transhipment*, depending on their role in the network. A general com-

[1] The terms "nodes" and "vertices" will be treated interchangeably in this work.

munication network is composed of a large number of nodes, which may not necessarily be connected to all other nodes of the network. The idea is that a node also works as a connection point between different incoming and outgoing paths. The communication among the entities is possible by transmission media (also called "links" or "edges"). Usually, links are associated to variable capacities. In a single communication network, links may have different types of facilities and associated capacity.

The importance of telecommunication networks has dramatically increased over the past few years. We have been facing evolving technologies which offer a variety of services such as audio, video and general data transmission. Moreover, the increasing demand from customers and the concern about service reliability in order to avoid customer complaints require grades of organization, leading to hierarchical telecommunication networks. The basic idea is that priorities are associated to sets of customers so that network providers can offer higher levels of service in both serviceability (e.g., high bandwidth) and survivability (failure protection) to certain key customers.

In this context, quality of service (QoS) [Srikitja et al., 1999] arises to provide special guarantees to different applications or customers. QoS refers to the ability of a network to deliver predictable results. Elements of network performance within the scope of QoS often include availability (uptime), throughput, latency and failure rate. A set of QoS metrics is discussed along this work. We show how issues like delay, load balancing, resilience and vulnerability can be taken into account while designing communication networks.

Network survivability is another key issue that may not be disregarded in the design of networks. Under failure conditions, protection mechanisms enable the network to maintain maximum connectivity and quality of service. These mechanisms lie in the topological level, protocol design or even additional bandwidth allocation. In the topological level, for instance, a two-connected network is robust against random single link/node failures. Other examples include the usage of dynamic routing protocols to reroute traffic against network dynamics during the transition of network dimensioning or equipment failures. A bandwidth allocation mechanism proactively allocates extra bandwidth to avoid traffic loss under failure conditions.

Complex networks are found in the real world in different areas of science, including the Internet, WWW, neural networks, friendship relationships, among others [Newman, 2003; Watts and Strogatz, 1998; Faloutsos et al., 1999; Thadakamalla et al., 2004; Albert et al., 1999]. They are many times characterized by a non-trivial topology and present interesting features which may be useful in designing communication networks. One of these features concerns the low cost for sending information (or a packet, a commodity) through the network. The small average path length is directly related to a small data communication latency. Thereby, engineered networks could take advantage of being modelled to present specific complex features, to improve their overall efficiency. Several network models were proposed and studied in an attempt to represent elements of a system and their relationships [Lewis, 2009; Newman, 2003; Watts, 2004; Thadakamalla et al., 2008]. Early models include regular [Lewis, 2009] and random networks [Erdös and Rényi, 1959; Gilbert, 1959]. In the late 1990s, new network models were proposed, and complex network concepts began to be formalized.

One important stage while creating engineered networks concerns the topological design of the network, in which critical decisions must be taken. This problem is referred as a *"network design problem"* in the Operations Research (OR) community [Magnanti and Wong, 1984]. This stage may involve determining where to place the components and how to connect them. Moreover, most of the time, fixed and variable costs are involved, giving rise to many optimization problems. Operations Research techniques and solutions have been playing a fundamental role across a wide range of network design problems. Considering the set of requirements to be met while designing such networks, an optimization phase is certainly of great importance. The fulfillment of these requirements is becoming increasingly challenging, once entities have been demanding high standards of service, including hierarchy, serviceability, survivability and extreme efficiency. Optimization techniques have been successfully applied in designing engineered networks [Magnanti and Wong, 1984; Resende and Pardalos, 2005; Chinneck et al., 2009].

1.2 Objectives

The main goal of the work described in this thesis is to design communication networks by means of optimization techniques in an exact approach. Therefore, the following issues compose our general objectives:

- Proposal of alternative mathematical formulations to model three network design problems, such that they can be based on network flows or path variables.

- Performance comparison between a Branch-and-bound algorithm for compact formulations and a Branch-and-price algorithm devised by applying a Delayed Column Generation approach to the problem.

- Study of different branching rules while dealing with the enumeration tree in the Branch-and-price algorithm.

- Evaluation of acceleration strategies while solving a Delayed Column Generation algorithm, such as stabilization and column management.

Based on these issues, we are investigating how state of the art algorithms may be improved or complemented by alternative approaches. In this context, the specific objectives of this thesis are:

- Propose an exact solution approach to design a hierarchical network with resilience and delay guarantees. The implied solution must deal with different sets of nodes which have different requirements, single failure resilience constraints and hop constraints to assure a maximum delay.

- Propose an exact solution to design optical networks considering routing, wavelength assignment, traffic grooming, protection and quality of service.

- Propose an exact algorithm to design efficient communication networks based on complex network features. By optimizing desired metrics such as shortest path length and maximum vertex degree, the overall efficiency of engineered networks may be improved.

1.3 Contributions

The contributions of this thesis were partially published/submitted and are listed below.

- Souza et al. [2008], published in the *XL Simpósio Brasileiro de Pesquisa Operacional (SBPO'08)*. In this work, a GRASP-based algorithm to generate small world topologies is proposed.

- Souza et al. [2009], published in the *First IEEE International Workshop on Network Science For Communication Networks (NetSciCom'09) in conjunction with IEEE Infocom (INFOCOM'09)*. This work addresses the problem of designing complex networks based on two mathematical formulations and a column generation algorithm.

- Souza et al. [2010b][1], presented at the *10th INFORMS Telecommunications Conference*. This work concerns a column generation algorithm for the resilient multilevel hop-constrained network design problem.

- Souza et al. [2010c][1], presented at the *Journées de l'Optimisation/Optimization Days 2010 (JOPT'10)*. This work concerns a column generation algorithm for the resilient multi-level hop-constrained network design problem.

- Souza et al. [2010a], published in the *IEEE GLOBECOM Workshop on Complex and Communication Networks (CCNet'10)*. In this work, a Branch-and-price algorithm for the design of complex networks is developed.

- Souza et al. [2012a], published in the *Handbook of Optimization in Complex Networks: Theory and Applications*. In this chapter we review optimization algorithms to design complex communication networks.

- Souza et al. [2012b], to be published in the *IEEE Symposium on Computers and Communication (ISCC'12)*. In this work, we compare solutions for the routing and wavelength assignment problem with traffic grooming, protection and quality of service in WDM optical networks.

- Work[1] submitted to the *European Journal of Operational Research*. In this paper we propose three formulations for the resilient multi-level hop-constrained network design problem, evaluate algorithms to solve the problem and prove the equivalence of two distinct formulations.

In parallel with his research topic, the student has been participating as a co-author in other works related to her research topic: complex communication networks. As a result of this iteration another work has been published and is listed below.

- Guidoni et al. [2010], published in *IEEE Global Communications Conference (GLOBECOM'10)*. This work studies the channel assignment problem over a Heterogeneous Sensor Network with small world features.

- Guidoni et al. [2012], to be published in *IEEE Symposium on Computers and Communication (ISCC'12)*. This work proposes a framework based on small world features to provide QoS in Heterogeneous Sensor Networks.

1.4 Outline

This thesis is organized as follows. In Chapter 2, a background of the methods adopted in our methodology is provided, along with the main concepts on communication networks that we study in this work. We start giving a contextualization of Linear and

[1]This work was developed during a sandwich program at Université de Montréal.

Integer Programming, followed by a Delayed Column Generation overview. We also introduce the Branch-and-price algorithm based on a combination of Delayed Column Generation and Branch-and-bound methods. Finally, we introduce some concepts on communication networks.

Chapters 3, 4 and 5 concern three different network design problems in the context of communication networks. All of them are tackled using a similar methodology. Chapter 3 addresses the Resilient Multi-level Hop-constrained Network Design problem, which consists of designing a hierarchical network with resilience and delay guarantees. The problem is introduced and contextualized in the literature. Two compact mathematical formulations based on network flows are proposed as well as an extended formulation based on a exponential number of path variables. A Branch-and-price algorithm based on the last formulation is developed and computational experiments are run, allowing a comparative analysis on the efficiency of the Branch-and-price algorithm and a Branch-and-bound algorithm for the former formulations.

Chapter 4 concerns formulations based on network flows and path variables for the Grooming, Routing and Wavelength Assignment problem with Protection and QoS in WDM optical networks in order to minimize the number of wavelengths used. We compare the performance of different algorithms over real world instances.

Chapter 5 is dedicated to the Optimal Topology Design of Complex Networks problem. We review some models, heuristics as well as exact solution approaches based on Integer Programming methods to generate topologies owning complex features. Two mathematical formulations are proposed, such that the former is based on network flows and the latter corresponds to an extended formulation with path variables. A comparative study between a Branch-and-bound algorithm for the flow-based formulation and a Branch-and-price algorithm for the path-based formulation is also performed.

The future work along with the final remarks on applying optimization techniques to design communications networks are presented in Chapter 6.

Chapter 2

Background

This chapter is devoted to the fundamental aspects of Operations Research techniques and the basic concepts on communication networks. We start contextualizing Linear and Integer Programming. Next, we provide a general idea of the methods applied in our solution approach. The Delayed Column Generation method is presented as well as some related issues. In the following, we introduce the Branch-and-price algorithm. Finally, we review the main concepts on communication networks in order to provide a better understanding along this work.

2.1 Linear and Integer Programming

Linear Programming (LP) consists in the optimization of a linear objective function, subject to linear equality and inequality constraints. The idea of solving a linear programming problem is usually associated with a way to achieve the best outcome when different activities compete for a set of scarce resources. The conception of linear programming is credited to George Dantzig, who proposed the problem around 1947. In this context, the *simplex* algorithm [Dantzig, 1963] was also proposed by Dantzig in the late 1940s, for solving LP problems. The simplex method is widely known for its practical ability to solve diverse management decision problems. Moreover, the theoretical importance of the method cannot be disregarded since it is a basis for other methods in Integer and Nonlinear Programming [Bazaraa et al., 2004].

Integer Linear Programming (ILP) extends the Linear Programming in the particular case that variables are all required to be integers. In contrast to Linear Programming, which can be usually solved efficiently, Integer Programming problems are still a challenging field of research. These decision problems modelled as Integer Programs are classified as NP-hard. Even the special cases, where variables are required to be 0 or 1 (binary integer programming) or when only some of the variables are required to be integers (mixed integer programming), are generally also NP-hard. This means that there are no known polynomial time algorithms to solve (optimally) a general Integer Programming problem.

Several approaches have been proposed to solve Integer Programs in the last decades. Early work in this direction accounts for the cutting-plane method introduced by Gomory [1958]. Another approach is the well known Branch-and-bound (BB) algorithm proposed by Land and Doig [1960], which consists of an implicit enumeration procedure of all candidate solutions. These pioneer methods gave rise to other sophisticated approaches such as the Branch-and-cut (BC) and Branch-and-price (BP) algorithms. Roughly speaking, these approaches are hybrids of Branch-and-bound plus cutting-plane and Branch-and-bound plus Delayed Column Generation, respectively. For further details regarding Integer Programming, see [Wolsey, 1998].

2.2 Delayed Column Generation

The Delayed Column Generation is a technique for solving Linear Programming problems, in which we do not need to consider all columns (variables) explicitly at once. In doing so, an extended reformulation of the problem (having exponentially many variables) works as the basis of the procedure. In order to derive lower bounds for a given formulation, we must deal with the excessive number of columns (variables) implicitly. Therefore, the method starts with a restricted set of columns and add new columns *on-the-fly*, as needed. This Linear Program is usually referred to as the Restricted Master Program (RMP). A new Restricted Master Program, enlarged with new columns is solved iteratively until no further columns need to be added. At this point, the Linear Program has been solved. Typically, the total number of columns at the end of the procedure is just a tiny fraction of the total number of columns. This method is known for providing stronger LP bounds compared to the linear relaxation of a compact formulation for many problems.

Let us now describe how new columns are generated along the procedure. First, we should associate our LP relaxation to its LP Dual. Consider that we have a basic feasible solution for the LP relaxation and its corresponding optimal dual solution. By the Duality Theory, we must check if dual constraints associated to the new columns that do not belong to the RMP are violated. While this is true, these columns must be included in the RMP, which will be re-optimized. Otherwise, the current solution solves the LP relaxation of the problem. The problem of finding new columns that violate the dual constraints is usually called *pricing problem*. The violation checking is based on the computation of the reduced cost of a new column. Every time a new RMP is solved, it enables dual prices for each of the constraints, which are used in the pricing problem in a mutual feeding strategy.

The Delayed Column Generation method will be explored in more detail in Chapters 3, 4 and 5, applied to specific problems. For further information, see [Desrosiers and Lübbecke, 2005; Lasdon, 1970; Barnhart et al., 1998].

2.2.1 Stabilization Methods

Stabilization methods for column generation emerged as an attempt to reduce the total time spent while solving the algorithm due to degeneracy and convergence difficulties. One of the problems faced in column generation arises when a primal solution is associated to multiple dual solutions. As the subproblems are totally dependent on the dual values, the choice of which dual values to use becomes a relevant issue.

Many stabilization methods have been proposed in the literature [Merle et al., 1997; Lübbecke and Desrosiers, 2005; Rousseau et al., 2007; Amor et al., 2009]. To understand how stabilization works, let us consider the interior point stabilization method proposed by Rousseau et al. [2007]. The main idea of this method is to select a dual optimal solution interior to the optimal face of the dual polyhedron rather than retrieving an extreme point. In order to achieve the centralization of dual values, several extreme points of the optimal dual polyhedron need to be generated, so that an interior point corresponding to a convex combination of all these extreme points may be computed. This is accomplished by solving an auxiliary Linear Programming problem that exploits complementary slackness conditions given by the optimal RMP primal-dual solution. The literature reports significant gains when such dual values are used to price columns.

2.2.2 Column Generation Primal Heuristics

Primal heuristics are generally referred to as methods based on truncated exact optimization procedures or constructive processes using relaxation [Barnhart et al., 1998; Joncour et al., 2010]. Being able to derive "good" primal feasible solutions, they become an effective algorithm when optimality is not the major concern.

A well known primal heuristic in column generation context is the restricted master heuristic. In this case, the formulation is solved as a static integer problem, limited to a set of columns which can be generated heuristically, or by the master itself, or even by a mixture of both. The main drawback of this approach is that feasibility is not guaranteed, and thus, an ad-hoc strategy needs to be used to repair infeasibility.

Rounding heuristics are another way to find approximated solutions. For this, the master LP solution is taken as a base for column selection. Greedy heuristics have also been proposed. They consist of iteratively adding a greedy selected column to the partial solution until a feasible solution is achieved.

The Branch-and-price tree may be used to develop a diving heuristic, which consists of a searching depth-first heuristic. Different from how the exact approach explores the tree, in a diving heuristic we do not need to be concerned in balancing the tree. At each node, a branch is heuristically selected based on greedy or rounding strategies. For additional details, see [Joncour et al., 2010].

2.3 Branch-and-price

Branch-and-price [Barnhart et al., 1998, 2000] considers a combination of the Branch-and-bound and Delayed Column Generation. In a simple view, the procedure consists in applying the Delayed Column Generation method to derive lower bounds to be used during the enumerative search. As mentioned before, the Delayed Column Generation is expected to provide strong LP bounds through the Branch-and-bound search, allowing a higher number of subproblems to be pruned by bounds. The fewer created subproblems, more efficient the algorithm tends to be.

A Branch-and-price algorithm starts solving the linear relaxation of the root node of the Branch-and-bound tree through the Delayed Column Generation procedure. As mentioned in the last section, when no more columns with negative reduced costs are found, the LP relaxation of the problem has been computed in the root node. If the LP solution is integer, it also solves the original integer problem. Otherwise, being fractional, we must resort to branching.

It should be clear that applying a traditional Branch-and-bound algorithm for the RMP obtained at the end of the column generation approach does not guarantee that an optimal (nor even feasible) solution to the problem will be found. In contrast to that, we embed the whole column generation procedure in an Branch-and-bound framework, leading to a BP algorithm, where new columns are likely to be generated at each node in the enumeration tree.

One key issue in the implementation of BP algorithms is how branching is performed [Vanderbeck, 2010; Koch et al., 2004]. Since at each node of the enumeration tree the pricing subproblems are called repeatedly and solving them accounts for most of the computing times in BP algorithms, ideally, the *branching rule* should not destroy the structure of the pricing subproblems. Another aspect while branching, involves how variables are chosen. Once the branching rule over a certain variable is defined, we must adopt a *branching policy*, in order to choose which variable on which to branch. This choice can be based on the fractional variable farthest or closest to integrality, or even at random.

Finally, in order to choose a node to explore from the list of outstanding nodes in the branching tree, many approaches may be considered. The main *node selection policies* include the breadth first, depth first and best bound. In the breadth first, the nodes of the tree are explored in the same order in which they were created. Depth first selection policy explores the last node created, going deeper in the BB tree. The best bound or best first chooses the node having the lowest value (in a minimization problem) of the LP relaxation among all BB nodes.

For a more complete review regarding Branch-and-price implementations, see [Barnhart et al., 1998, 2000; Desrosiers and Lübbecke, 2005].

2.4 Main Concepts on Communication Networks

Many network measurements have been established aiming to characterize the performance and general operational conditions of a network. The main metrics are closely related to the communication in the the network, i.e., the exchange of information (or packets). Some of them are listed in the following.

- *Delay*: time interval elapsed between the data departure time from the source vertex to the arrival time at the destination vertex.
- *Jitter*: time variation among packets arriving at a destination vertex.
- *Throughput*: data rate supported by the network.
- *Loss*: amount of data that did not reach its destination vertex.

The concept of quality of service emerged in an attempt to provide guarantees of performance in communication networks. We present two important requirements in the QoS context:

- *Serviceability*: the ability of a service to be obtained when requested, and continue to be provided without excessive impairment for a request. It can be associated to the former metrics presented: delay, jitter, throughput, congestion and so on.

- *Survivability*: the ability of a system to continue to function during and after a natural or man-made disturbance. It is usually associated to fault tolerance mechanisms. In this case, a system is said to be resilient and reliable. Otherwise, vulnerability begins to be an issue.

In order to provide survivability in a network, some mechanisms have been proposed.

- *Protection*: uses pre-assigned capacity to ensure survivability. Two schemes can be defined:
  - *Dedicated*: all the traffic is allocated twice in the network capacity, i.e., different protection paths do not share common resource.
  - *Shared*: multiple protection paths may go through common resources; when one protection path is activated, other protection paths that share common resources with it may have to be rerouted.
- *Restoration*: reroutes the affected traffic after failure occurrence by using available capacity.

In the following, we present the main concepts in Optical Networks:

- *Wavelength Division Multiplexing (WDM)*: is a technology which multiplexes a number of optical carrier signals onto a single optical fiber by using different wavelengths.
- *Lightpath*: sequence of optical "hops" defined by a physical path through which the optical signal bypasses intermediate nodes, creating a virtual connection between its end nodes.
- *Routing and Wavelength Assignment (RWA)*: problem of setting up lightpaths by routing and assigning a wavelength to each request that must be attended by the network.
- *Traffic Grooming*: technique to group several requests for traffic on the same wavelength.
  - *Static*: considers the case in which all requests are known in advance and do not change by long periods of time.
  - *Dynamic*: demands appear dynamically, according to a probability distribution.

Chapter 3

Resilient Multi-level Hop-constrained Network Design

In this chapter we investigate the Resilient Multi-level Hop-constrained Network Design (RMHND) problem, which consists of designing hierarchical telecommunication networks, assuring resilience against random failures and maximum delay guarantees in the communication. Three mathematical formulations are proposed and algorithms based on the proposed formulations are evaluated.

3.1 Introduction

In the telecommunication context, we have been facing evolving technologies which offer a variety of services such as audio, video and general data transmission. The increasing demand from customers and the concern about service reliability in order to avoid customer dissatisfaction require grades of organization, leading to a hierarchical telecommunication network. The basic idea is that priorities are associated to sets of customers so that network providers can offer higher levels of service in both serviceability (e.g., low delay, high bandwidth) and survivability (failure protection) to certain key customers.

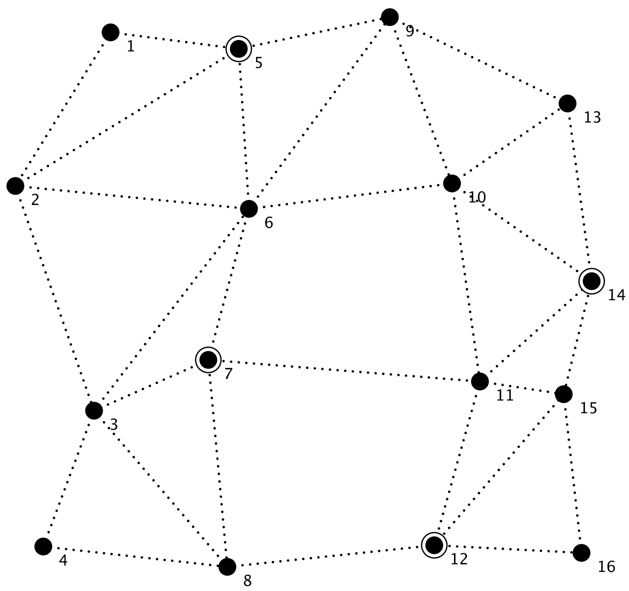
The importance of telecommunication networks has dramatically increased over the past few years. Consequently, Operations Research techniques and solutions have been playing a fundamental role across a wide range of telecommunication problems [Magnanti and Wong, 1984; Resende and Pardalos, 2005]. Considering the set of requirements to be met while designing such networks, the employment of an optimization phase is certainly of great importance.

This work aimed to study the application of Integer Programming techniques applied to the Resilient Multi-level Hop-constrained Network Design. The Multi-level Network Design (MLND), introduced by Balakrishnan et al. [1991], generalizes several well-known optimization models and addresses design decisions for hierarchical telecommunication, transportation, and electric power distribution networks. The nodes in this network have different levels of importance (priorities are associated to the nodes), requiring different levels of technologies in communication. Given an undirected graph whose nodes are partitioned into L levels, each edge can contain one of the L different levels of technology, with higher level requiring higher fixed costs. The goal is to select a connected subset of edges assigned to a technology level so that each source-destination pair communicate via its minimal necessary level or higher level, minimizing the total cost of the assigned technologies. We extended this problem to the RMHND, such that the subgraph implied by the selected edges provides two edge disjoint paths with up to H hops for each pair of nodes, i.e., it assures single-edge failure resilience and a maximum delay.

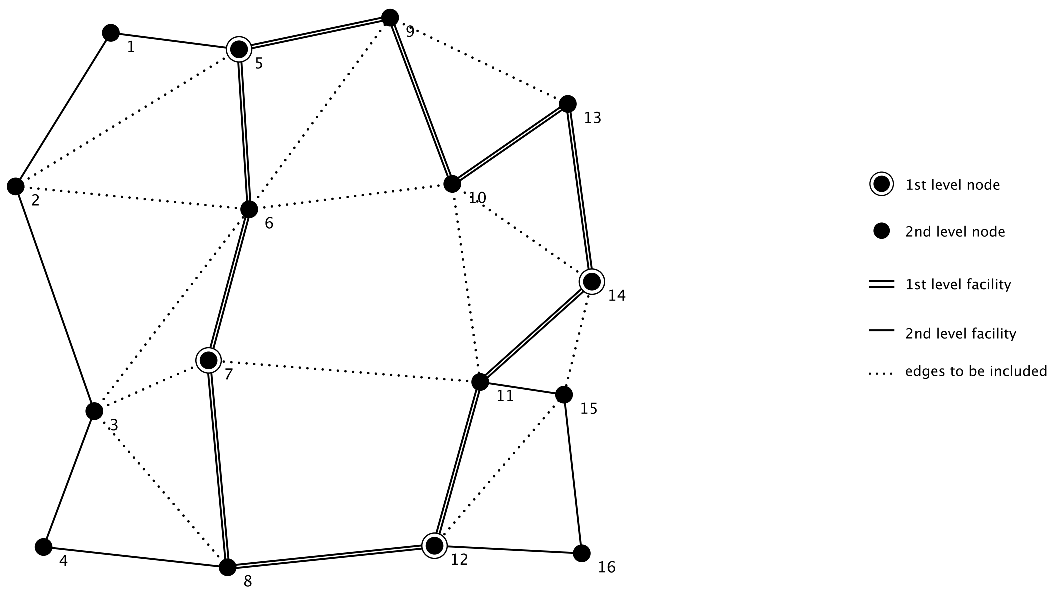
In the present study, we devise two compact formulations for the RMHND whose solution is accomplished by solving a Mixed Integer Program (MIP) to optimality through a Branch-and-Bound solver. Given the complexity of the problem, the main drawback of both compact formulations of a MIP is certainly the very limited size of the instances likely to be solved. In order to overcome this, we propose a column generation embedded in a Branch-and-bound algorithm, leading an exact Branch-and-price algorithm which has been applied successfully in the solution of large-scale problems [Lasdon, 1970; Barnhart et al., 1998].

Figure 3.1 shows a RMHND example for a network with 16 nodes, two sets of nodes and edges (two different levels) and a hop constraint where $H = 9$. Figure 3.1(a) depicts the original network, in which dotted lines represent the edges that might be included in the solution. Figure 3.1(b) displays the resulting network for the RMHND. Let us consider for instance, pair $(5, 14)$, where nodes 5 and 14 belong to the first level (the most priority level). Note that two edge disjoint paths using only first level technology edges are available, one with 4 hops (5-9-10-13-14) and another with 6 hops (5-6-7-8-12-11-14). Taking pair $(2, 16)$, where both nodes belong to the second level, we can also observe two edge disjoint paths. The first path, represented by (2-3-4-8-12-16) with 5 hops and the second path (2-1-5-9-10-13-14-11-15-16) with 9 hops, cross edges of its minimal required level (second one) or a higher level (the first level). It is important to point out that two edge disjoint paths are available for any pair of nodes.

The remainder of this chapter is organized as follows. In Section 3.2, we review some existing approaches in the literature to design multi-level networks. In Section 3.3, we introduce the Integer Programming formulations while, in Section 3.4, the



(a) Original network



(b) Solution network

Figure 3.1. RMHND Problem

Branch-and-price algorithm is described. Section 3.5 is devoted to present the equivalence between two formulations for the problem. Our computational experiments are presented and discussed in Section 3.6. Finally, the conclusion and future directions are given in Section 3.7.

3.2 Related Work

The Multi-level Network Design has theoretical importance since it generalizes the well-known Steiner Tree [Chopra and Rao, 1994] problem and the Hierarchical Network Design (HND) problem - defined by Current et al. [1986], which designates exactly two nodes of the network in the first level (also referred to as primary nodes) and the others in the second level. The MLND was first introduced by Balakrishnan et al. [1994b], which describes alternative model formulations for the problem and analyzes the worst-case performance for heuristics based upon Steiner and spanning tree computations. For the HND special case with only two primary nodes, the worst-case performance ratio of the heuristic is $4/3$. For the general case, the composite heuristic's worst-case performance ratio of $r+1$ depends on the worst-case performance ratio r of any Steiner network heuristic.

By the same time, Balakrishnan et al. [1994a] have developed an optimization-based heuristic methodology for solving the MLND problem. This method first applies certain preprocessing tests to reduce the problem by eliminating or installing primary or secondary technologies before solving the problem. The core of the method consists of a dual ascent algorithm to generate good linear programming based lower bounds and heuristic upper bounds. Computational results on large-scale problems (containing up to 500 nodes and 5000 edges) show that the method provides very good approximated solutions (guaranteed to be within 0.9% from optimality).

In [Balakrishnan et al., 1998], the survivability idea is incorporated to the MLND. For this, backup paths are provided for pairs of nodes belonging to the primary level. The authors propose and analyze the worst-case performance of tailored heuristics for several special cases of the two level problem. Depending upon the particular problem setting, the heuristics have worst-case performance ratios ranging between 1.25 and 2.6.

Formulations for the Two Level Network Design (TLND) - a particular case of MLND - are discussed by Gouveia and Telhada [2001]. The authors present an augmented arborescence formulation combining a directed formulation for the Steiner tree problem with a directed formulation for the spanning tree problem. They show that the linear programming value of the new formulation is proved to be theoretically weaker than the LP bound given by a flow based formulation, although for certain classes of instances the two LP bounds are quite close. In addition, a Lagrangian based relaxation is presented, where an arborescence minimization problem is solved as the relaxed subproblem. Computational results indicate that the Lagrangian relaxation based method is quite efficient, providing a reasonable alternative to handle the problem.

As an extension of [Gouveia and Telhada, 2001], a new formulation for the Multi-Weighted Steiner Tree (a general case of TLND) problem is proposed in [Gouveia and Telhada, 2008]. In the previous work, the authors propose a non-symmetric formulation for the problem in the sense that depending on the node selected as a root of the tree, the corresponding LP bounds could vary. Thus, a reformulation by intersection is proposed, obtained by the intersection of feasible sets of the models corresponding to each root selection for the problem. It is shown that the linear programming relaxation of the reformulation dominates the linear programming relaxation of the previous formulation for all possible root selections. They also present a Lagrangian relaxation scheme derived from the reformulation, with quite favourable results, on instances with up to 500 nodes and 5000 edges.

An optimal approach for the HND problem is devised in [Obreque et al., 2010] using a Branch-and-cut procedure. They show how to find valid cuts and how their separation works. A Branch-and-bound procedure completes the algorithm when no more cuts can be added. Computational results show that large instances are likely to be solved in a short CPU time.

