MULTI-LAYER NETWORK DESIGN PROBLEMS IN TELECOMMUNICATION

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

İNCİ YÜKSEL ERGÜN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE DOCTOR OF PHILOSOPHY
IN
INDUSTRIAL ENGINEERING

SEPTEMBER 2013

Approval of the thesis

MULTILAYER NETWORK DESIGN PROBLEMS IN TELECOMMUNICATION

Submitted by **İNCİ YÜKSEL ERGÜN** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy in Industrial Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen ______________________
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Murat Köksalan ______________________
Head of Department, **Industrial Engineering**

Prof. Dr. Ömer Kırca ______________________
Supervisor, **Industrial Engineering Dept., METU**

Assoc. Prof. Dr. Haldun Süral ______________________
Co-Supervisor, **Industrial Engineering Dept., METU**

Examining Committee Members:

Assoc. Prof. Dr. Sinan Gürel ______________________
Industrial Engineering Dept., METU

Prof. Dr. Ömer Kırca ______________________
Industrial Engineering Dept., METU

Assist. Prof. Dr. Cem İyigün ______________________
Industrial Engineering Dept., METU

Assoc. Prof. Dr. Oya Karaşan ______________________
Industrial Engineering Dept., Bilkent University

Assoc. Prof. Dr. Cüneyt Bazlamaçcı ______________________
Electrical and Electronics Engineering Dept., METU

Date: 04 September 2013

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: İnci, Yüksel Ergün

Signature :

ABSTRACT

MULTI-LAYER NETWORK DESIGN PROBLEMS IN TELECOMMUNICATION

YÜKSEL ERGÜN, İnci
PhD, Department of Industrial Engineering
Supervisor : Prof. Ömer KIRCA
Co-Supervisor: Assoc. Prof. Haldun SÜRAL

SEPTEMBER 2013, 223 pages

The telecommunication network design problem is to configure a telecommunication network of major hardware and their links in order to satisfy traffic demands and flows subject to a set of constraints arising from topology, capacity, and technology. Telecommunication network design has been studied in several disciplines and its literature is intricate. In this study, we classify the telecommunication network design problems in the literature from the perspective of operations research and review the network optimization problems to match design problems with optimization problems. Our review examines mainly decision problems, mathematical formulations, and effective solution methods for the relevant network optimization problems. We address the multilayer telecommunication network design problem consisting of networks with several layers working interdependently and investigate its sub-problems and capabilities of existing formulations. We suggest a novel mathematical formulation that models all layers using a single-mega network and incorporates various practical decision problems. Our computational experiments show that the problem instances with more than two layers, which are not computationally tractable with the existing formulations, can be solved using the NFF by general-purpose integer programming solvers. We also develop tailored solution algorithms based on Benders decomposition to solve the large telecommunication network design problems that cannot be handled by general solvers. Consolidating the available test problem instances in the literature, we perform extensive computational experiments on these instances to assess the behavior of the algorithms and to present favorable results.

Keywords: Network design, classification, telecommunication, multilayer telecommunication network design

ÖZ

ÇOK KATMANLI TELEKOMÜNİKASYON AĞ TASARIMI PROBLEMLERİ

YÜKSEL ERGÜN, İnci
Doktora, Endüstri Mühendisliği Bölümü
Tez Yöneticisi : Prof. Dr. Ömer KIRCA
Ortak Tez Yöneticisi: Doç. Dr. Haldun SÜRAL

EYLÜL 2013, 223 Sayfa

Telekomünikasyon ağ tasarımı problemi, ağın trafik talebinin karşılanması için ana donanımlar ve bağlantılarından oluşan bir telekomünikasyon ağının, topoloji, kapasite ve teknoloji kısıtları altında konfigüre edilmesini içerir. Telekomünikasyon ağlarının planlaması farklı disiplinlerdeki araştırmacılar tarafından çok çalışılmıştır ve literatürü oldukça karmaşıktır. Bu çalışmada, literatürdeki telekomünikasyon ağ tasarımı problemleri yöneylem araştırması bakış açısı ile sınıflandırılmış ve tasarım problemlerini optimizasyon problemleri ile eşlemek amacı ile tasarım problemleri taranmıştır. Literatür taraması, temel olarak karar problemlerini, matematiksel formülasyonlar ve ilgili ağ optimizasyonu problemleri için etkin çözüm yöntemlerini irdelemektedir. Çalışmamızda birbirlerine bağımlı olarak çalışan birden fazla ağ katmanından oluşan telekomünikasyon ağlarının tasarımını içeren çok katmanlı telekomünikasyon tasarım problemi işlenmiştir. Ayrıca, problemin alt problemleri belirtilmiş ve mevcut model ve formülasyonların yeterlilikleri araştırılmıştır. Tüm katmanları tek bir büyük ağ üzerinde modelleyen ve çeşitli pratik karar problemlerini birleştiren yeni bir matematiksel formülasyon önerilmiştir. Mevcut formülasyonlar kullanılarak hesaplama yapılamayan ikiden fazla katmanlı test problemlerinin önerilen matematik formülasyon ile genel amaçlı tamsayılı program çözücüler kullanılarak çözülebildiği görülmüştür. Ayrıca, genel amaçlı çözücülerin çözemediği daha büyük telekomünikasyon ağ problemlerini çözebilmek için Benders ayrıştırma metoduna dayalı ve probleme özel olarak uyarlanmış çözüm algoritmaları geliştirilmiştir. Bu algoritmaların davranışlarını değerlendirmek amacı ile literatürde mevcut test problemleri birleştirilerek kapsamlı hesaplama deneyleri yapılmış ve başarılı sonuçlar sunulmuştur.

Anahtar Kelimeler: Ağ tasarımı, sınıflandırma, telekomünikasyon, çok katmanlı telekomünikasyon ağ tasarımı

To Ekim

ACKNOWLEGMENTS

The time I have spent on my Ph.D. studies is most experiencing and enriching part of my professional life. This is an experience that I have learnt how the excitement of learning and contentment as a consequence of having the results of hard work wipe out the difficulty of research process that is even doubled by working for industry while doing the research. I am grateful to all people that supported me by any means during this time.

In the first place, I would like to express my gratitude to Prof. Dr. Ömer Kırca and Assoc. Prof. Dr. Haldun Süral for their precious supervision, advice and guidance throughout the study. I am deeply indebted to them for their patience, encouragement, and support.

I would like to thank to the thesis progress committee members Assoc. Prof. Dr. Oya Karaşan and Assist. Prof. Dr. Cem İyigün for their valuable comments and suggestions about the thesis study throughout the research. I also thank to Assoc. Prof. Dr. Sinan Gürel and Assoc. Prof. Dr. Cüneyt Bazlamaçcı for accepting to participate in my thesis committee and for their constructive suggestions about the thesis.

I am grateful to my manager Dr. Hüseyin Yavuz, who encouraged me to start to Ph.D. at the first place, for his constant help, support and motivation at all phases of my Ph.D. studies.

I would like to express my special thanks to my manager İsmet Atalar and my colleague Aybeniz Yiğit for their motivation and support during the thesis. I also thank to all my colleagues for their sympathy and support.

I would like to thank the company I work for, ASELSAN Inc., for organizationally encouraging and supporting my Ph.D. studies.

I would like to acknowledge that my Ph.D. thesis was supported by The Scientific and Technological Research Council of Turkey (TÜBİTAK), with National Scholarship Programme for Ph.D. Students.

I would like to express my gratitude to my cousin and my colleague Burcu Erol for diligently answered my questions about telecommunication networks and her motivation.

I would like to mention my eternal indebtedness to my parents Ümit Yüksel and Yusuf Yüksel, and my sister Sezin for their love, support and especially their persistent confidence in me. My grandparents deserve special mention for their support and love.

Last, but not the least, I would like to express my deepest gratitude for my husband Ekim, for his support and patience, and most importantly, for being there whenever I needed.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

CHAPTER 1

INTRODUCTION

Telecommunication technologies constitute a significant part in our daily lives and business regarding the usage of Internet and smart phones. According to Cisco Visual Network Index [1], global internet traffic in 2017 is forecasted to be 12 times as big as it was in 2007, and the number of online devices will be three times the world population in 2017. Therefore, huge investments would continue for the telecommunication infrastructure, mainly for the networks comprised of hardware and links that enable transmission of signals to meet this increasing demand. Since efficient design and expansion of these networks is an important concern, the telecommunication literature points out the telecommunication network design problem (TNDP) that is usually solved for effective and efficient design of telecommunication networks.

Telecommunication network design is to configure a network of major hardware and their links in order to satisfy traffic demands and flows subject to a set of constraints arising from topology, capacity, and technology. The TNDP is solved for either strategic decisions such as determining location of switching centers during installing a new telecommunication network or operational decisions such as determining how to route traffic demand through the network. Strategic decisions involve installation and operating costs while operational decisions are more technology specific in the sense that some performance measures such as reliability and network congestion are improved. In this thesis, we focus on strategic TNDPs.

The TNDP is studied by several disciplines such as electric-electronic engineering, computer science, and applied mathematics as well as operations research. Together with rapid evolution of telecommunication technology, this causes the TNDP terminology and content to be ambiguous, especially from the point of view of operations researchers. The literature is intricate and existing surveys are either specific to a distinct TNDP, telecommunication technology or old-dated not including the new problems emerging from new technological developments. We survey the TNDP literature focusing on the studies about strategic decisions from the point of view of operations research (OR). We identify attributes and classify TNDPs according to these attribute. We do not claim that the survey and classification is comprehensive as the TNDP literature is quite wide. However, the survey is comprehensive enough to highlight the essence of telecommunication network design problems and their connection to the network optimization problems. In this sense, we update the present surveys of classical network design problems and provide a guide from the OR perspective for linking telecommunication network design problems to network optimization problems. It also includes the new problem types to identify challenges and future research areas of telecommunication network design problems. The survey is presented in Chapter 2.

Network optimization problems are effective tools for modeling and solving TNDPs. In modeling and solving TNDPs, it is essential to know the decisions addressed by network optimization problems in order to benefit from the right network optimization problem. We provide an extensive survey on the network optimization problems that are used to solve TNDPs. It includes variants of the network optimization problems, connections between different types of network optimization problems, the main formulation structures, the solution methods and the capabilities of these solution methods. Network notation is unified for enabling comparison of inputs/outputs and decisions that can be made using a particular network optimization problem. In addition, the survey includes the recent studies about the network optimization problems, which are not included in the existing surveys. The survey is presented in Chapter 3.

The literature surveys on TNDPs and relevant network optimization problems reveal that TNDPs comprise of several subproblems. These subproblems can either be addressed by distinct network optimization problems or they can be modeled jointly. When the former is used, a sequential design process involving solution of subproblems sequentially is used. The latter needs an integrated design process which is computationally more expensive. The advances in computer technologies that increase computation power led to moving from sequential design, which usually results in suboptimality, to integrated design. Modeling subproblems jointly and solving them in an integrated way is practically more relevant as subproblems have mutual dependency.

Telecommunication networks can have different facilities according to the characteristics of the regions that they serve. Such kind of telecommunication networks are called multi-level networks and different regions serve as levels to the telecommunication network model. In addition, these networks may have several technologies along with having several layers and hence constitute multi-level and multi-technology telecommunication networks and called multi-layer networks [2]. Layers are abstraction of telecommunication networks on the same level having same technology. Therefore, in practice, telecommunication networks comprise of several network layers that are built on top of each other and work interdependently. Each network has its own technology and protocol, and they serve their own purposes. Some layers may even belong to different parties. Designing each layer through a sequential design process, which may result in suboptimal network designs, is the approach mainly used in the literature. The multilayer network design problem (MLNDP) that involves designing the network layers in an integrated way is a new problem in the telecommunication network design literature. In this thesis we propose a novel mathematical formulation that models all layers using a single-mega network and incorporates various practical decision problems. Our computational experiments show that the problem instances with more than two layers, which are not computationally tractable with the existing formulations, can be solved using the NFF by general-purpose integer programming solvers. The proposed mathematical formulation of the MLNDP and its network representation is presented in Chapter 4 with the result of the computational experiments.

In this study, multilayer telecommunication networks are generally addressed by the proposed model in Chapter 4. However, the examples and computational experiments focus on optical networks, e.g. SDH-over-WDM.

The MLNDP cannot be solved with general-purpose integer programming solvers for large networks. We develop tailored solution algorithms based on Benders decomposition to solve larger telecommunication network problem instances. We work with the original Benders decomposition, Benders decomposition within a branch and cut framework, and Ɛ-optimal Benders decomposition framework. In addition, we use several add-ons to algorithms to improve algorithms' performances. We perform computational experiments to observe the behavior of the algorithm to determine their weaknesses and strengths. According to the results of these computational tests, we improve the most promising algorithm. A Bender's like constraint generation method is used by Lardeux et al. [3] and Knippel and Lardeux [4] to solve the MLNDP. Fortz and Poss [5] use the same constraint generation method within a branch and cut algorithm framework. These three studies use a compact formulation called capacity formulation to model the MLNDP. Koster et al. [6] and Orlowski [7] use metric inequalities within a branch and cut and price framework to solve the MLNDP that is modeled by using flow formulation. Our algorithm differs from these implementations since we use an Ɛ-optimal Benders decomposition algorithm framework due to Geoffrion and Graves [8] together with repair and improvement heuristics. In addition, it is a tailored algorithm for our novel mathematical formulation that models the MLNDP using a single network graph representation instead of a multi-network graph representation in the literature. Consolidating the available test problem instances in the literature, we perform extensive computational experiments on these instances to assess the behavior of the algorithms and present favorable results. The algorithms and results for the computational experiments are given in Chapter 5.

The results are summarized and future research directions are listed in Chapter 6.

CHAPTER 2

NETWORK DESIGN PROBLEMS IN TELECOMMUNICATION

The telecommunication network design problem (TNDP) involves finding a suitable configuration of telecommunication network elements, which are hardware set on the nodes and links connecting these nodes so as to satisfy demand traffic, originating from nodes and flows through links. There are two types of objectives of the TNDP:

- To minimize installation and/or operating costs if the problem addresses strategic decisions such as location of switching centers when installing a new network.
- To minimize/maximize a performance measure if the problem addresses operational decisions such as how to route the traffic demand through the network, i.e. reducing network congestion.

The telecommunication networks are established and maintained under several constraints related to topology, hardware and link capacity, and hardware and link types that are mainly based on technology and quality standards. The main decisions are to determine locations, to select hardware and links, to allocate capacities on links, to provide survivability, and to route traffic flows. The problems focused on such decisions are referred as “classical problems” in network design. As telecommunication technology changes, some new problems have emerged like multilayer network design problems including virtual topology selection and wavelength assignment problems. If a telecommunication network is designed from scratch, it is called network deployment problem. When the network is to be expanded it is a capacity expansion problem. When the network is redesigned, it is called network update or redesign problem.

Telecommunication technologies constitute a significant part in our daily lives and business regarding usage of internet and smart phones. Huge investments expected to continue for telecommunication infrastructure, mainly for the design and expansion of the networks that enable transmission of signals to meet increasing demand, e.g. global internet traffic is estimated to increase twelve times between 2007-2017 by Cisco Visual Network Index [1]. The TNDP aims effective and efficient design of telecommunication networks. It is studied in electric-electronic engineering, computer science, applied mathematics and operations research. Terminology and the way problems are structured are not the same for these disciplines, which results in difficulties to follow.

In this paper, we survey the TNDP from the point of view of operations research (OR). We do not claim that the paper presents a complete survey of the telecommunication network design literature. However, it is comprehensive enough for OR. We classified the telecommunication network design problems, match with network optimization problems and present current issues and future challenges of telecommunication network design problems.

There are several surveys about the TNDP. Most surveys are related to a specific type of the TNDP. Gavish reviews formulations and algorithms for centralized computer networks [9] and local access networks [10]. Balakrishnan et al. [11] present a brief survey about local access network design problems. Local access networks are also reviewed in Carpenter and Luss [12]. Alevras et al. [13] review network dimensioning together with their connections to survivability and routing decisions. Telecommunication network grooming problem is reviewed in Barr et al. [14] and Zhu and Mukherje [15]. Routing and wavelength assignment problems are reviewed in Zang et al. [16]. Algorithms for solving this problem are reviewed in Choi et al. [17]. Dutta and Rouskas [18] review virtual topology design algorithms. Klincewicz [19] reviews the multilevel communication problems. Mehdi [20] reviews protection and restoration mechanisms for survivable telecommunication network design problems. Most recent review on multilayer network design problem is due to Orlowski's unpublished study [7]. There are a few reviews which do not focus on just one type of the TNDP, but multiple problem types like [21–24]. Surveys up to now are mostly dedicated to types of the TNDP. Gavish [10] and Minoux [21] are the firsts studies that relate the TNDP to the network optimization problems in addition to presenting basic formulations and solution methods for these problems. However, they are done more than twenty years ago and they do not include any of so-called new telecommunication network design problems. Our contributions with this survey are:

- updating the present surveys of classical network design problems,
- providing a classification about telecommunication network design problems including new problem types to identify challenges and future research areas of the TNDP.

2.1 General Telecommunication Network Structure

Telecommunication network structure has evolved since the end of 18th century. The first telephone networks appeared in USA after 1879. There are three major technologic developments that affected the telecommunication networks and make them evolve to the current status since 1960s. The first one is usage of digital signal transmission instead of analog signal transmission, which was started with an experimental phase in 1960s. The second one is wireless technology that started to be effective in 1980s. The last one is development of optical transmission components in 1990s. The brief history of telecommunication networks is presented in [25].

A generalized network structure is based on the relevant OR literature and fundamental telecommunication network structures. General telecommunication network structure involves multiple levels that are connected to each other in a hierarchical manner. The first level of hierarchy is called the backbone network and it serves urban space. Urban space is partitioned into local areas. Local areas are served by switching centers and communication among local areas is performed by backbone network. Switching centers are nodes of backbone network and links of backbone network are of high capacity with high transmission rates (backbone network nodes and switching centers are used interchangeably throughout the document). Switching centers connect local access areas to backbone network and they are also root nodes for local access networks. Local access areas are partitioned into service sections. The network between service sections that connect them to switching center is called local access network or primary network. Each service section is divided into terminal sections, where end users are connected to the network. Terminal sections are connected to service sections via secondary network and end users in terminal sections are connected via tertiary network. Thus, telecommunication networks are treated as multilevel networks. A general multi-level network structure is presented in Figure 1.

Another dimension of complexity in telecommunication networks is using multiple technologies in a single network. Multi-technological structure adds logical (virtual) network layers to telecommunication networks [2]. Practically, telecommunication networks are multilevel and multi-technology networks, and they are called multilayer networks. Detailed information about the general structure of telecommunication networks is presented in Appendix A.

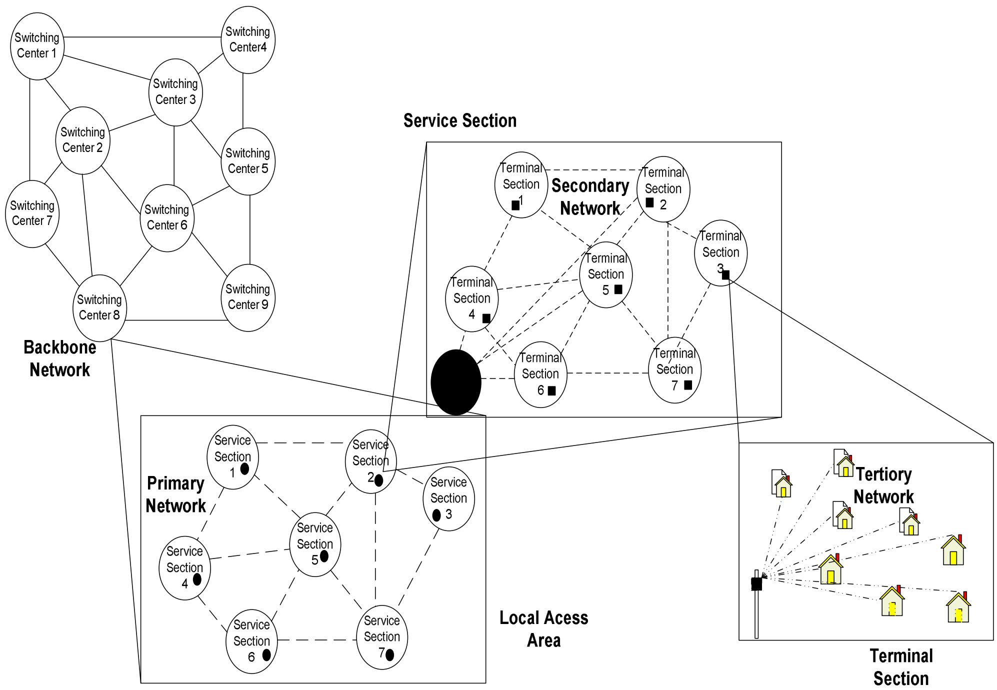


Figure 1. General Multi-Level Network Structure

2.2 Attributes of the TNDP

Complexity of the TNDP depends on the type of the suitable network optimization problem that is suitable for modeling and solving the TNDP. For example, if the TNDP can be formulated as the minimum spanning tree problem or the Steiner tree problem, the first model is easier to solve than the second model as the first problem is polynomially solvable while the second one is NP-hard. There are some features that assist to identify which network optimization problem is suitable for modeling and solving a particular TNDP. We describe the following attributes to classify the TNDP according to their complexity.

- *Capacity (CP): Capacitated (C) vs. uncapacitated (U)*
  Transmission links and hardware used at the nodes have limited capacity in real life. However, capacity limits can be large enough to ignore capacity constraints for some cases; hence utilization costs for links and hardware are used instead of capacity constraints for uncapacitated problems.
- *Network topology (tree topology vs. ring topology vs. mesh topology)*
  An imposed topology of telecommunication network is important as the topology affects complexity of the design problem. Thus, there is a trade-off between the complexity and advantages of network topology.
- *Flow Pattern (FP): Single commodity (S) vs. multicommodity(M)*
  Flow pattern of design problems is related to the origin and destination of demands. If there are different origin and destination points for traffic demands, multicommodity flow type is used.
- *Period (P): Single period (S) vs. multi period (M):* If it is needed to make topology design of telecommunication network within a time interval instead of designing the network for a single representative period, multi-period models are used. Multi-period models include time as a dimension for decision variables.
- *Facility Type (FT): Single facility (S) vs. multifacility(M)*
  Capacity of transmission links to be installed can be multiples of a single base unit while in some cases these capacities may be multiples of more than a single base unit constituting multifacility models
- *Single level vs. multilevel:* General telecommunication network structure is a hierarchical network structure consisting of levels, i.e. backbone and local access networks. These levels can be designed either in a sequential manner such that one level in the hierarchy is designed and the resulting solution is given as an input to the next level's design or in an integrated manner consisting of the multilevel telecommunication network design problem.
- *Single layer vs. multilayer:* Practical telecommunication networks consist of several layers. However, until recently, telecommunication network planning problems are solved and modeled for single layer networks since it is more complex to model multi-layer networks. The multi-layer telecommunication network topology design problem involves integrated design of different layers of the network i.e. basically to design physical layer and logical layers jointly.

2.3 TNDP Types

The TNDP involves several decisions to be made simultaneously. Hence the telecommunication network planning is a complex problem and in the literature the network design process is split into design phases. The problem is decomposed into subproblems each of which is related to a decision and a subproblem is solved in each design phase. Solutions of each design phase are an input to the next phase [10]. These subproblems are most of the time NP-hard [26]. Although decomposition principle makes problems solvable, solutions obtained after a sequential process are suboptimal. As computational capability of computers has been increasing, studies in the literature tend to integrate some subproblems of the telecommunication network design problem. In this section, subproblems of the telecommunication network topology design problem are reviewed.

The TNDP can be grouped into five types as backbone network design, local access network design, multilevel network design, and multi-layer network design and network expansion.

2.3.1 Local Access Network Design

Local access network constitutes the part of the general network where end users connect to the network. Local access networks constitute the most important part of the total cost of telecommunication networks.

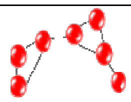
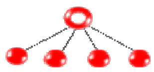
Topological design of local access networks involves a number of decisions. Decisions are related to the subproblems listed below:

- The concentrator location problem decides on the number of concentrators needed and where they should be placed. The facility location problem is a core problem for solving concentrator location problem [27].
- The terminal assignment problem is about how terminals should be assigned to concentrators. The minimum cost network flow problem and the concentrator location problem are used to solve the terminal assignment problem.
- The terminal layout problem decides on how terminals that are associated with a particular concentrator should be linked together. The minimum spanning tree problem and its variants, the Steiner tree problem, the minimum cost network flow problem and the concentrator location problem are used to solve the subproblem.
- The Telpak problem seeks what line capacities should be used on links between concentrators, and between concentrators and central site. The multilevel capacitated minimum spanning tree problem and the minimum cost network flow problem are used to solve the Telpak problem in local access network design.

Local access networks have several topologies as ring, star, and multidrop trees. The most reliable topology is star although its cost is highest. The cheapest solutions are attained when multidrop tree topologies are used, though they are not reliable. Ring topologies are more reliable than multidrop topology and cheaper than star topology.

Local access network design studies are presented in Table 1. The local access network design problem is a well-studied problem. Topological design of local access networks is investigated either as a network design problem from scratch or a network redesign problem, i.e. network expansion or network update problem. Main characteristics of the problem are given in [10–12]. Recent studies on the problem are mostly related with network expansion which is presented in detail in Section 3.4 and designing reliable networks [28], [29]. In addition, local access network redesign and local access network update problems are studied. The local access network update problem that involves technological specs of telecommunication infrastructure is presented by Chamberland [30] wherein a mixed integer programming model of the problem is given and a heuristic algorithm is proposed to solve the problem. The network redesign problem involves adding new capacity to the network while rearranging the existing capacity. Frantzeskakis and Luss present mixed integer model of local access network redesign and propose a heuristic algorithm to solve the design problem [31]. The local access network design problem is reviewed by Carpenter and Luss [12]. In addition, optimal methods for uncapacitated local access network design problem is surveyed and compared by Randazzo and Luna [32].

Table 1. Local Access Network Design Problem Papers[1]

CP	FP	P	FT	Topology	Source	Solution Approach	Computational Testing
C	S	S	M	---[2]	[33]	Exact solution by B&B	Algorithmic performance 64 instances[3] *(n,e)=(40,81)*
					[30]	Heuristic solution by TS	Algorithmic performance 120 instances, $n=420$
					[34]	Heuristic solution by a three-phase algorithm	Algorithmic performance 15 instances, $n=86745$
					[29]	Heuristic solution by SA	Algorithmic performance 27 instances[4], $n=90$
			S	---	[28]	Solution by constant factor approximation algorithm	---
					[35]	Heuristic solution by CH and IH	---
					[36]	Heuristic method of obtaining solutions	---
					[37]	Heuristic Solution by GA	Numerical example 1 instance, $n=16$
					[38]	Heuristic solution by CH (Esau-Williams heuristic)	----
					[9]	Heuristic solution by LR for degree constrained minimum spanning tree problem	Algorithmic performance 170 instance, $n=200$
						Exact solution by BD for capacity constrained minimum spanning tree problem[5]	Algorithmic performance $n=12$[6]

---

[1] B&B: Branch and Bound, TS: Tabu Search, SA: Simulated Annealing, CH: Construction Heuristic, IH: Improvement Heuristic, GA: Genetic Algorithm, LR: Lagrangian Relaxation, BD: Benders Decomposition, PSA: Parallel Savings Algorithm, SOGA: Second Order Greedy Algorithm, CG: Column Generation, BC: Branch and Cut, *n*: Number of nodes, *e*: Number of edges

[2] No imposed topology – the best topology is selected by the solution

[3] 12 randomly generated test problems, 5 instances for each problem type and instances ARPA, OCT, USA, and RING from [51], [270].