To the best of our knowledge, it was not found in the literature the problem of Multi-level Network Design with hop constraints and resilience against single-edge failure, as proposed in our work.

3.3 Mathematical Formulations

Let us now formally introduce the Resilient Multi-level Hop-constrained Network Design problem. For the sake of convenience, we define our problem over a directed graph to allow an easier transition over the different formulations. Given a directed graph $D = (N, A)$ with set of nodes N and arcs A, installation costs $\{c_{ij}^g = c_{ji}^g \geq 0 : \forall (i,j) \in A, \forall g \in G\}$ assigned to the arcs of A, where G denotes the set of different technologies, numbered from 1 to $|G|$. A technology g on arc (i,j) costs c_{ij}^g, with $c_{ij}^g > c_{ij}^{g'}$ if $g < g'$.

Let $K \subseteq N \times N$ be a set of requests; $K = \bigcup_{g \in G} K_g$, where K_g is the set of requests with technology g. The technology g of a request k is defined by the lowest level of the end-nodes of k. This means that the communication for request k must take place in the highest technology between its end-nodes. The RMHND problem consists in finding a subset of arcs $S \subseteq A$, such that each arc is assigned to exactly one technology and the subgraph of D implied by (N, S) provides two arc disjoint paths with length at most H for each request.

Three Integer Programming Formulations for RMHND are presented here. In the first two formulations, named Arc-Flow Formulation and Aggregated Hop-Indexed Formulation, connectivity between each pair of nodes is enforced through network flow [Ahuja et al., 1993] arguments. In the third one, Arc-Path Formulation, connectivity is guaranteed by imposing that two arc disjoint paths connecting every pair of nodes must be available in the subgraph of D implied by the selected arcs. These formulations are discussed next.

3.3.1 Arc-Flow Formulation (AFF)

Let us assume that, given $D = (N, A)$, A_j^- and A_j^+ respectively denote the set of arcs arriving and leaving $j \in N$. To model RMHND, we make use of the following sets of decision variables: (i) $x_{ij}^{kp} \in \mathbb{R}_+$, indicating the amount of flow of request k that passes through arc (i, j) which composes path $p \in \{1, 2\}$; (ii) y_{ij}^g, taking value 1 if arc $(i, j) = (j, i)$ is assigned to technology g and therefore included in the solution (0, otherwise). RMHND can now be stated as:

$$\min \sum_{(i,j)\in A: i<j} \sum_{g\in G} c_{ij}^g y_{ij}^g \tag{3.1}$$

s.t.

$$\sum_{p\in\{1,2\}}\sum_{j\in A_s^+} x_{sj}^{kp} \geq 2 \qquad \forall k \in K, s = source(k) \tag{3.2}$$

$$\sum_{p\in\{1,2\}}\sum_{i\in A_t^-} x_{it}^{kp} \geq 2 \qquad \forall k \in K, t = dest(k) \tag{3.3}$$

$$\sum_{i\in A_j^-} x_{ij}^{kp} - \sum_{l\in A_j^+} x_{jl}^{kp} = 0 \qquad \forall k \in K, \forall j \in N, j \neq source(k) \neq dest(k), \forall p \in \{1,2\}, \tag{3.4}$$

$$\sum_{g\in G} y_{ij}^{g} \leq 1 \qquad \forall (i,j) = (j,i) \in A, \tag{3.5}$$

$$\sum_{p\in\{1,2\}} x_{ij}^{kp} \leq \sum_{g'\leq g(k)} y_{ij}^{g'} \qquad \forall k \in K, \forall (i,j) = (j,i) \in A, \forall p \in \{1,2\}, \tag{3.6}$$

$$\sum_{(i,j)\in A} x_{ij}^{kp} \leq H \qquad \forall k \in K, \forall p \in \{1,2\}, \tag{3.7}$$

$$y_{ij}^{g} = y_{ji}^{g} \qquad \forall (i,j) \in A, \forall g \in G, \tag{3.8}$$

$$0 \leq x_{ij}^{kp} \leq 1 \qquad \forall k \in K, \forall (i,j) \in A, \forall p \in \{1,2\}, \tag{3.9}$$

$$y_{ij}^{g} \in \{0,1\} \qquad \forall (i,j) \in A, \forall g \in G. \tag{3.10}$$

Constraints (3.2)-(3.4) are flow balance constraints for each request k. Note that inequalities(3.2) imply that at least two units of flow will leave the source node of request k, while inequalities (3.3) indicate that every unit of flow for request k will arrive its destination node. Constraints (3.4) guarantee the flow conservation in transshipment nodes. Inequalities (3.5) assure that on each arc $(i,j) \in A$ at most one of the technologies is installed. Inequalities (3.6) couple flow and design variables, imposing that $g(k)$ flows are only allowed to cross g' arcs (arcs with higher technologies), where $g' \leq g(k)$. Note that inequalities (3.5) along with inequalities (3.6) impose that paths $p \in \{1,2\}$ are arc disjoint. Constraints (3.7) assure that paths implied by the flows are no longer than H. Constraints (3.8) impose whenever arc (i,j) is selected to be in a solution, so is (j,i). Finally, objective function (3.1) minimizes the total cost of selected arcs. Note that whenever arcs (i,j) and (j,i) are included in the solution, the cost c_{ij}^g is considered only once in the objective function.

Formulation (3.1)-(3.10) has $O(n^4)$ variables and constraints and therefore, only RMHND instances of limited size are expected be solved to proven optimality by BB algorithms based on it. Moreover, as we will see in the following, the LP bounds provided by the Arc-Flow Formulation may be improved by indexing flow variables with a hop counter, since constraints (3.7) are considered weak when we are dealing with the LP relaxation of the problem.

3.3.2 Aggregated Hop-Indexed Formulation (AHF)

In this formulation, previous constraints (3.7) are treated by an additional index on flow variables, as proposed in [Gouveia et al., 2006], except that we do not make use of "loop" variables. Through this indexing strategy, even when we are dealing with the LP relaxation of the problem, the paths imposed by the flows are guaranteed to have no more than H hops, making the bounds stronger than in the former formulation.

In addition to variables y_{ij}^g previously defined, our second formulation uses variables w_{ij}^{hk} indicating the amount of flow of request k that crosses arc (i,j) in the h^{th} hop of one of the paths. Thus, the problem can be stated as:

$$\min \sum_{(i,j)\in A: i<j} \sum_{g\in G} c_{ij}^g y_{ij}^g \tag{3.11}$$

s.t.

$$\sum_{j\in A_s^+} w_{sj}^{1k} \geq 2 \qquad \forall k \in K, s = source(k) \tag{3.12}$$

$$\sum_{h=1}^{H} \sum_{i\in A_t^-} w_{it}^{hk} \geq 2 \qquad \forall k \in K, t = dest(k) \tag{3.13}$$

$$\sum_{i\in A_j^-} w_{ij}^{hk} - \sum_{l\in A_j^+} w_{jl}^{h+1k} = 0 \qquad \forall k \in K, \forall j \in N, j \neq source(k) \neq dest(k),$$

$$h = \{1, ..., H-1\}, \tag{3.14}$$

$$\sum_{g\in G} y_{ij}^g \leq 1 \qquad \forall (i,j) = (j,i) \in A, \tag{3.15}$$

$$\sum_{h=1}^{H} w_{ij}^{hk} \leq \sum_{g'\leq g(k)} y_{ij}^{g'} \qquad \forall k \in K, \forall (i,j) = (j,i) \in A, \tag{3.16}$$

$$y_{ij}^g = y_{ji}^g \qquad \forall (i,j) \in A, \forall g \in G, \tag{3.17}$$

$$0 \leq w_{ij}^{hk} \leq 1 \qquad \forall k \in K, \forall (i,j) \in A, h = \{1, ..., H\}, \tag{3.18}$$

$$y_{ij}^g \in \{0,1\} \qquad \forall (i,j) \in A, \forall g \in G. \tag{3.19}$$

Flow balance constraints for each request k are given by constraints (3.12)-(3.14). Inequalities (3.12) ensure that at least two units of flow will leave the source node of request k, indexed by the 1^{st} hop. Inequalities (3.13) guarantee that every unit of flow for request k will arrive its destination node disregarding the number of hops used to reach it. Constraints (3.14) guarantee the flow conservation in transshipment nodes, assuring that each unit of flow leaving a node has an incremented hop index than in its arrival. Inequalities (3.15) assure that on each arc $(i, j) \in A$ at most one of the technologies is installed. Inequalities (3.16) couple flow and design variables, as in the previous formulation. Note that inequalities (3.15) along with inequalities (3.16) impose that the paths implied by the flows are arc disjoint. Constraints (3.18) impose that arcs (i, j) and (j, i) are symetric. The objective function (3.11) minimizes the total cost of selected arcs, as in the Arc-Flow Formulation.

Formulation (3.11)-(3.19) also has $O(n^4)$ constraints and far more variables ($O(n^5)$) than the Arc-Flow Formulation. Although the LP bounds provided by this formulation are tighter, once again, the size of RMHND instances expected to be solved is limited.

A third formulation, named Arc-Path Formulation is presented in the sequence. Despite having exponentially many variables, this formulation is suitable for the implementation of a Branch-and-price [Barnhart et al., 1998] method.

3.3.3 Arc-Path Formulation (APF)

In this formulation, we assume that P^k denotes the set of admissible directed paths connecting the endpoints of request k in D, including a hop constraint ($length(p) \leq H : p \in P^k$). The main idea of the Arc-Path Formulation to enforce connectivity and single arc failure resilience is to impose that, at least two paths connecting every request in the corresponding or higher level technologies must be available in the subgraph implied by the selected assignment whose cost we aim to minimize.

Assume that $a^p_{ij} \in \{0, 1\}$ indicates that arc (i, j) or (j, i) belongs to path p taking value 1 (0, otherwise). In addition to variables y^g_{ij} previously defined, we make use of the following set of decision variables: λ^k_p taking value 1 if path $p \in P^k$ is selected for request k (0, otherwise). The Arc-Path Formulation for RMHND is given by the Integer Program:

$$\min \sum_{(i,j) \in A: i<j} \sum_{g \in G} c^g_{ij} y^g_{ij} \tag{3.20}$$

s.t.

$$\sum_{p \in P^k} \lambda_p^k \geq 2 \qquad \forall k \in K, \tag{3.21}$$

$$\sum_{g \in G} y_{ij}^g \leq 1, \qquad \forall (i,j) \in A : i < j, \tag{3.22}$$

$$\sum_{p \in P^k} a_{ij}^p \lambda_p^k \leq \sum_{g' \leq g(k)} y_{ij}^{g'}, \qquad \forall k \in K, \forall (i,j) \in A : i < j, \tag{3.23}$$

$$\lambda_p^k \in \{0,1\} \qquad \forall k \in K, \forall p \in P^k, \tag{3.24}$$

$$y_{ij}^g \in \{0,1\} \qquad \forall (i,j) \in A : i < j, \forall g \in G. \tag{3.25}$$

The objective function (3.20) minimizes the total cost of the selected arcs. Constraints (3.21) ensure that at least two paths connecting each request will be selected. Inequalities (3.22) assure that on each arc $(i,j) \in A$ at most one technology is installed. Inequalities (3.23) imply that $g(k)$ paths are only allowed to cross g' arcs, where $g' \leq g(k)$, i.e., arcs with higher technologies. Following the same idea of the other formulations, inequalities (3.22) along with inequalities (3.23) guarantee that the selected paths are arc disjoint.

3.3.4 Considerations

For solving RMHND by a LP based BB algorithm, we have chosen the state-of-the-art commercial solver CPLEX [2011]. The advantage of the LP based BB approach is that little programming effort is needed, once one has in hand an Integer Programming solver like CPLEX.

However, the Arc-Flow Formulation and the Aggregated Hop-Indexed Formulation have $O(n^4)$ variables and constraints and $O(n^5)$ variables and $O(n^4)$ constraints, respectively. Therefore, even with the help of a highly sophisticated optimization package like CPLEX, only RMHND instances of limited size are expected be solved to proven optimality by LP based BB algorithms that rely on these formulations. Consequently, the drawback of this approach is that, in practice, only limited size instances of RMHND can be actually solved in a reasonable amount of time.

In order to overcome this difficulty, in the following, we present a Branch-and-price algorithm based on a reformulation for RMHND (Arc-Path Formulation). Roughly speaking, the procedure consists in applying the Delayed Column Generation method to derive lower bounds to be used in the search.

3.4 Branch-and-price Algorithm for the RMHND

The exponentially many columns in (3.20)-(3.25) does not refrain us to use the Arc-Path Formulation in an exact BB approach to solve RMHND. That could be attained by embedding a Delayed Column Generation in a BB algorithm.

3.4.1 Lower Bounds given by the Arc-Path Formulation

To understand how the LP bounds given by (3.20)-(3.25) are evaluated, let $\pi^k \geq 0$, $\beta_{ij} \leq 0$ and $\gamma_{ij}^k \leq 0$ be the dual variables associated with constraints (3.21), (3.22) and (3.23), respectively. The dual problem associated to the LP relaxation of (3.20)-(3.25) is given by:

$$\max \sum_{k \in K} 2\pi^k + \sum_{(i,j) \in A : i<j} \beta_{ij} \tag{3.26}$$

s.t.

$$\beta_{ij} - \sum_{g' \geq g} \sum_{k \in K_{g'}} \gamma_{ij}^k \leq c_{ij}^g \qquad \forall (i,j) \in A : i < j, \forall g \in G, \tag{3.27}$$

$$\pi^k + \sum_{(i,j) \in A : i<j} a_{ij}^p \gamma_{ij}^k \leq 0, \qquad \forall k \in K, \forall p \in P^k, \tag{3.28}$$

$$\pi^k \geq 0 \qquad \forall k \in K, \tag{3.29}$$

$$\beta_{ij}^k \leq 0 \qquad \forall k \in K, \forall (i,j) \in A, \tag{3.30}$$

$$\gamma_{ij}^k \leq 0 \qquad \forall k \in K, \forall (i,j) \in A. \tag{3.31}$$

$$\tag{3.32}$$

The LP relaxation of (3.20)-(3.25) can be computed as follows. Assume that sets of simple directed paths $C^k \subset P^k, \forall k \in K$ ($|C^k| << |P^k|$) are made available. Also, assume that the Restricted Master Problem

$$\min \sum_{(i,j) \in A : i<j} \sum_{g \in G} c_{ij}^g y_{ij}^g \tag{3.33}$$

s.t.

$$\sum_{p \in C^k} \lambda_p^k \geq 2 \qquad \forall k \in K, \tag{3.34}$$

$$\sum_{g \in G} y_{ij}^g \leq 1, \qquad \forall (i,j) \in A : i < j, \tag{3.35}$$

$$\sum_{p \in C^k} a_{ij}^p \lambda_p^k \leq \sum_{g' \leq g(k)} y_{ij}^{g'}, \qquad \forall k \in K, \forall (i,j) \in A : i < j, \tag{3.36}$$

$$\lambda_p^k \geq 0 \qquad \forall k \in K, \forall p \in C^k, \tag{3.37}$$

$$y_{ij}^g \geq 0 \qquad \forall (i,j) \in A : i < j, \forall g \in G. \tag{3.38}$$

has one basic feasible solution $\hat{y}, \hat{\lambda}$. Let $\hat{\pi}, \hat{\beta}, \hat{\gamma}$ be the corresponding dual optimal solution. If, for all requests k, no path $p \in P^k \setminus C^k$ violates the dual constraints

$$\hat{\pi}^k + \sum_{(i,j) \in A : i < j} a_{ij}^p \hat{\gamma}_{ij}^k \leq 0, \tag{3.39}$$

then, $\hat{y}, \hat{\lambda}$ solves the LP relaxation of (3.20)-(3.25) and the corresponding optimal LP function gives a lower bound on (3.20). Otherwise, for a given resquest k there must be a path in $P^k \setminus C^k$ that violates (3.39) that must be included in C^k. The new RMP, enlarged with the sets of paths associated to violated constraints (3.39), is re-optimized. The procedure goes on, until no inequality (3.39) is violated.

In our implementation, the initial sets C^k are generated as follows. In order to guarantee a feasible solution for the problem, two arc disjoint paths must be provided for each request $k \in K$. Therefore, we computed the first two minimum disjoint paths connecting request k in terms of number of hops (seeking a feasible solution). Whenever the length of the maximal path in the generated sets is no greater than H, (3.20)-(3.25) admits at least one solution. Otherwise, there is no feasible solution for the problem.

The pricing problem, consists in finding a path $p \in P^k \setminus C^k$ that violates (3.39) or proving that such path does not exist. Therefore, if there is no path for any request $k \in K$ yielding negative reduced costs $\bar{c}_p^k = -\hat{\pi}^k - \sum_{(i,j) \in A : i < j} a_{ij}^p \hat{\gamma}_{ij}^k$, the LP relaxation is solved. Since $\hat{\gamma}_{ij}^k \leq 0, \forall k \in K, \forall (i,j) \in A : i < j$ (and therefore $-\hat{\gamma}_{ij}^k \geq 0$), the pricing problems could be solved by any shortest path algorithm with resource constraints (note that in fact, it is a "longest" path, since $\hat{\gamma}_{ij}^k \leq 0$). In order to deal with the subproblems, we use the algorithm proposed by Feillet et al. [2004] to solve the associated Elementary Shortest Path Problem with Resource Constraints.

3.4.2 The Enumeration Tree

As previously described, when no more columns with negative reduced costs are found, the associated solution corresponds to the optimal value for the LP relaxation of the root node. If the LP solution is integer, it also solves the original problem (3.20)-(3.25). Otherwise, being fractional, an enumeration algorithm is required.

One important issue in the implementation of BP algorithms is how branching is performed. Since the pricing subproblems are iteratively solved along the BP procedure, ideally, the branching rule should not destroy the structure of the pricing subproblems. In this case however, it is irrelevant, as we use an exponential algorithm to deal with the subproblems, and the complexity will not increase.

We chose to branch on fractional y_{ij}^g variables. Next, we show that branching on variables y_{ij}^g will lead us to an optimal integer solution, and no branching in λ_p^k is needed, due to the property of elementary paths. Clearly, whenever λ^* is an optimal integral solution, variables y^* will also be. Although the contrary is not true, we can state that:

Proposition 1. *Given an optimal solution λ^*, y^* to RMHND with objective value O^*; if y^* is integral, λ^* is either integral or there exists an integral feasible solution with the same objective value.*

Proof. Suppose that y^* is integral. Let U_k^* be the set of variables λ for request k having positive values, i.e., $U_k^* = \{p \in P^k : \lambda_p^k > 0\}$. For each request k, if $|U_k^*| = 2$, there exist two non-zero variables assuming value 1, ensuring an integral solution. Otherwise ($|U_k^*| > 2$), one may select two variables among the elements of U_k^* and fix them to 1 and the others to 0. It is important to point out that as the paths represented by the selected variables must be arc disjoint, the selected couple cannot share common arcs. Once variables λ^* assume positive values, it is guaranteed that arcs belonging to the selected paths are included in the solution (remember that y^* is integral) and moreover, computed in the objective function. Therefore, we can obtain an integral solution with the same objective value O^*. To demonstrate that there exist two arc disjoint elementary paths among the elements of U_k^* for every request k, observe the following properties of the resulting subgraph implied by the solution of RMHND. Subgraph $D = (N, S)$ is bridgeless; otherwise, a flow of two units must be traversing the cut-edge (bridge) to guarantee the set of constraints, which is impossible since variables $y_{ij}^g \in \{0, 1\}$. Thus, we can state that $D = (N, S)$ is a two edge connected graph. According to Menger's theorem [1], for every pair source/destination there are

[1]Menger's theorem: Let G be a finite undirected graph and s and t two distinct vertices. The size

at least two edge disjoint paths in D. □

The idea of branching on y_{ij}^g variables is the creation of as many branches as the number of technologies $|G| + 1$. The first branch concerns the case in which all technologies g of arc $(i, j) = (j, i)$ are forbidden, i.e., arc $(i, j) = (j, i)$ is actually forbidden to belong to the solution. The other $|G|$ branches treat the cases where one of the technologies is imposed to be 1 and the remainder 0. The main advantage of this branching rule is that it does not affect the pricing subproblem, requiring only the elimination of the corresponding arc from the input graph when necessary.

The selection of the next fractional variable to branch on is performed at random. Moreover, the Branch-and-bound tree is explored choosing a node according to a *best bound policy*. According to our experiments, the best bound policy presents a better performance than the breadth first or depth first ones.

3.4.3 Stabilization and Acceleration Strategies

In order to evaluate the behaviour of our algorithm under stabilization, we apply an interior point stabilization method proposed by Rousseau et al. [2007]. The main idea of this method is to select a dual solution inside the optimal space rather than retrieving an extreme point. To obtain a point inside the polyhedron, one can simply define a random objective function and solve the associated dual problem imposing complementary slackness conditions. If the problem is solved with a simplex method, then the optimal solution obtained will always be an extreme point. Solving the dual problem for a number of random values, we obtain a set of extreme points, whose convex combinations will lead to an interior point that gives much more centered dual values. The only parameter to set is the number of points to identify in order to calculate an interior point of the dual optimal space. The trade-off is that a larger number of points will produce a more centered point, but also requires more computing time (since a LP has to be solved for each point).

Evaluating our experiments, it is observed that although the number of iterations of the procedure indeed decreases, the computational times are not improved. Different values for the number of points to be calculated were tested, but no advantage was achieved. The reason for that consists in the fact that the bottleneck of our problem is the time spent while solving the LP. Since the stabilization method is based on a number of runs of the associated dual LP, the gain obtained in the number of iterations of the column generation does not pay off the time spent to compute better dual values.

of the minimum edge cut for s and t (minimum number of edges whose removal disconnects s and t) is equal to the maximum number of pairwise edge independent paths from s to t.

A common acceleration strategy in column generation concerns the addition of a large set of good columns in the beginning of the procedure. As presented in Section 3.4.1, the set of initial paths for each request is initialized with only two arc disjoint paths. A larger set of paths was also tested to initialize the algorithm, but according to our computational results, this strategy does not improve our current solutions.

3.5 Equivalence of the LP Relaxations of AHF and APF

Theorem 1. *The LP relaxations of AHF and APF provide the same solution set and objective function values for RMHND.*

The proof for the theorem is based on showing that for each continuous solution to AHF (APF) there is a corresponding solution in APF (AHF) and both of them lead to the same continuous value for RMHND. Let F(.) be the feasible solution set for problem (.) and AHF_{LP} and APF_{LP} denote the LP relaxations of AHF and APF, respectively. Let the solutions in AHF_{LP} and APF_{LP} be represented by

$(w, y) \equiv (w_{ij}^{hk}, y_{ij}^{g}) \forall k \in K, \forall h \in I, \forall (i,j) \in A, \forall g \in G$ and

$(\lambda, \bar{y}) \equiv (\lambda_p^k, \bar{y}_{ij}^{g}) \forall k \in K, \forall p \in P^k, \forall (i,j) \in A : i < j, \forall g \in G,$

respectively. For $(w, y) \in \mathrm{F}(AHF_{LP})$ and $(\lambda, \bar{y}) \in \mathrm{F}(APF_{LP})$, consider $Z_{AHF}(w, y)$ and $Z_{APF}(\lambda, \bar{y})$ the corresponding linear objective function values of AHF and APF, respectively.

Proof. The equivalence of $\mathrm{F}(AHF_{LP})$ and $\mathrm{F}(APF_{LP})$ and the equality of the continuous solutions of AHF_{LP} and APF_{LP} are shown in three parts. □

Part 1. To show $\mathrm{F}(APF_{LP}) \subseteq \mathrm{F}(AHF_{LP})$, we take a solution $(\lambda, \bar{y}) \in \mathrm{F}(APF_{LP})$ and construct a solution $(w, y) \in \mathrm{F}(AHF_{LP})$.

Let b_{ij}^{hp} be a binary constant assuming value 1 if arc (i, j) is the h-th arc belonging to path p and 0, otherwise. It is easy to see that the binary constant a_{ij}^p presented in APF can be given by

$$a_{ij}^p = \sum_{h \in I} b_{ij}^{hp}, \tag{3.40}$$

since paths are acyclic. Given $(\lambda, \bar{y}) \in \mathrm{F}(APF_{LP})$, (w, y) can be constructed as follows.

$$w_{ij}^{hk} = \sum_{p \in P^k} b_{ij}^{hp} \lambda_p^k \qquad \forall k \in K, \forall (i,j) \in A, \forall h \in I \tag{3.41}$$

$$y_{ij}^g = \bar{y}_{ij}^g \qquad \forall (i,j) \in A, \forall g \in G \tag{3.42}$$

Claim 1. (w, y) constructed through Eqs. (3.41)-(3.42) is in $\mathrm{F}(AHF_{LP})$.

Proof of Claim 1. We show that (w, y) constructed through Eqs. (3.41)-(3.42) satisfies constraints (3.12)-(3.17).

(a) By substituting (3.41) into constraints (3.12), we get

$$\sum_{j \in A_s^+} \sum_{p \in P^k} b_{sj}^{1p} \lambda_p^k \geq 2, s = source(k)$$

As only one arc (s, j) can be the first arc of path p, the sum of binary constants b is said to be equal to 1 ($\sum_{j \in A_s^+} b_{sj}^{1p} = 1$).

Thus, we get

$$\sum_{p \in P^k} \lambda_p^k \geq 2 \qquad \forall k \in K$$

Since $(\lambda, \bar{y}) \in \mathrm{F}(APF_{LP})$, these inequalities hold by (3.21).

(b) By substituting (3.41) into constraints (3.13), we have

$$\sum_{h \in I} \sum_{i \in A_t^-} \sum_{p \in P^k} b_{it}^{hp} \lambda_p^k \geq 2, t = destination(k)$$

$$\equiv \sum_{i \in A_t^-} \sum_{p \in P^k} a_{it}^p \lambda_p^k \geq 2 \qquad \{\text{by (3.40)}\}$$

As exactly one arc (i, t) arriving in t must belong to path p, the sum of a is equal to 1 ($\sum_{i \in A_t^-} a_{it}^p = 1$).

Thus, we have

$$\sum_{p \in P^k} \lambda_p^k \geq 2 \qquad \forall k \in K$$

Since $(\lambda, \bar{y}) \in \mathrm{F}(APF_{LP})$, these inequalities hold by (3.21).

(c) By substituting (3.41) into constraints (3.14), we get

$$\sum_{i\in A_j^-}\sum_{p\in P^k} b_{ij}^{hp}\lambda_p^k - \sum_{l\in A_j^+}\sum_{p\in P^k} b_{jl}^{h+1p}\lambda_p^k = 0$$
$$\equiv \sum_{p\in P^k}\lambda_p^k\left(\sum_{i\in A_j^-} b_{ij}^{hp} - \sum_{l\in A_j^+} b_{jl}^{h+1p}\right) = 0$$

We can state that $\sum_{i\in A_j^-} b_{ij}^{hp} \in \{0,1\}$ and $\sum_{l\in A_j^+} b_{jl}^{h+1p} \in \{0,1\}$. Then we have two situations: (i) node j belongs to path p (both summations are equal 1), (ii) otherwise (both summations are equal 0).

Therefore,

$$\sum_{p\in P^k}\lambda_p^k \times 0 = 0$$

Equalities (3.14) hold, as the equations are null by the substitution.

(d) Constraints (3.15) can be written as

$$\sum_{g\in G}\bar{y}_{ij}^g \leq 1 \qquad \forall (i,j)\in A : i<j$$

Since $(\lambda,\bar{y}) \in \mathrm{F}(APF_{LP})$, these inequalities hold by (3.22).

(e) Constraints (3.16) can be written as

$$\sum_{h\in I}\sum_{p\in P^k} b_{ij}^{hp}\lambda_p^k \leq \sum_{g'\leq g(k)} \bar{y}_{ij}^{g'}$$
$$\equiv \sum_{p\in P^k} a_{ij}^p\lambda_p^k \leq \sum_{g'\leq g(k)} \bar{y}_{ij}^{g'} \qquad \forall k\in K, \forall (i,j)\in A : i<j \qquad \{\text{by } (3.40)\}$$

Since $(\lambda,\bar{y}) \in \mathrm{F}(APF_{LP})$, these inequalities hold by (3.23).

(f) By substituting (3.42) into constraints (3.17), we get

$$\bar{y}_{ij}^g = \bar{y}_{ji}^g$$
$$\equiv \bar{y}_{ij}^g - \bar{y}_{ji}^g = 0, \forall (i,j)\in A, \forall g\in G$$

Equalities (3.17) hold, as the equations are null by the substitution.

With (a), (b), (c), (d), (e) and (f) we show that $(\lambda,\bar{y}) \in \mathrm{F}(AHP_{LP})$, which proves Claim 1. □

Part 2. To show that $\mathrm{F}(AHF_{LP}) \subseteq \mathrm{F}(APF_{LP})$, we take a solution $(w,y) \in \mathrm{F}(AHF_{LP})$ and construct a solution $(\lambda,\bar{y}) \in \mathrm{F}(APF_{LP})$.

Given $(w,y) \in \mathrm{F}(AHF_{LP})$, construct $(\lambda,\bar{y})$ in the following manner

$$\bar{y}_{ij}^{g} = y_{ij}^{g} \qquad \forall (i,j) \in A, \forall g \in G \tag{3.43}$$

The set of paths λ is constructed through a flow decomposition algorithm[2] [Ahuja et al., 1993] as follows

Algorithm 1 Create set of paths P

Input: $\hat{w}$
Output: P

```
for all request k do
    while there are paths for k with positive flow do
        1. Find a simple path p_u from source(k) to dest(k) as follows.
            a. Pick and arc leaving source(k) at the first hop with positive flow (w_{sj}^{1k} > 0).
            b. Once you have w_{ij}^{hk}, find the next hop arc with positive flow, say w_{jl}^{h+1k}.
            c. Repeat (b) until node t is reached.
        2. Let f_u > 0 be the amount of flow of path p_u, i.e, f_u = min{w_{ij}^{hk} : (i,j) ∈ p_u}.
        3. Anti-augment flow f_u from all arcs (i,j) ∈ p_u.
        4. Declare p_u as a path in P with value f_u and remove all zero residual flow arcs.
    end while
end for
return P
```

Considerations regarding the proposed algorithm:

- Step (1) of the algorithm guarantees that a simple path from $source(k)$ to $destination(k)$ with positive flow is found at each iteration, so that the constructed path has no more than H hops, since the procedure is based on hop-indexed variables w.
- Although step (3) reduces the amount of flow in arcs belonging to p_u, the flow balance is still ensured in the residual network.

Since at least one arc is removed at each iteration of the algorithm for a request k, we have at most $m = |A|/2$ iterations for each request until it terminates.

Claim 2. $(\lambda, \bar{y})$ constructed through Eq. (3.43) and Algorithm (1) is in F(APF_{LP}).

Proof of Claim 2. We show that $(\lambda, \bar{y})$ constructed through Eq. (3.43) and Algorithm (1) satisfies constraints (3.21)-(3.23). □

[2] The original flow decomposition algorithm is designed to decompose arc flows into path and cycle flows. However, in this work we deal only with simple paths.

(a) We show that constraints (3.21) hold based on a recurrence relation in terms of the number of iterations performed by Algorithm (1).

Let u be the number of iterations in Algorithm (1), where $1 \leq u \leq m$. Considering that at each iteration a portion of the total amount of flow is extracted from solution w to solution λ for a request k, we can say that in the end of the procedure the amount of flow transferred for request k will sum to 2. Thus, we can state that

$$T(u) = \begin{cases} f_1 & \text{if } u = 1 \\ T(u-1) + f_u & \text{otherwise} \end{cases} \tag{3.44}$$

By solving 3.44, we have

$$\begin{aligned} T(u) &= T(u-1) + f_u && (3.45)\\ &= T(u-2) + f_{u-1} + f_u && (3.46)\\ &\dots && (3.47)\\ &= T(1) + f_2 + \dots + f_{u-1} + f_u && (3.48)\\ &= f_1 + f_2 + \dots + f_{u-1} + f_u && (3.49)\\ &= \sum_u f_u && (3.50) \end{aligned}$$

Constraints (3.21) hold by substituting flow-paths by λ-paths as follows

$\sum_u f_u \geq 2$

$\equiv \sum_{p \in P^k} \lambda_p^k \geq 2 \quad \forall k \in K$

(b) Constraints (3.22) can be written as follows

$\sum_{g \in G} y_{ij}^g \leq 1 \quad \forall (i,j) \in A : i < j$

Since $(w, y) \in \mathrm{F}(AHF_{LP})$, these inequalities hold by (3.15).

(c) Following the same idea of (a), constraints (3.23) can be written by means of the constructed flow-paths. Let d_{ij}^u is a binary constant indicating whether arc (i,j) belongs to the flow-path u. Substituting the constructed flow paths by λ-paths, we have

$$\sum_{u} d_{ij}^{u} f_u \leq \sum_{g' \leq g(k)} y_{ij}^{g}$$
$$\equiv \sum_{p \in P^k} a_{ij}^{p} \lambda_p^k \leq \sum_{g' \leq g(k)} \bar{y}_{ij}^{g} \qquad \forall k \in K, \forall (i,j) \in A : i < j$$

Part 3. We show that the given $(\lambda, \bar{y}) \in \mathrm{F}(APF_{LP})$ and the constructed $(w, y) \in \mathrm{F}(AHF_{LP})$ lead to the same objective function value, i.e., $Z_{APF}(\lambda, \bar{y}) = Z_{AHF}(w, y)$.

Claim 3. $Z_{AHF}(w, y) = Z_{APF}(\lambda, \bar{y})$

Proof of Claim 3. By construction of $y_{ij}^{g} (\forall g \in G, \forall (i,j) \in A)$ (see (3.42)), $Z_{AHF}(w, y)$ can be rewritten as

$$Z_{AHF}(w, y) = \sum_{g \in G} \sum_{(i,j) \in A : i < j} c_{ij}^{g} \bar{y}_{ij}^{g}$$

This expression corresponds to $Z_{APF}(\lambda, \bar{y})$.

□

Corollary 1. *The partial LP relaxations under variables w and λ of AHF and APF provide the same solution set and objective function values for RMHND. Once we define variables y to assume only binary values, the partial LP relaxations should be equivalent as well.*

3.6 Computational Experience

In this section, we present optimal solutions obtained with the BP algorithm. We also present the LP relaxations provided by all three formulations. For the Arc-Flow Formulation and the Aggregated Hop-Indexed Formulation, within a time limit of 3600 s, the optimal solutions found by the BB algorithm based on them are shown. All computational testings in this study were conducted on a Intel Core 2 Quad machine, with 2.5 GHz and 8Gb of RAM memory, running under Linux operating system. For all algorithms, CPLEX [2011] LP solver release 12.1 was used.

Different sets of instances are tested. The first sets of instances consist of artificial networks created using a random instance generator. The following parameters are required by the generator: $|N|$ - number of nodes, $|G|$ - number of technologies (or levels), $\hat{m}$ - array with the number of nodes in each level, (dx, dy) - area dimensions. Nodes are uniformly deployed over a random perturbed grid and assigned a different level at random, according to the values predefined in $\hat{m}$. It means that if we have $|G| = 2$ and the number of nodes in the first level is n, exactly n nodes must be chosen at random to belong to the first level. Arc costs are assigned based on the Euclidean distance (d) between their endpoints and different technologies have proportional costs. Accordingly, the costs are given by the formula $c_{ij}^{g} = \lfloor d \times \alpha \times \delta^{g} \rfloor$, where $\alpha \in [0.5, 1.5]$ with a uniform random distribution and δ^{g} is a constant factor to indicate how many times a technology costs more than the cheapest one.

Another set of instances, composed of real world networks, is studied. They present different topologies, allowing the algorithm analysis in diverse scenarios. In the same way, random costs are assigned to the arcs of these topologies and levels are attributed at random to the nodes. Table 3.1 details the characteristics of the instances, where $|N|$ is the number of nodes, $|A|$ is the number of arcs and $|K|$ corresponds to the number of requests. Note that we consider the whole set of requests, i.e., requests connecting every pair of nodes (this number is given by $|K| = |N| \times (|N| - 1)/2$).

Table 3.1. Real World Networks

instance	$\|N\|$	$\|A\|$	$\|K\|$
carrier	24	86	276
dora	15	52	105
eon	20	78	190
nsf	14	42	91
ring	15	42	105
sul	15	42	105

In Tables 3.2 and 3.3 we present the results of all three formulations for instances with 16 and 25 nodes respectively, a set of requests between all pairs of nodes, $|G| = 2$ and a squared area of 100×100 m. The first two columns of the tables, indicate the number of nodes in primary level $p_r \in \{2, 4, 8\}$ and the number of hops H in the interval $[hmin, hmax]$. Parameter $hmin$ corresponds to the minimum number of hops necessary to provide two arc disjoint paths for all requests while $hmax$ is the maximum number of hops that a path can assume, i.e., $|N| - 1$. In the next four columns, results attained by the Branch-and-bound algorithm based on the Arc-Flow Formulation are presented: GAP (the LP duality gap), t_{LP} (the time taken to evaluate the LP relaxation), followed by the number of instances that were proven optimality (#*opt*) and the respective time, t_{opt}. In the next four columns, similar entries are given for the Aggregated Hop-Indexed Formulation: the LP duality gaps (GAP) and the time taken to compute them (t_{LP}), followed by the number of instances that were proven optimality (#*opt*) and the respective time, t_{opt}. Results for the Arc-Path Formulation include the LP duality gaps (GAP), the time taken to compute them and the number of columns (#*col*) generated at the root node. Finally, in the last three columns, we provide the time taken to compute the optimal solutions given by BP (t_{BP}), the final number of columns (#*col*) and the number of nodes (*#nod*) explored in the enumeration tree. It is important to point out that each line corresponds to average values of 10 randomly generated instances owning the same features. The best computing times (when optimality is achieved) are highlighted in boldface.

Results in Tables 3.2 and 3.3 indicate that BP is able to solve all instances for each network size. On the other hand, it can be observed that the BB algorithm based on the Arc-Flow Formulation and the Aggregated Hop-Indexed Formulation is not capable to prove optimality for all tested instances within the time limit imposed (3600 s). Note that column *#opt* indicates the number of instances solved to optimality (from a set of 10 instances). Although the time taken to compute the bounds for the Arc-Flow Formulation is smaller in most of the cases compared to the times needed to compute them by the other two formulations, the provided gaps are weaker, reaching values up to 29.07%. On the other hand, the gaps provided by the Aggregated Hop-Indexed Formulation and the Arc-Path Formulation are stronger and identical, since these formulations are equivalent. However, the times taken to find optimal solutions (when the time limit is not achieved) by the Aggregated Hop-Indexed Formulation are higher than the Arc-Path Formulation for most of the tests.

Analyzing the results achieved by the three formulations, it can be observed that on average, the BP algorithm based on the Arc-Path Formulation has obtained the best computational times compared to a BB algorithm based on the other two formulations. The BB based on the Arc-Flow and Aggregated Hop-Indexed Formulations was not able to find optimal solutions for all instances, especially when the number of nodes (and consequently the number of requests) increases. Besides, although the bounds provided by the Aggregated Hop-Indexed Formulation are tighter than the bounds from the Arc-Flow Formulation, the overall behaviour of the latter is better, on average. One interesting result shows that the algorithm based on the Arc-Flow Formulation has a better performance as the number of hops H increases, while the algorithms for the other two formulations experience more difficulty. This occurs because the latter two formulations are based on "paths". As the number of hops increases, there are more possibilities of different paths for the Arc-Path Formulation and far more variables for the Aggregated Hop-Indexed Formulation. On the other hand, for small values of H, the BP algorithm experiences a better performance, since the set of paths feasible for the solution is more restricted, allowing less combinations to form different paths.

Table 3.4 presents results for real world instances previously described, with 3 levels (2 nodes in the primary level p_r, 4 nodes in the secondary level s_e and the others in the third level). The first two columns of the table correspond to the network instance and the number of hops H belonging to the first 5 values starting from the

Table 3.2. Summary of Random Instances with $|N| = 16; |A| = 48; |K| = 120; |G| = 2$

		Arc-Flow Formulation				Hop-Indexed Formulation				Arc-Path Formulation					
p_r	H	Linear Relaxation		Integer Solution		Linear Relaxation		Integer Solution		Linear Relaxation			Branch-and-price		
		GAP	t_{LR}(s)	$\#opt$	t_{opt}(s)	GAP	t_{LR}(s)	$\#opt$	t_{opt}(s)	GAP	t_{LR}(s)	$\#col$	t_{BP}(s)	$\#col$	#nod
2	6	11.9%	0.65	8	780.83	0%	0.2	10	**0.24**	0%	0.26	516.3	0.32	516.3	1
	7	6.2%	0.52	9	571.79	1.64%	0.5	10	3.13	1.64%	0.69	764.6	**1.93**	976.4	21.9
	8	3.8%	0.5	10	93.88	0.9%	0.82	10	6.08	0.9%	1.09	931.7	**2.42**	1,147.4	14.2
	9	3.3%	0.46	10	251.6	1.36%	1.05	10	62.53	1.36%	1.49	1,071.9	**3.62**	1,381.5	27.4
	10	1.18%	0.43	10	83.73	0.36%	1.37	10	56.81	0.36%	2.06	1,246.8	**3.31**	1,400.3	8.7
	11	0.81%	0.41	10	14.08	0.47%	1.75	10	147.56	0.47%	2.61	1,380.2	**4.33**	1,614.6	7.8
	12	0.46%	0.4	10	8	0.31%	2.15	10	269.55	0.31%	2.75	1,428.4	**4.49**	1,660.5	6.1
	13	0.41%	0.4	10	**3.68**	0.34%	2.56	9	755.72	0.34%	3.14	1,479	4.86	1,693.9	5.8
	14	0.11%	0.39	10	**0.73**	0.11%	3.11	10	555.01	0.11%	3.14	1,488.1	3.91	1,565.2	1.6
	15	0.11%	0.38	10	**0.67**	0.11%	3.95	7	1,386.14	0.11%	3.14	1,488.1	3.89	1,563.4	1.6
	avg	2.83%	0.45	9.7	180.9	0.56%	1.75	9.6	324.28	0.56%	2.04	1,179.51	**3.31**	1,351.95	9.61
4	6	19.74%	0.66	9	441.39	1.03%	0.2	10	**0.26**	1.03%	0.3	529.2	0.4	534.9	5.2
	7	10.54%	0.48	9	553.76	2.81%	0.49	10	1.91	2.81%	0.74	786.3	**1.65**	921.9	15.8
	8	7.75%	0.44	10	204.36	2.65%	0.84	10	7.99	2.65%	1.16	967.6	**2.98**	1,241.3	16.9
	9	3.63%	0.41	10	75.35	1.31%	1.15	9	397.52	1.31%	1.6	1,106.7	**3.42**	1,354.1	17.1
	10	2.21%	0.39	10	29.67	0.66%	1.45	10	121.07	0.66%	1.9	1,223	**3.83**	1,482.7	10.9
	11	0.83%	0.41	10	9.87	0.64%	1.52	10	251.2	0.64%	2.86	1,422.5	**4.98**	1,666.2	9.1
	12	0.35%	0.41	10	4.49	0.31%	1.87	9	462.64	0.31%	3.14	1,478	**4.41**	1,618.5	4
	13	0.35%	0.4	10	**3.96**	0.33%	2.29	10	455.55	0.33%	3.13	1,475.9	4.56	1,639.1	4.6
	14	0.27%	0.4	10	**0.92**	0.27%	2.8	10	612.72	0.27%	3.11	1,477.8	4.28	1,593.7	3.7
	15	0.27%	0.4	10	**0.98**	0.27%	3.32	8	1,147.77	0.27%	3.15	1,478	4.37	1,606.9	3.4
	avg	4.59%	0.44	9.8	132.48	1.03%	1.59	9.6	345.86	1.03%	2.11	1,194.5	**3.49**	1,365.93	9.07
8	6	29.07%	0.52	10	123.28	1.08%	0.21	10	**0.36**	1.08%	0.3	559.3	0.43	573.5	6.1
	7	20.78%	0.35	9	840.76	2.89%	0.48	10	**1.91**	2.89%	0.68	792.4	2.15	1,003.1	30.4
	8	13.27%	0.35	10	204.53	2.72%	0.77	10	7.12	2.72%	1.18	993.1	**2.77**	1,219.7	20
	9	9.21%	0.34	10	123.92	1.63%	1	10	24.56	1.63%	1.7	1,159.2	**3.87**	1,429	21.8
	10	6.03%	0.33	10	29.19	1.15%	1.24	10	97.83	1.15%	2	1,242.5	**4.3**	1,507.6	22.4
	11	0.47%	0.34	10	5.8	0.21%	1.24	10	89.26	0.21%	2.49	1,424.5	**3.57**	1,557.8	6.8
	12	0.08%	0.33	10	**0.67**	0.08%	1.44	10	268.7	0.08%	2.76	1,470.4	3.6	1,556.4	2.2
	13	0.08%	0.31	10	**0.67**	0.08%	1.66	10	398.14	0.08%	2.83	1,470.3	3.8	1,590.6	2.3
	14	0.08%	0.32	10	**0.72**	0.08%	1.88	9	462.66	0.08%	2.78	1,469.1	3.98	1,622.4	2.4
	15	0.08%	0.33	10	**0.7**	0.08%	2.27	8	1,171.87	0.08%	2.8	1,469.1	3.98	1,622.4	2.4
	avg	7.92%	0.35	9.9	133.02	1%	1.22	9.7	252.24	1%	1.95	1,204.99	**3.24**	1,368.25	11.68

minimum number of hops necessary for a feasible solution. As we are interested in hop-constrained guarantees, for now on we will focus on smaller values of H. The remaining columns are the same ones described earlier. Over again, it is possible to note that BP is able to find all optimal solutions unlike the other approaches. As before, the bounds provided by the Arc-Flow Formulation are weaker while stronger and identical bounds are found for the Aggregated Hop-Indexed and Arc-Path Formulations. It can be observed that for larger instances as `carrier` and `eon` the Arc-Flow Formulation and Aggregated Hop-Indexed Formulation could proven optimality only for a portion of the analysed instances.

The results show that the proposed BP algorithm is capable of solving medium size instances for the RMHND within reasonable times. It is also demonstrated that BP outperforms the BB based on the compact formulations. It is important to highlight why larger size instances will demand a higher effort to be solved. Observe that the hardness involved is not only related to the number of nodes in the network, but more importantly, the number of requests. Once we work with requests for all pairs of nodes, the task becomes harder as the number of requests belongs to $O(n^2)$.

Table 3.3. Summary of Random Instances with $|N| = 25, |A| = 80, |K| = 300, |G| = 2$

		Arc-Flow Formulation				Hop-Indexed Formulation				Arc-Path Formulation					
		Linear Relaxation		Integer Solution		Linear Relaxation		Integer Solution		Linear Relaxation			Branch-and-price		
p_r	H	GAP	t_{LR}(s)	$\#opt$	t_{opt}(s)	GAP	t_{LR}(s)	$\#opt$	t_{opt}(s)	GAP	t_{LR}(s)	$\#col$	t_{BP}(s)	$\#col$	#nod
2	8	10.58%	7.84	1	3,477.27	0.54%	2.97	10	235.16	0.54%	12.09	2,530.9	**20.63**	2,914.1	9.5
	9	7.14%	6.53	0	3,600	2.21%	14.41	0	3,600	2.21%	23.38	3,244.6	**125.68**	5,883.7	99.8
	10	4.71%	4.41	0	3,600	2.1%	29.84	0	3,600	2.1%	42.07	4,181.8	**265.1**	8,391.9	129.6
	11	4.06%	6.7	0	3,600	2.27%	51.03	0	3,600	2.27%	61.01	4,966.3	**622.6**	12,106.2	243.7
	12	3.02%	4.31	0	3,600	1.67%	95.91	0	3,600	1.67%	77.96	5,604.1	**568.34**	12,648.6	124.9
	13	2.44%	4.21	0	3,600	1.48%	191.12	0	3,600	1.48%	97.35	6,323.9	**900.12**	14,817.5	156.8
	14	1.62%	4.18	1	3,313.65	0.95%	270.28	0	3,600	0.95%	119.54	7,088.9	**664.53**	13,896	59.2
	15	1.38%	4.38	3	2,810.84	0.93%	352.96	0	3,600	0.93%	142.78	7,923.6	**899.13**	16,511.4	74.8
	16	0.96%	4.15	4	2,497.03	0.6%	473.44	0	3,600	0.6%	160.4	8,535.6	**585.01**	14,713	28.3
	17	0.75%	3.96	5	2,324.84	0.5%	663.62	0	3,600	0.5%	177.91	9,011.8	**655.18**	14,906.7	24.1
	18	0.48%	4.01	6	1,745.16	0.31%	556.04	0	3,600	0.31%	195.32	9,423.5	**511.53**	13,994	12.6
	19	0.44%	4.02	8	1,200.93	0.31%	793.56	0	3,600	0.31%	204.42	9,838.2	**602.99**	15,175.6	13.3
	20	0.36%	4.21	9	674.94	0.27%	1,038.87	0	3,600	0.27%	222.49	10,274.3	**570.5**	14,992.1	9.5
	21	0.36%	3.84	8	853.98	0.32%	995.32	0	3,600	0.32%	241.55	10,678.9	**675.99**	15,931.6	11.5
	22	0.18%	4.08	10	**292.33**	0.17%	1,143.44	0	3,600	0.17%	251.19	10,897.9	519.02	14,035	7.6
	23	0.01%	4.69	10	**63.38**	0.01%	1,141.27	0	3,600	0.01%	267.5	11,227	361.19	12,638.2	3.4
	24	0.01%	3.39	10	**154.12**	0.01%	1,442.73	0	3,600	0.01%	269.54	11,235	363.46	12,639.8	3.4
	avg	2.26%	4.64	4.41	2,200.5	0.86%	544.52	0.59	3,402.07	0.86%	150.97	7,822.72	**524.18**	12,717.38	59.53
4	8	13.57%	6.2	0	3,600	1.45%	2.74	9	510.88	1.45%	11.35	2,483.5	**22.61**	2,985.1	12.4
	9	8.88%	5.74	0	3,600	3.13%	12.81	0	3,600	3.13%	23.3	3,293	**116.11**	5,832.6	85.8
	10	6.46%	4.23	1	3,272.86	2.78%	24.32	0	3,600	2.78%	37.32	4,009	**182.6**	7,233.6	60.7
	11	4.02%	3.73	2	3,299.34	1.84%	28.13	0	3,600	1.84%	45.95	4,498.8	**306.2**	9,108.1	110.4
	12	3%	3.38	2	3,299.04	1.09%	75.09	0	3,600	1.09%	60.27	5,084.9	**205.48**	8,384.4	25.4
	13	1.09%	3.2	3	2,886.26	0.72%	134.92	0	3,600	0.72%	77.14	5,777.1	**319.73**	10,526.3	39.8
	14	0.54%	3.58	7	2,017.67	0.36%	150.85	0	3,600	0.36%	89.42	6,280.3	**325.36**	9,825	24.3
	15	0.49%	3.22	7	1,322.33	0.43%	155.62	0	3,600	0.43%	112.25	6,896.4	**425.98**	11,507.2	31.3
	16	0.33%	3.04	8	1,024.82	0.3%	285.19	0	3,600	0.3%	120.65	7,087.8	**350.81**	10,515.2	16
	17	0.32%	3.27	8	791.76	0.31%	389.64	0	3,600	0.31%	116.7	7,054.5	**447.32**	11,486.5	23.2
	18	0.16%	3.62	10	364.11	0.16%	415.78	0	3,600	0.16%	113.91	6,980.2	**307.1**	9,980.5	10.3
	19	0.15%	3.63	9	390.74	0.15%	544.1	0	3,600	0.15%	115.89	6,975.2	**334.52**	10,540.2	9.5
	20	0.12%	3.32	9	373.06	0.12%	588.11	0	3,600	0.12%	116.63	7,030.7	**242.93**	9,445.6	4.6
	21	0.11%	3.64	10	**77.39**	0.11%	792.24	0	3,600	0.11%	116.84	7,046	235.31	9,417.8	4.6
	22	0.11%	3.03	10	**20.41**	0.11%	766.84	0	3,600	0.11%	117.37	7,009	206.28	8,812.3	4
	23	0.11%	3.6	10	**66.62**	0.11%	987.93	0	3,600	0.11%	117.34	7,009	201.24	8,807.1	4
	24	0.11%	2.86	10	**12.12**	0.11%	1,212.57	0	3,600	0.11%	117.07	7,009	205.79	8,822.9	4
	avg	2.33%	3.72	6.24	1,554.03	0.78%	386.29	0.53	3,418.29	0.78%	88.79	5,972.02	**260.9**	9,013.55	27.66
8	8	20.05%	7.05	0	3,600	2.08%	3.23	9	624.23	2.08%	11.47	2,514.7	**25.68**	3,076.8	28
	9	14.97%	5.61	0	3,600	3.58%	12.2	0	3,600	3.58%	23.43	3,350.6	**119.19**	5,821.6	101.2
	10	11.21%	5.18	0	3,600	4.04%	23.33	0	3,600	4.04%	41.17	4,349.5	**427.2**	9,306.7	211.3
	11	7.84%	5.75	0	3,600	3.03%	35.63	0	3,600	3.03%	52.85	4,902.4	**502.89**	10,438.3	148.3
	12	5.88%	5.08	0	3,600	2.71%	76.53	0	3,600	2.71%	72.14	5,577.3	**707.64**	11,461.1	131.1
	13	3.45%	4.33	1	3,429.84	1.82%	132.39	0	3,600	1.82%	90.94	6,362.8	**398.36**	11,474.3	70.2
	14	2.45%	3.89	4	2,733.46	1.13%	158.9	0	3,600	1.13%	104.85	6,922.8	**376.27**	11,002.4	18.7
	15	0.74%	4.51	7	1,614.71	0.5%	169.15	0	3,600	0.5%	115.79	7,531.7	**221.06**	9,688.8	9.8
	16	0.47%	4.14	7	1,135.9	0.42%	225.77	0	3,600	0.42%	127.8	7,886.1	**214.15**	9,334.6	4.4
	17	0.1%	4.01	9	499.47	0.09%	266.95	0	3,600	0.09%	135.23	8,154.6	**201.58**	9,571.2	4.6
	18	0.05%	5.24	9	390.35	0.05%	388.48	0	3,600	0.05%	133.22	8,104.1	**188.51**	9,243.4	3.5
	19	0.05%	4.41	10	**18.03**	0.05%	380.46	0	3,600	0.05%	128.44	8,006.5	188.76	9,264.9	2.8
	20	0.04%	4.41	10	**127.01**	0.04%	475.4	0	3,600	0.04%	128.12	8,025.7	182.68	9,129.9	2.6
	21	0.04%	4.12	10	**20.65**	0.04%	634.64	0	3,600	0.04%	129.1	8,026.4	189.96	9,301.1	2.7
	22	0.04%	4.48	10	**13.52**	0.04%	656.4	0	3,600	0.04%	129.1	8,084.8	189.78	9,373	2.8
	23	0.04%	3.65	10	**26.11**	0.04%	780.09	0	3,600	0.04%	128.98	8,084.3	187.31	9,364.2	2.8
	24	0.04%	4.12	10	**23.45**	0.04%	865.01	0	3,600	0.04%	129.01	8,084.3	190.86	9,414.8	3
	avg	3.97%	4.7	5.71	1,648.97	1.16%	310.86	0.53	3,424.95	1.16%	98.92	6,704.04	**265.4**	9,192.18	43.99

3.7 Conclusion and Future Work

In this chapter, we presented a Branch-and-price algorithm for solving the Resilient Multi-level Hop-constrained Network Design problem. Our motivation concerns in devising an exact approach to an important problem in the telecommunications field, comprising issues related to resilience against random failures and delay guarantees. Extensive computational experiments were held to check the performance of the proposed algorithm. Our computational experience demonstrates that with the proposed

Table 3.4. Summary of Real Instances with $|G| = 3, p_r = 2, s_e = 4$

		Arc-Flow Formulation				Aggregated Hop-Indexed Formulation				Arc-Path Formulation					
instance	H	Linear Relaxation		Integer Solution		Linear Relaxation		Integer Solution		Linear Relaxation			Branch-and-Price		
		GAP	t_{LR}(s)	*#opt*	t_{opt}(s)	*GAP*	t_{LR}(s)	*#opt*	t_{opt}(s)	*GAP*	t_{LR}(s)	*#col*	t_{BP}(s)	*#col*	#nod
carrier	8	8.79%	6.44	0	3,600	1.73%	6.9	4	2,684.82	1.73%	19.68	3,253.5	**209.53**	5,293.5	302.7
	9	5.97%	4.48	1	3,508.3	1.72%	12.15	0	3,600	1.72%	31.07	4,129.8	**497.81**	7,514.5	359
	10	2.9%	4.73	2	3,219.15	0.53%	22	0	3,600	0.53%	44.24	5,103.6	**122.44**	7,258.9	23.2
	11	1.69%	4.08	7	1,785.84	0.34%	42.45	0	3,600	0.34%	55.26	5,816.8	**157.66**	7,635.3	19
	12	1.55%	3.77	6	1,714.66	0.34%	52.28	0	3,600	0.34%	62.6	6,002.7	**257.01**	8,280.2	27.4
	avg	4.18%	4.7	3.2	2,765.59	0.93%	27.15	0.8	3,416.96	0.93%	42.57	4,861.28	**248.89**	7,196.48	146.26
dora	6	8.98%	0.51	9	485.54	2.11%	0.37	10	2.82	2.11%	0.77	813.3	**2.02**	1,042.4	23.8
	7	3.19%	0.47	10	39.61	1.64%	0.6	10	7.17	1.64%	1.13	1,011.7	**3.16**	1,373.6	24.2
	8	1.51%	0.4	10	10.03	1.03%	0.83	10	13.58	1.03%	1.33	1,120.9	**3.03**	1,358.4	17
	9	1.08%	0.39	10	4.29	1.04%	0.93	10	28.52	1.04%	1.59	1,190.6	**3.42**	1,447.2	14.6
	10	1.04%	0.39	10	**0.87**	1.01%	1.4	10	182.95	1.01%	1.56	1,174.2	3.33	1,453.6	13
	avg	3.16%	0.43	9.8	108.07	1.37%	0.83	10	47.01	1.37%	1.28	1,062.14	**2.99**	1,335.04	18.52
eon	6	6.09%	2.75	5	2,156.45	1.22%	1.67	10	41.01	1.22%	5.99	2,037.9	**16.82**	2,584.9	60.2
	7	3.95%	1.88	10	994.26	0.88%	2.87	9	1,041.38	0.88%	9.92	2,754.2	**18.34**	3,375.9	15
	8	1.59%	1.82	10	412.16	0.4%	5.18	2	2,930.23	0.4%	12.58	3,219.7	**23.93**	3,911.6	8.6
	9	0.48%	1.62	9	440.26	0.27%	6.15	1	3,465.15	0.27%	15.22	3,681.9	**20.66**	4,035.2	4.3
	10	0.39%	1.69	10	39.3	0.37%	8.84	0	3,600	0.37%	18.83	4,123.5	**28.51**	4,648.3	5.8
	avg	2.5%	1.95	8.8	808.49	0.63%	4.94	4.4	2,215.55	0.63%	12.51	3,163.44	**21.65**	3,711.18	18.78
nsf	5	12.29%	0.33	10	35.62	0.25%	0.11	10	0.2	0.25%	0.13	363.9	**0.18**	368.1	3.4
	6	6.13%	0.28	10	19.09	1.64%	0.26	10	1.64	1.64%	0.26	480	**0.54**	543.3	12.6
	7	3.48%	0.26	10	15	1.55%	0.39	10	2.44	1.55%	0.4	586.6	**0.95**	708.3	15.8
	8	1.97%	0.23	10	9.91	1.3%	0.52	10	5.44	1.3%	0.52	637.6	**1.06**	747.7	14.6
	9	1.4%	0.23	10	2.38	1.03%	0.71	10	27.67	1.03%	0.57	669.6	**1.18**	790.1	12.2
	avg	5.05%	0.27	10	16.4	1.15%	0.4	10	7.48	1.15%	0.38	547.54	**0.78**	631.5	11.72
ring	8	0.54%	0.19	10	**0.44**	0.11%	0.3	10	0.54	0.11%	0.4	508.3	0.51	510.6	1.8
	9	0.09%	0.18	10	**0.42**	0.01%	0.4	10	0.75	0.01%	0.52	527.3	0.62	531.1	1.4
	10	0%	0.18	10	**0.36**	0%	0.5	10	0.98	0%	0.52	538.9	0.63	538.9	1
	11	0%	0.18	10	**0.36**	0%	0.61	10	1.3	0%	0.53	531.2	0.61	531.2	1
	12	0%	0.18	10	**0.36**	0%	0.73	10	1.54	0%	0.52	532.8	0.62	532.8	1
	avg	0.13%	0.18	10	**0.39**	0.03%	0.51	10	1.02	0.03%	0.5	527.7	0.6	528.92	1.24
sul	4	4.83%	0.04	10	0.48	0.15%	0.02	10	**0.04**	0.15%	0.04	203.4	0.05	207.3	2.2
	5	2.12%	0.03	10	0.22	0.82%	0.04	10	**0.08**	0.82%	0.06	250.8	0.09	267.9	4.2
	6	0.21%	0.04	10	**0.07**	0.21%	0.05	10	0.1	0.21%	0.07	272.3	0.09	275.3	1.4
	7	0.21%	0.03	10	**0.07**	0.21%	0.06	10	0.14	0.21%	0.08	272.4	0.09	275.4	1.4
	8	0.21%	0.03	10	**0.07**	0.21%	0.07	10	0.2	0.21%	0.07	272.9	0.09	275.9	1.4
	avg	1.52%	0.03	10	0.18	0.32%	0.05	10	0.11	0.32%	0.06	254.36	**0.08**	260.36	2.12

Branch-and-price method we could solve more problems whose dimensions are out of reach for BB algorithms based on compact formulations.