[4] Three networks, nine combinations of SA parameters, total of 27 instances

[5] These are network optimization problems to solve subproblems of local access network design subproblems.

Table 1 (Cont'd)

CP	FP	P	FT	Topology	Source	Solution Approach	Computational Testing
					[39]	Heuristic solution by subgradient optimization and augmented Lagrangian-based procedure	Numerical example 1 instance, *n=16*
					[40]	Heuristic solution by PSA	See [10]
					[41]	Exact solution by LR	Algorithmic performance 75 instance, *n=200*
					[42]	Heuristic solution by SOGA	Algorithmic performance *n=120*
					[10]	Heuristic solution by PSA for tree topology	Algorithmic performance *n*=400
					[43]	Exact solution by stabilized CG	Algorithmic performance, 74 instances, *n*=2965
					[44][7]	Heuristic solution by PSA	Algorithmic performance *n*=100
	M	S	M	---	[45]	Exact solution by BD within a BC framework	Algorithmic Performance 96 instances, *n*=67

In recent studies, the local access network design problem is solved jointly with the backbone network design problem which is presented in Section 5 in detail. Local access networks are centralized networks where there is a root node from which traffic flows to and from terminal nodes. Hence there is a single source node for the network. So that in most studies, local access networks are either modeled as trees or single commodity minimum cost networks. The other network optimization problems used are the minimum spanning tree problem, the Steiner tree problem and the single commodity minimum cost network flow problem.

2.3.2 Backbone Network Design

Backbone network constitutes the first level of hierarchy in general telecommunication network structure. Backbone network consists of switching centers as nodes and links between these switching centers where the links are high capacity links with high transmission rates.

Decisions related to the backbone network design problem are,

- choosing locations of backbone nodes,
- choosing the type of processor used in each backbone node, and
- backbone routing and capacity assignment problem on deciding which links will

[6] The results are disappointed even for 12 nodes.

[7] Includes selection for backbone nodes.

connect backbone network nodes and routing.

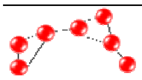
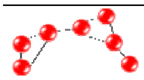
Links with high capacity and high transmission rates lead to sparse network topologies where only a few paths exist between nodes increasing possibility of service disruptions due to a single node or link failure. Thus, survivability is important for backbone networks. Level of survivability of the network is closely related to the network topology. In addition, the network topology affects complexity of subproblems solved for backbone network design.

Backbone network design studies are listed in Table 2.

Backbone networks can be of fully-interconnected, mesh or ring topology. In a fully-interconnected (fully-meshed) backbone network, each node is connected to every other node in the network by a link [19]. In such a network topology, transmission is fast since each node is connected to any other node with exactly one link. This topology is the most reliable topology, although it is most expensive one. Fully-meshed networks are used when main performance criteria are fast response and system reliability such as military networks [46]. If one or more links are not present in a fully interconnected network, network type becomes mesh network. In a mesh network, traffic between two backbone nodes may be routed via other backbone nodes [19]. Mesh topology is a reliable topology though it is expensive. Survivability is an important issue for optical networks and fully-interconnected topology is an expensive solution for maintaining survivability. Ring topology is for optical networks to satisfy survivability constraints. In a ring topology, each node has two paths to any other node, which means that if a failure of any node or any link occurs in one path, the traffic can be routed using other path. Ring topology is reviewed in [47].

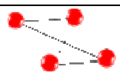
Early studies on the backbone network design problem focus on the capacity assignment and routing problem [48–50]. Network components are either assumed to be reliable [50] or 2-connectivity is used for reliability of the network [49]. Heuristic methods such as branch exchange (BXC), concave branch elimination (CBE) and cut saturation (CS) are discussed in these studies. In addition, Kershenbaum, Kermani, and Grover [49] propose a heuristic called MENTOR and Altinkemer and Yu [50] propose Lagrangian relaxation procedure to solve the problem. Amiri and Pirkul use Lagrangian relaxation to solve the routing and capacity assignment in backbone computer communication networks [51], [52]. In more recent studies, integer linear programming techniques are proposed to solve the problem [53]. However, as the problem size increases these techniques become impractical [54]. Hence, near optimal solutions are found using metaheuristics for design of backbone networks, such as simulated annealing [55], [56], tabu search [57], evolutionary algorithms [58–60] and ant colony optimization [61] or other heuristics [54], [62].

Table 2. Backbone Network Design Papers[8]

CP	FP	P	FT	Topology	Source	Solution Approach	Computational Testing
C	M	M	M	---[9]	[52]	Exact solution by LR	Algorithmic performance 88 instances, *(n,e,k)=(32, 60, 992)*
		S	M	---	[51]	Exact solution by LR	Algorithmic performance 79 instances, *(n,e,k)=(32, 60, 992)*
					[63]	Exact solution by CP	Algorithmic performance 126 instances, *(n,e,k)=(15, 34, 21)*
					[64]	Exact solution by BC[10]	Algorithmic performance 36 instances, *(n,e)=(27,51)*
			S	---	[50]	Exact solution by LR	Algorithmic performance 80 instances, *(n,e,k)=(30, 100,400)*
					[65]	Heuristic solution by a two-phase approach[11]	Algorithmic performance 20 instances, *(n,e,k)=(26, 30,650)*
					[66]	Heuristic solution by a two-phase approach[12] and exact solution method by LR	Algorithmic performance (Number of instances and their characteristics are not reported explicitly)
					[53][13]	Exact solution by BC	Algorithmic performance 12 instances, *(n,e)=(15, 22)*
					[49]	Heuristic solution by MENTOR (a local search heuristic)	Numerical example 1 instance, *(n,e,k)=(6,15,15)*
					[67]	Heuristic solution by GA	Algorithmic performance[14]
					[48]	Heuristic solution by BXC, CBE and CS	Algorithmic performance 26-node ARPANET topology

[8] LR: Lagrangian Relaxation, CP: Cutting Plane, BC: Branch and Cut, GA: Genetic Algorithm, BXC: Branch Exchange, CBE: Concave Branch Elimination, CS: Cut Saturation, n: Number of Nodes, e: Number of Edges, k: Number of Commodities
[9] No imposed topology – the best topology is selected by the solution
[10] Robust optimization
[11] A sequential approach to solve survivable network design problem that optimizes working capacity then optimizes spare capacity.
[12] A sequential approach to solve survivable network design problem that optimizes working capacity then optimizes spare capacity.
[13] Study includes local access network design, only backbone network design part is reported.
[14] Only one network topology whose only figure is provided is used with different telecommunication network components and GA parameters.

Table 2 (Cont'd)

CP	FP	P	FT	Topology	Source	Solution Approach	Computational Testing
	S	S	M	---	[13]	Exact solution by CP	Algorithmic performance 32 instances, (*n,e,k*)=*(17, 62, 106)*
U	M	S	S		[62]	Heuristic solution by two greedy heuristics	Algorithmic performance, 35 instances, *(n, e)=(50, 1225)*

Survivability is the main concern of backbone network when solving the capacity assignment and routing problem in recent studies ([54], [65], [66], [68], [69] and a significant portion of recent studies is specific to the technology used in the backbone network. Since backbone networks usually use optical fibers, these studies are mostly related to the Wavelength Division Multiplexing (WDM) network and Internet Protocol (IP) network design [54], [67]. Some of these studies solve the topology network design of backbone network problem jointly with classical problems such as survivability and routing or with new problems such as virtual topology design, and routing and wavelength assignment [69], [70].

Backbone networks are distributed networks where traffic flows from several source nodes to several sink nodes. Hence, the minimum cost multicommodity commodity network flow problem is generally used to model the backbone network design problem. If there is a predetermined topology, then topological constraints are taken into account. In addition, survivability is maintained by adding several side constraints to the model.

2.3.3 Multi-level TNDP

The multi-level TNDP involves installing links with different transmission rates between nodes depending on their demands and capacities where links with higher transmission rates are more expensive. The backbone network-local access network constitutes a two-level network while some local access networks consisting of more than one facility type, i.e. links with different transmission rates, constitute multi-level networks.
General telecommunication network structure is comprised of backbone and local access networks. As computation power increases, approaches tend to solve backbone and local access network design problems jointly instead of using a sequential approach. As it is stated, backbone networks are sparse networks with high-speed, high-capacity links that have expensive installation costs while local access networks are less sparse networks with cheaper links having slower transmission rates. Thus, joint topology design of backbone network and local access network involves network design with more than one facility i.e. links having different transmission rates and capacities. In order to use links with different transmission rates in a single network, installation of some hardware that make the transmission rate conversion i.e. concentrators or multiplexers, are required. Then, the problem involves the trade-off between installing expensive links with high-capacity and high transmission rate and installing concentrators on nodes while using slower links.

The two-level telecommunication network design problem, which is also called hierarchical design, refers to joint design of backbone network and local access network design problem

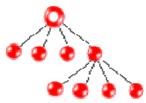
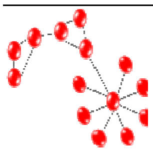
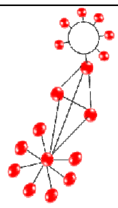
[71–73]. The two-level telecommunication network design problem involves several decisions and it can be decomposed into several subproblems. The subproblems related with the two-level TNDP can be listed as [71]:

- Hub location (or selection)
- Clustering of nodes
- Interconnection of nodes in the backbone network and cluster networks
- Routing in backbone network and cluster networks

The problems listed above are closely related to the subproblems of backbone network design problem and the local access network design problem.

Studies for the multi-level TNDP are listed in Table 3.

Table 3. Multilevel TNDP Papers[15]

CP	FP	P	FT	Topology	Source	Solution Approach	Computational Testing
C	M	S	M		[74]	Exact solution by BC	Algorithmic Performance 40 instances, *(n,e,le)*= (1000, 3500,4)
					[75]	A MIP model is proposed	---
				---[16]	[76]	Exact solution by BC	Algorithmic Performance 36 instances, *(n,e,le)*= (1000, 25000,2)
					[77]	Exact solution by BC	Algorithmic Performance 15 instances, *(n,e,le)*= (2500, 62500,2)
			S	---	[78]	Heuristic solution by iterative problem decomposition, clustering and local optimization	Algorithmic Performance 40 instances, (*n, le)*= (1000, 5)
					[79]	Heuristic solution (five methods ranging from exhaustive search to local search)	Algorithmic Performance >4000 instances, (n,le)=(100.000, 2)
	S	S	M		[46]	Heuristic solution by greedy algorithm and TS	Algorithmic Performance 200 instances, *(n,le)*=(400, 2)

[15] BC: Branch and Cut,TS: Tabu Search, BCP: Branch and Cut and Price, LR: Lagrangian Relaxation, DP: Dynamic Programming, *n*: Number of Nodes, *e*: Number of Edges, *le*: Number of Levels

[16] No imposed topology – the best topology is selected by the solution

Table 3 (Cont'd)

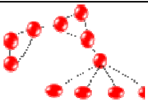
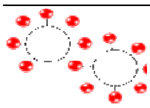
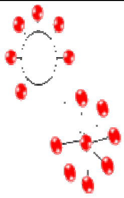
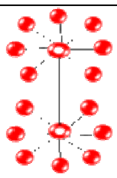
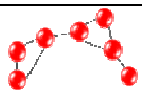
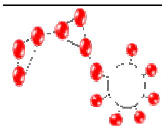
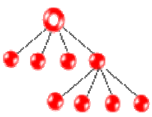
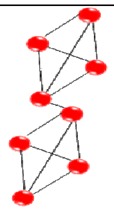
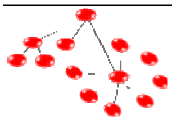
CP	FP	P	FT	Topology	Source	Solution Approach	Computational Testing
			S		[80]	Heuristic solution by decomposing the problem and solve subproblems by different heuristics	Case Studies, 2 cases, (n_1,le_1)=(42,2) and (n_2,le_2)=(84,2)
					[81]	Heuristic solution	Algorithmic Performance 110 instances, *(n,le)*=(120, 3)
					[82]	Exact solution by BC	Algorithmic Performance 542 instances, *(n,le)*=(300, 2)
					[83]	Heuristic solution (decomposes problem and solves the subproblems with heuristics iteratively)	Algorithmic Performance 13 instances, *(n,le)*=(4500, 2)
					[84]	Exact solution with Cplex 9.0	Case Studies 3 cases *(n,le)=(337, 3)*
U	M	S	S		[72]	Exact solution with BCP	Algorithmic Performance 17 instances, *(n,le)=(300,2)*
	S	S	M	---	[85]	Heuristic solution by a 3-phase algorithm	Algorithmic Performance 46 instances, *(n,e, le)=(189,297,2)*
					[86]	Exact solution by branch and price (BP)	Algorithmic Performance 24 instances, *(n,e, le)=(100,125,2)*
					[87]	Exact solution by a dual-based algorithm	Algorithmic Performance 87 instances, *(n,e, le)=(500,5000,2)*
					[88]	Heuristic solution by solving subproblems using the Steiner tree heuristics	---
					[89]	Exact solution by LR	Algorithmic Performance 60 instances, *(n,e, le)=(100,1237,2)*

Table 3 (Cont'd)

CP	FP	P	FT	Topology	Source	Solution Approach	Computational Testing
			S	---	[73]	Exact solution by DP	Algorithmic Performance 4 instances, *(n, le)=(1240,3)*
					[90]	Exact solution by BP	Algorithmic Performance 24 instances, *(n,e, le)=(25,300,,2)*
					[91]	Heuristic solution by decomposing the problems into two subproblems	Algorithmic Performance 48 instances, *(n, le)=(50,2)*

Earlier studies for the two-level TNDP are reviewed by Klincewicz [19]. A wherein most studies in the literature use special topologies such as star-star, ring-star, etc., to exploit benefits of special structures such as limited sets of costs and limited constraints. Recent studies on the problem also use special topologies such as star-star [83], ring-star [82], ring-ring [81], fully interconnected-fully interconnected [90] and mesh-mesh [71].

A basic model and an extended model for the two-level TNDP based on the fixed charge network design problem are presented in [71]. Thomadsen and Stinsen [72] present a MIP formulation for the generalized fixed charge network design problem which solves the hub selection, the backbone network design and the backbone network routing problems given a set of clusters. They propose a branch-cut-and-price algorithm to solve the problem [72]. In addition, Koch and Wessaly use the capacitated Steiner arborescence problem to formulate the two-level TNDP [84]. Rosenberg proposes a dynamic programming algorithm that solve the hierarchical topological network design problem with single node survivability [73]. Rosenberg reports that this algorithm is used for high level planning of two high speed packet switched networks of approximately 30 and 60 nodes by AT&T.

The two level network design problem, multilevel network design problem, and the multi-tier tree problem can be used [74], [87–89], [92]) to solve the two-level and multi-level TNDP. In addition, [93] and Mateus, Cruz and Luna [85] propose the multi-level network optimization problem to solve the multi-level TNDP. This formulation is very close to the multicommodity network flow with gains formulation that is also used by Balakrishnan et al. [11] with a layered graph representation to model the local access network consisting of multiple facilities. However, the former formulation also accounts for location decisions in addition to dimensioning decisions and the conversion ratio between consecutive layers in the representation is taken as 1:1 unlike the latter formulation.

In some studies, the decomposition principle is used to solve the hierarchical network design problem for overall backbone and local access network design optimization. The problem is decomposed into subproblems and the subproblems are solved in a sequential or iterative manner instead of solving the multi-level network design problem [78], [85], [91].

2.3.4 Network Expansion

The TNDP is not limited to deployment of new networks. The network expansion problems are more important than deployment problems from the practical point of view as networks need to be expanded to satisfy the increasing demands over time and network providers usually keep existing infrastructure while expanding networks. In addition, decrease in unit cost of hardware such as switches and transmission facilities as technology develops drives network expansions [94].

Network expansion can be (i) network capacity expansion involving only increasing capacity of existing links and hardware located at existing nodes, and (ii) network topology expansion involving installation of new links and hardware located at the existing nodes.

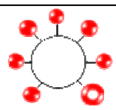
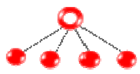
Telecommunication network expansion problems are presented in Table 4.

There are two alternative methods for network expansion. The first method is to expand capacity of transmission links and the other is to use concentrator devices at nodes. This is the trade-off of cable expansion and processor installation in telecommunication network capacity expansion [11], [95].

The network expansion problem exhibits economies-of-scale since cost of transmission and switching equipment decreases as added capacity increases [96], [97]. This brings the trade-off between adding large capacities before it is needed in order to exploit the economies-of-scale property and cost of installing capacity before it is needed [97]. In addition, cost of these equipment decreases as technology develops bringing another trade-off between network congestion costs and losing revenue producing opportunities that may arise with change in technology [94]. Hence, time is an important factor for network expansion problem and the problem is studied for multi-period cases [94], [98–100] in addition to single period cases [11], [95], [101–108]. Although the multi period network expansion problems are more realistic, the single period problems are studied because the multi period problems are hard to solve and the single period problem may give insights about the multi period case and solution methods developed to solve the single period model may be used as building blocks to solve multi period models [11].

The topology and capacity expansion problem is solved to determine how a given capacitated network is expanded by installing more capacity for the network to meet the traffic demand between origin and destination nodes such that the total of capacity installation and routing cost is minimized. The problem is a well studied problem. For the early work on the problem the book due to Freidenfelds [96], the review due to Luss [97] and papers due to Zadeh [109], and Christofides and Brooker [110] can be viewed. For a recent review on capacity expansion problem and survivable capacity expansion problem, unpublished study of Sivaraman [111] is referred.

Table 4. Network Expansion Papers[17]

CP	FP	P	FT	Topology	Source	Solution Approach	Computational Testing
C	M	M	S	---[18]	[94]	Heuristic solution by Lagrangian-based heuristics	Algorithmic Performance 3 instances
	S	M	M	---	[100]	Heuristic solution by DP and shortest path algorithm	Algorithmic Performance 5 instances, $p=10$
					[98]	Heuristic solution by decomposition and iterative procedure	Numerical Example 1 example, $(n,p)=(18,10)$
					[99]	Heuristic solution by decomposition	Numerical Example 1 example[19]
					[106]	Heuristic solution	Algorithmic Performance 972 instances, $(n,p)=(110,4)$
					[107]	Heuristic solution by a local search algorithm integrated with GA	Algorithmic Performance 324 instances, $(n,p)=(110,4)$
		S	M		[112]	Heuristic solution by TS	Algorithmic Performance 100 instances, $n=250$
			S		[102]	Exact solution by CPLEX 10.2	Algorithmic Performance 270 instances, $n=500$
					[101]	Exact solution by an enhanced DP using valid inequalities within LR framework	Algorithmic Performance 3 instances, $n=41$
					[95]	Exact solution by CPLEX 7.1	Algorithmic Performance 37 instances, $n=200$
					[104]	Exact solution by limited CG	Algorithmic Performance 25 instances, $n=200$
					[105]	Exact solution by DP	Algorithmic Performance 27 instances, $n=1000$

Chamberland and Sanso [112] point out that some of the studies on network expansion problem involve only some portion of the overall network expansion i.e.:

- capacity expansion [11], [98], [99], [101], [111] or topology expansion [112], [113],
- local access network expansion [11], [98], [99], [101] or backbone network

[17] DP: Dynamic Programming, GA: Genetic Algorithm, TS: Tabu Search, CG: Column Generation, n: Number of nodes, p: Number of periods

[18] No imposed topology – the best topology is selected by the solution

[19] Only results are reported, no detail is provided regarding the example instance

expansion [111], [112], [114], [115].

In addition, the joint topological and capacity expansion problems are solved for overall network expansion [94], [102].

The capacities to be installed on the links are not always of the equal bundles i.e. unit capacity can change, constituting multi-facility network expansion problem [98], [100], [116].

The local access network expansion problem has attracted considerable attention. The paper due to Balakrishnan et al. [11] is important for local access network expansion problem since the main aspects of the problem is given in detail and the assumptions done in this study are stated by a number of other papers such that [95], [102–105], [115]. Balakrishnan et al. [11] formulated the network expansion problem as the multi-commodity network flow with gains problem. In addition, Balakrishnan, Magnanti and Wong [101] show that the problem is NP-hard.

2.3.5 Multi-layer TNDP

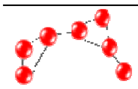
In practice, telecommunication networks involve more than one technology; hence telecommunication networks have multi-technology networks. A multi-level and multi-technology telecommunication networks are called multi-layer telecommunication networks. Hence, each layer involves a single technology and facility type [2]. Therefore, telecommunication networks comprise of many subnetworks in practice, which are organized in a manner that a subnetwork is built on top of another subnetwork and the physical components of the networks constituting the lowest network. Each subnetwork in this structure has its own technology and protocol in order to serve its own purpose [117].

Telecommunication services like internet and telephone are given by service providers and large companies use their private networks for their own telecommunication services. These services constitute traffic networks, which are also called application service networks. Generally, these service providers and companies lease physical telecommunication facility from other network providers; hence become customers of physical facility network providers. Physical networks provide transportation of traffic network and are called transport network [118]. Several service types like internet and telephone exist for traffic networks and several technology alternatives exist for transport networks, such that ATM, SDH, SONET and WDM. A telecommunication network may contain more than two transport networks having different technologies such that SDH/SONET over WDM networks. For detailed information about these technologies, the book due to Pioro and Medhi [118] and the review due to Plante and Sanso [119] are referred.

Multilayer TNDP studies are listed in Table 5.

Traffic and transport networks, which have a server-client relationship, constitute the upper and lower layers of a telecommunication network, respectively. In a multi-layered network, a layer serves as the client for its lower layer such that capacity needed for satisfying demand for traffic layer of a network is the demand for the transport layer and transport layer's capacity must satisfy this demand.

Table 5. Multilayer TNDP Papers[20]

CP	FP	P	FT	Topology	Source	Solution Approach	Computational Testing
C	M	S	M	---[21]	[117]	Sketch of an exact solution by BCP is proposed	---
					[3]	Exact solution by Bender's like constraint generation	Algorithmic Performance 12 instances, *(n,e1,e2, k la)*=(8,13,28,28,2)
					[4]	Exact solution by Bender's like constraint generation	Algorithmic Performance 13 instances, *(n,e1,e2, k la)*=(10,18,45,45,2)
					[120]	Exact solution by BC[22]	Algorithmic Performance 67 instances, *(n,e2, la)*=(60,102,1)
					[5]	Exact solution by BC using Bender's like constraint generation	Algorithmic Performance 41 instances, *(n,e1, la)*=(14, 22, 2)
					[121]	Heuristic solution by a 3-phase algorithm after decomposing the problem into subproblems	Numerical Example 2 instances, *(n,e1, e2, e3, la)*=(23,30, 30, 30, 3)
					[122]	Exact solution by BC	Algorithmic Performance 28 instances, *(n,e1,e2, la)*=(37,57,5096, 2)
					[7]	Exact solution by BCP	Algorithmic Performance 6 instances, *(n,e1,e2, la)*=(67,87,690, 2)[23]
					[6]	Exact solution by BC with problem specific preprocessing	Algorithmic Performance 6 instances, *(n,e1,e2, la)*=(17,26,674, 2)

In practice, a service provider may prefer to use one or more transport network providers. Likewise, a transport network provider may serve for different service providers with the

[20] BCP: Branch and Cut and Price, BC: Branch and Cut, *n*: Number of Nodes, *ei*: Number of Edges in Layer *i*, *k*: Number of Commodities, *la*: Number of Layers
[21] No imposed topology – the best topology is selected by the solution
[22] The transport network is assumed to be fixed in this study, hence the study focused on survivable routing of demands in traffic layer.
[23] Only useful upper bounds can be generated for this instance with test instance specific preprocessing

same transport network. Hence, the multi-layered structure of telecommunication networks provides the modularity needed for management of the networks.

The basic concepts of the multilayer telecommunication networks are the physical and logical links, demands and grooming, node hardware, cost, routing and survivability [7], [117].

The telecommunication networks comprised of several technologies which operate interdependently. The granularities of the data streams used by each technology are different from each other and a technology may use more than one level of granularity. The process of combining small granularity signals to a coarse granularity signal is called multiplexing and the opposite process is called demultiplexing. Since each layer use different technology and each technology has its own protocol, the data is encapsulated into another protocol each time it is transmitted to a different layer. Hence, the data is routed by "grooming paths" which is formed by multiplexing the data at the beginning node and demultiplexing it again at the end node of the path. The grooming paths cannot be accessed until the end of the path meaning that the data that has been multiplexed cannot be demultiplexed until the end of the grooming path. Therefore, a grooming path in a layer addresses a link in the upper layer, which is called a logical link [117]. In a two-layered network case, the lower layer is comprised of optical fibers or copper cables between nodes, while a link between two nodes in the upper layer is a path in the physical layer between these two nodes. Grooming layers are also called lightpaths if the underlying physical network is an optical network. Generalizing this to a multi-layer network case, we see that multilayer routing has a nested structure such that the uppermost layer's links are the paths in the neighboring lower layer whose links are paths in its lower neighbor layer and so on. Then the demands of the uppermost layer constitute an artificial layer which is on top of all layers [118].

The cost of multilayer networks is incurred from node hardware, logical links and physical links as switching and converting devices (cross-connects, wavelength converters, multiplexers/ demultiplexers), terminating devices (line cards, ports), and transmission equipment (fibers, radio links, leased lines, and optical amplifiers), respectively. Technology affects the exact cost structures. Further information about how the node hardware (switching and converting devices and terminating cards) works according to different technologies is presented in [7] and [117].

The notion of logical link brings the complexity of the multilayer networks, though this notion makes it possible to design multilayer networks sequentially from top to bottom by designing each layer as a single layer and defining each layer's demand as the capacity of its the upper neighbor layer. The sequential design procedure is used for multilayer network design until some studies propose the integrated multilayer network design methods recently. The sequential design is tractable and computationally easier than the integrated design but there are some drawbacks of the sequential design [7], [117]:

- Two logically link disjoint paths found by sequential design does not need to be physically disjoint, thus sequential design violates survivability conditions.
- The cost value found by sequential design may not be optimal.

- Coordination of routings in different layers in sequential design is important. If it cannot be done sufficiently, it may lead to unnecessary capacity to be installed for some physical links and some links to be overused resulting in delays or increase in failure probability.

Multilayer network design problem is a new problem type compared to other TNDP types and it has been studied for just over a decade since the study of Dahl et al. [123]. Orlowski and Wessaly [117] present an introduction about multilayer networks with technological examples and propose an integer programming model for multilayer network design problem. They included a sketch of an algorithmic scheme for solution of the model but they include neither a proper solution method nor any computational results. In addition, the most recent review for multilayer network design problem is presented in Orlowski's unpublished work [7]. Orlowski divided the studies into two groups as studies that solve routing and logical network design problem given physical network, and that make integrated multilayer network design, i.e. the physical network is designed together with logical network [7]. Pioro and Medhi present some basic formulations of the multilayer network design problem for different design options [118]. They present a review of the studies related to multilayer network design problem. Plante and Sanso [119] provide a typology for multi-technology multi-service broadband network synthesis, which serves as a review on technological considerations when designing a multilayer network.