Some future directions for investigation include the study of acceleration methods for the column generation algorithm and the implementation of other branching rules. Also, we intend to combine our BP algorithm with a heuristic in order to achieve good integer solutions early. At last, heuristics can be developed for the problem to compare to our current exact approach.

Chapter 4

Grooming Routing and Wavelength Assignment with Protection and QoS in WDM Optical Networks

In this chapter we investigate the Grooming Routing and Wavelength Assignment (GRWA) with protection and QoS in WDM mesh optical networks. This problem consists of setting up lightpaths by routing and assigning wavelengths to each arc aiming to attend a set of requests, allowing traffic grooming and ensuring resilience and QoS in the communication. Mathematical formulations are proposed and discussed, along with algorithms specially develop for the problem.

4.1 Introduction

Wavelength Division Multiplexing (WDM) [Mukherjee, 1997, 2006] and optical fiber are promising technologies which have emerged to accommodate the explosive traffic growth in telecommunications networks. Optical networks have a large transmission capacity in the order of tens of Tbps. However, the existence of a high capacity physical media does not imply high standard of quality in communication if the network is not used properly. Thus, optimization techniques play an important role in the design of these networks in order to exploit the full potential they offer.

By using WDM technology, the transmission capacity in the order of Tbps can be divided into multiple non-overlapping frequency or wavelength channels. Each WDM channel can operate at different rates, in the order of Gbps. Thus, optical fiber networks support tens of wavelength channels, each of which has transmission rates of more than one Gbps (OC-48, OC-192). In addition, requests can vary from low rate to very high demands. Thus, it is possible to further increase the channel utilization, grouping several requests for traffic on the same wavelength. This technique is known as Traffic Grooming [Zhu and Mukherjee, 2003; Mukherjee, 2006]. In this case, traffic requests are expressed as bandwidth reservation of standard granularities: OC-1, OC-3, OC-12, OC-48, for example.

Given a set of connections, the problem of setting up lightpaths by routing and assigning a wavelength to each request is called the Routing and Wavelength Assignment (RWA) problem [Zang et al., 2000; Somani, 2005]. Considering the diversity of capacities of the requests, this traditional approach of the problem has a drawback of sub-utilization of the network. For instance, consider a transport backbone of OC-192 capacity ($\approx$ 10 Gbps). In the absence of traffic grooming devices, an usual approach would be to take an entire lightpath for a single OC-1 (1 OC = 51.85 Mbps) connection, leaving most of the capacity of the lightpath unused.

The Grooming, Routing and Wavelength Assignment problem [Zhu and Mukherjee, 2003; Barr et al., 2006] has been proposed in order to handle the design of such networks, also dealing with the wide range of capacities a connection may request. By applying traffic grooming, two or more incoming connections can be sent out on the same lightpath. Therefore, traffic grooming increases the utilization of network bandwidth as long as the connections respect the bandwidth constraint of the lightpath. Future networks tend to demand requests with larger capacities, even exceeding the capacity of a wavelength. These cases are also met in our proposed solution.

The aggregation or disaggregation of traffic at a source, destination or intermediate node requires the use of optical multiplexers (each multiplexer has an installation cost) to add/drop the traffic of a wavelength. Therefore, the route of a given request is composed of "optical hops" defined by a physical path through which the optical signal bypasses intermediate nodes, creating a virtual connection between its end nodes. The sequence of "hops" followed by an optical request defines a lightpath [Mukherjee, 2006]. While solving the GRWA problem, the installation costs must be taken into account, leading to the minimization of the number of wavelength channels used to ensure that traffic requests are attended.

The aggregation of multiple requests in order to reduce the number of wavelengths assigned can create very long routes for the requests. Thus, to facilitate the delivery of certain requests with guarantees of QoS, one should set a maximum bound on the number of hops of the routes. Thus, it ensures a limit on the delay of service requests.

In today's networks, fault tolerance and Quality of Service (QoS) are important measures that should not be disregarded. Considering that usually only a single link may failure at a given time, finding backup paths that are arc-disjoint with the original working path of a request, provides resilience to the network. However, providing two paths (working and backup) for each request, i.e., dedicated protection, the bandwidth reservation in the network resources is two times the original traffic demand. Quality of service may be associated to the ability to provide different priority to different users or data flows, or to guarantee a certain level of performance to a data flow. Thus, to balance the tradeoff between fault tolerance and resource allocation, different level of QoS may be established in order to provide resilience to certain key users or data flows. Besides, QoS becomes much more important when the network capacity is insufficient.

This chapter aims to contribute with solutions to the GRWA problem with protection and QoS in optical networks. Mathematical formulations are proposed for the problem along with a branch-and-price algorithm and a column generation-based heuristic. We evaluate the performance of the proposed approaches and analyze the role played by the protection and QoS constraints.

The following sections are organized as follows. Section 4.2 describes the related work. In Section 4.3, a formal definition of the problem is given. In Section 4.4, two formulations are proposed for the problem and Section 4.5 describes the proposed branch-and-price algorithm along with some computational results. Section 4.6.2 presents the heuristic and computational results. Section 4.7 concludes the paper.

4.2 Related Work

Grooming techniques can be classified into many categories: static traffic grooming vs. dynamic traffic grooming, non bifurcated flow vs. bifurcated flow and single-hop traffic grooming vs. multi-hop traffic grooming. The static traffic grooming considers the case in which all requests are known in advance and do not change by long periods of time. In contrast, the dynamic traffic grooming addresses the case where demands appear dynamically, according to a probability distribution. In the non bifurcated flow, the demands must follow a single path from its source to its destination node. In the bifurcated flow, a request may be split and routed through one or more paths. The bifurcated flow turns the problem less complex, since it is easier to accommodate fractional demands in the remaining capacities of wavelengths. At last, the single-hop traffic grooming restricts a request to use a single lightpath and the multi-hop traffic grooming allows a request to use multiple concatenated lightpaths. Therefore, the bandwidth of a lightpath can be shared by traffic of different source-destination pairs. In this work and following related work, we are interested in the static grooming, non bifurcated flow and multi-hop traffic grooming variation.

The GRWA problem is addressed in [Zhu and Mukherjee, 2002; Hu and Leida, 2004; Vignac et al., 2009; Raghavan and Stanojević, 2011; De et al., 2010]. For a review in traffic grooming, see [Zhu and Mukherjee, 2003]. In [Zhu and Mukherjee, 2002], the authors propose a ILP formulation with the objective of improving the network throughput, along with two heuristics: Maximizing Single-Hop Traffic (MST) and Maximizing Resource Utilization (MRU). While the ILP formulation is able to solve instances with up to 6 vertices, the heuristics can find feasible solutions to a 15-vertices network; however, this work does not address the protection problem.

A decomposition method that divides the GRWA into two smaller problems - GR problem and WA problem - is proposed in [Hu and Leida, 2004]. The two problems can be solved much more efficiently if taken separately. Under some special conditions, the method can produce optimal solutions for the GRWA problem, otherwise, approximate solutions. They also show how protection may be added to the problem, but do not present computational results for that. The main drawback of this work is that the logical topology is defined in advance by a restricted set of established lightpaths, while usually the problem must design the logical topology and the routes altogether.

Reformulations and decomposition approaches are developed in [Vignac et al., 2009] for the GRWA problem. For realistic size instances (networks with 14 and 20 vertices, but thousands of requests), they provide solutions with optimality gap of approximately 5% on average within two hours of computing time. The main drawbacks of this study is that some specific restrictions are assumed: (i) the overall length of the physical route of each request is at most the length of the third elementary shortest path between its source and destination, (ii) restrictive grooming scenarios are considered. Moreover, this work does not also address the protection problem.

Improvements on the results obtained by Zhu and Mukherjee [2002] are shown in [De et al., 2010]. The authors in [De et al., 2010] also investigate the problem of GRWA with the objective of maximizing the network throughput, but using a different approach, which is based on the clique partitioning problem. They use the same instances presented by [Zhu and Mukherjee, 2002], and show that the proposed algorithm called Traffic Grooming based on Clique Partitioning (TGCP) provides higher throughput than MST and MRU. This work does not address the protection problem either.

In [Raghavan and Stanojević, 2011], a branch-and-price approach is proposed to deal with the RWA problem with non bifurcation flow. They use column generation embedded into a branch-and-bound framework to find optimal solutions for the problem. This work does not consider traffic grooming and protection. Besides, they only solve instances optimally, with up to 7 nodes and 42 demands.

Failures can occur due to fiber cuts from some natural event and therefore should not be neglect. Protection mechanisms have been proposed in [Ou et al., 2003; Ou et al., 2004; Yao and Ramamurthy, 2005; Jaekel et al., 2008]. Two schemes define the type of protection: shared and dedicated. In [Ou et al., 2003], a shared protection solution is proposed, so that backup paths can share resources as long as their corresponding working paths are unlikely to fail simultaneously. The authors propose effective heuristics for the problem. This work addresses the dynamic context of GRWA.

A dedicated protection scheme is proposed in [Ou et al., 2004], in the dynamic context of GRWA. They propose two protection approaches and developed effective heuristics for both of them.

In [Yao and Ramamurthy, 2005], survivable mechanisms are applied to the problem of traffic grooming in WDM networks. In order to maximize network performance, ILPs and heuristics are proposed. Heuristics are more scalable than ILPs finding solutions for a 24-vertex network.

An efficient ILP formulation for the problem of resilient traffic grooming in WDM networks is presented by Jaekel et al. [2008]. The proposed method is able to generate optimal solutions for networks with up to 14 vertices.

4.3 Problem Definition

The GRWA problem with protection and QoS, which we will call GRWA-PQoS is NP-hard as it can be reduced to RWA [Somani, 2005]. We also consider the following features: static traffic, asymmetric traffic, non bifurcated flow, multi-hop traffic grooming, dedicated protection and all nodes equipped with optical cross-connect. This means that each arc of the physical topology can work as a single lightpath in the network. With static traffic (when the entire set of requests is known in advance), the GRWA-PQoS problem can be formulated as a integer linear program.

This problem is defined as follows. Let a directed graph $D = (V, A)$, where V is the set of vertices and A is the set of arcs connecting the vertices. Given a set of wavelength channels W, the capacity of each wavelength C^w, a set of requests K, where a request $k = (s, t, d^k)$ is given by a triplet with a source vertex s, a destination vertex t and a demand d^k, the goal is to minimize the total number of wavelengths used for all arcs, meeting the demands of the requests without exceeding the capacities of channels and ensuring the protection/backup on working paths, i.e., two arc disjoint paths must be selected for each request (with a limited number of hops H).

4.4 Mathematical Formulations

The GRWA-PQoS can be modelled as an integer linear programming problem based on a network flow formulation [Ahuja et al., 1993]. The solution of such a formulation can be performed only for networks with a small number of nodes and wavelength channels. Advanced optimization techniques allow the development of exact algorithms based on a reformulation of the problem, increasing the scalability. Thus, we present a path-based reformulation in order to apply column generation and BP in the solution of instances of real-world network sizes.

4.4.1 Arc-Flow Formulation

Given $D = (V, A)$, A_j^- and A_j^+ denote the set of arcs arriving and leaving node $j \in V$, respectively. To model the problem, we use the following decision variables: (i) $x_{ij}^{kp} \in \mathbb{R}_+$, indicating the amount of flow for request k and path p, that passes through arc (i, j), (ii) $w_{ij} \in \mathbb{Z}$, indicating how many wavelengths are assigned to arc (i, j). The problem can be stated as:

$$\min \sum_{(i,j)\in A} w_{ij} \tag{4.1}$$

s.t.

$$\sum_{p \in \{1,2\}} \sum_{j \in A_s^+} x_{sj}^{kp} \geq 2, \qquad \forall k \in K, s = source(k), \tag{4.2}$$

$$\sum_{p \in \{1,2\}} \sum_{i \in A_t^-} x_{it}^{kp} \geq 2, \qquad \forall k \in K, t = dest(k), \tag{4.3}$$

$$\sum_{i \in A_j^-} x_{ij}^{kp} - \sum_{l \in A_j^+} x_{jl}^{kp} = 0, \qquad \forall k \in K, \forall p \in \{1,2\}, j \neq source(k) \neq dest(k), \tag{4.4}$$

$$\sum_{p \in \{1,2\}} x_{ij}^{kp} \leq 1, \qquad \forall k \in K, \forall (i,j) \in A, \tag{4.5}$$

$$\sum_{k \in K} \sum_{p \in \{1,2\}} d^k x_{ij}^{kp} - C^w \, w_{ij} \leq 0, \qquad \forall (i,j) \in A, \tag{4.6}$$

$$\sum_{(i,j) \in A} x_{ij}^{kp} \leq H, \qquad \forall k \in K, \forall p \in \{1,2\}, \tag{4.7}$$

$$0 \leq x_{ij}^{kp} \leq 1, \qquad \forall k \in K, \forall (i,j) \in A, p \in \{1,2\}, \tag{4.8}$$

$$w_{ij} \geq 0 \;\; integer, \qquad \forall (i,j) \in A. \tag{4.9}$$

Formulation (4.1)-(4.9) states the problem as a network design problem in which the objective function aims to minimize the total number of wavelengths used in the arcs of the network. Constraints (4.2)-(4.4) impose flow balance conditions for each pair source/destination of request k. Note that (4.2) ensures that two paths are established from the source vertex of the request, while (4.3) ensures that two paths reach the destination vertex of request k. Constraints (4.4) guarantee the flow conservation in transhipment vertices. Inequalities (4.5) ensure that the paths of a request are arc disjoint. Constraints (4.6) couple flow and integer variables, requiring that the sum of the demands that go through an arc is less than or equal to the sum of the capacities of all the wavelengths in that arc. Note that in fact, the requests are not associated to an specific lightpath assigned with a wavelength. They are only allocated in one of the wavelengths of an arc in the network. Constraints (4.7) imply that paths have a maximum number of hops H, for delay guarantees. Finally, constraints (4.8) and (4.9) provide the bounds of the variables.

4.4.2 Arc-Path Formulation

Given a request $k \in K$, we assume that P^k denotes the set of all simple directed paths connecting the endpoints of that request in D. In addition to integer variables w_{ij} defined previously, our second model uses binary variables $\{\lambda_p^k : k \in K, \forall p \in P^k\}$

(taking value 1 if path $p \in P^k$ is selected, 0, otherwise) to choose the paths. Let's assume that $a_{ij}^p \in \{0, 1\}$ denotes a binary parameter that indicates whether arc (i, j) belongs ($a_{ij}^p = 1$) or not ($a_{ij}^p = 0$) to path p. The arc-path formulation is given by:

$$\min \sum_{(i,j) \in A} w_{ij} \tag{4.10}$$

s.t.

$$\sum_{p \in P^k} \lambda_p^k \geq 2, \qquad \forall k \in K, \tag{4.11}$$

$$\sum_{p \in P^k} a_p^{ij} \lambda_p^k \leq 1, \qquad \forall k \in K, \forall (i, j) \in A, \tag{4.12}$$

$$\sum_{k \in K} \sum_{p \in P^k} d^k a_{ij}^p \lambda_p^k - C^w \, w_{ij} \leq 0 \qquad \forall (i, j) \in A, \tag{4.13}$$

$$\lambda_p^k \in \{0, 1\} \qquad \forall k \in K, p \in P^k, \tag{4.14}$$

$$w_{ij} \geq 0 \ \ integer \qquad \forall (i, j) \in A. \tag{4.15}$$

As in the previous model, the objective function (4.10) minimizes the total number of wavelengths used in the arcs of the network. Constraints (4.11) guarantee the selection of two paths for each request $k \in K$. Inequalities (4.12) ensure that the working and protection paths are arc disjoint for each request $k \in K$. Constraints (4.13) couple path and arc variables, requiring that the sum of the demands that go through an arc is less than or equal to the sum of the capacities of all the wavelengths in that arc. Note again that in fact, the requests are not associated to an specific lightpath assigned with a wavelength. They are only allocated in one of the wavelengths of an arc in the network. Constraints (4.14) and (4.15) define the bounds of the variables. Note that the constraint to ensure that paths have a maximum size H is not directly addressed in the model. This constraint will be considered in the pricing problem associated to this formulation.

4.5 Branch-and-price Algorithm for the GRWA-PQoS

From formulation (5.10)-(5.15), we can use column generation to evaluate LP bounds implied by dropping the integrality constraint of model (4.10)-(4.15).

4.5.0.1 Lower bounds implied by Column Generation

To understand how the LP bounds implied by (4.10)-(4.15) are evaluated, let us associate dual variables (π, γ, β) to constraints (4.11), (4.12) and (4.13), respectively. These variables will be used by the pricing problem, i.e., the problem able to identify new columns to be added to the RMP. The pricing problem should find the shortest path for each source/destination pair corresponding to the requests. Thus, at each iteration of the column generation method, the pricing problem is solved $|K|$ times. An algorithm to find the shortest path between two vertices in a graph such as Dijkstra can be applied. In order to check if the generated path is attractive or not to enter the RMP, the dual values from the last iteration of the RMP are used to price the paths. Mathematically speaking, one should verify if the dual constrains are violated:

$$\pi^k + \sum_{(i,j)\in A} a_p^{ij}\gamma_{ij}^k + \sum_{(i,j)\in A} d^k a_p^{ij}\beta_{ij} \leq 0, \forall k \in K, \forall p \in P^k \tag{4.16}$$

In our implementation, the initial sets C^k are generated as follows. In order to guarantee a starting feasible solution for the problem, two arc disjoint paths must be provided for each request $k \in K$. Therefore, we computed the first two minimum disjoint paths connecting request k in terms of number of hops (seeking a feasible solution).

4.5.1 Pricing Problem

To understand how the LP bounds implied by (4.10)-(4.15) are evaluated, let us associate dual variables to each constraint of the original formulation. These variables will be used by the pricing problem, i.e., the problem able to identify new columns to be added to the RMP. The pricing problem should find the shortest path for each source/destination pair corresponding to the requests. Thus, at each iteration of the column generation method, the pricing problem is solved $|K|$ times. An algorithm to find the shortest path between two vertices in a graph such as Dijkstra could be applied, if the paths were not restricted to a maximum size. As in our problem the paths are restricted to a maximum size H, we used an algorithm based on dynamic programming, which solves the shortest path problem with resource constraints [Feillet et al., 2004; Irnich and Desaulniers, 2005].

4.5.2 The Enumeration Tree

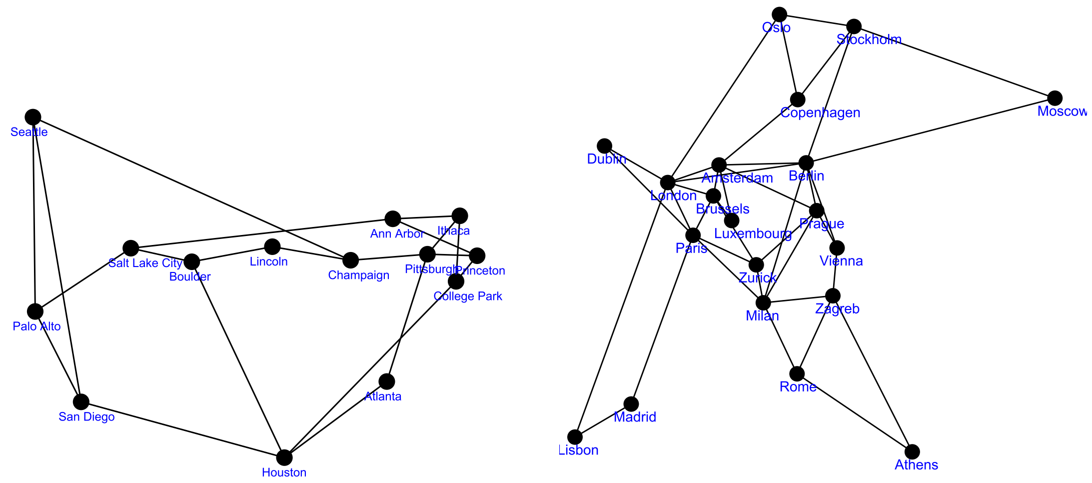
As stated before, if the LP solution achieved in the end of the column generation procedure is integer, it solves the original problem (4.10)-(4.15). Otherwise, we must resort to some kind of enumeration algorithm (to find an integer solution), such as Branch-and-price. We adopted a branching rule based on fractional variables w_{ij}. Thus, while the solution is fractional, we must choose one among all continuous variables w_{ij}, and create two subproblems, which will include a new constraint limiting the bounds of that variable. This branching rule is usually adopted in BB algorithms [Lasdon, 1970]. Thus, a new constraint is added to each subproblem: $w_{ij} \leq \lfloor w_{ij} \rfloor$ and $w_{ij} \geq \lceil w_{ij} \rceil$, respectively. The variable selection to start the branching is based on the fractional variable associated to the maximal integer unfeasibility (farthest from integrality). This means that among all the fractional variables, the variable closest to the value $x.5$ will be selected. In the case of ties, the first variable found is taken. The policy to select the next subproblem to be solved is based on the relaxation value of the node. Therefore, the subproblem with the lowest objective function for the relaxed problem is taken to be solved.

4.5.3 Computational Experience

In this section, we report computational results for the problem, obtained with both formulations/algorithms. Our aim is to illustrate the main advantages of one method over another for the problem. We used two well known real world network topologies, called NSF (National Science Foundation) [Krishnaswamy and Sivarajan, 2001] and EON (European Optical Network) [O'Mahoney et al., 1995]. In Figure 4.1 topologies are shown for both networks. The NSF network is composed by 14 vertices and 21 edges. The EON network consists of 20 vertices and 39 edges.

Different numbers of requests $|K| \in \{30, 60, 90\}$ were tested with different granularities (types OC-1, OC-3 and OC-12). The capacity of each wavelength was defined as OC-192 and each arc supports 12 wavelengths. Three increasing values for the number of hops H were used. The lowest value is the minimum number of hops to ensure feasibility in the solution. For example, with $H = 5$, it should be possible to find two disjoint paths of up to 5 hops for each request.

All tests reported in this section were conducted with an Intel Core 2 Quad Xeon with 2GHz and 8GB of RAM, running under Linux operating system. CPLEX release 12.1 was used for both algorithms (BB and BP restricted master program). A time limit of two hours ($7,200$ seconds) was set for CPLEX BB.



(a) NSF (b) EON

Figure 4.1. Small Network Topologies

Tables 4.1, 4.2 and 4.3 summarize the results for NSF and EON networks, with 30, 60 and 90 requests, respectively. The first two columns of the table correspond to the network instance and the number of hops H. In the next four columns, results attained by the Branch-and-bound algorithm based on the Arc-Flow Formulation are presented: GAP (the LP duality gap), t_{LP} (the time taken to evaluate the LP relaxation), followed by the integer solution obtained after a time limit of $7,200$ seconds: GAP and time (t_{int}) in seconds. Results for the Arc-Path Formulation include the LP duality gaps (GAP), the time taken to compute them and the number of columns ($\#col$) generated at the root node. Finally, in the last three columns, we provide the time taken to compute the optimal solutions given by BP (t_{BP}), the final number of columns ($\#col$) and the number of nodes ($\#nod$) explored in the enumeration tree. The best computing times (when optimality is achieved) are highlighted in boldface.

Figures 4.2, 4.3 and 4.4 illustrates the computational time demanded by each algorithm. In Figure 4.2, figures 4.2(a) and 4.2(b) correspond to the NSF and EON networks, for $H \in \{5,6,7\}$ and $H \in \{6,7,8\}$, respectively. A set of 30 requests of types OC-1, OC-3 and OC-12 were randomly generated for each network. In the NSF network, both algorithms were able to find the optimal solution to the problem for all

Table 4.1. Summary of NSF and EON for 30 requests

instance	h	Arc-Flow Formulation: Linear Relaxation GAP	t_{LP}(s)	Integer Solution GAP	t_{int}(s)	Arc-Path Formulation: Linear Relaxation GAP	t_{LP}(s)	$\#col$	Branch-and-Price t_{BP}(s)	$\#col$	#nod
nsf	5	12.5%	0.27	0%	153.26	3.91%	0.08	146	**0.22**	156	3
	6	3.45%	0.3	0%	264.68	3.45%	0.18	214	**2.76**	332	43
	7	0%	0.3	0%	8.87	0%	0.18	204	**1.39**	232	4
eon	6	8.89%	1.16	0%	7,200	3.6%	2.03	639	**22.09**	1,280	79
	7	4.65%	0.98	0%	7,200	2.75%	3.72	890	**142.07**	2,879	329
	8	2.38%	0.66	0%	7,200	2.26%	5.58	1,083	**132.07**	3,657	175

Table 4.2. Summary of NSF and EON for 60 requests

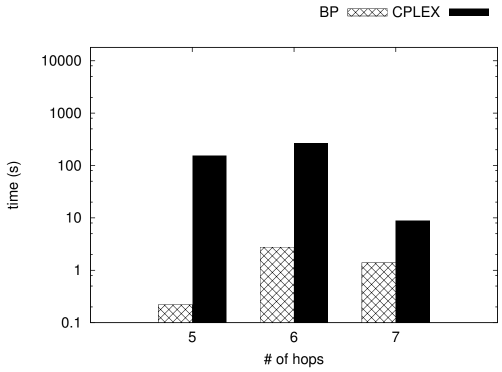
instance	h	Arc-Flow Formulation: Linear Relaxation GAP	t_{LP}(s)	Integer Solution GAP	t_{int}(s)	Arc-Path Formulation: Linear Relaxation GAP	t_{LP}(s)	$\#col$	Branch-and-Price t_{BP}(s)	$\#col$	#nod
nsf	5	14.53%	1.38	0%	7,200	3.92%	0.23	267	**0.83**	305	5
	6	8.5%	1.28	0%	742.51	2.9%	0.83	423	**6.08**	543	21
	7	6.67%	0.86	0%	7,200	3.33%	0.95	456	**1.48**	456	1
eon	6	11.3%	7.31	4%	7,200	3.32%	9.1	1,177	**846.72**	3,427	861
	7	6.67%	4.08	4%	7,200	2.22%	12.23	1,374	**1,058**.14	4,872	166
	8	4.55%	3.13	2%	7,200	2.98%	26.6	2,145	**1,866.12**	9,548	413

Table 4.3. Summary of NSF and EON for 90 requests

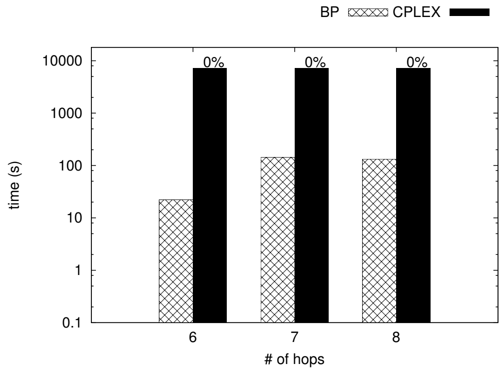
instance	h	Arc-Flow Formulation: Linear Relaxation GAP	t_{LP}(s)	Integer Solution GAP	t_{int}(s)	Arc-Path Formulation: Linear Relaxation GAP	t_{LP}(s)	$\#col$	Branch-and-Price t_{BP}(s)	$\#col$	#nod
nsf	5	14.53%	3.47	3%	7,200	1.96%	0.41	385	**1.65**	452	10
	6	11.01%	5.16	0%	7,200	4.37%	1.26	564	**9.71**	775	33
	7	5.32%	7.17	3%	7,200	3.08%	2.65	761	**25.52**	1,123	29
eon	6	11.92%	23.46	6%	7,200	3.3%	21.54	1,764	**883.62**	4,364	303
	7	7.68%	38.44	4%	7,200	4.13%	43.18	2,369	**4,066.49**	10,475	407
	8	3.53%	43.61	7%	7,200	1.8%	81.64	3,322	**1,895.96**	9,648	56

instances. However, the time required by the BP algorithm is much smaller than the time of CPLEX (logarithmic scale). In the EON network, BP algorithm was able to find optimal solutions while CPLEX has reached the maximum time limit. The number above the bars indicates the *gap* (distance to the optimal solution) of the best solution found by CPLEX when the time limit has been reached. In this case, optimality was not proven. One should note that the time required to solve EON network is greater than NSF network, due to their dimensions.