Borne et al. [120] propose a node arc formulation and a path formulation for the survivable two layer network design problem with modular link capacities, continuous flow and bifurcated routing. The study focuses on survivable routing of traffic in traffic layer while they assume the transport layer is fixed, hence the handled problem does not need an integrated multilayer network design. It is mentioned in the paper that they developed a branch and cut, and a branch and cut and price algorithm to solve the problem. Lardeux, Knippel and Geffard. [3] and Knippel and Lardeux [4] use the same formulation. They use dual of arc-path formulation in order to model the multilayer network design problem with modular link capacities, bifurcated routing and step increasing cost function. They propose a Bender's decomposition like constraint generation procedure which is similar to solution procedure of Gabrel, Knippel and Minoux [124] and tested the proposed algorithm for up to 10 node-2 layer networks. Fortz and Poss [5] use branch and cut algorithm to improve the method used by Lardeux et al. [3].

Kubilinskas [125] proposes an iterative approach to solve a two-layer network design problem with 1+1 protection. Elastic demand case is also considered in this study. Mattia [122] uses metric inequalities and proposes heuristic algorithms to solve the compact formulations based on metric inequalities.

Orlowski et al. [126] propose branch-and-cut approaches for the multilayer network design problem. Orlowski [7] provides mixed integer programming models of a two layer with modular link capacities in both layers which assumes continuous flow and bifurcated routing for cases with and without restoration. In addition, he proposes a branch and cut algorithm using the results found in [126] to solve the models which gives 1% optimality gap for up to 17 nodes with restricted logical layer such that only two or three logical links are admissible between each pair of nodes in the physical layer and useful dual bounds for up to 67 networks with restricted logical layer and with test instance specific preprocessing. Orlowski claims that they solved the most difficult multilayer network design problem modeled in the literature.

Most studies in the literature are focused on strategic decisions related to the multilayer telecommunication network design and have an objective function of minimizing the installation costs. However, some of the studies focus on network performance and use objective functions such as minimization of delay, minimization of congestion, minimization of total lost traffic. For the latter type of problems, objective functions serve to take more operational decisions than strategic, and the related studies are more technology dependent. Most relevant of such studies belongs to Raghavan and Stanojevic [127]. Raghavan and Stanojevic [127] propose MIP formulations and a branch-and-price algorithm to minimize the total lost traffic in a WDM network.

Multilayer network design problem involves the following subproblems for optical telecommunication networks [128]:

- Physical topology design problem: Determining the nodes that telecommunication hardware is located, capacity and type of hardware, nodes which are to be connected by fiber optic cables, and capacity of the cables given traffic demand,
- Logical topology design problem: Determining number of lightpaths (logical links) to be established between node pairs and routing of traffic over the established lightpaths, given node hardware at each node of the network, capacity of lightpaths and traffic demand [128].
- Traffic grooming problem: When bandwidth of traffic requests is lower than capacity of lightpaths, low-granularity requests are bundled into high-granularity flows by using multiplexing. This situation is practically common in optical networks since variety of its services is high and not all these services have the same bandwidth. The problem of locating hardware for grooming and routing flow on lightpaths is known as traffic grooming problem [128], [129].
- Lightpath routing problem: Given physical topology, i.e. node pairs that are connected with fiber optic cables, and logical topology, i.e. number of lightpaths between the node pairs, determining routing of lightpaths on the logical topology [127], [128].
- Routing and wavelength assignment problem: If network does not involve any wavelength converters, than each lightpath in a fiber optic cable must be assigned to a distinct wavelength. Then, if there is no wavelength conversion facility at all nodes of the network, lightpath routing problem is solved together with wavelength assignment problem [128].

These subproblems were solved sequentially before the last decade because of computational intractability of the models with the available technology then. However, since 1999 [123], integrated solution of two or more subproblems is being studied. Physical topology design, logical topology design and lightpath routing problems are solved jointly in most studies [3–5], [7], [117], [120], [122], [125], [126]. Raghavan and Stanojevic [127], and Stanojevic solve the logical topology design problem and the lightpath routing problem jointly [128]. The traffic grooming problem and the lightpath routing problem are also solved jointly [129–132].

Routing and wavelength assignment problem is not valid under the assumption that all nodes can make wavelength conversion. For the networks that have no wavelength conversion ability, it must be guaranteed that a lightpath uses the same wavelength from its beginning node to the end node and each lightpath in a fiber use different wavelengths. Many WDM networks have sparse wavelength conversion ability, however, modeling sparse wavelength conversion increases the difficulty of the formulation [128]. In that case, wavelength assignment and wavelength converter location problem is usually solved after a solution is found to the multilayer network design problem [7].

Many of the technologies used in the telecommunication networks support heterogeneous granularity flow such that SDH/SONET networks has different granularity flows called virtual containers. In this case, only a single technology can be designed as multilayer network design. Traffic grooming is used to convert low granularity flows into high granularity flows providing more efficient usage of bandwidth. However, in the literature, the traffic grooming problem is either solved as a separate problem or jointly with the lightpath routing problem to make operational decisions [15], [129–131] or, it is modeled but not solved, i.e. computational experiments are performed using two-layer networks, or models are given for two-layer networks and simply stated that they are extendible to multilayer networks [3–5], [7], [117], [120], [122], [125–128]. The latter studies do not include any computational results related to traffic grooming problem. So that, to the best of our knowledge, there is no study that handles all of the subproblems in an integrated fashion.

2.4 Network Problems and Telecommunication Network Design Problems

The solution methods of the TNDP are closely related to the solution methods of network optimization problems since network optimization problems are main tools for modeling the TNDP. Hence, it is important to see the relationship between telecommunication network design problems and network optimization problems. This relationship is presented in Table 6. The studies in the literature are exemplified and referenced in the table.

Table 6. Telecommunication Network Design Problems vs. Network Design Problems

Network Optimization Problems			**TNDP**	**Remarks**
Tree	Minimum spanning tree (MST)	Uncapacitated MST	Terminal layout	Optimal solution may include some links with high flow while some links have quite low flow
		Degree constrained MST	Terminal layout [39]	Degree constraint represents the capacity of processor device installed on the node
		Capacitated MST	Terminal layout [37]	Links have limited capacity flow

Table 6 (Cont'd)

Network Optimization Problems			TNDP	Remarks
		Multicenter capacitated MST	Choosing locations of backbone nodes Terminal assignment Terminal layout [44]	The related problem is to assign terminal nodes to backbone nodes which involves solving several problems jointly where there are several root nodes with different capacity
		Multilevel capacitated MST	Terminal layout [133] Telpak [134]	The difference from CMST is arcs are of different capacity
	Steiner tree		Terminal layout [135] Multilevel network design [136]	Although NP hard problem, it is used for being sure about existence of intermediate nodes between terminals and the root node.
Flow	Minimum Cost Single Commodity Flow Problem	Multiterminal network flow problem with heterogeneous terminals	Terminal layout [21] Telpak [35]	Terminals are different from each other in terms of traffic they generate while communication lines can vary in capacity and cost
		Telpak problem	Terminal layout Telpak [35]	Multiterminal network flow problem with heterogeneous terminals that uses staircase cost function
		One terminal Telpak problem	Terminal layout Telpak [36]	Telpak problem that involves one type of terminal in terms of traffic requirements
	Minimum cost multicommodity flow problem	Linear cost function case Linear with fixed cost case Piecewise linear concave cost function case Step increasing cost function case	Topology design - selection of nodes and edges of a network [127] Dimensioning - determine capacity of links [45] Routing and capacity assignment problem in backbone networks[51] Multilayer network design [122] Capacity expansion [102]	Solves the two basic problems of topology design and dimensioning jointly

Table 6 (Cont'd)

Network Optimization Problems		TNDP	Remarks
Flow	Multicommodity network flow with gains	Terminal assignment problem Terminal layout problem Telpak problem Concentrator location problem Capacity expansion problem [137]	Solves subproblems of local access network design problem jointly when more than one technology is used in links and some type of hardware is needed to convert the signals transmitted by links of different technology
Location	Concentrator location problem	Concentrator location problem Terminal assignment Terminal layout problem Multilevel network design problem [19], [27], [138]	Solves the given problems jointly. The network design problem can shown to be equivalent to constrained Steiner tree problem
	Capacitated facility location problem	Concentrator location problem Terminal assignment Terminal layout problem [19], [27], [138]	Concentrator location problem with a star-star topology

Several critical observations can be made using Table 6. First of all, the tree topology is used for TNDP when the TNDP is difficult for general topologies, in order to take the advantage of the simplicity that tree topology brings. In the earlier studies, tree topology hence the tree problems are seen more commonly especially for topology design problems. As the computing power increases in time, the other topologies, as ring and mesh, have come into stage with the increasing importance of survivability concept. However tree topology is still important as advances in technology bring more complexity to the telecommunication network design problems. Comparing the network optimization problems, the capacitated minimum spanning tree and its variants are used very commonly, however Steiner trees are more realistic that the existence of the transmission nodes needs to be certain for routing the demand. The tree problems and location problems are generally used for subproblem solutions. As the solution strategy moves from sequential design to integrated design, flow problems gain importance. The multi-tier tree, the multicommodity flow problem with gains and the multilayer network design problem are more complex network optimization problems to meet special requirements that arise as telecommunication technologies advance and networks become multilevel, multilayer, etc.

2.5 Conclusion

In this study, TNDPs are reviewed from the point of view of OR. Main network design problems are introduced with recent studies in the literature. The solution methods and their solution capabilities provided in the literature are presented. The studies which are grouped under the problem types are also classified according to some problem features that we think affect the complexity of the TNDP as they affect the underlying network optimization problem.

The classification of TNDP reveals that although the local access network design problems are well studied, the multifacility local access network design problem is still challenging. It is seen that tree topology is the most studied topology for the TNDP especially in the earlier studies as it simplifies solution of problems and other topologies are harder to solve than tree topology. Capacity expansion of backbone networks and local access networks are studied more than designing backbone network and local access network from scratch in recent studies. In addition, the integrated design of backbone and local access networks are studied instead of designing these networks one by one in recent studies. We observe that multiperiod cases of most of the TNDP are not studied very well although multiperiod case is more realistic than single period case. Survivability is an important issue for backbone networks. In recent studies, survivable backbone network design and backbone networks with mesh topology are studied with technology specific constraints.

Most studies in the literature are either too general to reflect the real life situations, or too specific to the technology. There are a few that balance the specifications in technology and generality. The general telecommunication network design models are mainly related to the strategic design decisions where the information about the inputs are not known in detail while the technology specific models are mainly specific to the case the model is defined for. There are some models that try to include all design considerations about the problem independent from the technology and most of them have not been solved. The OR researchers mostly study the general models, while the technology specific models are mostly studied by electrical and electronics engineers. It is worth to emphasize that the general problems are mainly related with strategic decisions like the location of node hardware and their connections while the specific models are mainly related to operational decisions such as capacity assignment, reliability, and routing of the traffic demand.

The economies-of-scale characteristics of telecommunication investments constitute one of the driving forces of the research of the TNDP, but reflecting these characteristics into the network optimization models makes the models difficult to solve. The cost functions are concave in the real life because amount of money paid to investments decrease as amount of investment increases. The problems with concave costs are very difficult to solve, so that step increasing cost functions are used to approximate the concave cost functions to preserve the economies of scale characteristic by introducing modularity concept of transmission links. Note that this modularity concept drives the research for the multifacility TNDP. However, the problems with step increasing cost functions are difficult to solve, too. In addition, there is a trade-off between installing all equipment at the beginning and installing equipments in a time period gradually. The former option takes the advantage of economies of scale as it decreases installation costs by installing a large amount of facility, though the operating costs increase. This option leads spare capacity installation before it is needed. The latter option is related to prepare a multiperiod installation/expansion/update schedule for telecommunication networks. This approach uses the advantage of using new technology as advances in technology decrease the equipment costs. However, congestion costs may increase between periods due to increase in the traffic demand. This trade-off shows that the multiperiod telecommunication design problems are more realistic than the single period ones. As a consequence, if the TNDP is desired to be solved appropriate to the real life, the problem should be multi-facility, multi-period and with a concave or step-increasing cost function.

The TNDP are evolving from the point of view of proposed solution methods. The earlier studies tend to decompose the telecommunication network problem into subproblems and sequentially solve these subproblems. The recent studies mainly focus on integrated solutions of subproblems and different telecommunication network design problems together. This drives the researches like survivable capacity expansion problem, multilevel network design problem, multi layer network design problem, etc.

Even if the proposed model to solve the telecommunication network problem is not realistic, it is still investigated since the real life problem is too complicated to solve and the proposed models may give some insights to solve the real life problem. A good example for this observation is the single period network expansion problems that are used to get insights to solve the multiperiod network design problems.

The demand matrices used to solve the telecommunication network design problems are deterministic most of the time. Especially for strategic decisions, a forecast of the traffic demand is done and estimated demand values are used to solve for node locations and connections between nodes. Note that, a telecommunication network has to be feasible in terms of routing the traffic demand and the strategic decisions are mainly related to find the best network configuration in terms of installation, leasing and operational costs between the feasible network configurations. Thus, using a single demand matrix in strategic decisions is acceptable in this context, though there are some shortcomings of this approach such that some of capacity may be idle at the beginning. When operational decisions like how the routing is done on the network or self healing capabilities are to be planned, robustness of the networks become more important than finding a minimal cost network that satisfy a single forecasted demand matrix. It is observed through the review that the robust network design techniques are begun to be used recently.

We listed some of the critical points regarding the TNDP:

- Improvements in computation power of computers lead a shift in solution approaches of the TNDP from decomposing problems into subproblems and solving them sequentially to joint solution of the subproblems, especially in 2000s.
- Survivability is an important issue in topological design of telecommunication networks as the necessary degree of survivability changes the physical topology of the network. However, incorporating survivability issues during the TNDP extends the network design problem and solution of the problem gets more difficult.

- Even though the topological design of telecommunication networks is a classical problem, it is still important for expansion and update of current telecommunication networks.
- Most studies in the literature assume that there is no uncertainty in the traffic demands unlike the real-life situation.
- Telecommunication networks involve multiple layer structure including IP networks and core networks of cellular networks. Multilayer network design problem has been used recently for the TNDP bringing joint design of virtual topology along with the physical topology as planning each layer apart from the others leads to suboptimal solutions and even infeasible solutions if survivability exists. The existing multilayer TNDP are complex to solve realistic multilayer instances for integrated design. Efficient heuristics and exact algorithms are needed for telecommunication networks with more than two layers. In addition, there are many topics related to multilayer network design problem that have just begun to be studied or have not been studied yet including single path routing, integral flow, multi-hour traffic and multi-service network traffic, robustness and energy efficiency.
- Capacity expansion problem is a difficult problem since it is a general case of network loading problem and hence the minimum cost multicommodity flow problem with a step increasing cost function. In addition, for the local access network design problems, it is solved as an extension of another difficult problem, the capacitated minimum spanning tree problem. It is observed that heuristic algorithms are proposed multi-period, multi-facility capacity expansion problems, but there is a lack of exact algorithms for this problem type. Survivability is incorporated with capacity expansion in a very few studies, and no exact procedure is proposed to solve capacity expansion problem for survivable networks.
- The multifacility and multiperiod problems with concave or step increasing cost functions are more appropriate to reflect the economies of scale characteristic of the TNDP. Hence, the multicommodity network design problem with concave cost and step increasing cost function is important to solve realistic telecommunication network design problems. If the performance of the solutions proposed to these problems is considered, it is seen that further research is needed for better solutions.

CHAPTER 3

NETWORK OPTIMIZATION PROBLEMS IN TELECOMMUNICATION NETWORK DESIGN

Network optimization problems are the tools for modeling and solving the TNDPs. The complexity and solution difficulty of a TNDP depends on the network optimization problem that is used to model and solve it. There are a number of TNDPs that involve various decisions and variables which do or do not map directly to a network optimization problem. Hence, it is important to know what is modeled by network optimization problems, how they are modeled and solved in order to model and solve TNDPs in an efficient and effective way. In this chapter, we surveyed network optimization problems in telecommunication network design. We reviewed the network optimization problems in telecommunication network design and unified the notation of the network optimization problems in the TNDP to see the variation of input and output of network optimization problems. We do not claim that this chapter presents a comprehensive survey of network optimization problems used in telecommunication. However, it is comprehensive enough to be used by an OR researcher to be informed about the basic network optimization problems used in telecommunication network design. Our main purpose is to provide a toolbox of network optimization problems to be used modeling and solving TNDPs.

In the literature, there are several reviews about network optimization problems in telecommunication [97], [135], [139–148]. These reviews include a certain network optimization problem type. In this study, we surveyed the network optimization problems together with their connection to telecommunication network design problem types. We unify the notation of the network optimization problems in telecommunication network design to enable comparison of inputs, outputs and decisions that can be made using a particular network optimization problem. In addition, this survey emphasizes the recent studies about the network optimization that are not included in the existing surveys.

3.1 Minimum Spanning Tree Problem

The minimum spanning tree problem is a fundamental problem in design of computer communication networks [9].

The related network design problem is the centralized network design problem, where a given set of terminals have to be connected through transmission lines to a central computer or data processing center and each terminal should be connected to the center [21]. Thus, the problem is basically related with terminal layout problem.

Minimum spanning tree problem is defined on a graph $G = (I, A)$ such that

$\tilde{I} = \{1,2,..,N\}$ is the set of terminal nodes, node 0 is the root node, $I = \tilde{I} \cup \{0\} = \{0,1,2,..,N\}$ is the set of nodes and A is the set of arcs between nodes. The cost of connecting nodes i and j by an arc (i,j) is denoted by c_{ij} where $c_{ij} \geq 0$ and $c_{ij} = c_{ji}$.

In [9], Gavish presented a formulation for the minimum spanning tree problem, which is used as a basis for formulating network design problems. Gavish proves that the formulation solves the minimum spanning tree problem by showing that the resulting graph does not contain any cycles and the number of arcs in the resulting graph is equal to the number of nodes minus one. The formulation given in [9] is as follows:

$$Min \qquad Z = \sum_{i=1}^{N} \sum_{\substack{j=0 \\ j \neq i}}^{N} c_{ij} y_{ij} \qquad (1.1)$$

$$subject\ to$$

$$\sum_{\substack{j=0 \\ j \neq i}}^{N} y_{ij} = 1 \qquad i = 1,...,N \qquad (1.2)$$

$$\sum_{\substack{j=0 \\ j \neq i}}^{N} x_{ij} - \sum_{\substack{j=1 \\ j \neq i}}^{N} x_{ji} = 1 \qquad i = 1,...,N \qquad (1.3)$$

$$x_{ij} \leq (n-1) y_{ij} \qquad i = 1,...,N, \quad j = 0,...,N \ and \ j \neq i \qquad (1.4)$$

$$x_{ij} \geq 0 \ and \ y_{ij} = 0 \ or \ 1 \qquad (1.5)$$

where

- $y_{ij} = \begin{cases} 1, & arc\ (i,j)\ is\ included\ in\ the\ optimal\ solution \\ 0, & otherwise \end{cases}, \quad y_{0j} = 0, \quad \forall j \in I$
- x_{ij} = flow of commodity on arc (i,j) connecting the nodes i and j

Let x_{ij}^* and y_{ij}^* be the optimal solution to the problem. Define the graph $G = (I,B)$ that is associated with the optimal solution such that I is the node set and $B = \{(i,j) \mid y_{ij}^* = 1\}$ is the arc set. The objective function (1.1) is to minimize the total link costs. Constraint (1.2) guarantees that the components in $G = (I,B)$ that might contain cycles could be composed of either a simple cycle or a one or more disconnected simple cycle with sub trees leading into it. (1.3) is flow conservation constraint and (1.4) guarantees that x_{ij}^* could be positive if and only if $(i,j) \in B$. Note that, the optimal solution cannot contain any cycles since constraint (1.3) is violated.

The history of algorithms to solve the minimum spanning tree problem is given in Graham and Hell [146]. On a more recent review, Bazlamacci and Hindi [149] compares performance of algorithms to solve the minimum spanning tree problem.

The variations of the problem include degree constrained minimum spanning tree, capacitated minimum spanning tree, multi-center capacitated minimum spanning tree and multi-level multi-center capacitated minimum spanning tree.

3.1.1 Degree Constrained Minimum Spanning Tree Problem

The degree constrained minimum spanning tree problem involves the minimum spanning tree problem with upper bounds on the number of arcs incident to the nodes of the tree [9].

Related network design problem is the centralized network design problem with upper bounds on the number of links that can be installed incident to each node. This can be considered as a capacity constraint since the number of links adjacent to the nodes represents the capacity of processor device installed on the node i.e., the capacity of concentrator.

The problem can be formulated by adding the following constraint to the formulation given for the minimum spanning tree problem [9].

$$\sum_{i=1}^{n} y_{ik} + \sum_{i=0}^{n} y_{ki} \leq q_k \qquad \forall k \in S \tag{2.1}$$

where

- q_k is the upper bound on the number of links that can be installed incident to node k for $\forall k \in S$ and S is a subset of nodes.

Gavish proposes a Lagrangian based algorithm for the degree constrained minimum spanning tree problem [9].

3.1.2 Capacity Constrained Minimum Spanning Tree Problem

Optimal solution of centralized network design using uncapacitated minimum spanning tree may include some links with a large flow while some other links with flow that is quite low depending on the structure of the tree obtained. This result may give an expensive solution if the average capacity usage of the network is considered [21]. In order to prevent this, a capacity constraint is imposed on minimum spanning tree. The capacity constraint limits the total flow on each link. This capacity constraint is in fact imposed by the hardware in the root node, such that each port of the hardware in the root node has a limited capacity which is notated as Q and each sub tree connected to these ports cannot have traffic flow more than this capacity.

The weights are thought to be flows on the links. If unit weights are assumed, then the capacitated minimum spanning tree problem with equal weights is obtained. This problem is a special case of the degree constrained minimum spanning tree problem [10].

The related network design problem is the centralized network design problem where nodes are to be connected to a single computer center or data processing unit with links which have limited capacity of flow. The problem is basically related with the terminal layout problem, mentioned in Section 2.3.1.

The CMST problem is defined on a graph $G = (I, A)$ such that $\tilde{I} = \{1, 2, .., N\}$ is the set of terminal nodes, node 0 is the root node, $I = \tilde{I} \cup \{0\} = \{0, 1, 2, .., N\}$ is the set of nodes and A is the set of arcs between nodes. Q is an upper bound on traffic that each link can carry, Q is the capacity of each port on root node. The cost of connecting nodes i and j by an arc (i, j) is denoted by c_{ij} where $c_{ij} \geq 0$ and $c_{ij} = c_{ji}$.

The CMST problem is formulated as the CMST problem with unitary demand and the CMST problem with non-unitary demand. Unitary demand case occurs when every terminal produces the same amount of traffic, which is equivalent to the case where each terminal has unit demand; otherwise it is called non-unitary demand. In fact, for the unitary demand case the capacity constraint limits the number of nodes in any multi-point line by a fixed value such that the number of nodes in any multi-point line cannot be greater than the certain value.

The formulations and the solutions proposed to solve the CMST problem differ according to the demand structure, unitary or non-unitary demand cases.

i. Unitary Demand Case

Unitary demand case for the CMST problem with capacity $Q = 1$ is trivial. The CMST problem with $Q = 2$ can be solved as a weighted matching problem [150]. The case with a capacity greater than or equal to the number of terminal nodes *n* is clearly equivalent to the uncapacitated minimum spanning tree problem. However, *2<Q<N/2* is NP-Hard [151].

A zero-one integer programming formulation, a single commodity formulation, a multicommodity formulation and a hop-indexed formulation of the unitary demand CMST are proposed. The zero-one integer programming formulation is used by Altinkemer to formulate the multi-center capacitated minimum spanning tree problem [44].

The single commodity formulation of the CMST problem is presented by Gavish [152]. The formulation is obtained by putting the following capacity constraint instead of constraint (1.4) where $(d_0,\ d_1, \ldots,\ d_N)$ is a vector such that $d_0 = 0$ and $d_i = 1$ where $i = 1, \ldots,\ N$.

$$y_{ij} \leq x_{ij} \leq (Q - d_i) y_{ij} \qquad i = 0, \ldots, N; j = 1, \ldots, N \qquad (3.1)$$

The multicommodity formulation of the CMST problem is also presented by Gavish. The formulation is based on multicommodity flow variables x_{ijk}, such that

$$x_{ijk} = \begin{Bmatrix} 1, & \textit{if a unit commodity } k \textit{ flows on link } (i,j) \\ 0, & \textit{otherwise} \end{Bmatrix}$$

The multicommodity formulation assumes there is a unit flow of commodity k that starts at node p and terminates at the center, node 0, $k = 1, \ldots, N$. d_i is the amount of traffic that has to be transferred between node i and node 0, for $\forall i \in \tilde{I}$ [10].

$$\textit{Minimize} \qquad Z = \sum_{i=1}^{N}\sum_{j=0}^{N} c_{ij} y_{ij} \tag{4.1}$$

subject to

$$\sum_{j=0}^{N} y_{ij} = 1 \qquad i = 1,\ldots,N \tag{4.2}$$

$$\sum_{j=1}^{N} x_{jik} - \sum_{j=0}^{N} x_{ijk} = \begin{cases} -1, & i = k,\ k = 1,\ldots,N \\ 0, & i \neq k,\ i = 1,\ldots,N,\ k = 1,\ldots,N \\ 1, & i = 0,\ k = 1,\ldots,N \end{cases} \tag{4.3}$$

$$x_{ijp} \le y_{ij} \qquad \forall i, j = 1,\ldots,N,\ k = 1,\ldots,N \tag{4.4}$$

$$\sum_{j=1}^{N}\sum_{p=1}^{N} x_{jik} d_k \le Q - d_i \qquad i = 1,\ldots,N \tag{4.5}$$

$$y_{ij} = 0 \ or \ 1 \qquad \forall i, j \tag{4.6}$$

$$x_{ijk} = 0 \ or \ 1 \qquad \forall i, j, k \tag{4.7}$$

The objective function (4.1) minimizes the total link costs. Constraint (4.2) guarantees that each node has a link leading out of it. Constraint (4.3) is the multicommodity network flow constraints, i.e., one unit of flow of commodity k that begins in node k, $k = 1,\ldots,n$ terminates at the root node. When link (i, j) is not part of the links selected by the design y_{ij} variables, (4.4) restrict the flow on link (i, j) to zero. Constraint (4.5) satisfies node and link capacity restrictions by ensuring that the flow entering node i is less than $Q - d_i$. For each end user node a path of links exists between it and the center node is guaranteed by constraints (4.2) - (4.4). Constraints (4.2) - (4.4) together with the summation of constraints (4.2) ensure that y_{ij} variables form a tree [10].

It is shown that the linear programming relaxation of zero-one integer programming formulation is tighter than linear programming relaxation of multicommodity flow formulation in Gavish [10].

Gouveia and Martins [140] proposed a “hop-indexed” single-commodity flow model that generalizes a well-known single-commodity flow model due to Gavish. The formulation uses the “hop’ index t, $t \in T$, $T = \{1,\ldots,Q\}$ such that if an arc is in position t in a feasible solution, it means that there are $t-1$ arcs to reach the root node. t ranges from 1 to Q since the position of any arc cannot be greater than the capacity, Q, in a feasible solution of the CMST problem.

$$\textit{Min} \qquad Z = \sum_{i=0}^{N}\sum_{j=1}^{N}\sum_{t=1}^{Q} c_{ij} y_{ijt} \tag{5.1}$$

subject to

$$\sum_{i=0}^{N}\sum_{t=1}^{Q} y_{ijt} = 1 \qquad j = 1,\ldots,N \tag{5.2}$$

$$\sum_{i=0}^{N} x_{ijt} - \sum_{i=1}^{N} x_{ji,t+1} = \sum_{i=0}^{N} x_{ijt} \qquad j = 1,\ldots,N;\ t = 1,\ldots,Q-1 \tag{5.3}$$

$$y_{ijt} \le x_{ijt} \le (Q - t + 1) y_{ijt} \qquad i = 0,\ldots,N;\ j = 1,..,N; t = 1,..,Q \tag{5.4}$$

where

- $y_{ijt} = \begin{cases} 1, & if\ arc\ (i,\ j)\ is\ in\ position\ t\ in\ the\ optimal\ solution \\ 0, & otherwise \end{cases}$

- x_{ij} = the amount of flow produced by the root going through arc (i, j) if this arc is in position t.