In Figure 4.3, 60 requests should be attended by each network. Figure 4.3(a) indicates that optimal solutions were achieved by the BP algorithm, while CPLEX was able to find the optimal solution just one time. In the other two cases, the time limit was reached and optimality was not proven. In Figure 4.3(b), BP reaches all the optimal solutions and the time limit is reached for all cases tested with CPLEX. In all



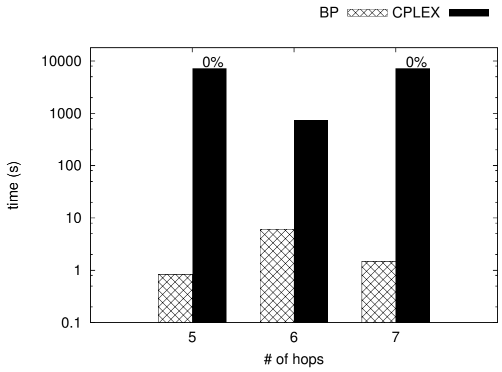
(a) NSF



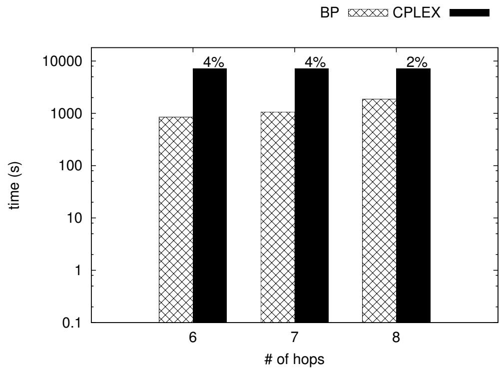
(b) EON

Figure 4.2. 30 Requests of Types OC-1, OC-3 e OC-12

cases there are gaps for the feasible solutions obtained by CPLEX.



(a) NSF

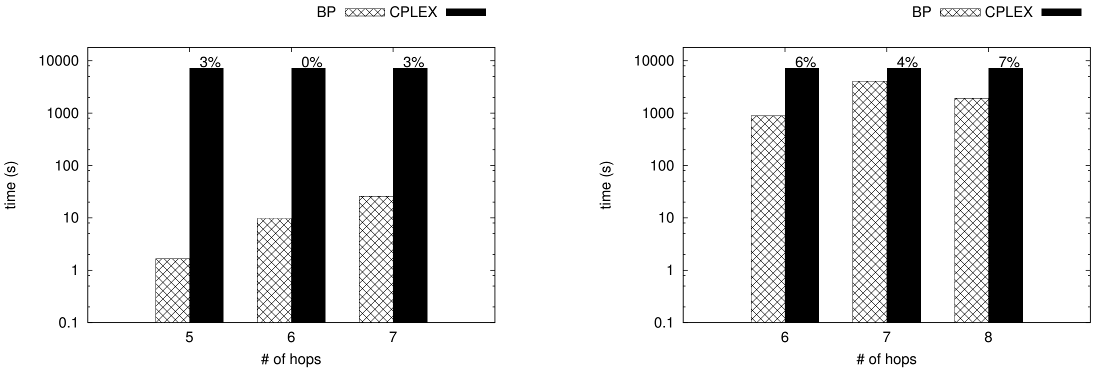


(b) EON

Figure 4.3. 60 Requests of Types OC-1, OC-3 e OC-12

Similar results are obtained when the number of requests is set to 90. It may be noted that in all cases tested the BP algorithm was able to find the optimal solution of the problem while CPLEX reaches optimality only when the instance has a small size and the number of requests is small. By increasing the number of requests, the times demanded by both algorithms increases. In the case of CPLEX, it is observed that gaps tend to increase with larger network sizes and a higher number of requests.

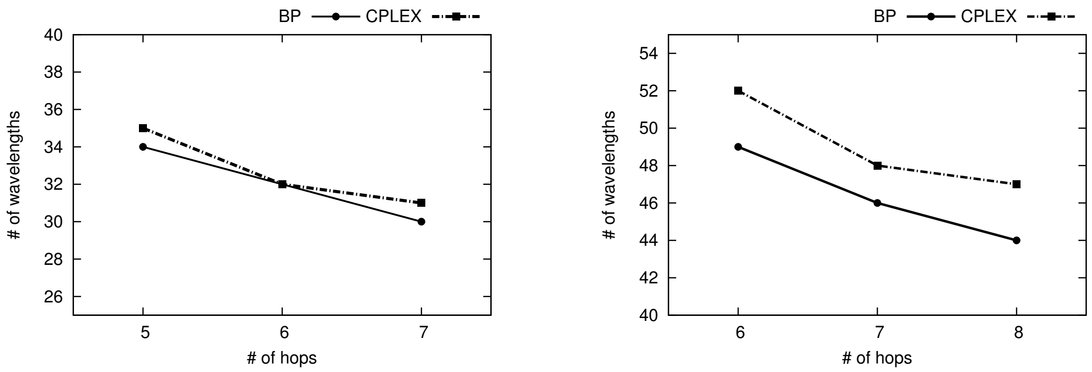
Figure 4.5 shows the number of wavelengths allocated to each of the networks, for 90 requests. It can be observed that as the number of hops increases, the number of wavelengths decreases. This behavior is due to a higher number of wavelengths demanded to ensure feasibility on the paths, if H is more restricted. Allowing longer paths, the effect of traffic grooming is more evident, demanding a smaller number of wavelengths in the network. CPLEX and BP curves do not coincide because the



(a) NSF (b) EON

Figure 4.4. 90 Requests od Types OC-1, OC-3 e OC-12

solution found by CPLEX in these cases was not optimal, due to the time limit applied.



(a) NSF (b) EON

Figure 4.5. Objective Function for 90 requests

4.6 Column Generation-based Heuristic

In order to address larger instances with a significantly higher number of requests, we present a column generation-based heuristic. In the former formulations, QoS was tackled by a threshold in the number of hops that a path could assume. According to preliminary tests, this constraint does not apply when the number of requests is increased so that the network has a high load. Thus, QoS is tackled under another view from now on. We evaluate the performance of the proposed approach and analyze the role played by the protection and QoS constraints. Formulation (4.10)-(4.15) is the basis of our column generation-based heuristic, except that we do not consider the number of hops constraint anymore.

4.6.1 QoS constraints

Considering that QoS may be associated to the ability to provide different priority to different users or data flows, or to guarantee a certain level of performance to a data flow, we state two variants of the proposed mathematical formulation, in which different levels of QoS apply. Constraints (4.11) seek for a solution that provides two arc-disjoint paths for each request $k \in K$. This approach guarantees that the traffic will not be lost if a single failure occur in any arc of the network. On the other hand, the number of wavelengths necessary to provide this protection will be much larger. It may be the case that only a key set of requests must be protected, aiming to decrease the number of wavelengths needed. For that, for a set of key requests, we maintain constraints (4.11), while for another set of requests, we allow that only the working path be attended (therefore, we substitute constraints (4.11) by $\sum_{p \in P^k} \lambda_p^k \geq 1 : \forall k \in K' \subset K$). Finally, if we do not want to provide protection to any request, constraints (4.11) will read $\sum_{p \in P^k} \lambda_p^k \geq 1 : \forall k \in K$.

4.6.2 Heuristic

A commonly used heuristic to obtain primal solutions is known as *restricted master heuristic* [Joncour et al., 2010]. The main idea is to solve the LP relaxation of the problem through the column generation procedure described above and solve the remaining RMP with integrality constraints on the variables.

Thus, the restricted master IP is defined by the columns that were generated while solving the master LP. Solving a static IP over these columns allows us to reach a feasible solution and evaluate how close this solution is from the optimal. This step can take a long time depending on the size of the IP, i.e., how big is the set of columns generated during the master LP execution. Therefore, we have established a time limit on the IP execution.

4.6.2.1 Pseudo-code

Algorithm 2 describes the main steps followed by the proposed heuristic. In line 1, the RMP is created. Line 2 sets the initial columns to guarantee a feasible solution in the starting RMP. Variable *improved* in line 3 indicates whether the algorithm should stop due to the lack of attractive columns to be added to the RMP. Variable *column* in line 4 starts empty. In line 5, *solution* receives the first (relaxed) solution of the RMP. Line 6 indicates that while there is at least one attractive column, the procedure will not stop. From line 8 to line 15, we repeat the same procedure for each request

$k \in K$: update the dual values in the pricing problem (line 9), solve the princing problem seeking for a new column (line 10), check if the column is attractive (line 11) and add it to RMP in this case (line 12). After that, if a new column is found, we optimize the RMP again (line 17). When no more columns need to be added to the RMP, we solve an IP with the columns in the RMP, to get an integer solution.

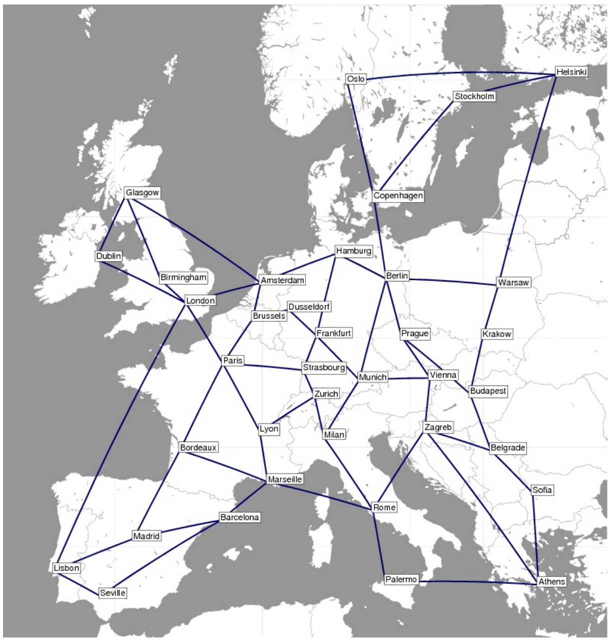
Algorithm 2 Column Generation-based Heuristic

```
1: RMP ← createRMP();
2: setInitialColumns(RMP);
3: improved ← true;
4: column ← ∅;
5: solution ← LPSolver(RMP);
6: while improved do
7:     improved ← false;
8:     for all k ∈ K do
9:         updateDualValues();
10:        column ← PricingSolver();
11:        if f(column) < 0 then
12:            addToRMP(column);
13:            improved ← true;
14:        end if
15:    end for
16:    if improved then
17:        solution ← LPSolver(RMP);
18:    end if
19: end while
20: solution ← IPSolver(RMP);
21: return solution;
```

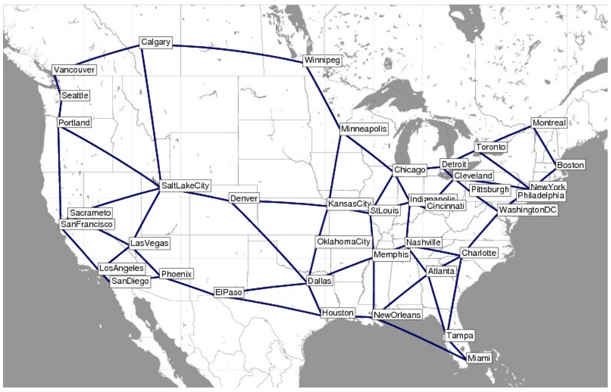
4.6.3 Computational Experience

In this section, we present computational results for the problem, obtained with our column generation-based heuristic. Besides NSF and EON networks, we also used two larger networks available at SNDlib (http://sndlib.zib.de/), called COST266 and JANOS-US-CA. The COST266 network is composed by 37 vertices and 57 edges, while the JANOS-US-CA network is formed by 39 vertices and 61 edges. Both networks are illustrated in Figure 4.6.

In order to create our test instances, different numbers of requests $|K| \in \{500, 1000, 1500, 2000\}$ were generated for each topology with different granularities (types OC-1, OC-3, OC-12 and OC-48). The capacity C^w of each wavelength was set to OC-192 and each arc admits $|W| = 64$ wavelengths. Parameter M was defined as



(a) COST266 (37 nodes and 57 edges)



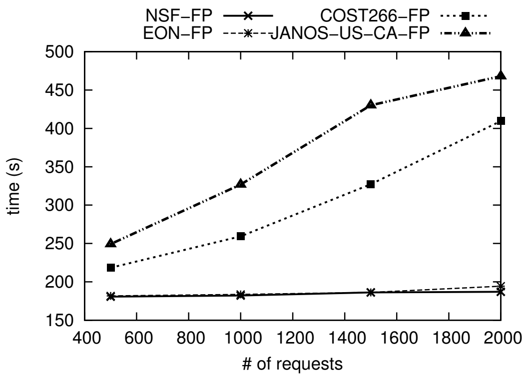
(b) JANOS-US-CA (39 nodes and 61 edges)

Figure 4.6. Large Network Topologies

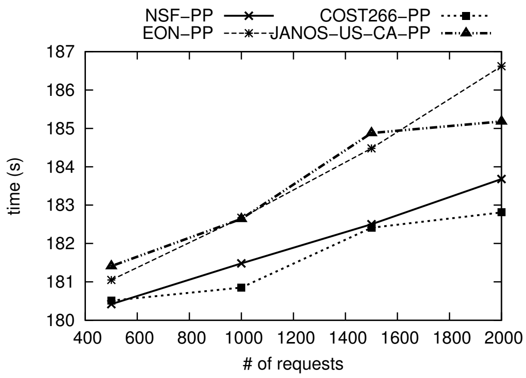
the number of requests in each case. All tests reported in this section were conducted with an Intel Core 2 Quad Xeon with 2GHz and 8GB of RAM, running under Linux operating system. CPLEX release 12.1 was used. A time limit of 180 seconds was set for solving the restricted master IP of the heuristic.

We evaluate three different scenarios, which we call Full Protection (FP), Partial Protection (PP) and None Protection (NOP). In the FP scenario, we consider that all requests should be protected by a backup path. In the PP scenario, only a subset of the requests should be protected. The subset was defined to be the first half of the requests. Finally, in the NOP scenario, we admit that all requests have only the working path allocated.

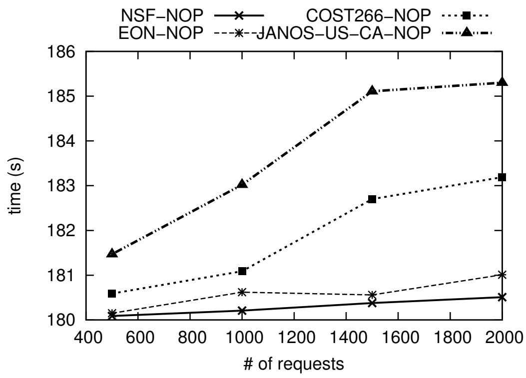
Figure 4.7 shows the computational time demanded by the algorithm to obtain solutions for the problem. It can be observed that as the the number of requests increase, the time taken to compute the solutions also increase. Moreover, for the same number of requests, larger topologies spend more time to execute (see Figures 4.7(a) and 4.7(c)), except for the partial protection scenario (see Figure 4.7(b)). This occurs due to the randomness in the choice of the set of requests to be protected. As the network topologies are different, we only set the number of requests to be protected, but they were chosen randomly for each topology.



(a) Full Protection



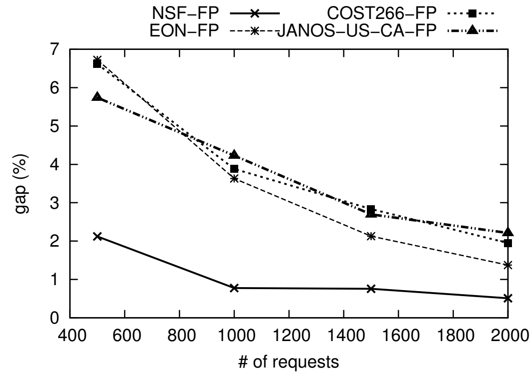
(b) Partial Protection



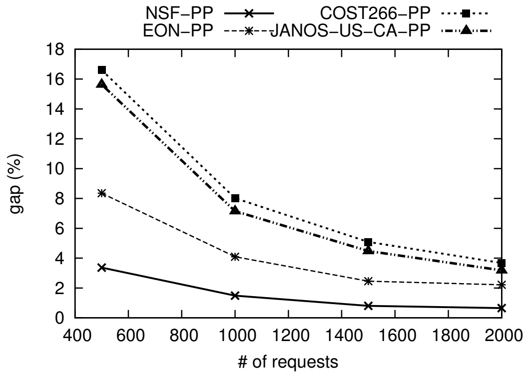
(c) None Protection

Figure 4.7. Computational Time

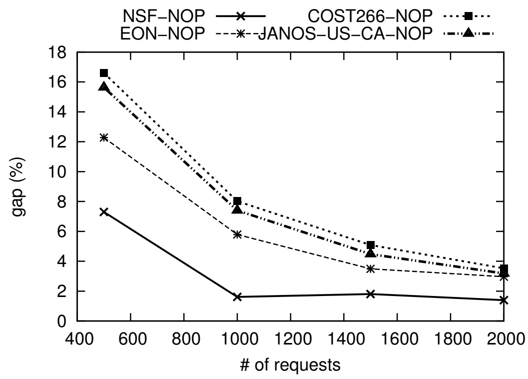
Figure 4.8 presents the gap value achieved in each solution, i.e., how close the solution is from the optimal value. It can be noted that in the full protection scenario, the gaps are less than 7%, while in the partial protection and none protection scenarios they are less than 18%. It seems that the grooming works better with a higher traffic in the network than in a low traffic demand, leading to smaller gap values when full protection is chosen. It is important to point out that as the number of requests increases, the gap values improve significantly. For all scenarios, gap values are less than 4%, for 2000 requests. In general, smaller networks reached smaller gaps than larger networks.



(a) Full Protection



(b) Partial Protection

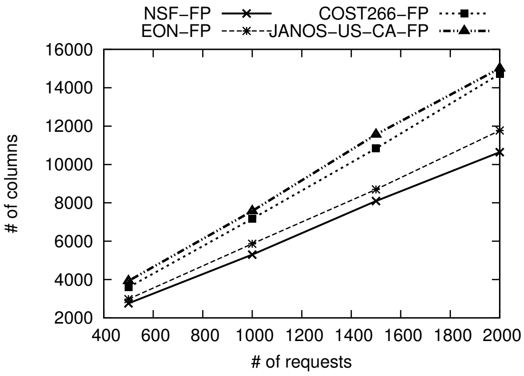


(c) None Protection

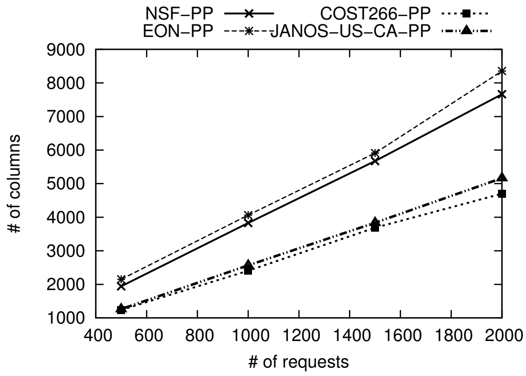
Figure 4.8. GAP

In Figure 4.9, the number of columns generated by the algorithm is analyzed. It can be observed that as the number of requests increase, the number of columns also increase. Besides, larger networks lead to a higher number of columns (see Figures 4.9(a) and 4.9(c)), except again for the partial protection scenario (see Figure 4.9(b)), in which the smaller networks generated more columns due to the randomness of the requests needing protection.

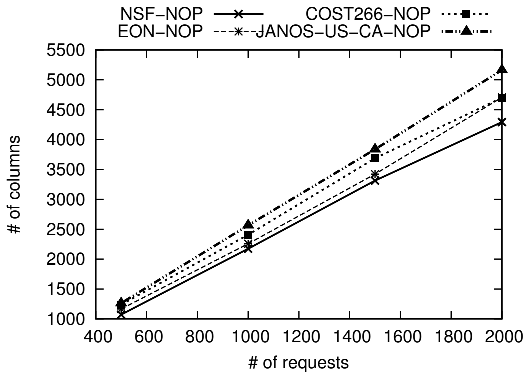
In Figure 4.10, we show results regarding the number of wavelengths used and the traffic lost for the three scenarios. The traffic lost is calculated based on the failure of a single edge. For example, suppose that edge $e = (i, j) = (j, i)$ fails; this implies that all the traffic going through arcs (i, j) and (j, i) will be lost. We show the average value of traffic lost, among all edges of the network. From Figure 4.10(a), the following can be observed: (i) the number of used wavelengths increases as the number of demands increases, for all networks and scenarios; (ii) the number of wavelengths increases if



(a) Full Protection



(b) Partial Protection



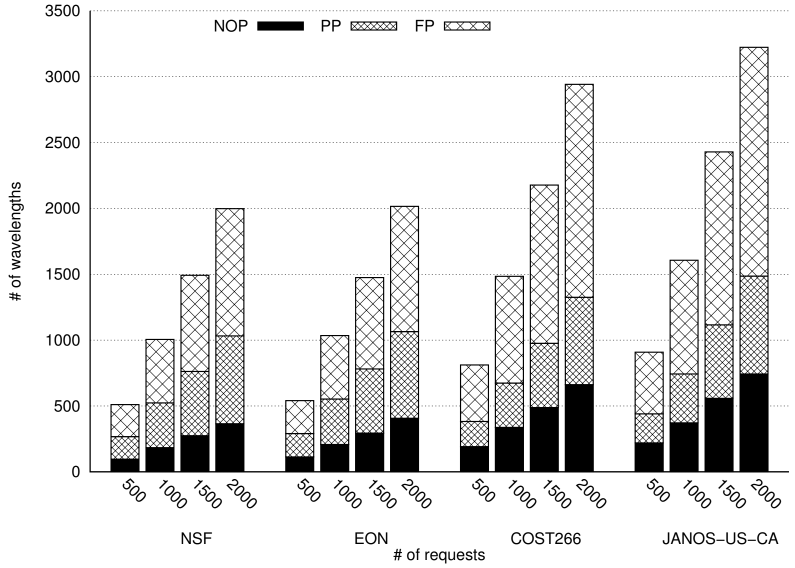
(c) None Protection

Figure 4.9. # of Columns

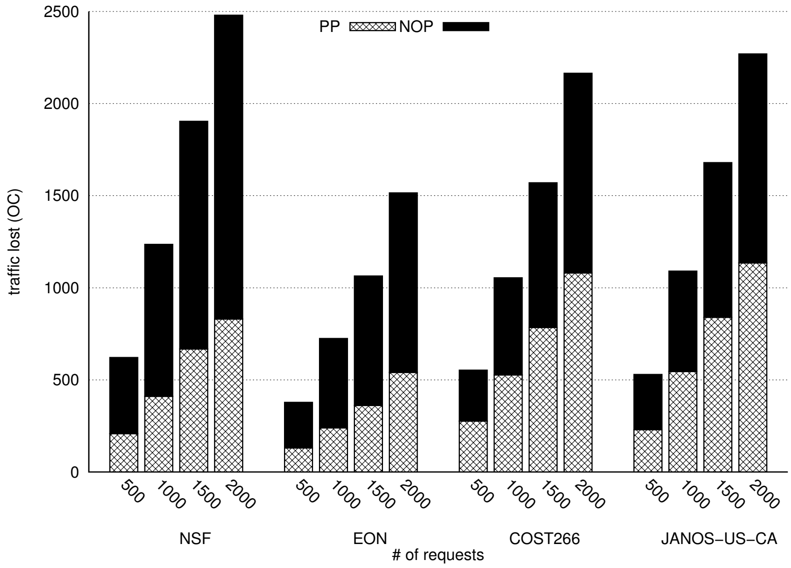
a type of protection is added to the problem. Thus, the full protection uses more wavelengths than the partial protection, which uses more wavelengths than the no protection scenario. In contrast, Figure 4.10(b) shows that the full protection does not lead to traffic lost, while the partial and the no protection scenarios both have traffic lost. When no protection is adopted, the traffic lost is even more accentuated. As one can note, there is a tradeoff between the number of wavelengths used and the traffic lost in case of failure. Also, it can be observed that NSF network has the highest lost rate. This occurs due to the total traffic amount be the same for all networks; thus, in larger networks, the traffic is more balanced among all edges, which is not the case in NSF.

4.7 Conclusions

In this chapter we presented two formulations for the GRWA-PQoS problem in WDM optical networks. It was shown that exact algorithms developed specifically for the



(a) Wavelengths



(b) Traffic Lost (OC)

Figure 4.10. # of Wavelegths vs. Traffic Lost

problem by advanced optimization techniques such as column generation and branch-and-price led to the solution of real-world networks with moderate number of requests (which is practical many times). In addition, computational results show that the times required by the proposed BP algorithm are feasible in all cases studied. It can be concluded that the proposed approach was able to design solutions for traffic engineering in WDM optical networks using optimization techniques efficiently.

We also proposed a Column Generation-based heuristic that finds good solutions within a small computational time for small and large topologies and a high number of requests. We show that the quality of our solutions does not deteriorate as the number of requests increases, instead, the quality is improved. Finally, we evaluate how protection constraints can provide different levels of QoS in the network. We show that there exist a tradeoff between the number of wavelengths used and the traffic lost in case of failure. It can be concluded that the proposed heuristic was able to design solutions for traffic engineering in WDM optical networks using optimization techniques. Future works include the develop of a more elaborated heuristic, in which new columns should be added after the end of the relaxation procedure. This approach may lead to better solutions, as a larger set of columns will be taken into account. Another point is the study of the dynamic approach, mixing optimization and simulation.

Chapter 5

Optimal Topology Design of Complex Networks

In this chapter we study the Optimal Topology Design problem (OTDP) of Complex Networks. We apply optimization techniques in the design of efficient communication networks based on complex networks features. Different formulations are proposed and analyzed to generate networks with non-trivial topology features.

5.1 Introduction

The Network Science concept has its roots in graph theory dating back to 1730s. However, it is evolving, since it reemerged in the late 1990s as a new science [Lewis, 2009]. Even though a definitive description of its meaning is still open, there are many ways to define it, for sure. Network science involves the study of the theoretical foundations of network structure, dynamic behavior and its application to many subfields [Barabási et al., 2000; Albert and Barabási, 2002; Newman, 2003; Watts, 2004; Thadakamalla et al., 2008]. Thus, the study of topics like structure, topology, emergence, dynamism, autonomy and so on is certainly of great importance in the field. In particular, the origin of real world complex networks is constantly a topic of interest, as an attempt to clarify what kind of natural processes take place in these networks. On the other hand, there is a conjecture whether such complex topologies normally appear as a result of some optimization processes. A challenge can be to identify these processes and properties and apply them to manage or to design the network from scratch.

Complex systems are found in real world in different areas of science, including the Internet, WWW, neural networks, friendship relationships, among others [Newman, 2003; Watts and Strogatz, 1998; Faloutsos et al., 1999; Thadakamalla et al., 2004; Albert et al., 1999]. All these networks are large scale networks and very different from traditional network problems explored by operations researchers. Also, they have an intense amount of activities and behaviors that cannot be fully explained. Since then, networks have been used to model and simulate complex interactions among elements of a system, providing support for a better understanding and analysis. Surely, it is important to emphasize that complex systems differ from complicated systems [Thadakamalla et al., 2008]. Large scale systems can be considered complicated, although their components and behaviors are well known. However, complex systems show diverse behaviors, not always known or predictable.