The objective function (5.1) gives the minimum total link cost. (5.2) ensures that each node has a link that goes into it. Constraint (5.3) is a generalization of the flow conservation constraints that accounts for the position of the arcs. Constraint (5.3) states that a flow coming into a node via an arc that is in position t leaves the node with arcs that are in position $t+1$. Constraint (5.4) is capacity constraint.

It is shown in [140] that the linear program relaxation of the hop-indexed CMST formulation is at least as tight as the single commodity formulation of the CMST.

ii. Non-unitary demand case

In a recent study, an arc based formulation of the non-unitary CMST is presented by Uchoa et al. [153]. The formulation is defined on a directed graph $G = (I, A)$, such that $A = \{(i, j) : i \in I, j \in I\}$. The cost of installing a link on arc a is notated by c_a and the binary variable y_a is defined as

$y_a = \begin{Bmatrix} 1, & arc\, a\ is\ included\, in\, the\, optimal\, solution \\ 0, & otherwise \end{Bmatrix}$. The formulation is given below:

$$Minimize\ \sum_{a \in A} c_a y_a \qquad (6.1)$$

$$subject\ to$$

$$\sum_{a \in \{(j,i): j \in I\}} y_a = 1 \qquad \forall i \in I \qquad (6.2)$$

$$\sum_{a \in \{(i,j): i \in I \setminus S,\ j \in S\}} y_a \geq \left\lceil \frac{\sum_{i \in I} d_i}{Q} \right\rceil \qquad \forall S \subseteq I \qquad (6.3)$$

$$x_a \in \{1, 0\} \qquad \forall a \in \qquad (6.4)$$

(6.1) is the objective function that minimizes the total arc costs. Constraint (6.2) is called "in-degree constraints" and states that exactly one arc must enter each non-root-vertex. Constraint (6.3) is called "capacity cuts" and states that at least k(S) arcs must enter each set S.

However, the formulation above cannot capture the knapsack like aspects of the CMST [153]. Uchoa et al. [153] proposed a formulation where the variables are associated to q-arbs. This structure arises from a relaxation of the capacitated prize-collecting arborescence problem in order to make it solvable in pseudo polynomial time. The q-arb formulation is obtained by adding the following constraints to directed graph formulation where all possible q-arbs are numbered from 1 to B:

$$\sum_{j=1}^{p} b_a^j \lambda_j - y_a = 0 \qquad \forall a \in A \qquad (7.1)$$
$$\lambda_j \geq 0 \qquad j = 1,...p \qquad (7.2)$$

where

- b_a^i is the number of times that arc a appears in the i[th] q-arb.

Constraint (7.1) imposes that y must be a weighted sum of arc-incidence vectors of q-arbs [153] and the λ_j in (7.1) are the non-negative weights used in (7.1).

Uchoa et al. present capacity indexed formulation of the CMST problem due to Gouveia [154]. Uchoa et al. [153] proposes a new formulation by combining the capacity indexed formulation and the q-abs which results in a stronger formulation. In addition, Uchoa et al. propose some valid inequalities [153].

The CMST problem and the associated constrained minimum spanning tree problems are NP Hard [151], [155], [156]. Several methods including exact and heuristic methods are developed to solve the capacitated minimum spanning tree problem. The greedy heuristic algorithms are divided into two groups: first order greedy algorithms (FOGA) and second order greedy heuristic algorithms (SOGA).

- *FOGA:* A first order greedy algorithm can be considered as a construction heuristic which builds a spanning tree by adding one arc at a time to a partial tree.
- *SOGA:* A second order algorithm involves using a different algorithm, usually Esau-Williams' savings heuristic or parallel savings heuristic due to Gavish and Altinkemer [40], to explore subproblems which are formed by adding some constraints to the original problem as fixing some nodes to be included to or excluded from the optimal solution [148].

The methods proposed up to 1999 are surveyed in Gouveia and Martins [140]. The exact solution methods are summarized in Table 7 and heuristics are summarized in Table 8 such that the emphasis is made on the studies after 1999 and some important work done before 1999. Besides the survey that Amberg et al. included in their paper contains some heuristic and exact procedures [148].

The computational experiments are performed mainly with the problem sets in ORLIB due to Beasley. The problem sets are called tc and te, where tc represents the problems whose root node is in the center of the other nodes and te represents the problems whose root node is in the corner, i.e., at the end of the node scatter. Hall [152] states that placing the root node in the center makes the capacity constraints less restrictive compared to placing the root node in the corner.

Table 7. The Capacitated Minimum Spanning Tree - Exact Methods

Paper	Solution Method	Solution Capability	New Formulation	New Valid Inequalities
[153]	Branch-cut and price	Computational results on benchmark instances from the OR-Library show very significant improvements over previous algorithms. Several open instances could be solved to optimality. CMST problems having up to 200 nodes with capacities 200, 400 and 800 can be solved by the method	-Arc based formulation -q- arbs formulation -Capacity indexed formulation -Capacity indexed formulation combined with q-arbs	- Homogeneous Extended Capacity Cuts
[157]	Cutting plane algorithm: two improvements added to the cutting plane algorithm proposed by Gouveia and Martins [158] A new set of inequalities that can be seen as hop-indexed generalization of the well known generalized subtour elimination (GSE) constraints An improved separation heuristic for the original set of GSE constraints	The problems tested in [158]are used in addition to two new instances with 120 nodes are tested.	- Hierarchical hop-indexed single commodity flow formulation	-Hop-ordering constraints -Generalized subtour elimination constraints -Hop indexed generalization of generalized subtour elimination constraints

Table 7 (Cont'd)

Paper	Solution Method	Solution Capability	New Formulation	New Valid Inequalities
[41]	Lagrangian Relaxation method where the subproblem is a directed spanning tree with a degree constraint on the root. To solve the master problem, a cut-and-column generation algorithm based on analytic centers is proposed.	The proposed method is compared to Gouveia and Martin's cutting plane algorithm [140] and iterative procedure [158] and with Hall's cutting plane algorithm [159]. The numerical results indicate that the proposed algorithm outperforms the proposed algorithms in directed case.	-Degree based model for capacitated minimum directed spanning tree problem	- φ(S) cuts where φ(S) is the minimum number of disjoint sub trees of S needed to cover its load.
[158]	Cutting plane algorithm which uses hierarchical hop indexed formulation: several levels of aggregation of the original formulation in [140] yielding a hierarchy of hop-indexed LP models which suggests an iterative method for computing lower bounds for the CMSTP and iteratively transforms a given model into a more disaggregated model with a tighter relaxation.	The tests are performed for problems with 41, 81 and 121 nodes.	-Hierarchical hop-indexed single commodity flow formulation	-Hop-ordering constraints generalized subtour elimination constraints

Table 7 (Cont'd)

Paper	Solution Method	Solution Capability	New Formulation	New Valid Inequalities
[140]	Cutting plane	CMST problems having up to 80 nodes with capacity 20 can be solved by the method. The best improvements are obtained for the tightly capacitated instances with the root in the corner which correspond precisely to the cases which have been considered hard by most of the best lower-bounding schemes known to date.	- Hop-indexed single commodity flow formulation	- Hop-ordering constraints generalized subtour elimination constraints
[39]	Augmented Lagrangian relaxation	The lower bounds found and the lower bounding scheme was the best found for nearly ten years before better ones are found.		
[152]	Dantzig-Wolfe decomposition Lagrangian relaxation	Lagrangian relaxation outperformed Dantzig-Wolfe decomposition.	- Single-commodity flow model (tighter than the one in [9])	
[9]	Bender's decomposition	The procedure has been tested on problems varying in size from n = 6up to n = 12, with very disappointing results since number of cuts generated was very large. Method leads a useful result of identifying valid inequalities for the LP characterization of the problem.	- Single-commodity flow formulation	- Generalized cut constraints

The computational experiments are performed mainly with the problem sets in ORLIB due to Beasley. The problem sets are called tc and te, where tc represents the problems whose root node is in the center of the other nodes and te represents the problems whose root node is in the corner, i.e., at the end of the node scatter. Hall [159] states that placing the root node in the center makes the capacity constraints less restrictive compared to placing the root node in the corner.

Table 8. The Capacitated Minimum Spanning Tree - Heuristic Methods

Paper	Solution Method	Solution Capability
[160]	Variable Neighborhood Search (VNS) approach which uses three different neighborhood types	With up to 1280 nodes indicate especially on instances With many nodes per cluster significant advantages over previously published metaheuristic approaches.
[161]	Combined neighborhood search and branch and bound technique	Computational experiments contain only test problems with 41 nodes.
[162]	An enhanced version of the well-known second order algorithm of Karnaugh [163] with inclusion of backward steps and some memory features. The proposed algorithm differs from the Karnaugh's second order algorithm such that when there is no improvement, the algorithm makes look-behind strategy to perform a backward step. This backward step is established by the dropping of a constraint from the accumulated set of constraints.	The algorithm is tested with problems with up to 160 nodes and capacity 20.
[164]	A fast approximate reasoning algorithm, which is based on the Esau–Williams savings heuristic and fuzzy logic rules	Test sets with 10, 20, 30, 40, 50 and 60 vertices were randomly generated. For each test set capacity bounds are taken as 7, 9 and 11.
[165]	Ant colony optimization (ACO) algorithm The algorithm exploits two important problem characteristics: (i) the CMST problem is closely related to the capacitated vehicle routing problem (CVRP), (ii) given a clustering of client nodes that satisfies capacity constraints, the solution is to find a MST for each cluster, which can be done exactly in polynomial time	The algorithm is tested for the problems with 40 and 80 nodes with capacities 5, 10, and 20.

Table 8 (Cont'd)

Paper	Solution Method	Solution Capability
[37]	Genetic algorithm	N/A
[166]	Approximation algorithm	(γ+2)-approximation ratio for the CMST problem is obtained where γ is inverse Steiner ratio and the ratio is 3.1548 for Euclidian plane and 3.5 for rectilinear plane. This is an improvement over the current best ratio of 4 for this problem.
[167]	A neighborhood structure which is used by a local search strategy is proposed. In addition a GRASP with path relinking heuristic is proposed.	The proposed algorithm is tested for te and tc problems with 40 to 160 nodes. The GRASP heuristic using a memory-based local search strategy improved the best known solution for five out of the six largest benchmark problems
[168]	Implemented GRASP and tabu search algorithms with the following multi-exchange neighborhood schemes: - exchanges of single nodes among several sub trees. -exchanges that involve multiple sub trees.	The algorithm was tested with benchmark test problems up to 200 nodes with capacity of 200 and 100 nodes with capacity of 400
[169]	Adaptive reasoning technique: iteratively solve the CMST by using the EW heuristic, and at each iteration modify a set of additional constraints	The tests are performed with 40 node problems using arc capacities of 3, 5, and 10; and the 80 node problems with arc capacities of 5, 10, and 20. The problem sets are tc where the root node in the center of the nodes and te where the root node is in the end of node scatter. These are benchmark problems from OR Library. The results are compared to methods due to Esau-Williams [38]-EW, Sharaiha et al. [170]-TS, Amberg, Domshke and Voß [148]AMB, Karnaugh [163] -KAR, Kershenbaum, Boorstyn, Oppenheim [42] - KBO and Gavish and Altinkemer [40]-IPSA(improved version of the parallel savings heuristic). The proposed algorithm outperformed the methods for tc and te problem sets.

Table 8 (Cont'd)

Paper	Solution Method	Solution Capability
[170]	Tabu search Neighborhood structure: Tree-Based T' is a neighbor of T if it is obtained from T by performing cut and paste operations, that is, to cut the whole sub tree or a part of a sub tree and then connecting (paste) it to the root node or to some other sub tree	
[148]	Tabu search and simulated annealing Neighborhood structure: Node-Based T' is a neighbor of T if it is obtained from T by changing a pair of nodes between two components of T, or by shifting one node from one component to another. For the unit-demand version of the problem, the complexity order of the algorithm is O(nK2) if only shift moves are considered and O(nK3) if exchange moves are also allowed.	The procedure is tested for problems with 40 and 80 nodes with capacity up to 20. it is reported that the procedure improved some of the known lower bounds by 1996 and improved EW heuristic on the average 3-4 %.
[10]	Heuristic based on constructing TSP tour over nodes. An error analysis of QITP heuristic shows that there can be a polynomial time algorithm for tour partitioning. Modified PSA algorithm to handle heterogeneous demand: PSA with dummy nodes	QITP heuristic is of O(E) where E is the number of edges. For non unitary demand case, EW heuristic outperforms PSA with dummy nodes.
[171]	Heuristic based on constructing TSP tour over nodes: Q Iterated Tour Partitioning (QITP) heuristic	The heuristic is $O(n^2)$ and guarantees to have a worst case error bound of 4.
[40]	Best node procedure Examines all components at once unlike FOGA and SOGA type algorithms	Solutions for test problems up to 400 nodes are reported with computation times of two to three times the EW time. The proposed algorithm yields 2-5% improvements for unitary demand case while for non-unitary case, it performs poorly. Time complexity is $O(n^2 \log_2 n)$
[172]	Clustering	Gives inferior solutions than the Esau-Williams algorithm for similar amounts of computing times

Table 8 (Cont'd)

Paper	Solution Method	Solution Capability
[42]	Second order greedy algorithm which improves an initial feasible solution by restricting some arcs to be in the solution	2% better than Esau-Williams heuristic with compatible computation time.
[163]	A second order algorithm uses a greedy algorithm to obtain a feasible solution within each subproblem subset of solutions, among which are evaluated in the algorithm/ The subproblems are generated by adding constraints i.e. inclusion/exclusion of a node or arc. The proposed second order greedy algorithm for CMST that employs repeated calls of a modified Esau-Williams procedure based on a look-ahead strategy used for a tentative inclusion of a constraint to the problem performed each iteration	Algorithm is tested problems with at most 150 nodes with a largest capacity of 20.
[173]	A first order greedy heuristic algorithm which builds a spanning tree by adding one arc at a time to a partial tree	A parameterization method is suggested which yields 1 to 5% improvement over Esau-Williams heuristic but the computation times get 3 to tenfold over Esau-Williams heuristic. Has a computational complexity of $O(n^2 \log n)$
[38]	A first order greedy heuristic algorithm which builds a spanning tree by adding one arc at a time to a partial tree	Often used as a benchmark heuristic and has a computational complexity of $O(n^2 \log n)$

3.1.3 Multi-center Minimum Spanning Tree Problem

For the capacitated minimum spanning tree problem, it is assumed that all the nodes are connected to a single node which is the central computer or data processing center, while in the multi-center capacitated spanning tree problem there are more than one center and sub trees formed are connected to a set of multiple nodes [10].

The related problem is to assign user nodes to backbone nodes. The decisions related to assigning user nodes to backbone nodes accounts for the following [10]:

- Choosing backbone node locations out of a possible set of candidate locations,
- Making an assignment of user nodes to those backbone nodes (terminal assignment problem),
- Determining connections among nodes (terminal layout problem).

The multi-center capacitated minimum spanning tree problem finds tree networks where the possible location of backbone nodes and the user node locations with traffic generated at each user node are known in advance. The problem is NP hard since the cost structure of links connecting to a processor installed on a backbone node, processor capacity and link capacities might be backbone node dependent since different capacities for backbone nodes may exist on different locations.

The multi-center capacitated minimum spanning tree problem forms sub trees each of which generates at most a predetermined amount of traffic, rooted at one out of many backbone nodes with a minimum total cost. Each user node must be served by exactly one of possible backbone nodes [10].

The formulation is given by Altinkemer [44] where the number of backbone nodes is M (thus $N-M$ is the number terminal nodes). This formulation is based on the zero-one integer programming formulation of the CMST problem.

$$Min \qquad Z = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} c_{ij} y_{ij} \tag{8.1}$$

$subject\ to$

$$\sum_{j=1}^{N-1} y_{ij} + \sum_{j=i+1}^{N} y_{ij} \geq 1 \qquad i = M+1, \ldots, N-1 \tag{8.2}$$

$$\sum_{j=1}^{N-1} y_{jN} \geq 1 \tag{8.3}$$

$$\sum_{i=1}^{N-1} \sum_{j=i+1}^{N} y_{ij} = N - M \tag{8.4}$$

$$\sum_{i \in S} \sum_{\substack{j \in S \\ j > i}} y_{ij} \leq \left|S\right| - \tilde{L}_S \qquad \forall S \subseteq \left\{M+1, \ldots, N\right\}, \left|S\right| \geq 2 \tag{8.5}$$

$$y_{ij} = 0\ or\ 1 \qquad i = 1, \ldots, N-1\ and\ j = i+1, \ldots, N \tag{8.6}$$

where

- $S \subseteq \{1, \ldots, N\}$ is a subset of nodes with cardinality greater than or equal to 2 i.e. $\left|S\right| \geq 2$,
- $d_i, i = M+1, \ldots, N$ is the expected of traffic generated at node i
- $\tilde{L}_s = \max(L_s, 1)$ and L_s is the optimal solution to bin packing problem in which bins have a length of Q and items that have to be packed into these bins have length of d_l, $l \in S$

(8.1) is the objective function that minimizes the total cost of the links. Constraint (8.2) ensures that each terminal is connected to some other node. Node N is guaranteed to be connected to at least one of the other nodes by constraint (8.3). Constraint (8.4) ensures that there will be $N-M$ arcs in the final solution. Constraint (8.5) guarantees that the capacity restrictions are satisfied and there will not be any loops in the final solution [44].

The solutions proposed for the multi-center capacitated minimum spanning tree problem are given in Table 9.

Table 9. Solution Methods for the Multi-center Capacitated Minimum Spanning Tree Problem

Paper	Solution Method	Solution Capability
[44]	Parallel savings heuristic	The proposed algorithm is tested for the problems with100 nodes with arc capacity varying from 100 to 200. The number of center varies from 2 to 7. The algorithm is compared to Esau-Williams heuristic [38].The proposed algorithm outperformed EW algorithm in 130 of 131 problems.
[26]	Iterated tour partitioning heuristic and optimal partitioning heuristic	No computational experiment is included; the algorithms are generalized for multi-center case. The worst case performance of both of the heuristics is proved to be 3 –(2/q) in the equal weight case and 4-(4/q) in the unequal weight case where q is the capacity restriction

3.1.4 Multi-level Minimum Spanning Tree Problem

In the capacitated minimum spanning tree problem, a fixed capacity is associated with every arc composing a feasible tree by paying its full cost without considering if all the capacity used on this link. Considering different capacities on the arcs is a natural extension of the CMST problem and it is treated as the Multi-level Capacitated Minimum Spanning Tree (MLCMST) problem. The problem is first stated by Gamvros et al. [174] as multi-level capacitated minimum spanning tree problem. As Gamvros et al. mention that the problem has not been given much attention and the most closely related problems in the literature to the MLCMST are local access network design in which the network topology is not necessarily to be a tree [174] and Telpak problem [134]. Gamvros et al. [175] state that the formulation of the Telpak problem presented by Gavish in [9], which is restricted to be a tree is, is in fact a multi-level capacitated minimum spanning tree problem formulation.

The related network design problem is basically same as the related network design problem of the CMST, which is a centralized network design problem where the nodes are to be connected to a single computer center or data processing unit with capacitated links. The problem is basically related with the terminal layout problem, mentioned in Section 2.3.1.

Three different formulations of the problem are given in Gamvros et al. [175]. These are single commodity formulation, enhanced single commodity formulation, and multicommodity formulation. Single commodity formulation is equivalent to the formulation for the Telpak problem restricted to tree in Gavish [9]. The problem is defined on a graph $G=(I,A)$ where I is node set and A is arc set. Node 0 in node set is the root node, the rest are the terminal nodes. d_i is the integer traffic requirement of node i to be transmitted to center node. $L=\{0,1,\ldots,L\}$ is the set of facility types to be installed on arcs with capacities $q^0 < q^1 < ... < q^L$ and the cost of installing a type l facility between nodes i and j is c_{ij}^l.

i. Single Commodity Formulation - SCF:

$$Min \qquad Z = \sum_{(i,j)\in A} \sum_{l=0}^{L} c_{ij}^l s_{ij}^l \qquad (9.1)$$

$$subject\ to$$

$$\sum_{j:(i,j)\in A} x_{ji} - \sum_{m:(i,m)\in A} x_{im} = \begin{cases} -d_i & if \quad i \neq 0 \\ D_0 & if \quad i = 0 \end{cases} \qquad \forall i \in I \qquad (9.2)$$

$$x_{ij} \leq \sum_{l=0}^{L} q^l s_{ij}^l \qquad \forall (i,j) \in A \qquad (9.3)$$

$$\sum_{(i,j)\in A} \sum_{l=0}^{L} s_{ij}^l = 1 \qquad \forall i \in \tilde{I} \qquad (9.4)$$

$$\sum_{l=0}^{L} \left(s_{ij}^l + s_{ji}^l \right) \leq 1 \qquad \forall \left\{i,j\right\} \in E, i,j \neq 0 \qquad (9.5)$$

$$s_{ij}^l \in \left\{0,1\right\} \qquad \forall (i,j) \in A, l \in L \qquad (9.6)$$

$$x_{ij} \geq 0 \qquad \forall (i,j) \in A \qquad (9.7)$$

where

- d_i = the amount of traffic supplied by node i (i.e. for unitary demand case, d_i=1 for $\forall i \in I$)
- D_0 = the amount of traffic demanded by the root node $\left\{0\right\}$ (i.e. for unitary demand case, $D_0 = N - 1$)
- $s_{ij}^l = \begin{Bmatrix} 1, & 1\ if\ \ a\ facility\ of\ type\ \ l\ \ is\ installed\ on\ arc\ \ (i,j) \\ 0, & otherwise \end{Bmatrix}$
- x_{ij} = the amount of flow on the arc connecting nodes i and j

(9.1) is the objective function that minimizes the total link cost. Constraint (9.2) ensures that demand of each node is sent to the central node. It is guaranteed by constraint (9.3) that the flow sent on an arc is less than the capacity of the facility installed on that arc. Existence of exactly one arc, and one facility type, directed out of node i is ensured by (9.4). Constraint (9.5) ensures that no more than one facility is installed between two nodes, and the facility is used in only one direction [175].

ii. Enhanced Single Commodity Formulation - ESCF:

The LP relaxation of the single commodity formulation is weak so they strengthen the formulation by

- adding the following constraint since there must be a flow on (i,j) if there is a facility installed on arc (i,j). In addition, if the facility installed on arc (i,j) is of type $l > 0$, then there must be at least $q^{l-1} + 1$ units of flow on the arc [175]:

$$x_{ij} \geq y_{ij}^0 + \sum_{l=1}^{L} q^{l-1} y_{ij}^l \qquad \forall (i,j) \in A$$

- and replace constraint (9.3) with the following ones since for any arc (i,j) for which node j is not the center node and a facility type L is installed on, the flow on arc (i,j) is less than or equal to $(q^L - D_i)$ [175] in order to come up with the enhanced single commodity formulation.

$$x_{i0} \le \sum_{l=0}^{L} q^l y_{i0}^l \qquad \forall (i,0) \in A$$

$$x_{ij} \le (q^L - d_i) y_{ij}^L + \sum_{l=0}^{L-1} q^l y_{ij}^l \qquad \forall (i,j) \in A, j \ne 0$$

iii. Multicommodity Flow Formulation – MCF:

They created commodity for each terminal node with a supply of one at the terminal node and a demand of one the central node. In the given notation, origin of commodity *k* is node *k* and the destination of commodity *k* is node 0.

$$Min \qquad \sum_{(i,j)\in A} \sum_{l=0}^{L} c_{ij}^l s_{ij}^l \qquad (10.1)$$

$$subject\ to$$

$$\sum_{j:(j,i)\in A} x_{ji}^k - \sum_{m:(i,m)\in A} x_{im}^k = \begin{cases} -1 & if \quad i = k \\ 1 & if \quad i = 0 \\ 0 & otherwise \end{cases} \quad \forall i \in I, \forall k \in K \qquad (10.2)$$

$$x_{ji}^k \le y_{ij} \qquad \forall (i,j) \in A, \forall k \in K \qquad (10.3)$$

$$x_{ij}^k \le \sum_{l=0}^{L} q^l f_{ij}^l \qquad \forall (i,j) \in A \qquad (10.4)$$

$$\sum_{k\in K} d_k x_{i0}^k \le \sum_{l=0}^{L} q_l s_{i0}^l \qquad \forall (i,0) \in A \qquad (10.5)$$

$$\sum_{k\in K} d_k x_{ij}^k \le (q_L - W_j) s_{ij}^l + \sum_{l=0}^{L-1} q_l s_{ij}^l \qquad \forall (i,j) \in A, j \ne 0 \qquad (10.6)$$

$$\sum_{j\in I} y_{ij} = 1 \qquad \forall i \in \tilde{I} \qquad (10.7)$$

$$y_{ij} + y_{ji} \le 1 \qquad \forall \{i,j\} \in E, j \ne c \qquad (10.8)$$

$$\sum_{l=0}^{L} s_{ij}^l = x_{ij} \qquad \forall (i,j) \in A \qquad (10.9)$$

$$\sum_{k\in K} d_k x_{ij}^k \ge s_{ij}^0 + \sum_{l=0}^{L-1} (q_{l-1} + 1) s_{ij}^l \qquad \forall (i,j) \in A \qquad (10.10)$$

$$y_{ij} \in \{0,1\} \qquad \forall (i,j) \in A \qquad (10.11)$$

$$s_j^l \in \{0,1\} \qquad \forall (i,j) \in A \qquad (10.12)$$

$$x_{ij}^k \ge 0 \qquad \forall (i,j) \in A, k \in K \qquad (10.13)$$

where

- x_{ij}^k is flow of commodity k on arc (i,j)
- $y_{ij} = \begin{cases} 1, if\ any\ type\ of\ facility\ is\ installed\ on\ arc\ (i,j) \\ 0, otherwise \end{cases}$

(10.1) is the objective function that minimizes the total link costs. Constraints (10.2), (10.3), (10.6) and (10.7) together ensure a directed tree network topology. The total traffic on any arc is less than the capacity of the facility installed on that arc is guaranteed by constraints (10.4) and (10.5). Constraint (10.8) guarantees that only one facility is installed on an arc, if and only if the arc is part of the MLCMST tree. Constraint (10.9) ensures that if the facility at least $q_{l-1}+1$ units [175].