Complex networks can be defined as large scale networks with an intricate relationship among their components and many degrees of freedom in the possible actions of components [Alderson, 2008]. In this context the concept of complex is based on behaviors exhibited by the network that arises naturally and unplanned. On the other hand, a complicated network is also a large scale network where the components and the rules governing its functioning are known [Thadakamalla et al., 2008]. In this case complex is associated with the difficulties to solve the problems using traditional approaches, including the computational complexity.

Given the network structure with their components, interactions and constraints, the objective is to optimize a well known function resulting from that structure. This solution allows the user to control and design the network.The models consider costs, performance, resource and design constraints. In complex networks we have the inverse problem, or the reverse engineering, where the objective is to know how the observed structure supports a perceived function [Alderson, 2008].

Complex networks are many times characterized by a non-trivial topology and present interesting features which may be useful in designing engineered networks. One of these features concerns the low cost for sending information (or a packet, a commodity) through the network. Thereby, computer, communication and transportation networks, just to name a few, could take advantage of being modelled to present specific complex features, to improve their overall efficiency. On the other hand, it is expected that the structure and function of a complex network can be interpreted from some optimization process. Consequently, the existing complex network can also be redesigned or restructured or a new network can be designed from scratch.

Regarding these considerations, the objective of this chapter is the investigation of how optimization strategies could be applied in the context of complex communication networks. We show that a given network can be tuned to satisfy a desired property or set of properties and to avoid others. This tuning can be guided by a set of patterns previously defined. This is possible by changing the structure level, i.e., modifying the physical topology of the network to get improvements in a function.

This chapter is organized as follows. Section 5.2 presents the main network metrics used in complex theory. In Section 5.3, different network structures and models are introduced; in particular, we present the regular, random, small world and scale free concepts of networks. Section 5.4 presents optimization models and algorithms to create complex network topologies while Section 5.5 describes the computational experience. Section 5.7 explores alternative approaches and in Section 5.8, the related work is presented. Final remarks are given in Section 5.9.

5.2 Measurements of Complex Networks

Complex systems have been modelled through network representation, making possible the analysis of topological features using informative measurements. A central issue in the study of complex systems is understanding the relationship between system structure and function. Network metrics are therefore of great importance while investigating network representation, characterization and behavior. This section is devoted to the presentation of the key measurements of networks which will be discussed along the chapter.

Let an undirected graph $G = (V, E)$ where V is the set of vertices and E is the set of edges connecting the nodes. The *degree* of a vertex $i \in V$ is the number of edges incident to vertex i and the *degree distribution* is the probability distribution of these degrees over the whole network. The *density* of a graph is the ratio between the number of edges and the upper bound on the number of edges.

A path connecting two vertices i, $j \in V$ is said to be minimal, if there is no other path connecting i to j with fewer links. Accordingly, the *average path length* of G is given by the average number of links in all shortest paths connecting all pairs of vertices in V. The graph *diameter* is the maximum shortest path length between all pairs of vertices in V. The *clustering coefficient of a vertex* i is the ratio between the number of edges between neighbors of vertex i and the upper bound on the number of edges between them. For instance, given $i, j, k \in N$ and assuming that edges $(i, j), (i, k) \in E$, the clustering coefficient defines the probability that (j, k) also belongs to set E. The *clustering coefficient of a graph* is the average value of the clustering coefficients of all vertices in G. The *betweenness centrality* of a vertex i is associated with an importance measure, based on the number of shortest paths between other pairs of vertices that include vertex i. The *global efficiency* of a network quantifies the efficiency in sending information between vertices, assuming that for a pair of vertices i and j it is

Table 5.1. Network Metrics

Metric	**Formula**
Order	$n = \|V\|$
Size	$m = \sum_{i,j \in V : i<j} a_{ij}$ where $a_{ij} = \begin{cases} 1, & \text{if vertices } i \text{ and } j \text{ are connected} \\ 0, & \text{otherwise} \end{cases}$
Degree	$d_i = \sum_{j \in V} a_{ij}$
Degree Distribution	$P(k) = \frac{n_k}{n}$ where n_k is the number of vertices with degree k
Density	$\rho = \frac{2m}{n(n-1)}$
Average Path Length	$L = \frac{1}{n(n-1)} \sum_{i,j \in V : i \neq j} d_{ij}$ where d_{ij} is the distance between vertices i and j
Diameter	$D = max\,\{d_{ij}\}\,, \quad \forall i, j \in V, i \neq j$
Clustering Coefficient of a Vertex	$C_i = \frac{2e_i}{r_i(r_i - 1)}$ where e_i is the number of edges between neighbors of i and r_i is the number of neighbors of vertex i
Clustering Coefficient of a Graph	$CC = \frac{1}{n} \sum_{i \in V} C_i$
Betweenness Centrality	$B_i = \sum_{s,t \in N : s \neq t} \frac{\sigma(s,i,t)}{\sigma(s,t)}, \quad s \neq i, t \neq i$ where $\sigma(s,i,t)$ is the number of shortest paths between vertices s and t that pass through vertex i and $\sigma(s,t)$ is the total number of shortest paths between s and t
Global Efficiency	$E = \frac{1}{n(n-1)} \sum_{i,j \in V : i \neq j} \frac{1}{d_{ij}}$

proportional to the reciprocal of their distance. Table 5.1 summarizes the mathematical formulas for the main network metrics outlined above. See [Costa et al., 2007] for a complete review of measurements.

5.3 Network Models

Several network models were proposed and studied in an attempt to represent elements of a system and their relationships [Lewis, 2009; Newman, 2003; Watts, 2004; Thadakamalla et al., 2008]. In the following sections, different network structures and the models used to generate them are introduced. Network models can be classified as either static or dynamic. Three static and one dynamic network models are discussed.

5.3.1 Regular Networks

Regular networks are characterized by an associated regular graph structure where the connections between vertices follow a common pattern. They have some theoretical importance since other models can be derived by rewiring the regular networks. Early work on the design of complex networks [Watts and Strogatz, 1998; Newman and Watts, 1999] contrasted the features of regular and random graphs. A regular network with vertices of degree k is called a k-regular network or regular network of degree k. Sparse k-regular networks are known to own high average path length and high clustering coefficient. As we will show in Section 5.3.3, k-regular networks are used in the design of small world networks.

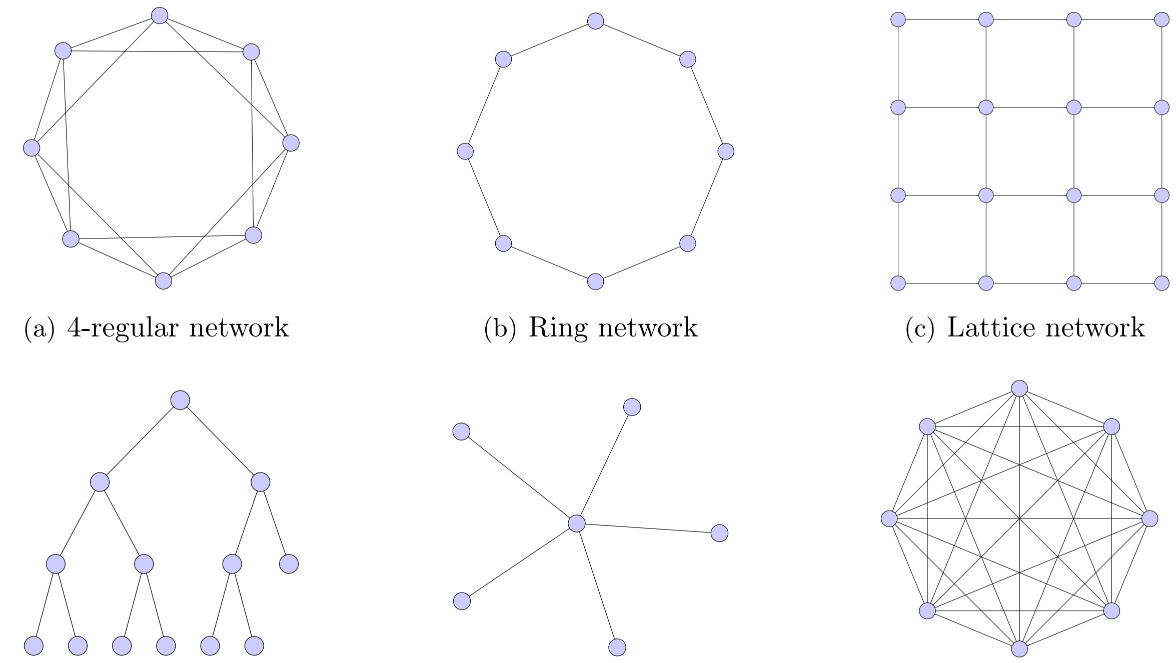
Other examples of regular networks include rings, lattices, n-ary trees, stars and full or complete graphs [Lewis, 2009]. Ring networks are a special case of a connected k-regular network, in which $k = 2$. In a lattice, the vertices are placed on a grid and connected to their immediate neighbors. A n-ary tree consists of a connected network without cycles, with a root vertex and each vertex which is not a leaf having at most n children. A star network is a special tree, where every vertex is connected to the root. In a full or complete network there is an edge between every pair of vertices. Figure 5.1 illustrates regular network structures. Figure 5.1(a), in particular, depicts a 4-regular ring network.

5.3.2 Random Networks

Random networks have been studied since the 1950s, when they were independently defined by Erdös and Rényi [1959] and Gilbert [1959]. A random network is generated by a random process, in which a set of edges are added at random between pairs of vertices belonging to the network. The class of random networks contrasts directly to that of regular networks mainly in the structure aspect, being a useful baseline for comparison.

The main idea of the Gilbert [1959] random network model is to add edges independently with probability p $(0 < p < 1)$ from the $n(n-1)/2$ potential edges of an undirected graph. Let $G(n, p)$ denote a graph G with n vertices and an associated probability p. One may note that the number of edges of a network added according to the $G(n, p)$ model is not known in advance. Moreover, the total number of possible graphs sums up to $2^{n(n-1)/2}$. On average, the resulting network ends up with $m = p[n(n-1)/2]$ and p also corresponds to the density of the network.

Another random network model, also known as ER model, was proposed by Erdös and Rényi [1959]. Unlike Gilbert's procedure, the ER model is characterized



(a) 4-regular network (b) Ring network (c) Lattice network

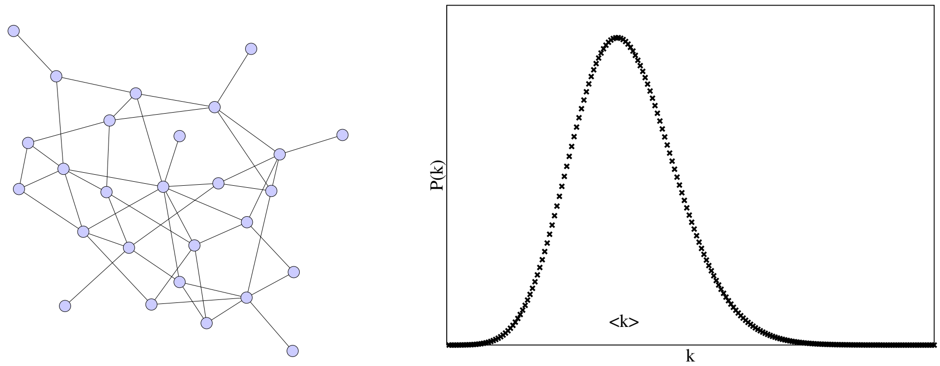
(d) Binary tree network (e) Star network (f) Full network

Figure 5.1. Regular Networks

by generating networks with a previously known fixed number m of edges. In the $G(n,m)$ model, equal probability is assigned to all graphs with exactly m edges. In other words, considering a stochastic process that starts with n vertices and no edges and at each step adds one new edge chosen uniformly from the set of missing edges, $G(n,m)$ represents a snapshot at a particular time (m) of this process.

The difference between ER and Gilbert models is that the ER model generates a network with a certain number of edges while the Gilbert model generates a network with a defined density. However, in both models the probability that a given vertex has degree k approaches a Poisson distribution for $n >> 1$, i.e., $P(k) =< k >^k e^{-k}/k!$, where $< k >$ is the average vertex degree. This means that random graphs tend to be homogeneous in vertex degree as the majority of the vertex degrees are close to the average value. The randomness attached to this class of networks induces properties based on two of the metrics presented above, small average path length and small clustering coefficient. Figure 5.2 shows an example of a random network (5.2(a)) and its Poisson degree distribution curve (5.2(b)).

For a long time, random networks were widely studied and used to model complex systems. Indeed, real world networks present an average path length close to the average path length of a random network with the same number of edges. However, in the last few years it was noticed that general properties of real world networks are



(a) Random network (b) Poisson distribution

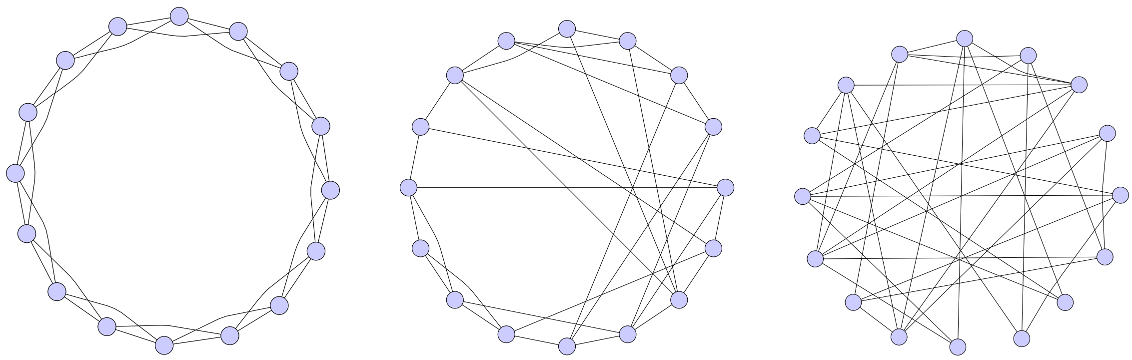
Figure 5.2. Random Network

quite different from those of random networks. For instance, the clustering coefficient of a network found in nature is remarkably larger than the clustering coefficient of a random network with the same number of vertices and edges. It seems that real networks present a kind of local interaction not observed in random networks. Furthermore, the typical degree distribution found in nature is significantly different from a Poisson distribution. Thus, new models were developed as Small World and Scale Free networks.

5.3.3 Small World Networks

The small world concept, first introduced from Milgram's experiment [Milgram, 1967], showed that the "world is small" because a person can reach all other people in the world, directly or indirectly, through few intermediaries. In [Watts and Strogatz, 1998; Newman and Watts, 1999], the authors formalized the small world concept and defined small world networks. The small world phenomenon is found in several networks like the Internet, telephone calls, road maps, food chain, electric power grids, metabolic processing networks and in social environments [Watts, 2004; Kleinberg, 2000; Watts et al., 2002]. Small world graphs are intriguing, because, among other reasons, they share characteristics of regular graphs (high clustering coefficient) and random graphs (small average path length).

In [Watts and Strogatz, 1998], a simple procedure to generate a small world network based on rewiring edges of the network was proposed by Watts and Strogatz (WS model). The WS model of a small world network is described as follows. Given a k-regular ring graph $G = (V, E)$ where each node is connected to its first k neighbors, rewiring edge $(i, j) \in E$ according to a probability p ($0 \leq p \leq 1$), consists in randomly replacing one of its endpoints i or j by another vertex q. After all edges of E are attempted to be rewired, one at a time, multiple edges and loops are not allowed, another type of graph may emerge from this process. Depending on which probabilities are used, the graph obtained may exhibit small world features. An example of the rewiring procedure is presented in Figure 5.3. Starting with the 4-regular graph ($p = 0$) depicted in Figure 5.3(a), two other types of graphs can be obtained. If a small value of p is used, a small world structure like that in Figure 5.3(b) arises. On the other hand, if larger values of p are used, random graphs like that in Figure 5.3(c) may appear.



(a) Regular network (b) Small world network (c) Random network

Figure 5.3. WS Small World Network

A similar procedure grounded in a slight improvement of the method was proposed by Newman and Watts [1999]. Instead of rewiring edges, the addition model starts with a k-regular graph $G = (V, E)$ and then adds new edges, according to probability p. As in the rewiring process, depending on the probability p, different graph structures may appear: small world graphs if p is small and random graphs if larger values of p are used.

This class of networks presents a high clustering coefficient for small values of p as the procedure starts with a regular graph, which has a high clustering coefficient value. However, the average path length falls significantly, since the random rewiring or addition of an edge works as a shortcut in the network, decreasing the distance among the vertices. This particular feature is extremely profitable in communication networks and indeed resides in real world networks such as social networks. As the global efficiency of the network (defined in Table 5.1) is based on the distance among its elements, the overall efficiency of small world networks is said to be improved comparing to regular networks.

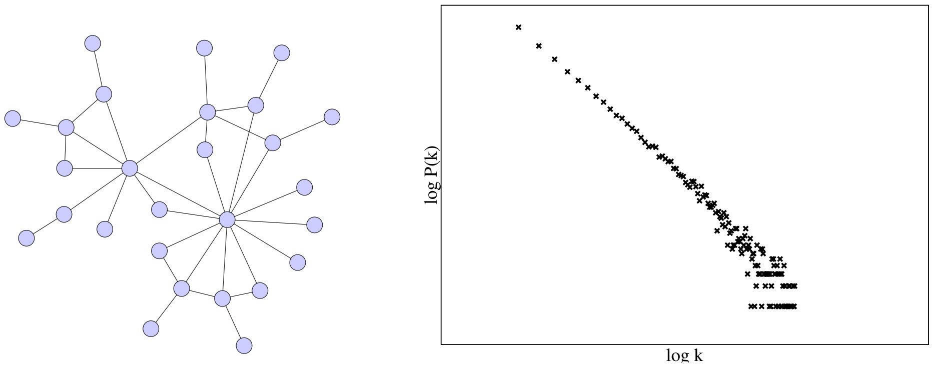
5.3.4 Scale Free Networks

The dynamic behavior of real world systems leads to the emergence of another important class of networks, known as scale free [Newman, 2003; Barabási et al., 2000; Albert and Barabási, 2002; Faloutsos et al., 1999]. Static models formerly presented are not able to capture the constant growth of a large scale network or how to attach new vertices and to connect them to existing ones. Scale free networks, in contrast to random ER graphs that follow a Poisson distribution, are characterized by a power-law degree distribution, in which there is a small number of high-degreed vertices and a large number of low-degreed vertices. The few vertices with high degree are usually called *hubs*, resulting in a network with skewed degree distribution. A power-law distribution follows the form $P(k) \sim k^{-\gamma}$, where k is the degree ($1 < k < \infty$) and γ is an exponent ($2 < \gamma < 3$).

Barabasi and Albert [1999] introduced the idea of evolving networks and addressed the origin of this power-law degree distribution in many real networks in a pioneering work. As previously noticed, contrary to the idea that real networks could be represented by random networks, it was proven that many real networks obey a power-law degree distribution instead of a Poisson distribution. Examples include the WWW, the Internet, railroads, market networks, phone call networks, and protein-protein interaction networks [Newman, 2003], just to name a few.

The Barabási-Albert network, also called BA network, is generated through a constructive procedure known as *preferential attachment*. The preferential attachment is biased (not random), in which new vertices entering the network do not connect uniformly to existing nodes, but attach preferentially to vertices of a higher degree. This model better represents the evolving real world networks, creating hubs in an unequal addition of new components. It starts with a small number (m_0) of connected vertices and assume that every time step a new vertex is added, and $m \leq m_0$ edges are connecting the new vertex to m different vertices already present in the network.The preferential attachment is incorporated assuming that the probability Π_i that a new vertex will be connected to the existing vertex i depends on the degree k_i of that vertex, so that $\Pi_i) = k_i / \sum_j k_j$. After t time steps, the model leads to a random network with $t + m_0$ vertices and mt edges.

The main property of this class of networks is the extremely high hub degree. This also means that the betweenness values of these vertices tend to be high, since they can participate in many paths connecting vertices in the network. The average path length of this class grows as $\log(n)/\log(\log(n))$ and thus displays the small world property. A linear relationship between clustering coefficient versus number of edges



(a) Scale free network (b) Power-law distribution

Figure 5.4. Scale Free Network

Table 5.2. Network Models' Features

Network Model	Features
Regular Networks	- Small average path length - High clustering coefficient
Random Networks	- Small average path length - Small clustering coefficient - Poisson degree distribution
Small World Networks	- Small average path length - High clustering coefficient
Scale Free Networks	- Small average path length - High betweenness centrality - Power-law degree distribution (small number of high-degreed vertices and high number of low-degreed vertices)

is found in a scale free network. This means that the clustering coefficient increases linearly with density ($CC \sim \mathrm{O}(\rho)$). At last, it was observed that random failures do not affect the usual operation of the network as the majority of vertices are those with a small degree and the likelihood that a hub be affected is almost minimum. On the other hand, if the vertices chosen to quit the network are specific, it may be turned into a set of isolated graphs easily. A scale free network is presented in Figure 5.4(a), while Figure 5.4(b) shows a power-law degree distribution.

5.3.5 Summary of Network Models

As presented above, each network model has specific features according to the particular structure of the network. Table 5.2 summarizes the features for each one of the network models and allows us to identify the differences among them.

As mentioned before, complex and engineered network principles follow different paradigms. Engineered networks are explored from individual components perspective related to well known metrics such as delay, resilience and efficiency. This means that for a long time, engineering problems are solved through application-driven research. On the other hand, complex theory has a more general approach, dealing with statistical metrics independent of the application, but intrinsically related to the dynamics of the network. Optimization techniques may be applied in both contexts. In order to apply optimization strategies in the complex network scenario, we must resort to some kind of association among metrics of both sides. To illustrate this idea, let us take a telecommunication network as motivation to establish a comparison. Although engineering metrics have been used for a long time as optimization criteria, it is possible to explore complex metrics to achieve similar objectives. Thus, four widely known metrics in traffic engineering for telecommunication networks were chosen: delay, load balancing, resilience and vulnerability. Table 5.3 presents the relation between engineering and complex metrics.

Table 5.3. Engineering Metrics × Complex Metrics

Telecommunication Metric	**Complex Metric**
Delay	- Small average path length - High betweenness centrality
Load balancing	- High average path length - Small betweenness centrality
Resilience	- High average path length - High clustering coefficient - Poisson degree distribution
Vulnerability	- Small average path length - High betweenness centrality - Power-law degree distribution

5.4 Optimization Models for Complex Networks

As pointed out in Section 5.3, early work in complex systems has focused mainly on how complex networks can be obtained by means of stochastic algorithms. In contrast, this section is dedicated to introduce optimization models representing networks that, if solved to optimality by exact solution algorithms, allow such networks to exhibit complex features. Our goal is to show that some complex features (such as small path length, high clustering coefficient and power-law degree distribution), often desirable for engineered networks outside the complex network domain, may arise as a result of deterministic optimization processes and algorithms.

In order to attain the goals we have set, we start with a network given by a directed graph $G = (V, A)$ having costs assigned to its arcs and state the *core* optimization problem we deal with in the section, the Optimal Topology Design Problem (OTDP). We formulate OTDP as two different Integer Programs and describe algorithms for solving each of them. Advantages of one program (and associated algorithm) over the other are also highlighted. We close the section indicating how small modifications in the core optimization problem allow us to obtain engineered networks where other complex features arise.

5.4.1 The Optimal Topology Design Problem

The Optimal Topology Design Problem (OTDP) for complex networks is defined as follows. Given a complete directed graph $G = (V, A)$ (with set of vertices V and arcs A), costs[1] $\{c_{ij} \geq 0 : \forall (i,j) \in A, i < j\}$ assigned to the arcs of A, the total number of edges D, and a budget B, OTDP consists in defining a subset $S \subseteq A : \sum_{(i,j)\in S} c_{ij} \leq B$, such that the subgraph (V, S) of G is connected and exhibits complex network features. By minimizing the average path length among all pairs of vertices under a limited budget, complex features may arise as we show in the sequence.

An Integer Program to model an optimization problem like OTDP can be defined in many different ways, depending on our choices to select decision variables, to state the constraint set and the objective function (the function we wish to minimize). Usually, there is a close connection between the way the model is formulated and the algorithms we devise to solve it.

Two Integer Programming Formulations for OTDP are proposed. In the first one, named Arc-Flow Formulation, connectivity between each pair of vertices is enforced through network flows [Ahuja et al., 1993]. In the second, the Arc-Path Formulation, connectivity is guaranteed by imposing that one path connecting every pair of vertices must be available in the subgraph of G implied by the arcs selected in a solution. In the next two sections, we discuss the two formulations. Each one leads to a different exact algorithm.

5.4.2 Arc-Flow Formulation

Let us assume that, given $G = (V, A)$, A_j^- and A_j^+ respectively denote the set of incoming and outgoing arcs in $j \in V$. To model OTDP, two sets of decision variables are used: (i) $x_{ij}^{st} \in \mathbb{R}_+$, indicates the amount of flow that leaves vertex s in direction

[1]Note that when $c_{ij} = 1 : \forall (i,j) \in A, i < j$ the problem turns into polynomial.

to vertex t, that passes through arc (i,j); (ii) $y_{ij} \in \{0,1\}$, taking value 1 if arc (i,j) is selected to belong to S (0, otherwise). OTDP can be stated as:

$$\min \sum_{s \in V} \sum_{t \in V} \sum_{(i,j) \in A} x_{ij}^{st} \tag{5.1}$$

s.t.

$$\sum_{j \in A_s^+} x_{sj}^{st} = 1 \qquad \forall s,t \in V, s \neq t, \tag{5.2}$$

$$\sum_{i \in A_t^-} x_{it}^{st} = 1 \qquad \forall s,t \in V, s \neq t, \tag{5.3}$$

$$\sum_{i \in A_j^-} x_{ij}^{st} - \sum_{k \in A_j^+} x_{jk}^{st} = 0 \qquad \forall s,t,j \in V, s \neq t, s \neq j, t \neq j, \tag{5.4}$$

$$\sum_{(i,j) \in A} c_{ij}\, y_{ij} \leq B \qquad i < j, \tag{5.5}$$

$$\sum_{(i,j) \in A} y_{ij} = D \qquad i < j, \tag{5.6}$$

$$x_{ij}^{st} + x_{ji}^{st} \leq y_{ij} \qquad \forall s,t \in V, \forall (i,j) \in A : i < j, \tag{5.7}$$

$$0 \leq x_{ij}^{st} \leq 1 \qquad \forall s,t \in V, \forall (i,j) \in A, \tag{5.8}$$

$$y_{ij} \in \{0,1\} \qquad \forall (i,j) \in A : i < j. \tag{5.9}$$

Formulation (5.1)-(5.9) states OTDP as a network design problem in which the linear objective function (5.1) can be defined to achieve efficient communication in a network. The constraints of the model guarantee the satisfaction of the basic conditions of existence of the network: topology connectivity and the limited budget for a given number of edges.

Objective function (5.1) minimizes the average path length of the whole network. Constraints (5.2)-(5.4) impose flow balance conditions for each pair of vertices s,t. Observe that (5.2) guarantees that one flow unity will be sent from s to t, for all pairs of vertices $s,t \in V$, while (5.3) ensures that the flow sent by s will arrive at destination vertex t. Equalities (5.4) assure the flow conservation in transshipment vertices. Constraint (5.5) guarantees that the selected arcs will not violate the budget. Equality (5.6) impose the total number of edges that must be included. Inequalities (5.7) couple flow and binary variables, imposing that an arc cannot be used to send flow if it is not included in the solution. Note that in model (5.1)-(5.9), variables x_{ij}^{st} were not imposed to assume binary values. However, due to constraints (5.7) and (5.9), whenever

$\{y_{ij} : (i, j) \in A : i < j\}$ variables assume integer values, $\{x_{ij}^{st} : (i, j) \in A, s, t \in V, s \neq t\}$ variables also do.

For solving OTDP by a LP based BB algorithm, we have chosen the state-of-the-art commercial solver CPLEX [2011]. The advantage of the LP based BB approach we just described for OTDP is that, after model (5.1)-(5.9) is stated and loaded into an optimization package, little additional programming effort is needed, once one has in hands an Integer Programming solver like CPLEX.

Formulation (5.1)-(5.9), however, has $O(n^4)$ variables and constraints. Therefore, only OTDP instances of limited size are expected be solved to proven optimality by LP based BB algorithms that rely on this formulation. In the following, we present a reformulation for OTDP that, despite having exponentially many variables, can lead to a specialized Branch-and-price algorithm.

5.4.3 Arc-Path Formulation

Given a pair of vertices $s, t \in V, s \neq t$, assume that P^{st} denotes the set of all simple directed paths connecting s to t in G. The main idea of the Arc-Path Formulation to enforce connectivity is to impose that, for every s, t, out of all paths in P^{st}, exactly one that uses only the arcs included in the solution will be used to compute the average paths we aim to minimize. Accordingly, in addition to binary variables y_{ij} defined previously, our second model uses binary variables $\{\lambda_p^{st} : s, t \in V, s \neq t, \forall p \in P^{st}\}$ (taking value 1 if path $p \in P^{st}$ is selected, 0, otherwise) to choose the paths. Assume that $a_{ij}^p \in \{0, 1\}$ denotes a binary parameter that indicates whether arc (i, j) belongs ($a_{ij}^p = 1$) or not ($a_{ij}^p = 0$) to path p. The Arc-Path Formulation for OTDP is given by the Integer Program:

$$\min \sum_{s \in V} \sum_{t \in V} \sum_{p \in P^{st}} d_p^{st} \lambda_p^{st} \tag{5.10}$$

s.t.

$$\sum_{p \in P^{st}} \lambda_p^{st} = 1 \qquad \forall s,t \in V, s \neq t, \tag{5.11}$$

$$\sum_{(i,j) \in A} c_{ij}\, y_{ij} \leq B, \qquad i < j, \tag{5.12}$$

$$\sum_{(i,j) \in A} y_{ij} = D \qquad i < j, \tag{5.13}$$

$$\sum_{p \in P^{st}} a_{ij}^p \lambda_p^{st} - y_{ij} \leq 0 \qquad \forall s,t \in V, \forall (i,j) \in A, \tag{5.14}$$

$$\lambda_p^{st} \in \{0,1\} \qquad \forall s,t \in V, \forall p \in P^{st}, \tag{5.15}$$

$$y_{ij} \in \{0,1\} \qquad \forall (i,j) \in A, \tag{5.16}$$

where d_p^{st} stands for the costs of the arcs in path p. As before, the objective function (5.10) minimizes the average path length among all pair of vertices. Convexity constraints (5.11) ensure that exactly one path connecting every pair of distinct vertices s,t will be selected. Inequalities (5.14) assure that if at least one selected path crosses arc (i,j), this arc should be included in the solution.

Despite the exponentially many variables (columns) in formulation (5.10)-(5.16), we can use a Delayed Column Generation to evaluate the Linear Programming bounds implied by dropping integrality constraints of model (5.10)-(5.15).

5.4.3.1 Lower Bounds in Column Generation

To understand how the LP bounds implied by (5.10)-(5.16) are evaluated, let us associate dual variables $\pi^{st} \in \mathbb{R}, \gamma \leq 0, \phi \in \mathbb{R}$ and $\beta_{ij}^{st} \leq 0$ to constraints (5.11), (5.12), (5.13) and (5.14), respectively. The LP Dual associated to the LP relaxation of (5.10)-(5.16) is given by:

$$\max \sum_{s \in V} \sum_{t \in V} \pi^{st} + B\gamma + D\phi \tag{5.17}$$

s.t.

$$\pi^{st} + \sum_{(i,j)\in A} \beta_{ij}^{st} \leq d_p^{st} \qquad \forall s,t \in V, s \neq t, \forall p \in P^{st} \tag{5.18}$$

$$c_{ij}\,\gamma + \phi - \sum_{(s,t)\in V} \beta_{ij}^{st} \leq 0 \qquad \forall (i,j) \in A, \tag{5.19}$$

$$\pi \in \mathbb{R}, \tag{5.20}$$

$$\gamma \leq 0, \tag{5.21}$$

$$\phi \in \mathbb{R}, \tag{5.22}$$

$$\beta \leq 0. \tag{5.23}$$

The LP relaxation of (5.10)-(5.16) can be computed as follows. Assume that sets of simple directed paths $C^{st} \subset P^{st}, \forall s,t \in V, s \neq t$ ($|C^{st}| << |P^{st}|$) are made available. Assume as well that, given the sets $C^{st}, \forall s,t$, the Restricted Master Program

$$\min \sum_{s\in V} \sum_{t\in V} \sum_{p\in C^{st}} d_p^{st} \lambda_p^{st} \tag{5.24}$$

s.t.

$$\sum_{p\in C^{st}} \lambda_p^{st} = 1 \qquad \forall s,t \in V, s \neq t, \tag{5.25}$$

$$\sum_{(i,j)\in A} c_{ij}\, y_{ij} \leq B, \qquad i < j, \tag{5.26}$$

$$\sum_{(i,j)\in A} y_{ij} = D \qquad i < j, \tag{5.27}$$

$$\sum_{p\in C^{st}} a_{ij}^p \lambda_p^{st} - y_{ij} \leq 0 \qquad \forall s,t \in V, \forall (i,j) \in A, \tag{5.28}$$

$$\lambda_p^{st} \geq 0 \qquad \forall s,t \in V, \forall p \in C^{st}, \tag{5.29}$$

$$y_{ij} \geq 0 \qquad \forall (i,j) \in A, \tag{5.30}$$

has one basic feasible solution $\hat{\lambda}, \hat{y}$. Let $\hat{\pi}, \hat{\gamma}, \hat{\phi}$ and $\hat{\beta}$ be the corresponding optimal dual solutions. If, for all pairs s,t, no path $p \in P^{st} \setminus C^{st}$ violates the dual constraints

$$\hat{\pi}^{st} + \sum_{(i,j)\in A} \hat{\beta}_{ij}^{st} \leq d_p^{st}, \tag{5.31}$$

then, $\hat{\lambda}, \hat{y}$ solves the LP relaxation of (5.10)-(5.15) and the corresponding optimal LP function gives a lower bound on the optimal value for (5.10). Otherwise, for a given pair s, t there must be a path in $P^{st} \setminus C^{st}$ that violates (5.31) that must be included in C^{st}. The new Restricted Master Problem, enlarged with the sets of paths associated to violated constraints (5.31), is reoptimized. The procedure goes on, until no inequality (5.31) is violated.

Let us now discuss how, for a given pair $s, t \in V$, the associated pricing problem is solved. Firstly, recall that d_p^{st} denotes the costs of the arcs in path p. Define the reduced cost of a path $p \in P^{st}$ as $\overline{c}_p^{st} := -\hat{\pi}^{st} + \sum_{(i,j)\in p}(1 - \hat{\beta}_{ij}^{st})$. In the previous definition, we allow ourselves the abuse of notation $(i, j) \in p$ to mean the arcs that are included in path p. Note that constraint (5.31) associated with path $p \in P^{st}$ is violated if and only if $\overline{c}_p^{st} < 0$. Instead of checking for the reduced cost of each possible path $p \in P^{st} \setminus C^{st}$, the pricing problem is formulated as an optimization problem as well. It involves finding a constrained shortest path connecting s to t in digraph D, under arcs costs given by $\{1 - \hat{\beta}_{ij}^{st} : (i, j) \in A\}$ (note that these costs are non negative since $\hat{\beta}_{ij}^{st} \leq 0$). The paths are constrained to use a budget that does not exceed B. If the sum of the optimal constrained path length plus $-\hat{\pi}^{st}$ is negative, the corresponding path has negative reduced cost, i.e., a violated constraint (5.31) has been found. For details on algorithms for the resolution of the Resource Constrained Shortest Path Problems see [Feillet et al., 2004; Irnich and Desaulniers, 2005]. For the particular case considered here, for each pair s, t the associated pricing problem can be solved in $O(Bn)$ time.

5.4.3.2 The Enumeration Tree

As stated before, if the LP solution achieved in the end of the Column Generation procedure is integer, it solves the original problem (5.10)-(5.16). Otherwise, being fractional, we must resort to some kind of enumeration algorithm.

One key issue in the implementation of BP algorithms is how branching is to choose a branching rule that do not destroy the structure of the pricing subproblems.

To illustrate, assume that λ_p^{st} is fractional and that branching on variable dichotomy (by imposing $\lambda_p^{st} = 1$ and $\lambda_p^{st} = 0$ on the two child nodes) is performed. While the first branching decision can be easily accommodated, the latter cannot. Note that for the first branch ($\lambda_p^{st} = 1$), it is sufficient to remove from the associated RMP all the other columns associated to paths that connect s, t and not to solve the subproblem defined by the pair s, t. However, the latter branch cannot be enforced by solving a simple shortest path problem. This is true since the same path could be regenerated, i.e., there is no guarantee that variable λ_{st}^p will not be regenerated again and again. Although this issue could be tackled by finding the next cheapest shortest path, as we go deeper in the enumeration tree, and k next-shortest paths must be found, solving the pricing subproblems become more and more inefficient.

In order to overcome the difficulty associated to the $\lambda_{st}^p = 0$ branch, we branch on variables y_{ij}. Based on variable dichotomy, it may be implemented imposing $y_{ij} = 1$ and $y_{ij} = 0$ on the two child nodes. This branching rule does not affect the pricing subproblem, requiring just the elimination of the corresponding arc from the input graph in the latter case.

The selection of which variable to branch on is based on the fractional variable associated to the maximal integer unfeasibility (farthest from integrality). This means that among all fractional variables, the variable closest to the value 0.5 is to be selected. In the case of ties, the last fractional variable found is taken. Preliminary tests indicated that this branching policy is more suitable for the problem than choosing the variable closest to one or at random. A depth-first strategy for selecting nodes from the list is also implemented, aiming to find feasible integer solutions earlier.

5.5 Computational Experience

In this Section, we report computational results for OTDP, obtained with both formulations/algorithms discussed previously. Our results are based on two networks: one with 12 vertices and 24 edges and another with 14 vertices and 28 edges. Arc costs $\{c_{ij} > 0 : (i,j) \in A : i < j\}$ are generated according to the Euclidian distance among vertices, supposing that the vertices are arranged symmetrically along a ring. Different budget values B in the interval $[B_{min}, B_{max}]$ are tested. Parameter B_{min} corresponds to the sum of the first D smaller costs on the edges, while B_{max} is the sum of the first D larger costs on the edges.

All computational results reported in this section were conducted with a Intel Xeon Core 2 Quad machine, with 2GHz and 8Gb of RAM memory, running under Linux operating system. CPLEX release 12.1 was used for both algorithms.

Table 5.4 provides computational results for both algorithms. The first two columns in the table indicate the number of nodes, n and the budget value B. In the next two columns, we report GAP (in % figures), the LP duality gap implied by the Arc-Flow formulation and t_{LP}, the time (in seconds) taken to evaluate the LP relaxation bound. The next column, named Integer Sol./t_{opt}(s) gives the time taken for the integer problem. In the next three columns, similar entries are given for the Arc-Path Formulation: the implied LP duality gap (GAP), the time t_{LP} (also in seconds) taken by the Delayed Column Generation, at the root node, to evaluate the bound, and, the number of columns (#*col*) generated at the root node. In the last four columns, we provide more detailed results for the Branch-and-Price search tree. They are: the optimal solution values (under headings *opt*), the total time t_{BP} (in seconds) spent in the search, the total number of columns (#*col*) priced out and, finally, the total number of nodes (#*nod*) investigated in the tree. A time limit of $3,600$s is applied for both algorithms.

Table 5.4. Summary of OTDP model

		Arc-Flow Formulation			Arc-Path Formulation						
n	B	Linear Relax.		Integer Sol.	Linear Relaxation			Branch-and-Price			
		GAP	t_{LR}(s)	t_{opt}(s)	GAP	t_{LR}(s)	#*col*	*opt*	t_{BP}(s)	#*col*	#nod
12	936	14.29%	0.36	**0.23**	0%	0.36	804	126	0.54	804	1
	978	3.08%	0.5	**5.45**	3.08%	0.44	715	125	31	2,165	1,223
	997	2.49%	0.4	**2.29**	2.49%	0.41	656	122	15.74	1,693	443
	1,025	2.75%	0.49	**10.89**	2.75%	0.36	591	119	17.53	1,322	565
	1,066	4.31%	0.52	**25.55**	4.31%	0.32	505	116	38.81	1,430	1,440
	1,126	4.42%	0.39	**213.52**	4.42%	0.2	436	113	363.36	2,248	16,283
	1,213	0.92%	0.42	**132.52**	0.92%	0.17	342	109	172.75	1,499	7,921
	1,340	0%	0.38	1.48	0%	0.13	282	108	**1.21**	498	28
14	910	21.43%	1.09	2.95	0%	1.31	1,959	196	**1.97**	1,959	1
	947	2.99%	1.29	226.18	2.99%	2.09	1,659	190	**127.93**	4,814	1,163
	964	2.79%	1.31	**179.75**	2.79%	2.5	1,512	185	214.66	4,608	661
	989	2.63%	1.44	**149.14**	2.63%	2.93	1,445	179	234.02	4,466	1,079
	1,025	3.81%	1.57	**276.19**	3.81%	2.75	1,273	174	951.83	5,115	7,951
	1,077	6.75%	1.63	2,566.83	*	*	*	*	*	*	*
	1,154	*	*	*	*	*	*	*	*	*	*
	1,266	*	*	*	*	*	*	*	*	*	*

* Not available, time limit exceeded.

Results in Table 5.4 indicate that in general, the BB outperforms the BP approach. The best computing times are highlighted in boldface. It can be observed that, increasing the budget B, the CPU time to solve the instances to prove optimality becomes larger. However, when budget values are too large, the LP bounds are very tight and the time taken to compute the solution drops significantly . The difficulty associated to this problem is partially observed by the gaps, i.e., how far the optimal solution is from the bounds provided by both formulations. It can be seen that for a very small budget (B_{min}), the bounds provided by the column generation approach is tighter. Although the number of columns generated by the BP is not high, the number of nodes explored in the enumeration tree can be quite high. It can be seen that none of the approaches fit well for this problem, as the number of nodes of the network increases.

Let us now investigate the subgraphs implied by the optimal solutions for different values of parameter B, trying to identify, in each of them, features that indicate improvements on the communication. In doing so, we attempt to establish how the

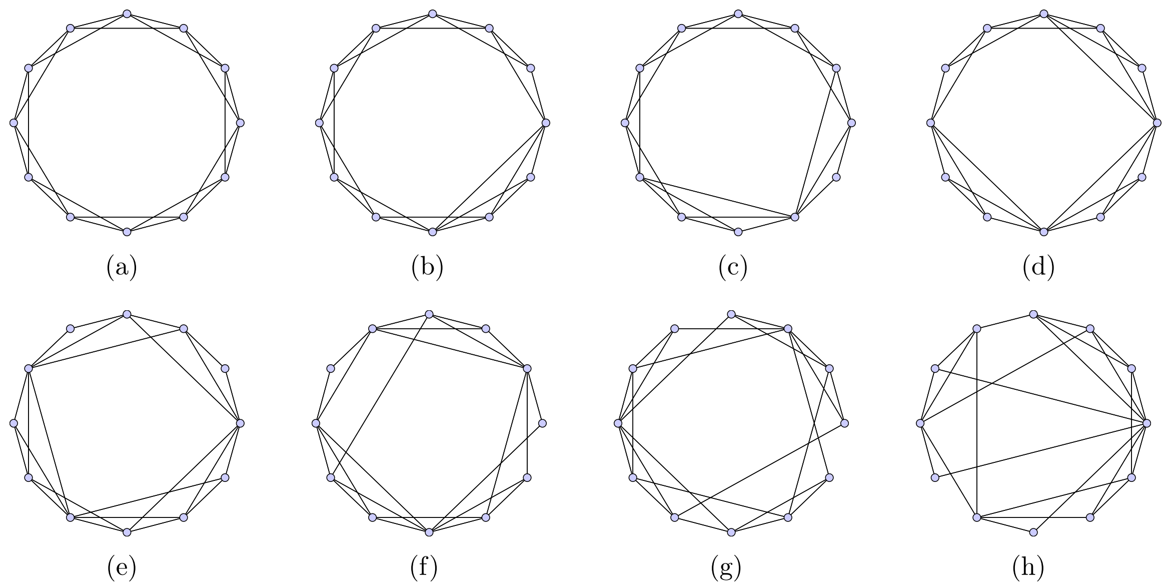


Figure 5.5. Evolution of Network Topologies for $|V| = 12$ and $D = 24$

efficiency in communication increases, as the budget B changes, as a consequence of the inherent optimization process.

In Figure 5.5, we present the optimal network topology found for each budget B, when $|V| = 12$ and $D = 24$. The smallest budget leads to a topology that can be regarded as a k-regular network structure, with $k = 4$ (see Figure 5.5(a)). As expected, when we assume B_{min} as the budget, only edges of very low cost are allowed to be included, so that a regular network emerges with a uniform degree and a high average path length. Increasing the budget values, it is possible to observe the presence of a few edges a little bit more longer (see Figures 5.5(b), 5.5(c), 5.5(d)), inducing the average path length of the network to decrease. For intermediate values of B, we also identify the appearance of hubs, associated to a power-law degree distribution (see Figures 5.5(e), 5.5(f), 5.5(g)). Finally, budget values near B_{max} clearly induce a main hub (Figure 5.5(h)), turning the average path length very small. In this sense, the budget value B plays here an important role to induce the reduction of the average path length of the network and the changes in the degree distribution of the vertices.

5.6 Creating Optimized Communication Networks

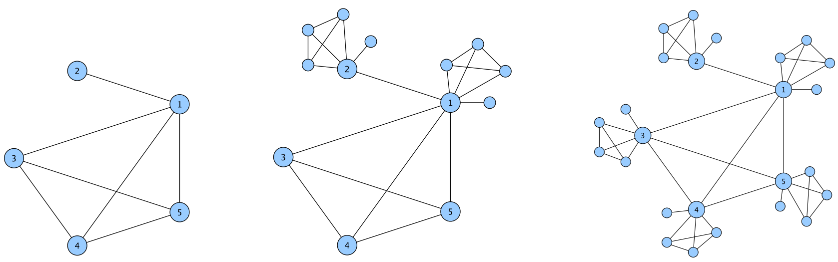
The main drawback of designing networks based on the mathematical models proposed above is that only instances of limited size are expected be solved. However, we present a simple deterministic construction procedure for larger optimized networks. Our procedure is based on the replacement of the vertices of our original network with other networks, resulting in a new also efficient communication network which have more vertices and edges than the original.

The procedure to create optimized communication networks can be divided in two steps:

1. Design a network topology (which we call original network) through the mathematical model just presented, for a given number of vertices, arcs, costs and a limited budget.

2. Expand the original network obtained in step (1), substituting each vertex by a “new network" (the new network corresponds to the original network itself).

Figure 5.6 illustrates the idea of step (2), for a network with 5 vertices. Let Figure 5.6(a) be the original network generated through step 1. Take the first vertex of the original network, say vertex 1. Substitute vertex 1 by the original network itself, so that the point of connection be the most connected vertex of the network (in this example, vertex 1). Take now vertex 2, and substitute it by the original network, allowing vertex 2 to be the most connected vertex (i.e., vertex 1). Figure 5.6(b) shows the result of both substitutions. The procedure should iterate along all vertices until the final network is reached (see Figure 5.6(c)).



(a) Original network (b) Partial network (c) Final network

Figure 5.6. Network Construction

The proposed procedure creates a new network from an original optimized network, having $|V|^2$ vertices and $(|V|+1) \times D$ edges. Scale invariance or self-similarity is an interesting property observed in complex systems [Thadakamalla et al., 2008]. This property claims that the structure of a system is similar regardless of the scale (such as fractals, for example). Following this idea, we say that the larger network obtained by the proposed procedure will preserve similar features of the original small network.

In the next section, we will present the results and discussion regarding the properties and features of the networks obtained through this construction procedure.

Figure 5.7 shows the achieved topologies when we apply the procedure described in Section 5.6 to the network topologies above and obtain new networks with $|V| = 144$ and $D = 312$. In a similar fashion of Figure 5.5, the topologies evolve from an apparently regular structure to a more interconnect structure, where hubs play a fundamental role to decrease the average path length of the networks.

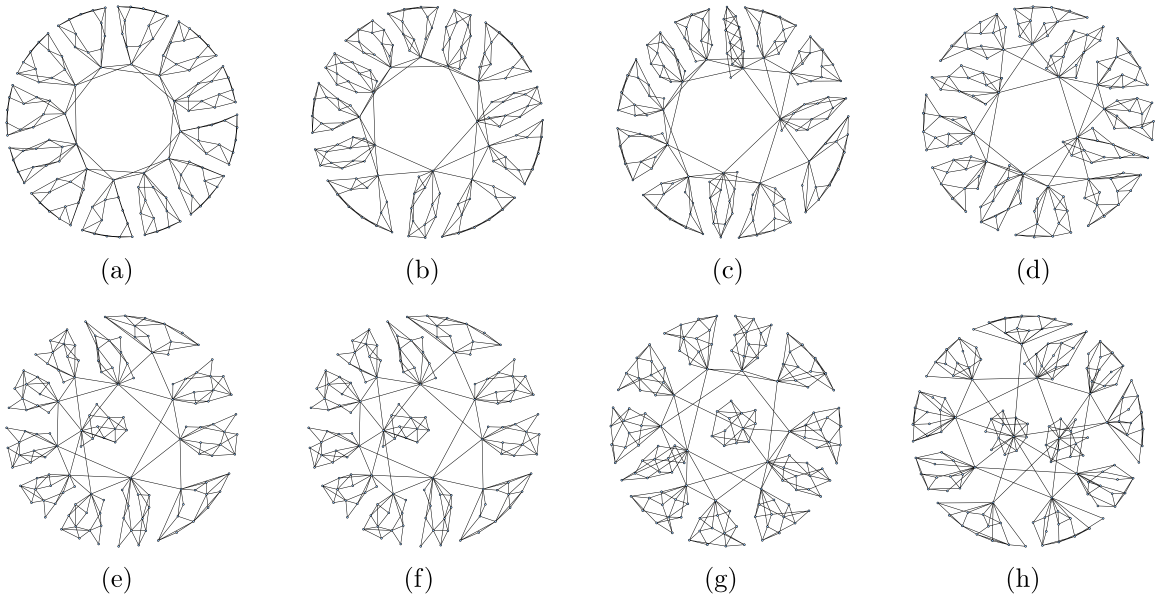
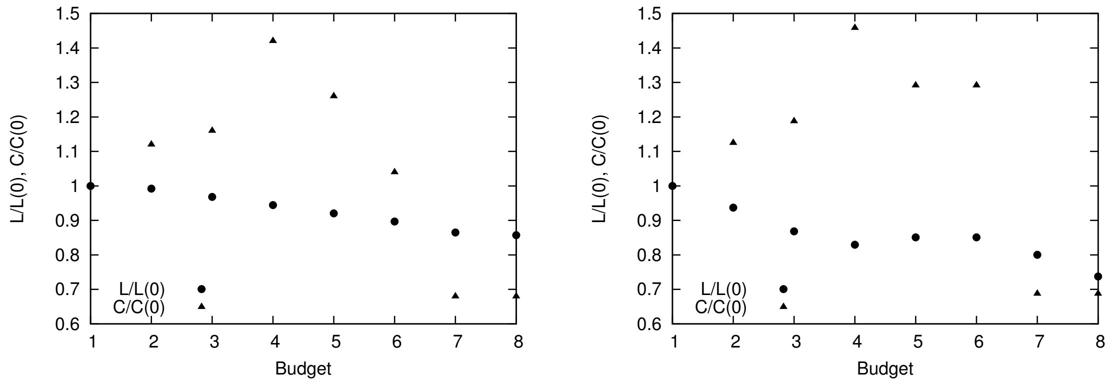


Figure 5.7. Evolution of Network Topologies for $|V| = 144$ and $D = 312$

In order to understand the behavior of the topologies as the budget value B is varied, it is important to analyze the network metrics found in those networks. For that, we calculated the average path length (L) and the clustering coefficient (C) of all topologies. As B_{min} lead us to a k-regular topology network, we take it as a basis to compare with further topologies. Thus, we say that the average path length of the basis network is given by $L(0)$ and the clustering coefficient is given by $C(0)$. We can now compute the ratio between the metrics of the generated topologies and the basis topology.

Figure 5.8(a) shows the evolution of the normalized metrics for our original network ($|V| = 12$ and $D = 24$) as the budget value B increases, while Figure 5.8(b) presents the same evaluation for the constructed larger network ($|V| = 144$ and $D = 312$). We can see that the normalized average path length slowly decreases as the budget value increases. As our original network is small, we cannot observe a sharp drop in this metric, but we can say that as the network becomes larger, the decreasing of the average path length will be more evident. On the other hand, the clustering coefficient presents a significant increase for intermediate budgets, getting lower again for larger budgets.



(a) $|V| = 12$ and $D = 24$

(b) $|V| = 144$ and $D = 312$

Figure 5.8. Path Length and Clustering Coefficient Analysis

In summary, it can be seen that although we moved from a 12 vertices network to another with 144 vertices, the properties of both networks remain very closer. Looking at Figures 5.8(a) and 5.8(b), it is possible to see that the behavior of the metrics in both figures is very similar. After evaluating these results, the following become apparent: (i) the increase of the budget allows longer range edges, reducing the average path length of the network; (ii) higher budget values induce the emergence of hub vertices in the network; and (iii) the creation of a larger network from an efficient small network leads to a network also efficient.

5.7 Exploring the Objective Function and Constraints

As mentioned before, depending on the desired complex network features and properties (network structure, function and the different metrics), different criteria may be explored in the objective function or even in the constraints of an optimization problem.

In what follows, minor modifications into the core optimization problem, OTDP, are introduced. These modifications account for different objective functions and constraints in order to capture other network features into our mathematical models. For simplicity, such functions and constraints are formulated to replace or to be appended in the Arc-Flow Formulation. Equivalent counterparts could be presented in terms of the variables of the Arc-Path Formulation. The optimization problems discussed next, thus, correspond to variants of OTDP, in which some complex metrics and network models are explored.

Generally, the formation of hubs (high-degreed vertices) is associated with a sort of vulnerability to external attacks and does not prioritize the resilience [Albert et al., 2000, 2004]. A practical way to avoid them is to impose constraints limiting vertex degrees from above. Expected topologies tend to show a regular structure as the vertex degrees will be close to the average value, a regular structure in its characteristic of high average path length and a small betweenness for all nodes. Compared to OTDP model (5.1)-(5.9), the Integer Program

$$\min \ \omega \tag{5.32}$$

s.t.

$$\sum_{j \in V} y_{ij} \leq \omega, \qquad \forall i \in V, \tag{5.33}$$

$$\tag{5.34}$$

$$\text{(5.2)-(5.9)}$$

makes use of a new decision variable ω that means the maximum degree of a vertex in a solution, in addition to x_{ij}^{st} and y_{ij} variables that keep their previously defined meaning. Note that constraints (5.33) assure that the degree of each vertex is no more than ω, the variable that it is minimized through objective function (5.32). Applying the complex metric to minimize the maximal degree of each vertex in the telecommunication context, implies to improve the load balance and the resilience aspects and consequently to reduce the vulnerability.

A third formulation can be proposed with the objective of decreasing the maximum distance (in number of hops) between pairs of vertices in the network and, therefore, leading to a network with smaller average length. Following this idea, the created network is characterized by the small world phenomenon and scale free properties since it can generate a subset of high-degreed vertices with high betweenness and a subset of low-degreed vertices. To attain this goal, we formulated the next model:

$$\min \ \chi \tag{5.35}$$

s.t.

$$\sum_{(i,j)\in A} x_{ij}^{st} \leq \chi, \qquad \forall s,t \in V \tag{5.36}$$

(5.2)-(5.9)

The model uses variable χ that represents the maximum number of arcs in any path connecting two vertices of the network. Constraints (5.36) assure that only paths with less than χ arcs can be used in a solution. The model can be seen as a telecommunication problem where the objective is to minimize delay, a well known quality of service (QoS) metric in the context of routing problems.

The objective functions of both problems use number of arcs to define ω and χ. Therefore, a mono-objective problem can also be obtained considering the sum of both functions. In this case, it is possible to balance the minimization of the maximum path length of the network and the maximum degree of the vertices in the same model:

$$\min \ \omega + \chi \tag{5.37}$$

s.t.

$$\sum_{j\in V} m_{ij} \leq \omega, \qquad \forall i \in V, \tag{5.38}$$

$$\sum_{(i,j)\in A} x_{ij}^{st} \leq \chi, \qquad \forall s,t \in V \tag{5.39}$$

(5.2)-(5.9)

Note that the objective function (5.37) seeks a trade off between average path length and vulnerability to external attacks, since it minimizes the sum of ω and χ. All the complex characteristics emphasized for both models are competing for the optimal structure. For large and sparse networks, the χ variable has a great influence. On the other hand, for small and dense networks the variable ω will predominate in the network structure.

In the next model, capacities $\{F_{ij} \geq 0 : (i,j) \in A\}$ are assigned to the use of each arc in G. The value F_{ij} limits from above the number of paths that make use of arc (i,j), in order to connect two vertices in a solution. Assume that a variable α is associated with the maximum occupation of any arc in the network. The occupation of an arc means the fraction of its capacity that is used in a solution.

$$\min \ \alpha \tag{5.40}$$

s.t.

$$\sum_{s\in V}\sum_{t\in V} x_{ij}^{st} \leq \alpha F_{ij}, \qquad \forall (i,j) \in A, \tag{5.41}$$

$$0 \leq \alpha \leq 1, \tag{5.42}$$

$$\text{(5.2)-(5.9)},$$

The model attempts to balance the flow distribution in the whole network minimizing the maximum occupation of an arc. Note that constraints (5.41) guarantee that the occupation of arc (i, j) is no more than α times its capacity F_{ij}. Solutions to this model tend to follow a regular structure, where the average path length cannot be too small since capacity constraints on the use of the arcs are now imposed and the betweenness is small for majority of the vertices. Another result is that the associated telecommunication network is likely to show more load balancing and resilience because the balanced flow distribution allows smaller losses in case of failure.