Multicommodity flow formulation is stronger than enhanced single commodity flow formulation [175]. Gamvros et al. report that GAP between the LP relaxation of MCF and its MIP solution is %0.6 less than GAP between the LP relaxation of ESCF and its MIP solution though the running time of MCF is two orders of magnitude greater than ESCF.

iv. Capacity Indexed Model

Martins et al. presented a capacity-indexed formulation for the MLCMST problem [134]. This model was proposed by Gouveia for the CMST problem [154] and was successful in branch-cut and price algorithm proposed in Uchoa et al. [153]. Martins et al. chose this model to solve subproblems in the GRASP heuristic that they propose. In order to use this model it is assumed that $q_i = i$ for $i = 1,...,L$ and the capacities increase from 1 to q_L by unitary increments. If these assumptions do not hold, artificial capacities are created such that the number of different capacities available is set to $P = \overline{Q}^L$, and then $\overline{Q}^1 = 1$, $\overline{Q}^2 = 2$, . . . , $\overline{Q}^P = q_L$.

$$Min \qquad \sum_{p=1}^{P}\sum_{i\in I}\sum_{j\in \tilde{I}} \overline{c}_{ij}^{p} y_{ij}^{p} \qquad (11.1)$$

$$subject\ to$$

$$\sum_{p=1}^{P}\sum_{i\in I} y_{ij}^{p} = 1 \qquad \forall j \in \tilde{I} \qquad (11.2)$$

$$\sum_{p=1}^{P}\sum_{i\in I} \overline{q}_{ij}^{p} y_{ij}^{p} - \sum_{p=1}^{P}\sum_{j\in \tilde{I}} \overline{q}_{ij}^{p} y_{ij}^{p} = d_i \qquad \forall j \in \tilde{I} \qquad (11.3)$$

$$y_{ij}^{p} \in \{0,1\} \qquad \forall i \in I, \forall j \in \tilde{I}, p = 1,...,P \qquad (11.4)$$

where

- $y_{ij}^{l} = \begin{cases} 1, if\ capacity\ l\ is\ installed\ on\ arc\ (i,\ j\ ) \\ 0, otherwise \end{cases}$
- $\overline{c}_{ij}^{p} = \begin{cases} \overline{c}_{ij}^{1}, if\ \ p = 1,...,q_1 \\ \overline{c}_{ij}^{l}, if\ \ p = q_{l-1}+1,...,q_1,\ l = 2,...,L \end{cases}$
- $\overline{q}_{ij}^{p}$ is the capacity installed on arc (i, j) with capacity index p

The objective function (11.1) minimizes the total link cost. Constraint (11.2) is in-degree constraint for an arborescence rooted at center node, $\{0\}$. Constraint (11.3) is the capacity balance constraint. These two constraints together ensure arborescence feasible to the multi-level capacitated minimum spanning tree problem [134].

Multicommodity flow formulation provides tighter linear relaxation bounds than the capacity indexed model. However, the capacity indexed model is shown to be most effective to solve the multi-level CMST problem to optimality [134].

Solution methods proposed for the multi-level capacitated minimum spanning tree problem are presented in Table 10.

Table 10. Solution Methods for the Multi-level Capacitated Minimum Spanning Tree Problem

Paper	Solution Method	Solution Capability
[134]	GRASP using an hybrid heuristic-subproblem optimization approach	The proposed algorithm is tested for the instances that are also used in computational experiments by Gamvros et al. [174]. The algorithm improved the best known solutions of 247 out of 250 problem instances.
[133]	Particle swarm optimization (PSO)	A specific instance of MLMCST problem is introduced and it is solved by global PSO, local PSO and the proposed hybrid PSO.
[175]	The following methods are proposed: -a savings-based construction heuristic is proposed - local search algorithms that use exponential size, node-based, cyclic and path exchange neighborhoods are developed -a hybrid genetic algorithm	The proposed algorithms were tested for problems with 20, 30, 50 and 100 nodes. In addition, construction heuristic and one of local search procedures are tested for problems with 150 nodes whereas genetic algorithm and the other local search procedure cannot be tested because of excessive computation time needed. At the end of computational experiments it is seen that genetic algorithm is robust and the best algorithm among the heuristics.
[174]	Evolutionary algorithm	They tested the algorithm for problems with 50 and 100 nodes and facility capacities of 1,3 and 10. The average GAP for 50 node problems is 9.95% and the GAP is 7.68% for 100 node problems.

3.2 Steiner Tree Problem

The Steiner tree problem involves a graph $G=(I,E)$ with vertex set I and edge set E where the vertices are partitioned into two groups as compulsory vertices V and Steiner vertices $I \setminus V$. The problem is to find a subset of edges such that all compulsory vertices are connected in the partial graph with a minimum total length. The optimal solution to this problem is known as an acyclic graph called Steiner tree.

If the Steiner tree problem literature is considered, two major classes of Steiner tree problems are distinguished [135]:

- Steiner's problem in metric spaces (the Euclidean Steiner problem and the rectilinear Steiner problem)
- Steiner problem in graphs

In telecommunication applications, mainly the graph theoretic version of the Steiner problem is referred. Thus, in this review, Steiner tree problems in graphs are addressed.

The Steiner tree problem is used to formulate local access networks as the minimum spanning tree problems for solving the terminal layout problem. Although the Steiner tree problem is NP-hard, optimal solutions to minimum spanning tree problem are found by Kruskal's algorithm. Because of necessity about the real existence of intermediate nodes between terminals and the root node, the Steiner tree problem is used for local access network design [22].

Mathematical formulations for the Steiner tree problem are surveyed in Goemans and Myung [176], and Polzin and Daneshmand [177]. Polzin and Daneshmand [177] list some frequently used formulations, whose relaxations are used to find lower bounds, provide the relationships among the present formulations and a proposed new formulation. In their review, Polzin and Daneshmand listed basic cut-based, flow-based, and tree-based formulations, as well as a relaxation based on multiple trees and an augmented flow relaxation. We included only the basic formulations of Steiner tree problem. For the rest of review due to Polzin and Daneshmand, the reader is referred to [177].

i. Cut Formulations

The directed cut formulation is due to Wong [178]. The directed cut formulation of the Steiner tree problem is defined on a directed graph $\vec{G} = (\tilde{I}, A, c)$ where vertex set $\tilde{I} = \{1,2,...,n\}$, the edge set is $E = \{\{i,j\} \mid i \in I, j \in I\}$ and edge weights are $c_{ij} = c((i,j)) > 0$. V is the set of compulsory vertices. The arc set is defined as $A = \{(i,j); (j,i) \mid \{i,j\} \in E\}$. The problem that uses the directed graph $\vec{G} = (\tilde{I}, A, c)$ is to find minimum weight arborescence with a terminal i.e. v_1 as the root that spans $V_1 = V \backslash v_1$.

The decision variable is a binary variable such that

$$y_{ij} = \begin{cases} 1, \; if \; (i,j) \, is \, in \, optimal \, solution \\ 0, \; otherwise \end{cases}.$$

$$Minimize \qquad \sum_{(i,j)\in A} c_{ij} y_{ij} \tag{12.1}$$

$$subject \; to$$

$$\sum_{(i,j)\in\{(i,j): i\in \tilde{I}\backslash S, j\in S\}} y_{ij} \geq 1 \qquad S \subseteq \tilde{I}, v_1 \notin S, S \cap V_1 = \varnothing \tag{12.2}$$

$$y_{ij} \in \{0,1\} \qquad \forall (i,j) \in A \tag{12.3}$$

The objective function (12.1) minimizes the total link cost. Constraint (12.2) is called the Steiner cut constraint that guarantees that in any arc set corresponding to a feasible solution, there is a path from v_1 to any other terminal [177].

The undirected cut formulation is due to Aneja [179]. It is defined on a graph $G = (\tilde{I}, E, c)$. The decision variable is a binary variable such that

$$y_{ij} = \begin{cases} 1, \; if \; \{i,j\} \, is \, in \, optimal \, solution \\ 0, \; otherwise \end{cases}.$$

$$\textit{Minimize} \sum_{\{i,j\}\in E} c_{ij} y_{ij} \tag{13.1}$$

subject to

$$\sum_{\{i,j\}\in\{\{i,j\}:i\in S, j\in \tilde{I}\setminus S\}} y_{ij} \geq 1 \quad S \subseteq \tilde{I}, S\cap V = V, \tilde{I}\setminus S \cap V \neq \varnothing \tag{13.2}$$

$$y_{ij} \in \{0,1\} \qquad \{i,j\}\in E \tag{13.3}$$

The objective function (13.1) and the constraint (13.2) are the undirected versions of (12.1) and (12.2).

ii. Multicommodity Flow Formulation

Multicommodity flow formulation is due to Wong in [178] which the quantity of the commodity k flowing through arc (i,j) is denoted by variable x_{ij}^k.

$$\textit{Minimize} \sum_{(i,j)\in A} c_{ij} y_{ij} \tag{14.1}$$

subject to

$$\sum_{(i,j)\in A} x_{ji}^k - \sum_{(i,j)\in A} x_{ij}^k = \begin{cases} 1, & v_t \in V_1;\ i = v_t \\ 0, & v_t \in V_1;\ i \in \tilde{I}\setminus\{v_1, v_t\} \end{cases} \tag{14.2}$$

$$y_{ij} \geq x_{ij}^k \qquad v_t \in V_1;\ (i,j)\in A \tag{14.3}$$

$$x_{ij}^k \geq 0 \qquad v_t \in V_1;\ (i,j)\in A \tag{14.4}$$

$$y_{ij} \in \{0,1\} \qquad (i,j)\in A \tag{14.5}$$

For each terminal $v_t \in V_1$, there is a flow of one unit of commodity t from v_1 to v_t and it is guaranteed by constraints (14.2) and (14.4). There is a path from v_1 to any other terminal in any arc set corresponding to a feasible solution and it is guaranteed by (14.2), (14.3) and (14.4) [177].

iii. Tree Formulations

The tree formulations are defined on a modified graph $G_0 = (I, E_0, c_0)$ that is formed by adding a new vertex 0 to vertex set $\tilde{I}$, connecting it to all vertices in $\tilde{I}\setminus V$ with zero cost edges and a fixed terminal vertex, i.e., v_1. The problem is to find a minimum spanning tree T_0 in G_0 such that every vertex in $\tilde{I}\setminus V$ adjacent to $\{0\}$ must have a degree of one [177]. Degree constrained tree formulation is due to Beasley [180].

$$\textit{Minimize} \sum_{\{i,j\}\in E} c_{ij} y_{ij} \tag{15.1}$$

subject to

$$\sum_{\{i,j\}\in E_0} y_{ij} = N \tag{15.2}$$

$$y_{0k} + y_{ki} \leq 1 \qquad k \in \tilde{I}\setminus V,\ \{k,i\}\in\{\{i,j\}: j\in \tilde{I}\} \tag{15.3}$$

$$\sum_{\substack{\{i,j\}\in E_0;\\ \{i,j\}\in S}} y_{ij} \leq |S| - 1 \qquad \varnothing \neq S \subseteq \tilde{I} \tag{15.4}$$

$$y_{ij} \in \{0,1\} \qquad \{i,j\}\in E_0 \tag{15.5}$$

The objective function (15.1) minimizes the total link cost. Constraints (15.2) - (15.5) together find a spanning tree T_0 in G_0 such that every vertex in $\tilde{I}\setminus C$ adjacent to 0 have a degree of one.

In addition directed version of the directed tree formulation and a rooted tree formulation is presented in Polzin and Daneshmand [177].

A hierarchy of linear programming relaxations of Steiner Tree problem formulations is presented in Polzin and Daneshmand. They proved that flow-class relaxation cannot be worse than the optimal value of a tree-class relaxation. As Polzin and Daneshmand report in[177], flow formulation and directed cut formulation are strictly stronger than undirected cut formulation, degree constrained tree and rooted tree formulations. They also report that rooted tree and degree constrained tree formulations are equivalent as well as flow formulation and directed cut formulation.

The Steiner tree problem is a special case network design problem. It can be formulated as a minimum cost multi-terminal (single commodity) network design problem. In order to formulate the Steiner tree problem as a multi-terminal (single commodity) flow problem, any node from the compulsory nodes is selected as the common source for all of the requirements. The other compulsory nodes are defined as sinks for the one source-one sink flows from the selected compulsory node to other nodes and the amount of flow is set equal to one. In addition, each edge is assigned a cost function with a fixed cost of edge length and a linear cost of zero. The optimal solution to the presented minimum cost flow problem formulation is the solution of the Steiner tree problem [21].

Reduction techniques are used to simplify the problem instances before solving the Steiner tree problems. A recent study of Polzin and Daneshmand [181] reviews the main algorithmic developments for the Steiner Tree problem.

The reduction techniques are proposed in Duin and Volgenant [182], [183]and in Balakrishnan and Patel [184]. More recent results for reduction techniques are presented in Uchoa et al. [153], [185], Duin [186], Polzin and Daneshmand [187], [188] and Polzin [189].

The reduction techniques are classified into two groups in Polzin and Daneshmand [188] as alternative-based and bound-based. Polzin and Daneshmand proposed partitioning as reduction technique [177] and they proposed extended reduction techniques [188].
Exact solution methods include branch and cut techniques [180], [190] and Lagrangian relaxation [191], Polzin and Daneshmand [192] use the directed cut formulation of Aneja [179] and Wong [178] in a dual ascent fashion with reduction techniques. They tested the algorithm with instances that are solved by other authors and not solved from Steinlib. They report that, for the solved instances, the algorithm is faster than the other algorithms by an order of magnitude and they solved 32 of the unsolved 73 instances [189], [193]. In addition, Fuchs et al. present dynamic programming algorithm for Steiner tree problem [194].

Heuristics proposed to solve Steiner tree problem is surveyed in Voß [195] and a classification scheme of the heuristics is presented in Voß [135]. Voß mentions the Steiner tree heuristics use the two main ideas that arise from two famous minimum spanning tree algorithms [135]. The ideas are as follows:

- Single component extension (corresponds to Prim's Algorithm for the MST): Start with a partial solution consisting of a single vertex i.e., root which is arbitrarily selected from the compulsory vertices V. Extend the initial partial solution to a feasible solution by inserting at most |V| shortest paths to compulsory vertices.
- Component Connecting (corresponds to Kruskal's Algorithm for the MST): Start with an initial solution consisting of compulsory edges only. Expand the initial partial solution to a feasible solution by repeatedly selecting components which are connected by shortest paths.

In addition to the classification scheme based on two main ideas, the algorithms differ in the number of build up steps in order to find feasible solution i.e., 1BASIC algorithms makes extension by adding single component each time, while kBASIC algorithms add k components at a time.

Voß presents a heuristic measure algorithm that is a unified approach based on component connecting idea [135]:

HEUM:
(1) Start with T = (V, Ø) comprising |V| basic vertices (sub trees of the final subgraph).
(2) While T is not connected,
*do choose a vertex v using a function **f** and unite the two components of T which are nearest to v by combining them with v via shortest paths (the vertices and edges of these paths are added to T).*

Changing the function ***f*** used in HEUM leads different construction heuristics:

- Shortest Distance Graph Heuristic (SDISTG)

$$f_1(v) = \min_{1 \le i \le \sigma} \left\{ d(v, T_0 \cap V) + d(v, T_i \cap V) \right\}$$

- Cheapest Insertion Heuristic (CHINS)

$$f_2(v) = \min_{1 \le i \le \sigma} \left\{ d(v, T_0) + d(v, T_i) \right\}$$

Variations of SDISTG and CHINS that are applied by 1BASIC or kBASIC with a single pass or multiple pass to calculate a feasible solution exist in the literature.

An improvement heuristic called minimum spanning tree and pruning is also presented in [135]:

MST+P: Minimum spanning tree and pruning
(1) Given a solution with vertex set V_T, construct an MST $T' = \{V_T', E_T')$ of the subgraph of G induced by V_T.
(2) While there exists a leaf of T' that is a Steiner vertex, do delete that leaf and its incident edge.

The heuristics are problematic since they cannot continue the search once they are trapped to a local optimum. In order to solve this problem, metaheuristics are also proposed to solve the Steiner tree problem.

- The pilot method is applied to the Steiner tree problem in Duin and Voß [196].
- Local search is used for solving the Steiner tree problem and the neighborhood definition used in local search procedures can be separated into two groups regarding the main ideas used [197]:

- edge oriented transformation
- node-oriented transformation

- Local search and population based metaheuristics also proposed for solving the Steiner tree problem:
    - GRASP (greedy randomized adaptive search procedure): Martins et al. [198] and Ribeiro et al. [199]

As a result, the Steiner tree problem is a well-studied problem. The computational experiments use the SteinLib library and the recent solutions of the Steiner tree instances can be accessed from http://steinlib.zib.de//steinlib.php.

3.3 Minimum Cost Single Commodity Flow Problem

The minimum cost single network flow problem is defined on a graph $G=(I,A)$, where I is the set of nodes and A is the set of arcs. There is a single commodity which originates from a source node s and terminates at a sink node t. d is the amount of flow that is to be transmitted from source node to sink node. The minimum cost single network flow problem is to find the minimum cost network that can transmit the demand for the commodity from its source node to sink node within the capacity constraints by determining which nodes to be connected by links and determining the links' capacity.

The formulation of the problem is as follows:

$$\textit{Minimize} \qquad C(x)=\sum_{(i,j)\in A} G(x_{ij}) \qquad (16.1)$$

subject to

$$\sum_{j\in I} x_{ij} - \sum_{j\in I} x_{ji} = \begin{cases} d, & i=s \\ -d, & i=t \\ 0, & otherwise \end{cases} \qquad \forall i \in I \qquad (16.2)$$

$$x_{ij} \le q_{ij} \qquad \forall (i,j)\in A \qquad (16.3)$$

$$x_{ij} \ge 0 \qquad \forall (i,j)\in A \qquad (16.4)$$

where nonnegative variable x_{ij} is the amount of flow on the arc (i,j).

The objective function (16.1) minimizes the total link cost. Constraints (16.3) ensures that flow on each link is less than or equal to the link's capacity. Constraint (16.2) is the flow conservation constraint.

3.3.1 The Multi-terminal Network Flow Problem with Heterogeneous Terminals

The main characteristic that distinguishes the multi-terminal flow problem from other type of problems is that the multi-terminal problem involves heterogeneous node and link types in terms of their capacity and cost.

The multi-terminal network flow problem can be thought as a multicommodity flow problem if the center is considered as the center being the source for all commodities and the other terminals are the sinks, i.e., if there are n nodes, the number of commodities is at most n-1. The problem can be reduced to a single commodity flow problem with a common source and different sinks [21].

The multi-terminal network flow problem addresses telecommunication networks where the terminals are different from each other in terms of traffic they generate, while the communication lines can vary in capacity and cost.

Multi-terminal network flow model is the typical model that arises when modeling centralized data processing networks. It is also related to almost all problem types told in this working paper. The relationships are given below:

- The case of stair case cost functions or piecewise linear cost functions with discontinuities: the Telpak problem and one terminal Telpak problem. Since the Telpak problem is the capacitated case of the minimum cost network flow problem with link costs, the multi-terminal network flow problem is also related with optimum rented lines problem.
- Concave cost function: this constitutes the more general and complex case and it is also called the minimum concave cost single commodity flow problem.
- Linear cost function with fixed cost case where linear cost is a function of flow and the fixed cost is the cost of installing links: the minimum cost fixed charge network flow problem, optimum network problem.
- The case with variable cost that depends on the flow is zero and the fixed cost of installing a link is equal to its length can be used to formulate the Steiner tree problem when the flow from any compulsory node to all other compulsory nodes are taken to be one.

3.3.2 Telpak Problem

Telpak problem is the multi-terminal network flow problem with heterogeneous terminals where the cost function is a staircase function or piecewise linear function with discontinuities [21].

In practice, terminals and end users of telecommunication systems are not homogeneous in terms of the traffic they generate and receive. When the terminals are identical in terms of the traffic, requirement for capacity of links is higher when it gets closer to the center. The links are available in different capacities and costs, i.e., the link cost vary according to its capacity [10]. The cost function used in Rothfarb and Goldstein [36] is show in Figure 2. As it is seen in the figure, the cost function of links is a staircase function, which represents the bulk units of flow as for some capacities the cost remains the same as capacity increases and economies of scale is observed as the additional cost per capacity decreases as capacity increase.

Telpak problem addresses network design problem in which there is a center and the terminals are to be connected to the center with a minimum cost network that meets the requirements where the cost of links depend on the traffic flow on each link as the cost of links vary with its capacity.

Gavish tells that Telpak problem is fundamental design problem in the local distribution system of telephone systems and it is especially important for local access networks [10].

Telpak problem is also important as it links some network design problems such that it can be formulated as a capacitated minimum spanning tree problem [10], its more general case is the minimum concave cost flow problem while its more specific case is the fixed charge network flow problem [21].

Telpak problem is treated as the multi-level capacitated minimum spanning tree problem by Gamvros et al. [174] and the most recent studies on this problem are done for the MLCMST problem.

The latest formulation (if we disregard the formulations done for the same problem with the name MLCMST), which is similar to the capacitated minimum spanning tree problem, is given in Gavish [10]:

$$Minimize \qquad \sum_{i=1}^{N}\sum_{\substack{j=0\\ i\neq j}}^{N} f_{ij}y_{ij} + \sum_{i=1}^{N}\sum_{\substack{j=0\\ i\neq j}}^{N} F_{ij}(x_{ij}) \tag{17.1}$$

subject to

$$\sum_{\substack{j=0\\ i\neq j}}^{N} y_{ij} = 1 \qquad i = 2,\dots,N \tag{17.2}$$

$$\sum_{\substack{j=0\\ i\neq j}}^{N} x_{ij} - \sum_{\substack{j=1\\ i\neq j}}^{N} x_{ji} = d_i \qquad i = 2,\dots,N \tag{17.3}$$

$$x_{ij} \le q_{ij}y_{ij} \qquad i = 1,\dots,N\,,\; j = 1,\dots,N \tag{17.4}$$

$$x_{i1} \le qy_{i1} \qquad i = 1,\dots,N \tag{17.5}$$

$$x_{ij} \ge 0 \qquad i = 1,\dots,N\,,\; j = 1,\dots,N \tag{17.6}$$

$$y_{ij} \in \{0,1\} \qquad i = 1,\dots,N\,,\; j = 1,\dots,N \tag{17.7}$$

where

- d_i = the traffic generated by terminal i, $d_i > 0$, for $\forall i \in I$
- $F_{ij}(y)$ =the minimal cost of a line (or line configuration) which connects nodes i and j and accommodates a traffic volume x_{ij}.
- q_{ij} =upper limit on line capacity between nodes i and j.
- Q = upper limit on the capacity of each port on the central switch
- f_{ij} = fixed setup cost for connecting nodes i and j .
- $y_{ij} = \begin{cases} 1, & if\ link\ (i,j)\ is\ used\ in\ the\ design \\ 0, & otherwise \end{cases}$
- x_{ij} = variable which represents the amount of flow on link (i,j)

In the formulation it is assumed that $F_{ij}(x) > 0$ for y > 0; it is $F_{ij}(x) = 0$ for $x = 0$.
Constraint (17.2) guarantees that each node has one link connecting it to any other node; (17.3) is flow conservation constraint, the flow on link (i,j) to zero if link (i,j) is not selected to be used in the design is restricted by (17.4) and (17.5) limits the amount of traffic handled by each port to less than the port capacity.

3.3.3 One Terminal Telpak Problem

One terminal Telpak problem is presented in Rothfarb and Goldstein [36] and has attracted much attention since then.

The problem is the special case of Telpak problem where only one type of terminal exists. Rothfarb and Goldstein considered only the traffic flow from satellite locations to a common facility. Even if there is only one type of terminal, i.e. the traffic requirements do not vary, the cost of network is still flow dependent as it is mentioned earlier that capacity need on links increases when the terminal location gets closer to the center. The problem is then can be formulated as a single commodity flow problem. The cost function is given in Figure 2.

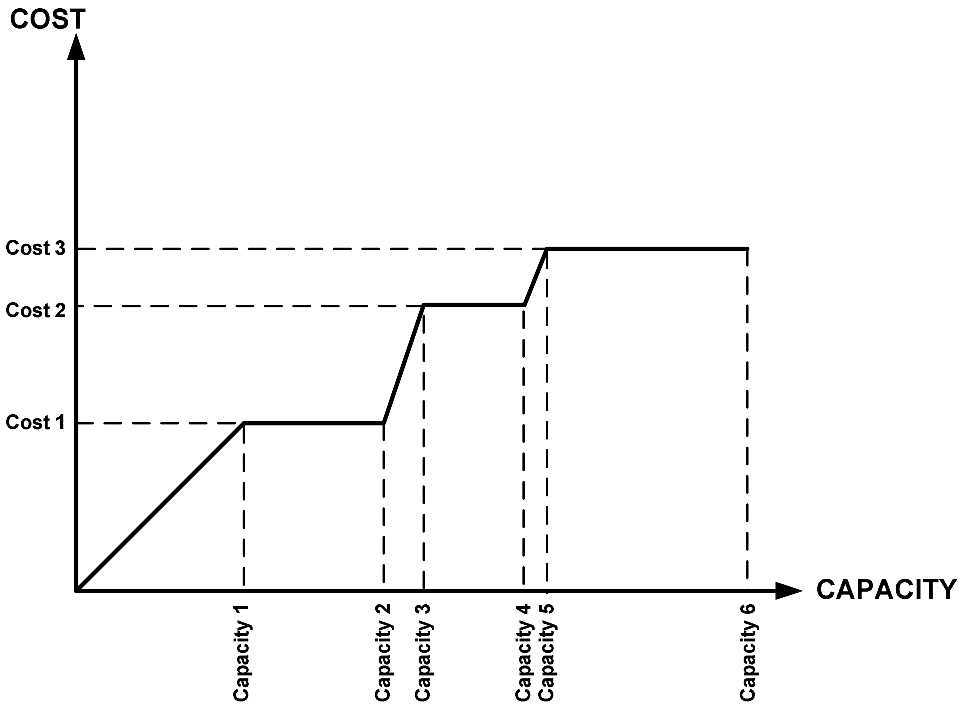


Figure 2. Cost Function of the One Terminal Telpak Problem [21]

3.4 Multicommodity Minimum Cost Network Flow Problem

The multicommodity minimum cost network flow problem is related to optimal design and dimensioning of telecommunication networks. The basic problem can be stated as given a list of traffic nodes and anticipated values for the volume of traffic between nodes, to build a network connecting sources and sinks which can handle the traffic flow requirements. Solving the basic problem provides a joint solution of the network topology and dimensioning problems [143].

The minimum cost multicommodity network flow problem is defined on a graph $G = (I, A)$, where I is the set of nodes and A is the set of arcs. The set of commodities is notated as K. Each commodity $k \in K$ originates from a source node denoted by $s(k)$ and terminates at a sink node denoted by $t(k)$. d_k is the amount of flow that is to be transmitted from source node to sink node for each commodity $k \in K$. The minimum cost multicommodity network flow problem is to find the minimum cost network that can transmit the demand for each commodity from their source nodes to sink nodes within the capacity constraints by determining which nodes to be connected by links and determining the links' capacity. The formulation of the problem is as follows:

$$Minimize \qquad C(x) = \sum_{k \in K} \sum_{(i,j) \in A} G(x_{ij}^k) \qquad (18.1)$$

$$subject\ to$$

$$\sum_{j \in I} x_{ij}^k - \sum_{j \in I} x_{ji}^k = \begin{cases} d_k, \ i = s(k) \\ -d_k, \ i = t(k) \\ 0, otherwise \end{cases} \qquad \forall k \in K, \forall i \in I \qquad (18.2)$$

$$\sum_{k \in K} x_{ij}^k \le q_{ij} \qquad \forall (i,j) \in A \qquad (18.3)$$

$$x_{ij}^k \ge 0 \qquad \forall (i,j) \in A, \ \forall k \in K \qquad (18.4)$$

where nonnegative variable x_{ij}^k is the amount of flow of commodity k, on the arc (i, j).

The objective function (18.1) minimizes the total link cost. Constraints (18.3) ensures that flow on each link is less than or equal to the link's capacity. Constraint (18.2) is the flow conservation constraint.

The MCMCF with switching equipment cost may be reformulated as MCMCF if each node in the graph is split into two nodes and two arcs are added between these two nodes with the cost of the original node [142].

The practical difficulty in solving the minimum cost multicommodity flow problem depends on the objective function. If it is acceptable to model the cost function by a linear cost function, the problem can even be reduced to shortest path problem which can be solved easily. However, if a more accurate cost function is needed, a step increasing cost function is used leading more difficult problems to solve in practice [143]. MCMCF problem can be classified into four groups according to the cost function used [143].

3.4.1 Linear Cost Function Case

Each cost function is of the following form then the MCMCF is formulated as a linear program.

$$G(x_{ij}^k) = c_{ij} \sum_{(i,j) \in A} x_{ij}^k \ with \ c_{ij} \ge 0$$

The solution methods of the MCMCF problem up to 2006 are summarized by Minoux [143] and up to 2009 are summarized by Kramer [200]. The solution techniques are classified in Kramer [200] as direct and decomposition methods. Direct methods are partitioning methods [201] and [202] and interior point methods [203–209]. Decomposition methods are also divided into groups as Dantzig-Wolfe decomposition [210–213], resource directive decomposition [214] and bundle based decomposition methods [215–217]. The Resource Directive Decomposition differs from Dantzig-Wolfe decomposition as it uses a variable to allocate the bundle resources for each commodity instead of pricing out which extreme points to include. Bundle decomposition is a specialized dual ascent method. In addition, augmented Lagrangian algorithm [218] and column generation [219] are used to solve the minimum cost multicommodity flow problem with linear cost.

One of the recent studies on linear minimum cost multicommodity flow problem is due to Larsson and Yuan [218]. They proposed an augmented Lagrangian algorithm that is a combined method of Lagrangian relaxation and penalty approach. They report that the augmented Lagrangian algorithm provides near optimal solutions (relative accuracy of 0.2%) to instances with over 3600 nodes, 14000arcs and 80000 commodities within reasonable computation times. Experimental study of [218] presents a comparison of primal partitioning code due to Castro and Nabona [220], bundle method due to Frangioni and Gallo [204] and Frangioni [216] and Dantzig-Wolfe decomposition.

3.4.2 Linear With Fixed Cost Case (Minimum Cost Fixed Charge Multicommodity Network Design Problem)

The cost function in this case is:

$$G\left(\sum_{(i,j)\in A} x_{ij}^k\right) = \begin{cases} 0, & if \sum_{(i,j)\in A} x_{ij}^k = 0 \\ f_{ij} + c_{ij} \sum_{(i,j)\in A} x_{ij}^k, & if \sum_{(i,j)\in A} x_{ij}^k > 0 \end{cases}$$

Although, this function is relevant to approximate more general cost functions and can capture economies of scale, the problem is NP-hard [21]. The single commodity case of this problem is also studied by Ortega and Wolsey [221] and Rardin and Wolsey [222].

Lagrangian relaxations of the capacitated minimum cost fixed charge multicommodity network design problems is presented by Gendron, Crainic and Frangioni [139]. They make computational experiments to compare branch and bound, weak relaxation of fixed charge problem, strong relaxation of fixed charge problem, tabu search due to Crainic, Gendreau and Farvolden [223] and the Lagrangian based resource decomposition heuristic that they proposed. They conclude that Lagrangian based resource decomposition heuristic outperforms the others and gaps are mostly in acceptable ranges (i.e. for 30 vertices, 700 edges and 100 commodities – 5.41% gap), however, some gaps are very large (i.e. for 30 vertices, 700 edges and 400 commodities – 18.56% gap). It is observed from the computational tests that the proposed algorithm gives worse results as the number of commodities increase.

Crainic, Frangioni and Gendron [224] surveyed Lagrangian based bounding methods multicommodity capacitated fixed charge problem. They combined different Lagrangian relaxations (shortest path relaxation and Knapsack relaxation) with bundle or subgradient method and compare the results of the methods. They conclude that when the number of commodities increases, CPLEX with nearopt option cannot solve the problem due to memory problems i.e., 400 commodities cannot be solved. They pointed out that subgradient method converges slowly while bundle method achieves the most progress in the first iterations.

Chouman, Crainic and Gendron [225] proposed a cutting plane algorithm for the problem. They improved the mixed integer programming model of the multicommodity capacitated fixed charge problem by adding valid inequalities. They make extensive computational experiments to compare the valid inequalities, the procedures to generate cut sets, alternative cutting plane algorithms and performance of the proposed cutting plane algorithm with state-of-the-art software CPLEX that implements branch and bound. They concluded that proposed algorithm outperformed CPLEX.

3.4.3 Piece-wise Linear Concave Cost Function Case

The cost function is $G(x_{ij}^k) = F_{ij}(x_{ij}^k)^\alpha \quad where\ 0 \leq \alpha \leq 1$ which is closer to reality than linear cost function and linear with fixed cost case. However, the MCMCF problems with concave cost functions of the given form are very hard to solve. The problem cannot be solved exactly except for small instances which have less than 15 nodes, 15 commodities and 10 possible paths for each commodity [143].

Say and Bazlamacci present a survey of the solution techniques of the problem by 2007 [226], [227]. Some of important techniques presented are Minoux's accelerated greedy algorithm [21], Yaged's linearization algorithm [228] and concave branch elimination due to Gerla and Kleinrock [48]. Bazlamacci and Hindi proposed extreme-point search and tabu search algorithms to solve the problem [229].

3.4.4 Step-Increasing Cost Function Case

A typical step increasing cost function is of the form [143]:

$$G\left(\sum_{(i,j)\in A} x_{ij}^k\right) = \begin{cases} 0,\ if \sum\limits_{(i,j)\in A} x_{ij}^k = 0 \\ c_{ij}^r, if\ \forall \sum\limits_{(i,j)\in A} x_{ij}^k \in \left]Q_{ij}^{r-1}, Q_{ij}^r\right], \forall r = 1,..., R(ij) \end{cases}$$

where $Q = \{q_{ij}^0, q_{ij}^1, ..., q_{ij}^{R(ij)}\}$ is defined as a finite set of capacities, where $R(ij)$ is the number of discrete capacities that can be installed on link $\{i,j\}, i \in I\ and\ j \in I$, is and the step increasing cost function on link $\{i,j\}, i \in I\ and\ j \in I$ is specified on the set Q such that $c_{ij}^0 = G(Q_{ij}^0)$, $c_{ij}^1 = G(Q_{ij}^1)$, ..., $c_{ij}^R = G(Q_{ij}^R)$ with $0 = c_{ij}^0 \leq c_{ij}^1 \leq ... \leq c_{ij}^R$ and $0 = q_{ij}^0 \leq q_{ij}^1 \leq ... \leq q_{ij}^R = q_{ij}$.

Note that, the cost function involves only fixed installation costs of the links and the flow cost is not included. This is a special case of the minimum cost capacitated network design problem which is called the *network loading problem*. The associated telecommunication network design problem is to find the minimum cost network that can meet the given point-to-point communication demand by installing capacitated facilities on the arcs. The problem arises especially when installing private networks that need to lease lines from private companies that are exclusive to their network in a bulk manner. Each facility's cost depends on its capacity [63] and [64]. Some special cases of this problem are studied according to the number of facilities used, i.e., the facilities are differentiated according to their capacities and links are to be installed with integer multiples of these capacities, thus ν_u in the cost function are integer multiples of these capacity values [21]. Single facility case is studied by Magnanti and Mirchandani [230] and Barahona [231]. Two-facility case is studied by Magnanti et al. [63] and three-facility case is discussed by Magnanti and

Mirchandani [230].

The *optimum rented lines network problem* is presented by Minoux [21] addresses the same problem as network loading problem such that

> "Suppose that a large communication company needs traffic exchange between plants and services scattered on a large area, i.e. all over one or more countries and the traffic requirement matrix is given. In order for the company to meet these requirements, the company can either use public transmission network that has linear costs proportional to the amount of the traffic flow or use a network of rented lines with a fixed cost depending on its capacity. The optimum solution is a combination of both choices since it is advantageous to use public transmission lines where low traffic requirements exist and use rented lines elsewhere."

The optimum rented lines problem can be formulated as the minimum cost multicommodity flow problem with link costs if the capacity of rented lines is ignored, i.e., capacity of rented lines are large enough that capacity constraints can never be binding. The more general case where more than one transmission types exist can still be formulated as the minimum cost multicommodity flow problem with link costs. The cost function is given in Figure 3, where the resemblance to the Telpak problem cost function is obvious [21].

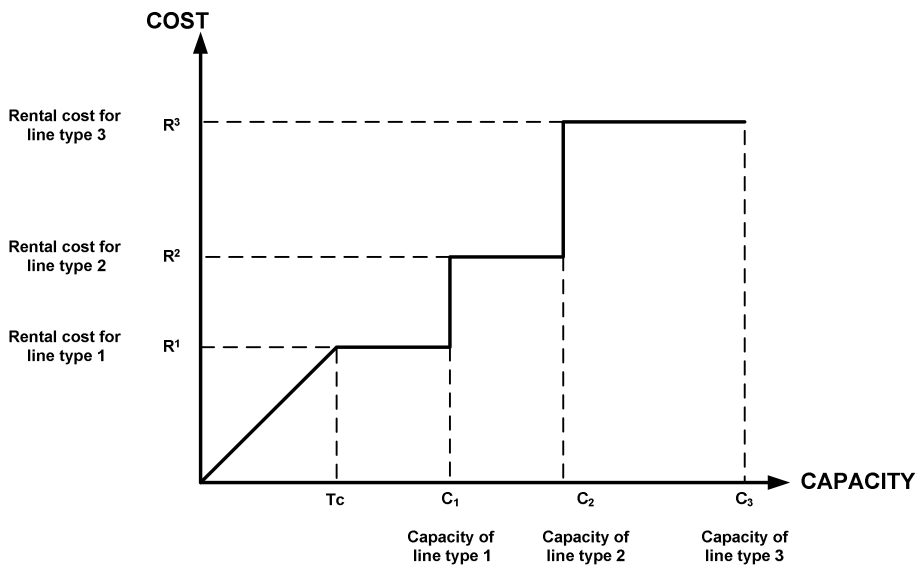


Figure 3. Cost Function of the Optimum Rented Lines Problem [21]

The exact solution methods for the minimum cost multicommodity problems with step-cost functions are surveyed by Minoux [143] and [142]. In addition, in [64] Altin gives a discussion of particular studies related to the network loading problem. According to these studies, common approach to such problems is to formulate the problem as a mixed integer linear programming, define strong valid inequalities in order to strengthen their linear programming relaxations and apply a branch and bound or a branch and cut approach to solve them. On the other hand, when a general increasing step cost function is used some other methods are proposed. One of them is due to Gabrel et al. [124]. The method is a specialization of Bender's decomposition. Gabrel et al. [124] and Minoux [142] formulated the general case of this problem as a pure 0-1 integer programming problem. For general case, heuristic methods and approximation methods are also investigated. A survey and a comparison of heuristic solutions of the problem are presented by Gabrel, Knippel and Minoux [232].

In the literature, the demand matrix is taken to be deterministic as Altin [64] reported, however, Altin uses uncertainty in demand matrix in this study.

3.4.5 Nonlinear Convex Cost Function Case

The nonlinear convex cost function case has implementations in telecommunications. The cost function has two components such that total cost is sum of installation costs of links and flow costs of each commodity on links. The difference from linear with fixed cost case is that the cost components are nonlinear convex functions.

The solution methods proposed are surveyed by Ourou et al. [144]. The methods used to solve the problem are flow deviation method, projection method, cutting plane method and proximal decomposition method.

3.5 Survivable Network Design Problem

Survivability is one of the main concerns in backbone network design problem. Survivability of a telecommunication network is defined as its ability to remain operational after some of network components fail. This ability is closely related to the components' reliability as well as topology of the network, i.e. how these components are linked to others and the management of the network, i.e. existence of a re-routing mechanism after a failure. In this section, we are interested in how to obtain a survivable network through the design by considering connectivity among nodes, protection and restoration mechanisms.

If the telecommunication network design involves only topological design of the network, i.e. which nodes should be selected and how they should be linked, then the topological survivability is in question. The topological survivability is obtained by satisfying some connectivity constraints in order to guarantee existence more than a specific number of edge-disjoint or node-disjoint paths between origin and destination pairs. Topological survivability does not involve re-establishment of resources in the failure cases since the topology design problem does not involve routing and capacity related decisions. Protection and restoration mechanisms differ from the topological design in that sense as protection and restoration design involves dimensioning of the network as well as routing decisions. Protection and restoration are the mechanisms that are used to maintain robustness of the network during failure states. Protection involves determining re-establishment of the network when a failure occurs before it happens by dedicating some spare capacity to be used in the case of failure. Restoration differs from the protection as restoration makes re-establishments by rerouting affected demands after failure. Note that protection is more expensive than restoration in terms of spare capacity but restoration needs to be managed during network operation. There are several types of protection and restoration which are surveyed by Pioro and Medhi [118]together with their applicability to several telecommunication technologies. In this study, we try to present some of the basic mathematical programming formulations related to basic methods used to achieve a survivable network. The surveys due to Grotschel et. al. [147], Kerivin and Mahjoub [141], Fortz and Labbe [233] and Medhi [20] can be viewed for detailed review of the problem, its special cases, polyhedral properties and solution methods.

Before going through the integer programming models for survivability, some definitions are presented. Consider an undirected graph $G = (I, E)$ where I is the set of nodes and E is the set of arcs. Define an *st-path* as a sequence of nodes and edges as $P_{st} = (i_0, e_1, i_1, e_2, ..., e_l, i_l)$ where $l \geq 1$, $(i_0, i_1, ..., i_l)$ are distinct nodes, e_k is an edge between nodes i_{k-1} and i_k and $i_0 = s$, $i_l = t$. The *st paths* $P_{st1}, P_{st2}, ..., P_{stl}$ are called node-disjoint if they do not include any common node other than nodes s and t, and they are called edge-disjoint if they do not share any common edge [141], [147]. c_{ij} is the cost for installing edge $\{i, j\}$ where $\{i, j\} \in E$.

Survivable network design problem (SNDP) is stated as finding the minimum cost network by selecting edges that satisfy the node-survivability and/or edge-survivability requirements stated in terms of node- connectivity and/or edge-connectivity. A general model for SNDP related to the node and edge connectivity is introduced by Grotshcel and Monma [68]. The modeled survivability requirements are stated in terms of three nonnegative integers r_{st}, rn_{st} and re_{st} defined for each pair of nodes as the designed network has (i) at least r_{st} of edge-disjoint st paths, and (ii) removal of at most rn_{st} nodes from $N \setminus \{s, t\}$ must leave at least re_{st} edge-disjoint st paths. This problem is discussed and surveyed by Grotshcel et.al. [147], who state that the model is too general to be used practically as it involves many restrictions simultaneously and in practice it cannot be applied because of lack of data. A less general model which is proposed by Grotschel et. al. [147] and surveyed by Kerivin and Mahjoub [141] uses the *connectivity type vectors.* A connectivity vector is notated as r and node-survivable network design problem-NSNDP (link-survivable network design problem-LSNDP) is defined as finding the minimum cost network by selecting edges such that the network contains at least $min\{r(s), r(t)\}$ node-disjoint (edge-disjoint) st-paths.

Steiner tree problem is a special case of the SNDP where for each s and t pair of compulsory nodes (previously notated as set V) $r_{st} = 1$ and $r_{st} = 0$ otherwise [141], [147]. Then, integer programming formulation of the LSNDP problem is obtained from the undirected cut formulation of the Steiner tree problem given with 13.1, 13.2 and 13.3 by

- adding the following constraint

  $$0 \leq y_{ij} \leq 1 \qquad \{i, j\} \in E \qquad (19.1)$$

- and changing right hand side value of 13.2 from 1 to

  $$min\left\{max\left\{r(i) \mid i \in S\right\}, max\left\{r(i) \mid i \in I \setminus S\right\}\right\}$$

The NSDP formulation is obtained by adding the following cut-set inequalities given by 20.1 to the LSNDP formulation:

$$\sum_{(i,j) \in \left\{(i,j): i \in \left(I \setminus U\right) \setminus S, j \in S\right\}} y_{ij} \geq r(s,t) - \left|U\right| \qquad \forall s,t \in I, s \neq \text{t}, \forall \varnothing \neq U \subseteq I \setminus \{s,t\}, \; \left|U\right| < r(s,t), \forall S \subseteq I \setminus U \qquad (20.1)$$

In order to make the connectivity constraints effective during operation, the traffic should flow in such a way that can benefit from existence of node-disjoint and edge-disjoint paths between origin and destination nodes. This can be provided by either diversification or reservation mechanisms. When diversification is used, only a certain percentage of flow is sent via a single path, hence in the case of single node or single path failure, a only a certain percentage of flow is failed to be sent to the destination node. In order to use diversification, routing must be an integral part of the telecommunication network design problem [145]. Since diversification involves precautions before failure happens, it can be thought as a protection mechanism. If capacity decisions are part of the telecommunication network design problem with routing decisions, reservation is used. Reservation involves adding network some spare capacity for flow between each origin and destination pairs in order to be used when a failure happens. Reservation provides survivability by assigning a percentage of the flow between each origin and destination pair to different edge-disjoint or node-disjoint paths in such a way that the network survives when any of the pre-defined possible failure states. Since reservation involves re-establishment of the connections after failure happens, it is a restoration mechanism.

3.6 Multi-tier Tree Problem (MTT)

In order to solve the multi-level telecommunication network topology design problem with tree-tree topology, multi-tier tree problems is used. The MTT problem is called multi-level network design problem (MLNDP) in some of the studies and it is a generalization of the Steiner tree. Note that the problem only solves the topology design problem with deciding which type of facility should be used on edges, but dimensioning of the network is not addressed by the problem since it does not account for edge capacities and traffic demands [74], [87–89], [92].

Facilities with higher capacity and higher transmission rates are called higher grade facilities. If there are T types of facilities, i.e. T grades, a network $G = (N, E)$ whose nodes are partitioned into T distinct nonempty sets such that $N_{t_1} \cap N_{t_2} = \varnothing \mid t_1, t_2 \in T$ and $\cup_{1 \le t \le T} N_t = N$ is given where nodes in N_t are called t-tier nodes and grade t facility is used for these nodes. Installation costs of edges differ according to the facility type selected. The MTT problem is to find the minimum cost network configuration by choosing grades to the edges such that an edge's grade must be at least the lowest grade of its neighbor nodes [92].

If there are only two types of facilities as primary and secondary, MTT problem becomes two-tier tree problem which is also called two level network design problem (TLNDP). The two-tier tree problem is formulated by using observation from the optimal solutions two-tier tree problem as the optimal solution is a spanning tree and in the optimal solution edges with primary facilities constitute a Steiner tree where primary nodes are compulsory nodes and secondary nodes are Steiner nodes. The formulation of MTT problem is obtained by extending these observations to the multi-tier case:

$$
\begin{aligned}
&minimize \sum_{l \in L} \sum_{\{i,j\} \in E} (c_{ij}^{l} - c_{ij}^{l+1}) x_{ij}^{l} \\
&subject\ to \\
&\qquad x_{ij}^{l-1} \leq x_{ij}^{l} \qquad for\, l = 2,3,...,L \ and \ \forall \{i,j\} \in E \\
&\qquad x_{ij}^{l} \in S_l \qquad for\, l = 1,2,...,L \ and \ \forall \{i,j\} \in E
\end{aligned}
$$

where

- S_l is a Steiner tree with nodes of grade l or higher as compulsory nodes and others as Steiner nodes
- $x_{ij}^{l} = \begin{cases} 1, if\ edge\ \{i,j\}\ is\ used\ in\ S_l \\ 0, otherwise \end{cases}$
- c_{ij}^{l} is cost for installing facility type l to the edge $\{i,j\}$, where $c_{ij}^{l} \geq c_{ij}^{l+1}$ land $c_{ij}^{L+1} = 0$.

3.7 Multicommodity Minimum Cost Network Flow with Gains

Multicommodity minimum cost network flow problem with gains (MCMCF-G) is another problem type that is used to solve the multi-level telecommunication network design problem. MCMCF-G problem is used when dimensioning and routing decisions given nodes of network while deciding the type of facility to be installed on nodes and links. The MCMCF-G problem is applied to the multi-level telecommunication problem via a layered representation of the telecommunication network such that nodes of the telecommunication network are repeated on each layer and the links in each layer represents the flow of a single facility type represented by the layer while links between layers represents the converters located at the nodes. Thus there is a flow gain on links between layers by the amount of the conversion rate among the facilities [11]. The model is proposed by Balakrishnan et.al. which is used for expansion of a local access network, however, only solution methods to special cases are specified since the resulting problem is very large to solve for practical problems.

The MCMCF-G problem differs from the MCMCF problem by the flow balance constraints such that the flow balance is provided in the MCMCF-G by multiplying flows by some coefficient called conversion ratio.

3.8 Facility Location Problem

When the network problem involves a great number of terminals with low individual traffic loads, a two-level hierarchy network is used to match traffic flows to the capacities of the transmission lines. The low speed lines are used to connect group of terminals and the high speed lines are used to connect the groups of terminals to the data processing center. Concentrators are the devices that convert speed of the transmission. The facility location problem is used to formulate the problem of finding the number of concentrators needed, where they should be placed, and how the terminals should be assigned to the concentrators in centralized teleprocessing and computer network design and local access network design, which is called concentrator location problem.

The UFL problem is given the set of terminals $\tilde{I}$ and the set of possible locations for the concentrators M, to determine the number and location of the concentrators and assign the terminals to these concentrators that require minimum cost involving installation of the concentrators and service of the concentrators to the assigned terminals [27]. The formulation of UFL problem is given below:

$$Minimize \quad \sum_{i\in\tilde{I}}\sum_{j\in M} c_{ij}x_{ij} + \sum_{j\in M} F_j y_j \tag{19.1}$$

$$subject\ to$$

$$\sum_{j\in M} x_{ij} = 1 \qquad \forall i \in \tilde{I} \tag{19.2}$$

$$x_{ij} \le y_j \qquad \forall i \in \tilde{I}, j \in M \tag{19.3}$$

$$x_{ij} \in \{0,1\} \qquad \forall i \in \tilde{I}, j \in M \tag{19.4}$$

$$y_j \in \{0,1\} \qquad \forall j \in M \tag{19.5}$$

where the cost of assigning terminal i to concentrator j is denoted as c_{ij}, the cost of installing a concentrator at location j is denoted as F_j and decision variables are as follows:

- $y_j = \begin{cases} 1, & if\ a\ concentrator\ is\ installed\ at\ location\ j \\ 0, & otherwise \end{cases}$
- $x_{ij} = \begin{cases} 1, & if\ terminal\ i\ is\ assigned\ to\ concentrator\ \ at\ location\ j \\ 0, & otherwise \end{cases}$

(19.1) is the objective function which minimizes the total cost of assigning terminals to concentrators and concentrator installation costs. Constraints (19.2) and (19.4) ensure that each terminal is assigned to exactly one concentrator. Constraint (19.3) ensures that any terminal is not assigned to a concentrator that is not installed [27].

In capacitated concentrator location problem, the set of terminals $\tilde{I}$ and the set of possible locations of concentrator M with capacities q_j for each $j \in M$ are given. Each terminal has known demand d_i for all $i \in \tilde{I}$. The problem is to determine the location of concentrators and assignment of each terminal to exactly one concentrator with minimum installation cost while keeping the capacities of the concentrators sufficient enough to meet the demand of the terminals. The formulation of the capacitated concentrator location problem is given below [27].

$$Minimize \quad \sum_{i\in\tilde{I}}\sum_{j\in M} c_{ij}x_{ij} + \sum_{j\in M} F_j y_j \tag{20.1}$$

$$subject\ to$$

$$\sum_{j\in M} x_{ij} = 1 \qquad \forall i \in \tilde{I} \tag{20.2}$$

$$\sum_{j\in M} d_i x_{ij} \le q_j y_j \qquad \forall j \in M \tag{20.3}$$

$$x_{ij} \in \{0,1\} \qquad \forall i \in \tilde{I}, j \in M \tag{20.4}$$

$$y_j \in \{0,1\} \qquad \forall j \in M \tag{20.5}$$

The objective function (20.1) is same as (19.1) and the constraints (20.2) and (20.4) are same as (19.2) and (19.4), respectively. Constrain (20.3) ensures that terminal i is assigned to concentrator j only if a concentrator is installed in location j and its capacity is enough to supply demands of all the terminals assigned to it [27]. The capacitated concentrator location problem having star/star topology is equivalent to the capacitated facility location problem with single source constraints ([19]and [21]).

As the computing performance increases, the backbone network topology design problem is begun to be solved jointly with other subproblems of telecommunication network topology design problem such that designing backbone network topology jointly with the local access network topology design which is solved by multi-level network design problem (see Section 3.6), and designing physical topology design of backbone network jointly with the design of virtual topology of backbone network which is called multi-layer network design (see Section 3.10).

3.9 Capacity Expansion Problem

Capacity expansion problem is a generalization of network loading problem which is a minimum cost multicommodity network flow problem with step increasing cost function. The problem is to find how a given capacitated network is expanded by installing more capacity in order to make the network meet the traffic demand between origin and destination nodes such that the total of capacity installation and routing cost is minimized. The capacity expansion problem generalizes the network loading problem such that network loading problem assumes zero initial capacities [234] and zero flow costs. The network loading problem is presented in Section 3.4.4.

3.10 Multi-layer Network Design Problem

The multi-layer network design problem is used to design multi-layer telecommunication networks. Practically, telecommunication networks consist of several network layers that are built on top of each other leading to multi-level network structure with multi-technology devices. In the multi-layer networks each layer use different technology and each technology has its own protocol. Data is encapsulated into another protocol each time it is transmitted to a different layer. Multiplexing and demultiplexing procedure is used for encapsulation such that data is multiplexed at the beginning node, it cannot be accessed until the end of the path and demultiplexed at the end node of the path. This path is called a *grooming path* and grooming paths are the main source of complexity of the multi-layer networks; because grooming paths have a nested structure like "paths in paths in paths…" [117]. From the definition, a grooming path in a layer addresses a direct link in the upper layer which is called a logical link. The upper layer which consists of logical links is also called logical layer or logical network, while the lower layer is called physical network or physical layer in a two-layer network. This notion of two-layer networks such that the lower layer called physical layer is comprised of fiber optic cables on links and node hardware on the nodes, and upper layer, called logical layer, is defined on physical layer as a link in the logical layer corresponds to path in physical layer, is used by almost all studies in the literature as network representation.

The two-layer representation is extended to multi-layer networks more than two layers such that each layer is added over the uppermost logical layer. Suppose that the network has three layers, the physical network consists of nodes I={A, B, C, D} and links E={{A, D}, {D, B},{B, C}} and the node set is same in all layers. The flow has to be routed from A to C in the uppermost layer. The different routing schemes are presented in Figure 4 using this network representation.