It may be the case that, under a very tight budget constraint, not all connections between pairs of vertices of V can be established. In such cases, it may be interesting to have a model that maximizes the number of pairs of vertices that actually have a path connecting them. The next model represents this case. The model uses $\{\theta^{st} : \forall s, t \in V, s \neq t\}$ variables to indicate whether or not there will be a directed path connecting s to t in the solution. The model reads:

$$\min \sum_{s\in V}\sum_{t\in V}(1 - \theta^{st}) \tag{5.43}$$

s.t.

$$\sum_{j\in A_s^+} x_{sj}^{st} = \theta^{st} \qquad \forall s, t \in V, s \neq t, \tag{5.44}$$

$$\sum_{i\in A_t^-} x_{it}^{st} = \theta^{st} \qquad \forall s, t \in V, s \neq t, \tag{5.45}$$

$$\theta \geq 0, \tag{5.46}$$

(5.4)-(5.9).

Note that the right hand side of the convexity constraints (5.44) and (5.45) may take a zero value. When that happens, no path to the corresponding pair of vertices will be available in the solution. The objective function (5.43) thus maximizes the number of pairs of vertices that can be connected. None of the complex metrics is been considered in this case, but in the context of telecommunications the objective is to maximize the number of requests accepted.

5.8 Related Work

High computing times involved in LP based Branch-and-bound algorithms (and in BP as well) preclude their use to solve huge network optimization problems. In such cases, we must resort to heuristics. Differently from exact approaches, heuristics are concerned in seeking good solutions, not necessarily the optimal one. They have been applied in the solution of several practical problems. The progress in that approach gave rise to a set of metaheuristics [Gonzalez, 2007; Talbi, 2009]. Metaheuristics are based on stochastic selection and iteratively seek for an improved candidate solution regarding a given measure of quality. Despite not guaranteeing that an optimal solution will be found, quite often they are capable of providing near-optimal solutions if properly implemented. In this section, we present related work in which metaheuristics were applied in order to create complex network topologies. In the following we present three models that, have proven their merit for generating networks with complex features and share the same principle: desired topologies are generated by means of randomized algorithms. It is important to emphasize that all models were developed independently and although each approach aimed to create complex topologies, each one formulated a different starting optimization problem in order to attain that goal.

5.8.1 Small World Optimization Algorithm

In order to achieve an efficient communication system in which the information among entities should be exchanged as fast as possible, one aims to minimize the average path length, taking into account that it is wasteful to wire everything to everything else. An optimization model based on simulated annealing [Kirkpatrick et al., 1983] is proposed for that purpose by Mathias and Gopal [2001]. The idea is to investigate whether the emergence of small world topologies could arise as a tradeoff between maximal connectivity (small path length) and minimal wiring (physical distance as a goal criterion).

The input k-regular graph is composed by vertices symmetrically placed along a ring, similar to the WS model. The size of the graph, as well as the total number of edges is fixed. Given a weighting factor $\mu \in [0, 1]$, the procedure minimizes the function $O = \mu L + (1 - \mu)W$, where L and W respectively denote the normalized average path length among all pairs of vertices and the normalized wiring cost (which corresponds to the Euclidean distance between pairs of vertices). The characteristic path length L is normalized by $L(0)$ (which is the path length in the corresponding k-regular network); W (which is a physical distance measure) is normalized by the total wiring cost that results when the edges at each vertex are the longest possible, namely, when each vertex is connected to its diametrically opposite vertex, and to the vertices surrounding it.

By weighting two goals into the objective function the optimization of either one or the other will result in two extremes. At $\mu = 0$, when the optimization lies on minimizing the cost of wiring edges, a regular network emerges with a uniform degree and a high average path length ($L \sim n$). On the other hand, at $\mu = 1$, when only the average path length is minimized, the resulting network is random ($L \sim \log n$). At intermediate values of μ, the emergence of hub vertices is observed, due to the contribution of L to the objective function. Moreover, due to the contribution of W, hubs tend to be formed by connections to the closest vertices. Also, in order to reduce the path length, hubs may appear connected. Thus, opposite to the WS model, in which the average path length decreases by the presence of long range shortcuts, the reduction here is due to a small fraction of significant hub vertices.

Analyzing metrics as average path length, clustering coefficient and wiring cost and comparing them to the metrics from WS model, the following became apparent: (i) in both models L shows a sharp drop related to the small word behavior, such that the drop caused by hub formation is much sharper; (ii) the drop observed in CC in the WS model is not valid for the optimized network, since hub formation keeps the CC at values higher than those for regular networks; (iii) the minimal wiring objective makes a clearly difference between the two models, as for larger values of μ the amount of wiring in the WS model is much greater than in the optimization model.

Mathias and Gopal [2001] conclude that the optimized networks are more clustered than corresponding regular networks, and have a smaller average degree of separation than their corresponding random graphs. Besides, small world topologies that arise from optimization consumes less wiring than their WS counterparts, being useful when wiring is expensive.

5.8.2 Scale Free Optimization Algorithm

An evolutionary algorithm for optimized network design is presented by Cancho and Sole [2001], combining into the objective function, the minimization of the graph density and the average path length. These objectives include two relevant aspects of network performance: the cost of physical links and the communication speed among entities. Observing that most complex networks are extremely sparse and exhibit the so-called small world phenomenon, a minimization procedure based on these two criteria was expected to lead to small world and hub formation features.

The proposed procedure consists in optimizing an energy function defined as $E(\phi) = \phi d + (1-\phi)\rho$, where $0 \leq \phi, d, \rho \leq 1$. ϕ is a parameter controlling the linear combination of d (average path length) and ρ (density), which are normalized accordingly. The minimization of $E(\phi)$ involves the simultaneous minimization of distance and number of links (which is associated with cost).

The algorithm proposed by Cancho and Sole [2001] works with discrete time intervals. Starting at time $t = 0$, the network is set up with a density $\rho(0)$ following a Poisson distribution of degrees (connectedness is enforced). At time $t > 0$, the graph is modified by randomly changing the state of some pairs of vertices. For instance, with probability ν, each a_{ij} can switch from 0 to 1 or vice-versa. The new adjacency matrix is accepted if $E(\phi, t+1) < E(\phi, t)$. Otherwise, a different set of changes is tested. The algorithm stops when the modifications applied are not accepted a given number of times in sequence. This number is a parameter of the algorithm, set up by the designers.

Depending on how density and path length are weighted into the objective function, four main types of networks can be found: (a) exponential networks, (b) scale free networks, (c) star networks and (d) dense networks. Analyzing some basic properties such as density, clustering coefficient and path length as a function of ϕ along with another measure defined as degree entropy, the authors identified four different phases, separated by three sharp transitions at $\phi_1^* \approx 0.25, \phi_2^* \approx 0.80$ and $\phi_3^* \approx 0.95$. Examining the degree distributions achieved by the procedure, it is possible to note that the transitions are easily explained since from (a) to (b) hub formation emerges, from (b) to (c) a hub competition leads to a central vertex and finally a dense graph (d) results when a progressive increase in the average degree of non-central vertices occurs and a sudden loss of the central vertex.

The results in [Cancho and Sole, 2001] suggest that preferential attachment networks (scale free) might emerge at the boundary between random attachment networks (a) and forced attachment (all vertices linked to a central vertex) networks (c). Exponential like networks appear when the path length is minimized under high density weight. When linking cost substantially decreases, the reduction of vertex-vertex distance is enforced heading to a complete graph for high values of ϕ.

5.8.3 Small World Topologies using GRASP

The problem of generating a small world topology treated in [Souza et al., 2008] consists in turning a regular graph into a small world graph with the principle of adding new edges to it (such as the addition model presented in section 5.3.3), minimizing its average path length. Instead of the probability p used in the stochastic model from the literature, the authors make use of an additional parameter B called budget, which defines the number of shortcuts (edges) that may be included in the graph. Therefore, the input parameters are the original regular graph and the budget value.

A GRASP approach is adopted for solving the studied problem. GRASP (Greedy Randomized Adaptive Search Procedure) is an iterative method for solving optimization problems proposed by Feo and ResendeFeo and Resende [1995], composed by two phases: a construction phase, in which a solution is built from scratch; and a refinement phase (local search), in which a local optima solution is reached. The best solution found during the GRASP iterations is returned as the result of the algorithm.

In the particular application considered here, at each iteration of the construction phase, a new solution is created by adding to the graph as many edges that can be fitted with the budget. A higher priority is given to edges offering best benefit (low cost and high impact in reducing the average shortest path length of the graph). However, a portion of randomness is also included in order to avoid a purely greedy behavior that allows the procedure to be applied repeatedly, in a multi-start scheme. The refinement phase, which is applied only to the best solution found in the construction phase, works as follows. For each edge in the current solution, we attempt to replace it by an edge that if included in the solution does not violate the budget and decreases the average shortest path length. Note that the evaluation of the benefit of the movement (the operation of replacing one edge by another) is very time consuming. That is the reason why in the proposed implementation of GRASP, the local search is applied only to the best solution, and not to all solutions provided by GRASP at the end of its first phase.

In Table 5.5, we show how the network obtained by the application of GRASP compares to the optimal one, in terms of their average path length. Optimal networks were obtained by means of the Branch-and-bound algorithm based on the Arc-Flow Formulation (named here LP BB) proposed in section 5.4.2 with an additional constraint set that forces every edge initially included in the regular graph to be included in the final solution as well. In doing so, the quality of the solution provided by GRASP can be compared to the optimal one.

For each value of B in our test bed, we report in Table 5.5, how much the solutions provided by GRASP and LP BB improve the average path length of the solution provided by the stochastic model from section 5.3.3. For example, for $B = 25$, the average shortest path length of the optimal solution is 7.7% inferior than the average shortest path length of the solution obtained by the stochastic model.

Table 5.5. Path Length Improvement

B	$n = 15$		B	$n = 30$	
	GRASP	LP BB		GRASP	LP BB
1	3.2%	3.2%	3	8.1%	8.5%
4	8%	9%	25	7.2%	7.7%
5	1.1%	2.2%	38	2.9%	3.5%
6	4.4%	7.1%	64	0%	0.6%
10	6.9%	6.9%	107	0%	0.6%
19	1.2%	1.2%	148	0%	0%
60	0%	0%	240	0%	0%

As one could expect, graphs generated by the optimization approaches present better values of the average path length (indicated by the improvement values) compared to the stochastic method, except when the budget value B is very high (in such cases, all approaches provided solutions with identical average path length). This means that using the same number of additional links, the optimization approaches are capable of finding small world networks with a smaller average path length, increasing the efficiency of the whole network.

5.9 Conclusion and Future Work

In this chapter, we presented different alternatives to design complex communication networks. Besides the stochastic methods from the literature, we focused on how optimization techniques both based on exact solution methods as well as in heuristics may be applied to generate complex communication networks. The main difference between both approaches concerns in the scalability provided by them.

Irrespective of how the optimization models are solved (through heuristics or exact methods), it can be observed that all optimization techniques have shown that features such as small path length, high clustering coefficient and power-law degree distribution can be achieved. It has been shown that the optimization of different criteria in the objective function and constraints leads to diverse complex network topologies.

Chapter 6

Conclusion and Future Work

The final remarks and future work are presented in this chapter. Section 6.1 concludes the thesis with a summary of accomplished work. Section 6.2 addresses possible future work to proceed the research on this study.

6.1 Final Remarks

In this thesis, we studied how to apply optimization techniques in the design of communication networks. The first problem investigated, named Resilient Multi-level Hop-constrained Network Design, concerns designing hierarchical telecommunication networks, assuring resilience against random failures and maximum delay guarantees in the communication. After, we investigated solutions to the Grooming, Routing and Wavelength Assignment problem with protection and quality of service in WDM optical networks. At last, we addressed to the Optimal Topology Design of Complex Networks, which consists of designing efficient communication networks based on complex networks features.

Three mathematical formulations were proposed for the Resilient Multi-level Hop-constrained Network Design problem. Through our computational experiments, it was shown that the bounds provided by the Arc-Flow Formulation are quite weak, taking high computational times while solving the problem via a Branch-and-bound algorithm. The lower bounds provided by the other two formulations are stronger and their equivalence is demonstrated by a formal mathematical proof. However, a Branch-and-price algorithm based on the Arc-Path Formulation proved to be more efficient compared to a traditional Branch-and-Bound algorithm for the Aggregated Hop-Indexed Formulation. Finally, we conclude that real world networks adapted to the problem are solvable to optimality by our Branch-and-price approach.

Two formulations for the Grooming, Routing and Wavelength Assignment problem with protection and quality of service in WDM optical networks were proposed. It was shown that exact algorithms developed specifically for the problem by advanced optimization techniques such as column generation and branch-and-price led to the solution of real-world networks. Our computational results show that the times required by the proposed BP algorithm are feasible in all cases studied. Moreover, in order to deal with larger instances (with a high number of requests), we proposed a column generation-based heuristic, that finds good solutions within a small computational time. Finally, we evaluated how protection constraints can provide different levels of QoS in the network. We conclude that there is a tradeoff between the number of wavelengths used and the amount of traffic lost in case of failure.

The Optimal Topology Design of Complex Networks was explored through two basic formulations and some variations. For this problem, the BB algorithm outperforms the BP approach. By exploring different objective functions and constraints based on complex network metrics, we may achieve diverse topologies, related to well known metrics from engineered networks, such as delay, load balancing, resilience and robustness.

The main contributions of this thesis project include different mathematical formulations for three interesting network design problems and a Branch-and-price approach for each of them. Through a comparative analysis, we showed the performance of different algorithms based on the proposed formulations. We demonstrate that the proposed Branch-and-price approach outperforms other traditional optimization tools for two of the problems.

6.2 Future Work

In this section, we describe the next activities that can be developed. Despite its theoretical importance, an exact approach is limited by the size of instances that can be tackled in practice. Thereby, this step includes the development of heuristics for all problems. We expect that good solutions can be reached for instances with at least one order of magnitude larger. Besides, many other particular aspects can be explored in all problems. In the following, we describe some future work for each of them.

- **RMHND**: A pure BP approach was devised to deal with this problem. Some future directions include the combination of the exact approach with a heuristic, in order to achieve good integer solutions early. Also, acceleration strategies may be added to our column generation algorithm. Improvements in this work can lead us to solve larger instances for the problem.

- **GRWA-PQoS**: This problem admits many variations. In a future step, features such as multi-domain, shared protection and dynamic traffic grooming can apply. The addition of new features can make the problem more practical and closer to a real scenario.
- **OTDP**: As the BP approach did not achieve good results for this problem, some improvements should be taken into account. For example, the addition of cuts to the problem could help the BP to present a better performance.

Bibliography

Ahuja, R. K., Magnanti, T. L., and Orlin, J. B. (1993). *Network Flows: Theory, Algorithms and Applications.* Prentice Hall, USA.

Albert, R., Albert, I., and Nakarado, G. (2004). Structural vulnerability of the north american power grid. *Physical Review E*, 69(2).

Albert, R. and Barabási, A.-L. (2002). Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74(1):47--97.

Albert, R., Jeong, H., and Barabási, A.-L. (1999). The diameter of the world wide web. *Nature*, 401:130.

Albert, R., Jeong, H., and Barabási, A.-L. (2000). The internet's achilles' heel: Error and attack tolerance of complex networks. *Nature*, 406:200--0.

Alderson, D. L. (2008). Catching the 'network science' bug: Insight and opportunity for the operations researcher. *Operations Research*, 56(5):1047--1065.

Amor, H. B., Desrosiers, J., and Frangioni, A. (2009). On the choice of explicit stabilizing terms in column generation. *Discrete Applied Mathematics*, 157:1167--1184.

Balakrishnan, A., Magnanti, T. L., and Mirchandani, P. (1991). The multi-level network design problem. Working papers 3366-91, Massachusetts Institute of Technology (MIT), Sloan School of Management.

Balakrishnan, A., Magnanti, T. L., and Mirchandani, P. (1994a). A dual-based algorithm for multi-level network design. *Management Science*, 40(5):567--581.

Balakrishnan, A., Magnanti, T. L., and Mirchandani, P. (1994b). Modeling and heuristic worst-case performance analysis of the two-level network design problem. *Management Science*, 40(7):846--867.

Balakrishnan, A., Magnanti, T. L., and Mirchandani, P. (1998). Designing hierarchical survivable networks. *Operations Research*, 46(1):116--136.

Barabasi, A.-L. and Albert, R. (1999). Emergence of scaling in random networks. *Science*, 286(5439):509–512.

Barabási, A.-L., Albert, R., and Jeong, H. (2000). Scale-free characteristics of random networks: the topology of the world-wide web. *Physica A: Statistical Mechanics and its Applications*, 281(1-4):69--77.

Barnhart, C., Hane, C. A., and Vance, P. H. (2000). Using branch-and-price-and-cut to solve origin-destination integer multicommodity flow problems. *Operations Research*, 48:318--326.

Barnhart, C., Johnson, E., Nemhauser, G., Savelsbergh, M., and Vance, P. (1998). Branch-and-price: column generation for solving huge integer programs. *Operations Research*, 48:318--326.

Barr, R. S., Kingsley, M. S., and Patterson, R. A. (2006). *Telecommunications Network Grooming*, chapter 29, pages 837–862. Springer US, New York, USA.

Bazaraa, M. S., Jarvis, J. J., and Sherali, H. D. (2004). *Linear Programming and Network Flows*. Wiley-Interscience.

Bondy, J. A. and Murty, U. S. R. (1976). *Graph theory with applications*. Macmillan, London.

Cancho, R. F. and Sole, R. V. (2001). Optimization in complex networks. Working Papers 01-11-068, Santa Fe Institute.

Chinneck, J. W., Kristjansson, B., and Saltzman, M. J. (2009). *Operations Research and Cyber-Infrastructure*. Springer Publishing Company, Incorporated, 1 edition.

Chopra, S. and Rao, M. R. (1994). The steiner tree problem i: Formulations, compositions and extension of facets. *Mathematical Programming*, 64:209–229. 10.1007/BF01582573.

Costa, L. F., Rodrigues, F. A., Travieso, G., and Boas, P. R. V. (2007). Characterization of complex networks: A survey of measurements. *Advances In Physics*, 56:167--242.

CPLEX (2011). www.cplex.com.

Current, J. R., ReVelle, C. S., and Cohon, J. L. (1986). The hierarchical network design problem. *European Journal of Operational Research*, 27(1):57–66.

Dantzig, G. (1963). *Linear Programming and Extensions*. Princeton University Press.

De, T., Pal, A., and Sengupta, I. (2010). Traffic grooming, routing, and wavelength assignment in an optical wdm mesh networks based on clique partitioning. *Photonic Netw. Commun.*, 20(2):101--112.

Desrosiers, J. and Lübbecke, W. (2005). A primer in column generation. In Desaulniers, G., Desrosiers, J., and Solomon, M., editors, *Column Generation*, pages 1--32. Springer US.

Diestel, R. (2005). *Graph Theory*, volume 173 of *Graduate Texts in Mathematics*. Springer-Verlag, Heidelberg, third edition.

Erdös, P. and Rényi, A. (1959). On random graphs i. *Publicationes Mathematicae Debrecen*, 6:290--297.

Faloutsos, M., Faloutsos, P., and Faloutsos, C. (1999). On power-law relationships of the internet topology. In *SIGCOMM '99: Proceedings of the conference on Applications, technologies, architectures, and protocols for computer communication*, pages 251--262, New York, NY, USA. ACM.

Feillet, D., Dejax, P., Gendreau, M., and Gueguen, C. (2004). An exact algorithm for the elementary shortest path problem with resource constraints: Application to some vehicle routing problems. *Networks*, 44(3):216--229.

Feo, T. A. and Resende, M. G. (1995). Greedy randomized adaptive search procedures. *Journal of Global Optimization*, 6:109--133.

Frank, H., Kahn, R. E., and Kleinrock, L. (1972). Computer communication network design: experience with theory and practice. In *Proceedings of the May 16-18, 1972, Spring Joint Computer Conference*, AFIPS '72 (Spring), pages 255--270, New York, NY, USA. ACM.

Gilbert, E. (1959). Random graphs. *Annals of Mathematical Statistics*, 30:1141--1144.

Gomory, R. E. (1958). Outline of an algorithm for integer solutions to linear programs. *Bulletin of the American Society*, 64:275--278.

Gonzalez, T., editor (2007). *Handbook of Approximation Algorithms and Metaheuristics (Chapman & Hall/CRC Computer & Information Science Series)*. Chapman and Hall/CRC, 1 edition.

Gouveia, L., Patricio, P., and Sousa, A. (2006). *Compact Models for Hop-Constrained Node Survivable Network Design: an Application to MPLS*, volume 33 of *Operations Research/Computer Science Interfaces Series*, chapter 9, pages 167--180. Springer, New York.

Gouveia, L. and Telhada, J. (2001). An augmented arborescence formulation for the two-level network design problem. *Annals of Operations Research*, 106(1-4):47–61.

Gouveia, L. and Telhada, J. (2008). The multi-weighted steiner tree problem: A reformulation by intersection. *Computers & Operations Research*, 35(11):3599–3611.

Guidoni, D. L., Boukerche, A., Souza, F. S., Mini, R. A., and Loureiro, A. A. (2010). A small world model based on multi-interface and multi-channel to design heterogeneous wireless sensor networks. In *Proceedings of the IEEE Global Communications Conference (GLOBECOM '10)*.

Guidoni, D. L., Boukerche, A., Villas, L. A., Souza, F. S. H., Mini, R. A., and Loureiro, A. A. (2012). A framework based on small world features to design hsns topologies with qos. In *Proceedings of the IEEE Symposium on Computers and Communication (ISCC'12)*.

Hu, J. and Leida, B. (2004). Traffic grooming, routing, and wavelength assignment in optical wdm mesh networks. In *Proceedings of the IEEE INFOCOM*, pages 495--501.

Irnich, S. and Desaulniers, G. (2005). Shortest path problems with resource constraints. In Desaulniers, G., Desrosiers, J., and Solomon, M., editors, *Column Generation*, pages 33--65. Springer US.

Jaekel, A., Bari, A., and Bandyopadhyay, S. (2008). Resilient traffic grooming for wdm networks. *J. Opt. Netw.*, 7(5):378--387.

Joncour, C., Michel, S., Sadykov, R., Sverdlov, D., and Vanderbeck, F. (2010). Column generation based primal heuristics. *Electronic Notes in Discrete Mathematics*, 36:695--702.

Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, 220(4598):671–680.

Kleinberg, J. M. (2000). Navigation in a small world. *Nature*, 406(6798):845--845.

Koch, T., Martin, A., and Achterberg, T. (2004). Branching rules revisited. *Operations Research Letters*, 33:42--54.

Krishnaswamy, R. and Sivarajan, K. (2001). Design of logical topologies: a linear formulation for wavelength-routed optical networks with no wavelength changers. *IEEE/ACM Transactions on Networking*, 9(2):186–198.

Land, A. H. and Doig, A. G. (1960). An automatic method of solving discrete programming problems. *Econometrica*, 28(3):497--520.

Lasdon, L. S. (1970). *Optimization Theory for Large Scale Systems.* McMillan Company, New York, USA.

Lewis, T. G. (2009). *Network Science: Theory and Applications.* Wiley Publishing.

Lübbecke, M. E. and Desrosiers, J. (2005). Selected topics in column generation. *Operations Research*, 53(6):1007--1023.

Magnanti, T. L. and Wong, R. T. (1984). Network design and transportation planning: Models and algorithms. *Transportation Science*, 18:1--56.

Mathias, N. and Gopal, V. (2001). Small worlds: How and why. *Physics Review E*, 63(2):021117.

Merle, O. D., Villeneuve, D., Desrosiers, J., and Hansen, P. (1997). Stabilized column generation. *Discrete Mathematics*, 194:229--237.

Milgram, S. (1967). The small world problem. *Psychology Today*, 2:60--67.

Mukherjee, B. (1997). *Optical Communication Networks.* McGraw-Hill.

Mukherjee, B. (2006). *Optical WDM Networks.* Springer.

Newman, M. E. J. (2003). The structure and function of complex networks. *SIAM Review*, 45:167--256.

Newman, M. E. J. and Watts, D. J. (1999). Scaling and percolation in the small-world network model. *Physical Review E*, 60(6):7332--7342.

Obreque, C., Donoso, M., Gutierrez, G., and Marianov, V. (2010). A branch and cut algorithm for the hierarchical network design problem. *European Journal of Operational Research*, 200(1):28–35.

O'Mahoney, M., Simeonidou, D., Yu, A., and Zhou, J. (1995). The design of a european optical network. *Journal of Lightwave Technology*, 13(5):817 –828.

Ou, C. S., Zhu, K., Zang, H., Sahasrabuddhe, L. H., and Mukherjee, B. (2003). Traffic grooming for survivable wdm networks -âĂŞ shared protection. *IEEE Journal on Selected Areais in Communications*, 21:1367--1383.

Ou, C. S., Zhu, K., Zhang, J., Zhu, H., Mukherjee, B., Zang, H., and Sahasrabuddhe, L. H. (2004). Traffic grooming for survivable WDM networks: dedicated protection [Invited]. *Journal of Optical Networking*, 3:50.

Raghavan, S. and Stanojević, D. (2011). Branch and price for wdm optical networks with no bifurcation of flow. *INFORMS J. on Computing*, 23(1):56--74.

Resende, M. G. and Pardalos, P. M., editors (2005). *Handbook of Optimization in Telecommunications*. Springer, New York, NY, USA.

Rousseau, L.-M., Gendreau, M., and Feillet, D. (2007). Interior point stabilization for column generation. *Operations Research Letters*, 35(5):660–668.

Somani, A. K. (2005). *Survivability and traffic grooming in WDM optical networks*. Cambridge University Press.

Souza, F. S. H., Cunha, A. S., and Mateus, G. R. (2009). Optimal topology design of complex networks. In *INFOCOM'09: Proceedings of the 28th IEEE International Conference on Computer Communications Workshops*, pages 278--283, Piscataway, NJ, USA. IEEE Press.

Souza, F. S. H., Cunha, A. S., and Mateus, G. R. (2010a). On the design of complex networks through a branch-and-price algorithm. In *IEEE GLOBECOM Workshop on Complex and Communication Networks*, pages 393--397.

Souza, F. S. H., Cunha, A. S., and Mateus, G. R. (2012a). *Optimization in Designing Complex Communication Networks*, volume 57 of *Springer Optimization and Its Applications*, chapter 1, pages 3--38. Springer, New York, USA.

Souza, F. S. H., Gendreau, M., and Mateus, G. (2010b). A column generation algorithm for the resilient multi-level hop-constrained network design. In *10th INFORMS Telecommunications Conference (extended abstract).*

Souza, F. S. H., Gendreau, M., and Mateus, G. (2010c). A column generation algorithm for the resilient multi-level hop-constrained network design. In *Journées de l'Optimisation/Optimization Days (abstract).*

Souza, F. S. H., Guidoni, D., and Loureiro, A. (2008). Uma abordagem baseada em grasp para geração de topologias small world. In *XL Simpósio Brasileiro de Pesquisa Operacional*.

Souza, F. S. H., Guidoni, D. L., and Mateus, G. R. (2012b). Formulations for the problem of routing and wavelength assignment (rwa) with traffic grooming, protection and quality of service on wdm optical networks. In *Proceedings of the IEEE Symposium on Computers and Communication (ISCC'12)*.

Srikitja, A., Tipper, D., and Medhi, D. (1999). On providing survivable qos services in the next generation internet. In *IN PROC. OF MILCOM*, pages 30602--97.

Talbi, E.-G. (2009). *Metaheuristics: From Design to Implementation*. Wiley Publishing.

Thadakamalla, H., Kumara, S. R. T., and Albert, R. (2008). *Complexity and Large-Scale Networks*, volume 4 of *Operations Research Series*, chapter 11. CRC Press, Taylor and Francis Group, Pennsylvania State University, University Park, USA.

Thadakamalla, H. P., Raghavan, U. N., Kumara, S., and Albert, R. (2004). Survivability of multiagent-based supply networks: A topological perspective. *IEEE Intelligent Systems*, 19:24–31.

Vanderbeck, F. (2010). Branching in branch-and-price: a generic scheme. *Mathematical Programming*, pages 1--46. 10.1007/s10107-009-0334-1.

Vignac, B., Vanderbeck, F., and Jaumard, B. (2009). Reformulation and decomposition approaches for traffic routing in optical networks. Research report, INRIA.

Watts, D. J. (2004). *Six Degrees: The Science of a Connected Age*. W. W. Norton & Company.

Watts, D. J., Dodds, P. S., and Newman, M. E. J. (2002). Identity and search in social networks. *Science*, 296:1302.

Watts, D. J. and Strogatz, S. H. (1998). Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440--442.

Wolsey, L. (1998). *Integer Programming*. John Wiley and Sons.

Yao, W. and Ramamurthy, B. (2005). Survivable traffic grooming with path protection at the connection level in wdm mesh networks. *J. Lightwave Technol.*, 23(10):2846.

Zang, H., Jue, J. P., and Mukherjee, B. (2000). A review of routing and wavelength assignment approaches for wavelength-routed optical wdm networks. *Optical Networks Magazine*, 1:47--60.

Zhu, K. and Mukherjee, B. (2002). Traffic grooming in an optical wdm mesh network. *IEEE Journal on Selected Areais in Communications*, 20(1):122--133.

Zhu, K. and Mukherjee, B. (2003). A review of traffic grooming in wdm optical networks: Architectures and challenges. *Optical Networks Magazine*, 4(2):55–64.