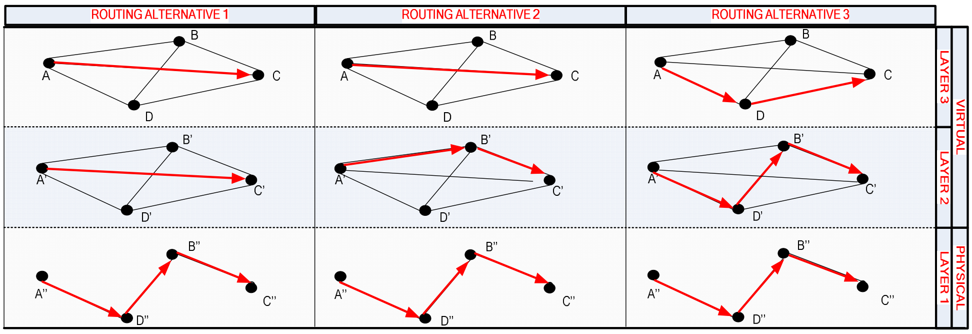


Figure 4. Alternative Routings for Demand From A to C

This representation is intuitive since it uses the server-client relationship between the layers as it is done when sequential design is made such that one layer is designed and needed capacity is used as demand for the next lower layer and so on. In addition, this representation provides the modularity needed by telecommunication network management since the owner of all layers may not be the same. However, from computational point of view, defining a different layer for all technologies is hard to deal with especially when the granularity of the flow in the network is not homogeneous, that is grooming is done in the network. In addition, to add some practical side constraints like survivability constraints, the relationship between the layers must be taken into account.

In the literature, formulations are given for two-layer networks which has a physical layer and a virtual layer. However, since the formulations are based on the fact that the capacity needed to route the demand for the upper layer is the demand for the lower layer, in theory the formulations can be extended to multi-layer network design problem by adding constraints related to the added layer itself and the relation of this layer with its lower layer such that the added layer is the logical layer and its lower layer is the physical layer in the two-layer network. However, all of the studies in the literature involve only results with two-layer network instances. In this section, existing formulations are given as two-layer network design formulations that can be extended to multi-layer formulations.

Multi-layer network design problem is either formulated by a multi-layer version of the multi-commodity flow formulation i.e. flow formulation or a compact formulation that eliminates flows, i.e. capacity formulation.

Both capacity and flow formulations use two different approaches to model logical links, namely explicit and implicit approaches:

- Implicit model: Physical routing of the logical links is not known at the stage of modeling.
    - the whole set of parallel physical paths between node pairs are represented by a logical link between these nodes
    - logical graph is simple and parallel logical links can be aggregated into a single link
    - Advantage: Implicit model interprets the logical link capacities as physical demands for physical layer which is in fact the top-down view of multi-layer networks. Capacity of upper layer is the demand for lower layer. This can be modeled by either multicommodity flow variables or metric inequalities.
    - Disadvantage: Node failures and existence of non-admissible physical links for some logical links cannot be modeled using implicit model. Therefore, implicit model is not flexible enough for modeling some important practical side constraints for telecommunication network design [7].

- Explicit model: For each layer logical layer l, a physical routing E_l and L_e are known in advance, i.e., before modeling
    - every logical link is associated to path in the physical layer
    - there are many parallel logical links between node pairs that are corresponding to different physical paths
    - Advantage: Node failures and existence of non-admissible physical links for some logical links can be modeled using explicit model. Explicit approach is flexible for modeling such practical side constraints for telecommunication network design [7].
    - Disadvantage: Exponential number of integer variables (capacity of logical links) would be defined if all possible admissible logical links are available. Therefore, as the number of nodes increase or the number of layers increase, it becomes computationally intractable to model the problem by explicit approach.

Flow formulation can further be divided into two groups as:

- Arc (link) path formulation/path flow formulation (PF): Main decision variable is a binary variable for indicating if the arc (link) is used by the path or portion of the capacity of arc used by the path
- Node-arc (edge) formulation/edge flow formulation (EF): Main decision variable is a flow variable for each arc (edge) between nodes i and j.

Notation and definitions of the variables and parameters used in the existing models in the literature are given in the Table 11.

Table 11. Notation and Definitions for the Base Problem

	Notation	Definition	Remark
Graphs	$G(I,E)$	the physical network	Undirected network
	$H(V,L)$	the logical network	Undirected network
Sets	$I=\{0,1,\ldots,N\}, \forall i \in I$	nodes to be connected on G	$\forall i \in I$ can operate either at one single layer or in both layers depending on the technological components they have. For P0 both networks have the same set of nodes: $I=V$ all the nodes of the network are equipped to operate at both levels
	$V=\{0,1,\ldots,M\}, \forall v \in V$	nodes to be connected on H	
	$E=\{\{i,j\} \mid i \in I, j \in I\}, e \in E$	potential physical links	Words edge, arc, and link are used synonymously
	$L=\{\{u,v\} \mid u \in V, v \in V\}, l \in L$	logical links	Lightpath and logical edge are used synonymously
	$L_{ij} \subseteq L,\ i,j \in I$	parallel logical links between nodes i and j	$L_{ii}=\varnothing\ and\ L_{ij}=L_{ji}$
	$K=\{1,2,\ldots,K\}, k \in K$	set of commodities	$\forall k \in K$ is a triple (s^k, t^k, d^k) as source and destination nodes and demand
	$E_l \subseteq E, l \in L$	physical links used by logical link $l \in L$	To use explicit logical link model, these sets must be known a priori.
	$L_e \subseteq L, e \in E$	logical links using edge $e \in E$	
Para-meters	U	size of a capacity module for logical links	every time we buy one unit or module of capacity for a logical (physical) link we get a capacity of U (B) on that link
	B	size of a capacity module for physical links	
	c_e^E	cost of installing one module of capacity on edge $e \in E$	

Table 11 (Cont'd)

	Notation	Definition	Remark
Para-meters	c_l^L	cost of installing one module of capacity on link $l \in L$	Cost of a logical link is related to the hardware that it begins and terminates. In some studies [117] this cost is directly calculated using the hardware model. In P0 no hardware model is used, hence this is an approximated cost.

3.10.1 Explicit Flow Formulation (EFF)

$$min \sum_{e \in E} c_e^E x_e + \sum_{l \in L} c_l^L y_l$$

$$s.to$$

$$\sum_{j \in I} \sum_{l \in L_{ij}} (f_{l,ij}^k - f_{l,ji}^k) = \begin{cases} d^k & if\ i = s^k \\ -d^k & if\ i = t^k, \\ 0 & o.w. \end{cases} \qquad i \in I\ and\ k \in K \qquad (21.1)$$

$$\sum_{k \in K} (f_{l,ij}^k + f_{l,ji}^k) \leq U y_l \qquad l = (i,j) \in L \qquad (21.2)$$

$$\sum_{l \in L_e} y_l \leq B x_e \qquad e \in E \qquad (21.3)$$

$$f \geq 0$$

$$x_e, y_l \in \mathbb{Z}_+ \qquad e \in E\, and\, l \in L$$

where

- x_e : number of capacity units installed on physical edge $e \in E$
- y_l : number of capacity units installed on logical edge $l \in L$
- $f_{l,ij}^k$: flow of commodity $k \in K$ directed from i to j on edge $l = (i,j) \in L$

In EFF, (21.1) is a flow conservation constraint in logical layer, (21.2) are capacity constrains for logical links and (21.3) are capacity constraints for physical layer that ensures that physical capacity is enough for supporting all the logical links.

The EFF formulation with edge flows (EFF-EF) is presented in [7], [120], [122], [235], [236]. It is solved by Bley et al. with an iterative approach involving exact MILP methods and combinatorial heuristics for test instances with two layers up to 50 nodes [236]. Koster et al. uses branch and cut with problem specific preprocessing to solve test instances with two layers up to 17 nodes under 1% relative gap [235].

The EFF formulation with path flows (EFF-PF) is presented in [117], [118], [120]. There are not any solution methods or computational results reported for EFF-PF up to our knowledge.

3.10.2 Implicit Flow Formulation (IFF)

$$min \sum_{e \in E} c_e^E x_e + \sum_{l \in L} c_l^L y_l$$

$s.to$

$$\sum_{l \in L} (p_{ij}^l + p_{ji}^l) \leq Bx_e \qquad e = (i,j) \in E \qquad (22.1)$$

$$\sum_{k \in K} (f_{l,ij}^k + f_{l,ji}^k) \leq Uy_l \qquad l = (i,j) \in L \qquad (22.2)$$

$$\sum_{j \in I} \sum_{l \in L_{ij}} (f_{l,ij}^k - f_{l,ji}^k) = \begin{cases} d^k & if\, i = s^k \\ -d^k & if\, i = t^k \\ 0 & o.w. \end{cases}, \quad i \in I \;\; and \;\; k \in K \qquad (22.3)$$

$$\sum_{j \in I:(i,j) \in E} (p_{ij}^l - p_{ji}^l) = \begin{cases} y_l & if\, i = s^l \\ -y_l & if\, i = t^l \\ 0 & o.w. \end{cases} \qquad i \in I \, and \;\; l \in L \qquad (22.4)$$

$$f, p \geq 0$$

$$x_e, y_l \in \mathbb{Z}_+ \qquad e \in E \, and \, l \in L$$

where

- p_{ij}^l : flow on physical layer corresponding to the routing of logical capacity installed on logical link $l \in L$ going from i to j on edge $e = (i,j) \in E$

(22.1) and (22.2) are capacity constraints for physical layer and for logical layer, respectively. (22.3) and (22.4) are flow conservation constraints for logical layer (flow of commodities) and for physical layer (lightpaths), respectively.

The IFF formulation with edge flows (IFF-EF) is presented in [5], [122]. There are not any solution methods or computational results reported for IFF-EF up to our knowledge.

The IFF formulation with path flows (IFF-PF) is presented in [5], [237]. Kubilinskas et al. proposed an iterative algorithm based on convex lexicographical maximization and reported results of two numerical examples with of two layers and logical and physical network nodes of 12 and 12, and 12 and 42.

3.10.3 Explicit Capacity Formulation (ECF)

$$min \sum_{e \in E} c_e^E x_e + \sum_{l \in L} c_l^L y_l$$

$s.to$

$$\sum_{l=(i,j) \in L} \alpha_{ij} U y_l \geq \sum_{k \in K} \pi_{D(k)}^k (\alpha) d^k \qquad \alpha \in Met_L \qquad (23.1)$$

$$\sum_{l \in L_e} y_l \leq Bx_e \qquad e \in E \qquad (23.2)$$

$$x_e, y_l \in \mathbb{Z}_+ \qquad e \in E \, and \, l \in L$$

where

- α is a function, such that $\alpha : A \rightarrow \mathbb{R}_+$ defines a **metric** on $H(V,L)$:

- Let G(N, A) be a graph, a function $\alpha : A \rightarrow \mathbb{R}_+$ defines a **metric** on G $\Leftrightarrow$ $\forall \alpha \geq 0$ and $\alpha_{ij} \leq \alpha_{ij}(P_{ij})$ where P_{ij} is the shortest path distance between i and j when α is used as weights.

- Met_L is the cone of all nonzero metrics defined on $H(V, L)$.

- $\pi^k_{D(k)}(\alpha)$ is defined as shortest path distance from origin to destination node of commodity $k \in K$ that is found by using $\alpha \in Met_L$ values as link weights.

Constraints (23.1) are the metric inequalities for logical layer which ensure that the capacity installed on logical links can support the demand. Constraints (23.2) guarantee that capacity installed on physical links can support the logical links.

Logical layer is the client of physical layer which means that capacity needed to support the demand for logical layer becomes the demand for physical layer. Therefore, logical links in logical layer behaves as commodities for physical layer.

The ECF is presented by Mattia in [122] and solved by branch and cut in the same study. The results are reported on test instances with two layers up to 37 nodes with a hop limit of 5 in logical layer and up to 12 nodes without hop limit.

3.10.4 Implicit Capacity Formulation (ICF)

$$min \sum_{e \in E} c^E_e x_e + \sum_{l \in L} c^L_l y_l$$

$$s.to$$

$$\sum_{l=(i,j) \in L} \alpha_{ij} U y_l \geq \sum_{k \in K} \pi^k_{D(k)}(\alpha) d^k \quad \alpha \in Met_L \qquad (24.1)$$

$$\sum_{e=(i,j) \in E} \alpha_{ij} B x_e \geq \sum_{l \in L} \pi^l_{D(l)}(\alpha) y_l \quad \alpha \in Met_E \qquad (24.2)$$

$$x_e, y_l \in \mathbb{Z}_+ \qquad e \in E \, and \, l \in L$$

where

- Met_E is the cone of all nonzero metrics defined on $G(I, E)$.

Constraints (24.1) and (24.2) are the metric inequalities for logical and physical layers.

The ICF formulation is presented in [3–5], [122]. Mattia proposed a branch and cut algorithm for solving ICF and test results with two layer test instances with up to 37 nodes are given [122]. A Bender's like constraint generation method is used by Lardeux et al. [3] and Knippel and Lardeux [4]. Fortz and Poss [5] also use the same constraint generation method but within a branch and cut algorithm framework. Lardeux et al. [3] report computational tests for two layer test instances with 6 and 9 physical layer nodes, Knippel and Lardeux [4] report results for two layer test instances with 6, 8 and 9 nodes, and Fortz and Poss [5] report the results of two layered test instances with 8 and 9 nodes. Fortz and Poss [5] also report solutions for six SNDLIB [238] instances up to 15 nodes.

There are six different formulations which differ from each other whether the formulation is a flow formulation or compact formulation that does not have any flow variable or whether logical links are implicitly or explicitly modeled or whether the flow formulation is an edge flow or path flow formulation. These formulations have several capabilities in terms of modeling different side constraints of the multi-layer telecommunication networks and being computationally tractable. In addition, outputs of these solutions have different detail of information that gives solution for different decisions. In order to discuss these differences, let us first define a base problem that all formulations can solve.

The base problem can be defined as "finding capacities for physical and logical links needed to carry demand with minimum installation cost for a two-layer network defined by graphs, sets, and parameters given in Table 11".

In the base problem,

- all physical paths are admissible as logical links,
- the same capacities can be installed on all logical links (one capacity module exists for all logical links),
- node hardware does not depend on the flow through a node,
- survivability is not considered (node failures and link failures are not considered)
- cost and capacity of routing and switching devices at the nodes are ignored (no node model is used).

Let's call the base problem P0 if the physical routing of logical links are not known a priori (implicit approach), else P1. Therefore, implicit formulations will solve P0 and explicit formulations will solve P1.

First of all, let's discuss flow formulation capabilities. Solutions of P0 and P1 will be equivalent, i.e. implicit and explicit approaches give equivalent solutions if,

- all physical paths are admissible as logical links,
- the same capacities can be installed on all logical links,
- node hardware does not depend on the flow through a node,
- node failures are not considered

The following can be modeled by both implicit and explicit approaches:

- several capacity types (modular capacities),
- a maximum number of logical links using a physical link (needed for post processing of the results for wavelength assignment)
- flow-independent node capacities (node model)
- physical link failures

For flow formulation whether path flow or edge flow formulation is picked for the mathematical model directly affects the index set of the model. The selection of path flow or edge flow formulation depends on the problem type. Path flow formulation can be used in dimensioning problems. Because in these problems, installed links are predetermined and only their capacities are decided. So which path uses which arc is known and the arcs can be indexed as such. For allocation problems, you may not know which arc is used by which path from the beginning. So that edge flow formulation is used if all possible paths are admissible [118].

Explicit node costs and capacities can be modeled by flow formulations. In order to incorporate such a node model in EFF and IFF formulations, the following constraints are added to the model [6], [7]:

$$z_i \leq 1 \qquad \forall i \in I \qquad (25.1)$$

$$Qz_i - \sum_{l \in L_{ij}} Uy_l \geq q_i \qquad \forall i \in I \qquad (25.2)$$

$$z_i \in \{0,1\} \qquad \forall i \in I$$

where,

- z_i is number of node hardware
- Q is capacity of node hardware
- q_i is amount of pre-defined slack capacity that is added to each node

Constraints (25.1) guarantee that at most one node hardware module is installed to each node. Constraints (25.2) guarantee that node module on each node $i \in I$ is enough to switch the traffic on logical links starting or ending at node $i \in I$. The right hand side of the inequality is usually taken as greater than zero, since in practice, an excess capacity for each node is planned as a remedy for not re-dimensioning the nodes in case of change occurring in routings.

3.10.5 Incorporating Survivability into the MLNDP Formulations

In the literature, there are three ways of defining survivability in the multilayer networks:

- 1+1 protection mechanism
- Diversification mechanism
- Predefining failure states of the network

1+1 Protection Mechanism

1+1 protection mechanism is commonly used by transport networks. It involves duplicating the demand for commodities to be protected and route these demand pairs via two nodes and link disjoint paths. In modeling, 1+1 protection mechanism is the same as 1+1 restoration mechanism. 1+1 protection provides survivability to single line card or port failures as well as single link failures, i.e., it provides survivability on logical layer as well as physical layer.

This mechanism is preferred by network operators and does not require reconfiguration for single link or node failure cases. However, the network requires much back up capacity.

In the literature, only flow formulations are used for modeling 1+1 protection. Because, capacity formulations cannot handle any routing constraints, they cannot be used for guaranteeing node and/or link disjoint routes for duplicated demands. So that, 1+1 protection can only be modeled by using flow formulations.

1+1 protection is hard to solve using mixed integer problems. The mathematical model involves either integer cycle variables or integer working and back up path variables. These variables are exponential in number and there is symmetry between working and backup variables which is problematic in solving the model. In addition, branching and pricing decisions affect each other if the models are attempted to be solved by branch and price such that a new column adds a new row [7].

Formulating Diversification Mechanism

Diversification ensures that at most a certain percentage of demand is routed through any node and link. It is proposed by Dahl and Stoer [123].

In order to come up with computational difficulties that 1+1 protection introduces a diversification mechanism is modeled as a relaxation of 1+1 protection in some studies. For example, Orlowski [7] models diversification as relaxation to 1+1 protection. He doubles the demands of commodities as in 1+1 protection and adds the following constraints to be sure that at most half of demand for a particular commodity is routed through any physical link or node. Note that Orlowski uses explicit edge-flow formulation.

$$\frac{1}{2}\sum_{l\in\delta_L(v)}(f^k_{l,ij}+f^k_{l,ji})+\sum_{l\in L^v}(f^k_{l,ij}+f^k_{l,ji})\le\frac{d^k}{2}\qquad \forall v\in I\ \ and\, k\in K \qquad (26.1)$$

$$\sum_{l\in L^e}(f^k_{l,ij}+f^k_{l,ji})\le\frac{d^k}{2}\qquad \forall e\in E\ \ and\, k\in K \qquad (26.2)$$

where

- $\delta_L(v)$ is the set of logical links $l\in L$ starting or ending at node v, $v\in I$.
- L_i is the set of logical links $l\in L$ containing node $v\in I$ as an inner node.

(26.1) provides link diversification and (26.2) provides node diversification. By these constraints, Orlowski deals with the following:

- Diversification against single link failures: The network is survivable against a single physical link failure.
- Diversification against multiple link failures: The network is survivable against multiple link failures since failure of a single physical link leads failure of multiple logical links in a multilayer network.
- Diversification against node failures: The network is survivable against node failures.

In single layer, link diversification constraints are dominated by node diversification constraints except for direct links between origin and destination. However, in multilayer networks, there can be some rare situations that link diversification constraints are not dominated by node diversification constraints. Thus, Orlowski [7] does not add all link diversification constraints from the beginning. He adds only link diversification constraints related to the direct links between origin and destination nodes of the commodities. The rest are added if needed through the solution procedure.

Diversification can be modeled in path flow formulation too. At this point, edge flow and path flow formulations that are equivalent for single layer survivable network design problem become unequal for multilayer survivable network design problem. Under diversification constraint, edge flow formulation is strictly stronger than path flow formulation if diversification constraints are used as the same as single layer survivable network design problem [7].

Predefining Failure State of the Network
A network that is able to route all demands in each and every possible failure state is defined as a survivable network. Modeling survivability by predefining failure states involves predefining the working node and link sets for each possible failure state and investigates the network that is optimal for all predefined failure states. Then, the scenarios are incorporated in the model as the index sets to the node and link sets and the related sets, i.e., metric of logical links and decision parameters, i.e., flow variables.

Mattia [122] and Kubilinskas [125] use this method to incorporate survivability to the mathematical models of multilayer networks.

Revisiting the types of formulations, let $s \in S$ be the index set for predefined failure states. The following changes are done for setting up the optimal network survivable:

- EFF: $s \in S$ is added as an index to vertex, logical link and commodity sets, i.e., sets V^s, L^s, and K^s and to the flow parameters, i.e., $f_{l,ij}^{ks}$.
- IFF: $s \in S$ is added as an index to vertex, physical link, logical link and commodity sets, i.e., sets V^s, E^s, L^s, and K^s and to the flow parameters, i.e., $f_{l,ij}^{ks}$ and p_{ij}^{ls}.
- ECF: $s \in S$ is added as an index to logical link and commodity sets, i.e., sets L^s and K^s.
- ICF: $s \in S$ is added as an index to physical link, logical link and commodity sets, i.e., sets E^s, L^s, and K^s.

3.10.6 Comparison of the MLNDP Formulations
Both edge-flow and path-flow formulations can be used to formulate P0 and P1. Their solutions are equivalent for base problems P0 and P1 with and without diversification mechanism for single link failures, although Path Flow Formulation is a strict relaxation of the edge-flow formulation for both P0 and P1 if the multiple link failures are modeled with diversification mechanism in the sense that every solution of the edge-flow version can be transformed into a path-flow solution with (at most) the same cost, but not necessarily vice versa [7].

The capability for modeling survivability mechanisms differs for the formulations. 1+1 protection mechanism and its relaxation diversification mechanism cannot be modeled using capacity formulation. Single link failures can be modeled by both the implicit and explicit flow formulations while for modeling multiple link failures, explicit representation must be used. Moreover, modeling node failures is possible only when edge flow formulation is used.

Modeling restrictions on admissible physical links that can be used by logical links is not possible with capacity formulation though edge flow formulation using the explicit approach and both edge flow and path flow formulations can be used to model this side constraint with implicit approach. However, edge flow formulation as well as capacity formulation cannot be used to formulate general routing restrictions like bound on number of hops on a physical path with both P0 and P1 while, path flow formulation is capable of formulating this side constraint for both P0 and P1. In addition, node cost and capacity, and unsplittable flow can be modeled explicitly by flow formulation while capacity formulation cannot model explicit node cost and capacity, and unsplittable flow. Routing costs cannot be modeled by the capacity formulation.

Comparison of the formulation types regarding their modeling capabilities is presented in Table 12.

Table 12. Comparison of Formulation Types

Modeling Capability	**EFF-EF**	**EFF-PF**	**ECF**	**IFF-EF**	**IFF-PF**	**ICF**
Admissible physical paths for logical links	√	√			√	
General routing restrictions (bound on hops)		√			√	
Unsplittable flow (single-path routing)	√	√		√	√	
Single link failure by 1+1 protection	√	√		√	√	
Multiple link failures by 1+1 protection	√	√				
Node failures by 1+1 protection	√					
Single link failure by diversification[24]	√	√		√	√	
Multiple link failures by diversification[25]	√	√				
Node failures by diversification	√					
Single link failure by failure states	√	√	√	√	√	√
Multiple link failures by failure states	√	√	√	√	√	√
Node failures by failure states	√	√	√	√	√	√
Explicit node cost and capacity	√	√		√	√	
Routing costs	√	√		√	√	

[24] EFF-EF and EFF-PF are equivalent [7]
[25] EFF-PF is a strict relaxation of EFF-EF [7]

Formulations do not differ from each other with respect to their modeling capabilities, their solution are different from each other in terms of the details provided. For example, capacity formulation and flow formulation solutions are not equivalent because flow formulation computes optimal capacities and the corresponding feasible routing at the same time explicitly. However, capacity formulation computes only the optimal capacities ensuring that with given capacities a feasible routing exists. Another LP is solved for a feasible routing corresponding to the optimal capacity if capacity formulation is used. In addition, implicit and explicit formulations differ from each other since logical link capacities are explicitly given in the solution for explicit formulations while an LP is needed to allocate aggregated logical link capacities found by the solution to individual logical links for implicit formulations.

As a summary, capacity formulation compared to flow formulation and implicit formulation compared to explicit formulation are more aggregated in terms of the solution details they provide since some LP's are needed to be solved for finding a routing in capacity formulation solution and finding logical link capacities explicitly in implicit formulation solution. Explicit edge flow formulation gives the utmost flexibility to model realistic survivability mechanisms compared to all other formulations although, the number of variables increases with the number of vertices and layers making the problem computationally intractable for moderate number of nodes with two-layers or small number of nodes for more than two layers. Therefore it is important to find a formulation that is both flexible to model practical side constraints and computationally tractable even if there are more than two layers.

3.11 Summary of Findings

We extracted a relationship scheme with the network optimization problems regarding the changes in telecommunication network design problem type. It is observed that the minimum spanning tree problem which is used for topology design of centralized computer networks can be used as an origin for the relationships as other problems can be extracted from minimum spanning tree problem by making some changes. The variations of minimum spanning tree problems form a basis for formulating the multi-terminal network design problem with terminals having heterogeneous traffic restrictions [9]. Other network design problems are for obtaining the minimum cost network topology design that can be derived from multi-terminal network flow problem with heterogeneous terminals since different cost functions used in the problem lead different problem types [21].

The relationships between these problem types are presented in Figure 5.

The minimum spanning tree problem can be called the fundamental problem in solving the classical network topology problem since all the other problems are linked to the minimum spanning tree problem. The capacitated minimum spanning tree problem is the minimum spanning tree problem with a single capacity constraint, i.e., if a single capacity constraint for all the links is added to the minimum spanning tree problem, the capacitated minimum spanning tree problem is obtained. The multi-center capacitated minimum spanning tree problem is the capacitated minimum spanning tree problem with more than one root node. The multilevel capacitated minimum spanning tree problem is the capacitated minimum spanning tree problem with multiple link capacities. If the tree restriction in the multi-level minimum spanning tree is removed, then the problem becomes the Telpak problem. The Telpak problem is a special case of fixed-charge network design problem and the minimum concave cost flow problem is a generalization of the Telpak problem. The Telpak problem is in fact a minimum cost multi-terminal network flow problem with a staircase cost function. The minimum cost multi-terminal network flow problem having a linear with fixed cost function is called the fixed charge network design problem. The Steiner tree problem is related to the minimum cost multi-terminal network flows problem since the Steiner tree problem can be formulated as the minimum cost multi-terminal network flow problem.

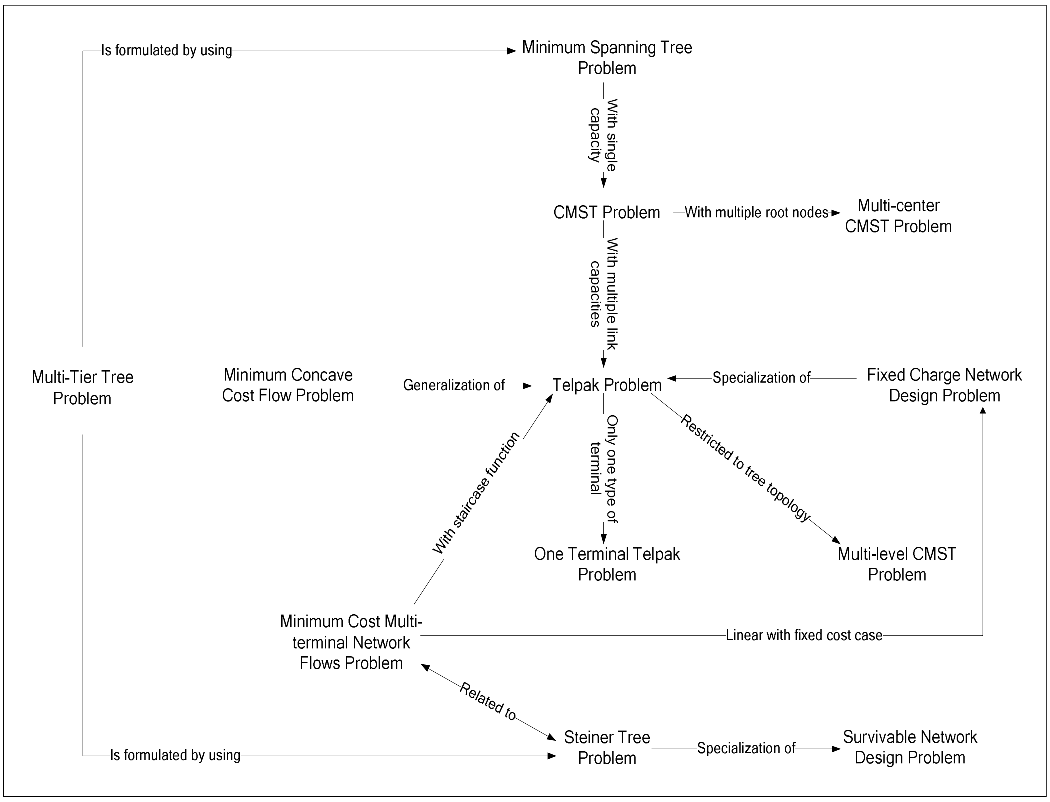


Figure 5. Relationships between Network Problems in Telecommunication Network Topology Design

The network optimization problems that are used to solve the telecommunication network design problems with connectivity constraints such as survivable network design problem and multi-tier tree problem are related to the Steiner Tree problem such that the multi-tier-tree problem and the survivable network design problem are generalizations of the Steiner tree problem.

The mathematical formulations, solution methods and solution capabilities of the network optimization problems used to solve the telecommunication network design are investigated in the study. A unified notation for sets, variables and parameters is used for the mathematical models of each of the problems given above. The notation is given in Table 13. This table serves as summary for sets, variables and parameters used for each problem presented in the paper as well as it presents a comparison among these problems regarding their inputs and outputs.

Table 13. Notation used for Network Optimization Problems in Telecommunication Network Design

		Minimum Spanning Tree	Steiner Tree	Flow Problems	Location Problems	Multi-Tier Tree Problem
SETS	Root node	$\{0\}$	$\{0\}$	----	----	----
	Set of terminal nodes	$\tilde{I} = \{1,2,\ldots,N\}$, $\forall i \in \tilde{I}$	$V = \{v_1,\ldots,v_N\}$	----	$\tilde{I} = \{1,2,\ldots,N\}$, $\forall i \in \tilde{I}$	----
	Set of nodes	$I = \{0,1,\ldots,N\}, \forall i \in I$			----	$I = \{0,1,\ldots,N\}$, $\forall i \in I$
	Set of edges	$E = \{\{i,j\} \mid i \in I, j \in I\}$, $e \in E$			----	$E = \{\{i,j\} \mid i \in I, j$ $e \in E$
	Set of arcs	$A = \{(i,j) \mid i \in I, j \in I\}$, $a \in A$		----	----	----
	Set of commodities	$K = \{1,2,\ldots,N\}, k \in K$			----	----
	Hop index	$T = \{1,2,\ldots,Q\}$, $t \in T$	---	---	---	----
	Set of facility types	$L = \{1,2,\ldots,L\}$, $l \in L$	---	---	---	$L = \{1,2,\ldots,L\}$, $l \in L$

Table 13 (Cont'd)

			Minimum Spanning Tree	Steiner Tree	Flow Problems	Location Problems	Multi-Tier Tree Problem
SETS	Capacity index		$P = \{1,2,\ldots,P\}$, $p \in P$	---	---	---	----
	Set of possible concentrator locations		---	---	---	$M = \{1,2,\ldots,M\}$, $m \in M$	----
PARAMETERS	Cost Cost	Cost of installing an arc	$c_{ij},\ i,j \in I$ $c_a,\quad a \in A$	$c_{ij},\ i,j \in I$	$c_{ij},\ i,j \in I$	$c_{ij},\ i,j \in I$	----
		Cost of installing a multi-facility arc	$c_{ij}^l,\ i,j \in I;$ $l \in L$	---	---	---	$c_{ij}^l,\ i,j \in I;\ l \in L$
		Capacity indexed cost of installing an arc	$c_{ij}^p,\ i,j \in I;$ $p \in P$	---	---	---	----
		Linear cost of an edge (flow dependent cost)	---	---	$c_{ij},\ i,j \in I$	---	----

Table 13 (Cont'd)

			Minimum Spanning Tree	Steiner Tree	Flow Problems	Location Problems	Multi-Tier Tree Problem
PARAMETERS	Cost	Fixed cost of an edge (cost of installing an edge)	---	---	$fe,\ e \in E$	---	----
		Minimal cost of installing an edge with capacity satisfying flow x	$f_{ij}(x),\ i,j \in I$	---	---	---	----
		Cost of installing a hardware on the node	---	---	---	$c_j,\ j \in I$	----
	Capa city	upper limit on the traffic that each edge can transmit	Q	---	Q	---	----

Table 13 (Cont'd)

			Minimum Spanning Tree	**Steiner Tree**	**Flow Problems**	**Location Problems**	**Multi-Tier Tree Problem**
PARAMETERS	Capa city	Upper limit on the number of the edges to be connected to each node (degree constraint – port capacity of node hardware)	$q_i,\ j \in I$	---	---	---	----
		Capacity of edge between nodes i and j	---	---	$Q_{ij},\ i,j \in I$	---	----
		Capacity of a link	---	---	$q_e,\ e \in E$	---	----
		Multi-facility capacity of nodes	$q_l,\ l \in L,$ $q_0 < q_1 < .. < q_L$	---	---	---	----

Table 13 (Cont'd)

			Minimum Spanning Tree	Steiner Tree	Flow Problems	Location Problems	Multi-Tier Tree Problem
PARAMETRS	Demand	Amount of traffic that has to be transmitted between origin and destination nodes	$d_i,\ i \in I$	---	$d_i,\ i \in I$	---	----
	Number of times that an arc appears in the i^{th} q-arb		$b_a^i,\ a \in A$	---	---	---	----
VARIABLES	Continuous Var.	Single commodity flow	$x_{ij},\ i,j \in I$	---	$x_{ij},\ i,j \in I$	---	---
		Hop-indexed flow	$x_{ijt},\ i,j \in I;\ t \in T$	---	$x_{ijt},\ i,j \in I;\ t \in T$	---	---
		Multi-commodity flow	$x_{ij}^k,\ i,j \in I;\ k \in K$	$x_{ij}^k,\ i,j \in I;\ k \in K$	$x_{ij}^k,\ i,j \in I;\ k \in K$	---	---
	Binary Var.	Installing an arc	$y_{ij},\ i,j \in I$ $y_a,\ a \in A$	$y_{ij},\ i,j \in I$	$y_{ij},\ i,j \in I$	$y_{ij},\ i,j \in I$	---
		Installing a hop-indexed arc	$y_{ijt},\ i,j \in I;\ t \in T$	---	---	---	---

Table 13 (Cont'd)

			Minimum Spanning Tree	Steiner Tree	Flow Problems	Location Problems	Multi-Tier Tree Problem
VARIABLES	Binary Var.	Installing a multi-level arc	$s_{ij}^{l},\ i,j \in I;$ $l \in L$	---	---	---	---
		Installing a capacity indexed arc	$y_{ij}^{p},\ i,j \in I;$ $p \in P$	$y_{ij}^{p},\ i,j \in I;\ p \in P$	$y_{ij}^{p},\ i,j \in I;$ $p \in P$	---	---
		Installing a multi-facility arc	---	---	---		$y_{ij}^{l},\ i,j \in I, l \in L$
		Installing a facility on a node	---	---	---	$y_j,\ j \in M$	---
		Existence of multicom modity flow	$y_{ijk},\ i,j \in I;$ $k \in K$	---	---	---	---

The solution methods and the solution capabilities of the network problems used to solve the telecommunication network design are presented in Table 6. A summary of the current state of the solution methods and their solution capabilities are presented.

Table 14. Solution Methods Developed for the Network Problems Used to Solve the TNDP

Network Problems		**Solution Methods**	**Solution Capabilities Obtained**
Minimum spanning tree [26]	Uncapacitated minimum spanning tree [9], [21], [146], [149]	Mostly heuristic methods used: Prim’s and Kruskal's algorithms	Polynomially solvable problem Bazlamacci and Hindi compared algorithms with randomly generated test instances having up to 16,000 nodes in [149]
	Degree constrained minimum spanning tree [9]	Lagrangian based algorithm	170 problems are tested for problems with number of nodes varying from 20 to 200, 5 to 25 of which have degree constraints of 2 to 16. The problems are solved to optimality except 3 of them that had gaps less than 10^{-3} [9]
	Capacitated minimum spanning tree [9], [10], [21], [37], [38], [41], [42], [140], [148], [150–153], [157–167], [169–173], [239], [240]	Exact methods: -Dantzig Wolfe decomposition -Bender's decomposition -Augmented Lagrangian relaxation -cutting plane -Lagrangian relaxation -Branch and price and cut Heuristic Methods: -FOGA -SOGA -GRASP -Tabu search -Variable neighborhood search (VNS) -Ant colony optimization -Genetic algorithm	Exact solutions: the best results are taken from branch and price and cut algorithm due to Uchoa et al. [153] up to now CMST problems having up to 200 nodes with capacities 200,400 and 800 can be solved by the method Heuristic solutions: Best solutions found up to 2008 are obtained by VNS due to Hu et al. [160]. The method is tested for problems with up to 1280 nodes.

Table 14 (Cont'd)

Network Problems		Solution Methods	Solution Capabilities Obtained
Minimum spanning tree	Multi-center capacitated minimum spanning tree [10], [26], [44], [241]	Approximation and heuristic methods used	Tests are performed for problems up to 100 nodes with capacities from 100 to 200 [10]
Minimum spanning tree [63]	Multi-level capacitated minimum spanning tree [9], [133], [134], [153], [154], [174], [175]	Heuristics: saving based construction heuristic, and Meta-heuristics: local search, evolutionary algorithm, particle swarm optimization, GRASP	Best solutions up to 2009 are obtained by GRASP due to Martins et al. [134]. The problem is tested up to 150 nodes.
Steiner tree [21], [22], [135], [153], [176], [177], [179–186], [188–191], [193–199], [242], [243]		Exact Methods (branch and cut techniques, Lagrangian relaxation, Dual ascent method), Heuristics: construction and improvement, and metaheuristics (tabu search, local search, pilot search, GRASP) are proposed In addition, reduction techniques such as classical reduction techniques, extended reduction techniques, partitioning as a reduction technique used to reduce the problem instance size	Exact Methods: Steiner tree problem is tested by instances from SteinLib. The solution capability of the exact methods depends on the problem instance and the recent results of the instances: http://steinlib.zib.de//steinlib.php. Extended reduction techniques and partitioning when used as reduction technique outperforms classical reduction techniques in terms of percent of edges that remain after reduction. Polzin [189] reports that there is an order of magnitude (2% – 24%, 38%) with the percent of remaining edges obtained by the fastest extended reduction technique they proposed and the fastest reduction techniques proposed up to 2003.

Table 14 (Cont'd)

Network Problems			Solution Methods	Solution Capabilities Obtained
Flow	Minimum Cost Single Commodity Flow Problem [21], [143]	Multi-terminal network flow problem with heterogeneous terminals		
		Telpak problem [10], [21], [26], [35], [36], [174]	the recent studies are done with the name `Multi-level CMST"-- See Multi-level CMST problem	
		One terminal Telpak problem [21], [36]		
	Minimum cost multicommodity flow problem [21], [142], [143]	Linear cost function case [168], [200], [201], [204–216], [218–220], [244]	Solution methods include: -Dantzig-Wolfe Decomposition -Lagrangian duality -resource decomposition -partitioning -Interior point methods -Resource Directive Decomposition -Bundle Decomposition -Augmented Lagrangian Algorithm -Column Generation	Larsson and Yuan [218] report that the augmented Lagrangian algorithm provides near optimal solutions (relative accuracy of 0.2%) to instances with over 3600 nodes, 14000arcs and 80000 commodities within reasonable computation times.
		Linear with fixed cost case [224], [225] [21], [139], [221–223]	Lagrangian based methods, Lagrangian resource decomposition method and cutting plane methods	The tests are performed for problems having 10-100 nodes, 35-700 edges and 10-400 commodities [225].

Table 14 (Cont'd)

Network Problems			**Solution Methods**	**Solution Capabilities Obtained**
Flow	Minimum cost multicom modity flow problem	Piecewise linear concave cost function case [21], [48], [226], [228]	Most effective techniques for circuit switching network design.: -Yaged's linearization technique -Minoux's greedy algorithms Most effective techniques for packet switching network design: -Gerla and Kleinrock's concave branch elimination -Gersht and Weihmayer's greedy -Stacey, Eyers and Anido's concave link elimination Modified Minoux's greedy algorithm and disaggregate local search are proposed by Say and Bazlamacci [226]	Extensive computational study is done by Say and Bazlamacci [226] involving the techniques listed as solution methods. They performed the computational experiments with small (25 nodes), medium (50 nodes) and large (75 nodes) sized networks. They concluded that the best method for the network design depends on the size, traffic and the cost function of the network.
		Step increasing cost function case [19], [21], [63], [64], [116], [124], [142], [143], [230–232], [234]	Exact Methods: branch and cut, branch and bound, bender's decomposition Heuristic Methods: link rerouting and flow rerouting algorithms, approximate solution with Bender's decomposition.	Exact solutions are tested for problems with -15 nodes, 34 arcs 21 commodities [230] -13 nodes, 7 nodes and 10 nodes [231] -15 nodes and 22 links; 16 nodes and 49 link (Bienstock and Gunluk, 1996) [116] Heuristic methods are tested for problems with 15-50 nodes [19]
	Multicommodity network flow with gains [11]		Dual ascent	-

Table 14 (Cont'd)

Network Problems			**Solution Methods**	**Solution Capabilities Obtained**
Flow	Multi-layer Network Design [118]	Explicit Flow Formulation EF: [7], [120], [122], [235], [236] PF: [117], [118], [120]	*Edge Flow (EF)* Exact Methods: branch and cut [228] Heuristic Methods: Iterative approach with MILP methods and combinatorial heuristic [229] *Path Flow (PF)* ---	*Edge Flow* Exact methods up to layers - under %1 gap with problem specific preprocessing Heuristic methods up to nodes. *Path Flow (PF)* ---
		Implicit Flow Formulation EF: [116], [118] PF: [5], [237].	*Edge Flow (EF)* --- *Path Flow (PF)* Heuristic Methods: an iterative algorithm based on convex lexicographical maximization	*Edge Flow (EF)* --- *Path Flow (PF)* Heuristic Methods: two layers and logical and physical network nodes of 12 and 12, and 12 and 42
		Explicit Capacity Formulation [122]	Exact Methods: branch and cut, [122]	Exact Methods: two layers -up to 37 nodes with a hop limit of 5 in logical layer and -up to 17 nodes without hop limit [122]
		Implicit Capacity Formulation [113], [114], [116], [118]	Exact Methods: Benders like constraint generation, branch and cut, Benders decomposition within branch and cut	Exact Methods: two layers -up to 37 nodes with a hop limit of 5 in logical layer -up to 17 nodes without hop limit [122]

Table 14 (Cont'd)

Network Problems		Solution Methods	Solution Capabilities Obtained
Location [9], [19], [21], [85], [154]	Uncapacitated facility location problem [27], [85], [91]	Dual based solution procedure Greedy heuristic, arc substitution Heuristic Lagrangian Relaxation and subgradient method	The tests for greedy heuristic and arc substitution heuristic performed for problems with 43-189 nodes and 68-297 edges [85]. The Lagrangian relaxation with subgradient algorithm is tested with problems having 30 and 50 user nodes and optimality gap is 0-2.9% [91]
	Capacitated concentrator location problem [27]	Valid inequalities Lagrangian based heuristics	

The minimum spanning tree problem is a polynomially solvable problem and can be solved by Prim's algorithm or Kruskal's algorithm. The degree constrained version of the minimum spanning tree problem is solved by a Gavish using a Lagrangian based algorithm [9]. The algorithm is tested by 170 problem instances having 20 to 200 nodes 2 to 25 of which have a degree constrained of 2 to 16. Gavish reports in [9] that the algorithm solves 167 of 170 problems to optimality. The maximum integer gap is in the order of 10^{-3} which can be considered as 0 for practical purposes. The computation time on the average is between 1 second and 6 seconds [9]. For the capacitated minimum spanning tree problem, to our knowledge, the best results obtained up to now using an exact solution method are due to Uchoa et al.[153]. They proposed a branch and cut and price algorithm and new cuts. The proposed algorithm using the new cuts reduced integrality gaps of some instances without increasing the solution time. The tests are performed using 126 test instances having up to 200 nodes and for 81 of the instances, the proposed branch and cut and price algorithm performs well, however in 45 of the instances the algorithm performs poorly. For these 45 instances a branch and bound algorithm over an arc formulation performs well [153]. The best heuristic method to solve the CMST problem is due to Hu et al. [160]. They proposed a variable neighborhood search algorithm that is tested for problems having up to 1280 nodes. The multi-center capacitated minimum spanning tree problem is solved by approximation and heuristic methods. The multi-center parallel savings algorithm is tested for problems having up to 100 nodes in Gavish [10]. The best solutions for the multi-level capacitated minimum spanning tree problem up to 2009 are obtained by GRASP heuristic due to Martins et al.[134]. They report that the proposed algorithm improved the best known upper bounds for almost all of the considered problem instances. They tested the algorithm for problems having up to 150 nodes.

Reduction techniques are used as a part of solution of the Steiner tree in order to decrease the problem size. Polzin reports that the extended reduction techniques and partitioning used as a reduction technique outperforms the pre-proposed reduction techniques in terms of the percentage of the remaining edges [189].

Larsson and Yuan proposed an augmented Lagrangian algorithm for the minimum cost network flow problem with linear cost function which provides near optimal solutions to instances with over 3600 nodes, 14000 arcs and 80000 commodities within reasonable computation times [218]. The most recent study on the minimum cost multi-commodity network flow problem having a linear cost function with fixed cost is due to Chouman, Crainic and Gendron [225]. They improve the mixed integer formulation of the problem by incorporating new valid inequalities into a cutting plane algorithm. They tested the proposed algorithms for 196 test problems having 10 to 100 nodes, 35 to 700 edges and 10 to 400 commodities. They investigated the impact of the new valid inequalities and compared the results with CPLEX solutions of the problem. They report that within 2 hours CPU time, 138 of the 196 problems are solved to optimality. In 10 hours CPU time, the 12 of the remaining 58 problems are also solved to optimality and the optimality gap for unsolved problems reach values close to 3%. For the minimum cost multi-commodity network flow problem with piecewise linear concave cost function, one of the recent studies is due to Say and Bazlamacci [226]. Extensive computational study is done by Say and Bazlamacci [226] involving Yaged's linearization technique and Minoux's greedy algorithms for circuit switching network design; and Gerla and Kleinrock's concave branch elimination, Gersht and Weihmayer's greedy algorithm, and Stacey, Eyers and Anido's concave link elimination for packet switching network design. In addition, they improved Minoux's greedy algorithm and proposed Modified Minoux's greedy algorithm and disaggregate local search algorithms. They performed the computational experiments with small (25 nodes), medium (50 nodes) and large (75 nodes) sized networks. They concluded that the best method for the network design depends on the size, traffic and the cost function of the network. The minimum cost multi-commodity flow problem with step increasing cost function is solved by exact methods such as branch and cut, branch and bound, and Bender's decomposition, and heuristic methods such as link rerouting algorithm, flow rerouting algorithm and approximate solution with Bender's decomposition. There are various computational test results reported in the literature for problems with 7 nodes to 15 [116], [230], [231].

Multi-layer network design problem has been studied since 1999. Two types of formulations are for multi-layer network design (i) flow formulation and (ii) capacity formulation. Flow formulations are complex and difficult to solve compared to capacity formulations, though they are more capable of modeling practical side constraints than capacity formulation. Most of the studies address the capacity formulation. Capacity formulation is solved by branch and cut [122], benders decomposition [3], [4] and benders decomposition within branch and cut framework [5]. The problem is solved for 37 node-two layer network with an hop limit of 5 in the physical layer and up to 17 nodes without such a limit [122]. Explicit flow formulation, which is the most complex but capable formulation, is solved heuristically for up to 50 nodes two layers [236]. It is also solved by branch and cut for up to 17 nodes 2 layer network with two logical links exist in the upper layer under 1% relative gap [235]. Exact and heuristic solution methods for the multi-layer network design problem are needed for solving large network design problems.

3.12 Conclusion and Future Challenges For Network Optimization Problems in telecommunication Network Design

We made a review of the network optimization problems that are used to solve the telecommunication network design problem and presented mathematical formulations, solution methods and performance of these methods.

During the study, we observed that there is a tendency to solve integrated telecommunication network design subproblems in recent studies with the increase in the computing power instead of using decomposition approach and solving basic problems. However, this does not reduce the importance of the basic network optimization problems since the integrated telecommunication network design problems are modeled by extending or combining the basic network optimization problems. The solution methods used to solve the integrated telecommunication network problems involve solving basic network optimization problems most of the time. Thus, polyhedral and algorithmic properties of basic network design problems are important to solve telecommunication network design problems even if the telecommunication network design problems get more complex.

To this extend, the multicommodity network flow problems are very important for telecommunication network design problems since routing, topology and capacity assignment decisions can be made jointly by using multicommodity network flow problems. The increasing significance of survivability in telecommunication networks brings the necessity to solve Steiner tree problems more efficiently. Algorithmic advances in solving minimum spanning tree problem and its variants lead to more efficient solution methods for more complex network design problems. Thus, the performance of available solution methods to basic network optimization problems determines the efficiency of solving telecommunication network design problems even if the problems get more complex.

Capacity expansion problem is a difficult problem since it is a general case of network loading problem and hence the minimum cost multicommodity flow problem with a step increasing cost function. In addition, for local access network design problems, it is solved as an extension of another difficult problem, the capacitated minimum spanning tree problem.

The multi-facility and multi-period problems with concave or step increasing cost functions are more appropriate to reflect the economies of scale characteristic of telecommunication network design problem. Hence, the multicommodity network design problem with concave cost and step increasing cost function is important to solve realistic telecommunication network design problems. If the performance of the solutions proposed to these problems is considered, it is seen that further research is needed for better solutions.

CHAPTER 4

A NOVEL MATHEMATICAL FORMULATION FOR MULTI-LAYER TELECOMMUNICATION NETWORK DESIGN (MLND) PROBLEM

Telecommunication networks comprise of many subnetworks in practice. They are organized in a manner that a subnetwork is built on top of another subnetwork and the physical components of the networks constitute the lowest network. Each subnetwork in this structure has its own technology and protocol in order to serve its own purpose. In this chapter, the motivation behind the multi-layer structure of the telecommunication networks is presented to explain the practical relevance and necessity to design telecommunication networks using multi-layer models. Then, the existing graph model that is commonly used in the multi-layer telecommunication network design is presented. A novel mathematical formulation and graph representation to model the multi-layer networks using a single network is proposed. The NFF can be generalized to any type of multilayer telecommunication network design network; however, in this study we used optical networks such as SDH-over-WDM to exemplify our network transformations and computational tests. We compare our model with existing formulations and discuss their solutions using sample solutions and detailed computational experiments.

4.1 Multi-layer Telecommunication Networks

In a recent study about telecommunication network architectures heterogeneity of the emerging infrastructures is defined as critical [2]. The study lists the following dimensions of heterogeneity:

- Multi-service
    - Refers to client experience when connecting to the edge of a network
    - Characterized by combination of physical port type, network transport instance, and performance characteristic
- Multi-technology
    - Deployment of multiple technologies to implement a network service
- Multi-level
    - Domains or network regions may operate in different routing areas and can be represented in an abstract manner across associated area/region boundaries.
- Multi-layer
    - An abstraction encompasses both concepts of multi-level and multi-technology as described above

The above dimensions point out a fundamental question, "What is the difference between the problems of multi-level and multi-layer network designs?"

- The multi-level network design problem involves a single technology with different facility types, i.e., the transmission rate differs for the links as primary facility and secondary facility. Plus multi-level network design problems in the literature mainly involve topology design so that the problem reduces to a location and connectivity problem. Finally, these problems include decisions about concentrator location.
- The multi-layer network design problem adds another dimension to multi-level network design problem as technology changes such that the protocols are different in each layer. The transition through technologies is done by multiplexing/ demultiplexing which results in the grooming paths.

There are several motivations to model the telecommunication networks using multi-layer structure. These are listed below:

4.1.1 Practical Motivation of Layered Networks (Administration Point of View - Modularity)

The layering concept facilitates the system management by providing modularity. For example, air traveling can also be modeled in a multi-layered fashion as seen in Figure 6. It is seen that each layer is implementing a service via its own internal-layer actions and each layer's action relies on services provided by the layer below. Regardless of how many connected flights are used to go from departure airport to arrival airport, the activities related with ticket, baggage, gate, etc. are done with a certain sequence [245].

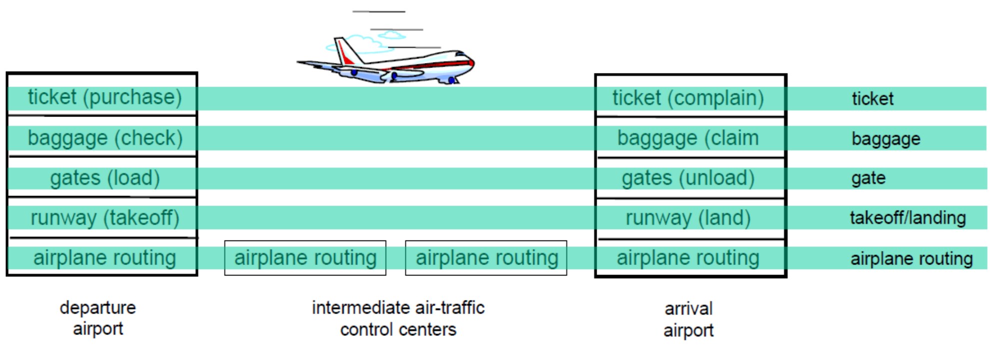


Figure 6. Multi-layer Network Analogy with Air Travel Process [245]

Modularity is essential for managing telecommunication networks. From administrative point of view, telecommunication networks are comprised of two main layers called traffic network and transport network as different layers of telecommunication networks. From the planning point of view, traffic and transport networks differ from each other since they are different in demand and cost structures, and time domain of operations is different for each network type.

The telecommunication services like internet and telephone are given by service providers. Large companies use their private networks for their own telecommunication services. These services constitute the traffic networks, also called application service networks.

Generally, these service providers and companies lease the physical telecommunication facilities from other network providers. Hence they become the customers of physical facility network providers. Physical networks provide transportation of the traffic network, called transport network. Several service types exist like internet and telephone, while several technology alternatives like ATM, SDH, SONET and WDM exist for transport networks. A telecommunication network may contain more than two transport networks having different technologies such as SDH/SONET over WDM networks.

In practice, a service provider may prefer to use one transport network provider instead of using more than one transport network providers. Likewise, a transport network provider may serve different service providers with the same transport network. Hence, the multi-layered structure of telecommunication networks provides modularity needed for network management.

Traffic and transport networks that have a server-client relationship constitute the layers of a telecommunication network. In a multi-layered network, the upper layer serves as the client for the lower layer such that the capacity needed for satisfying demand for traffic layer is the demand for transport layer and transport layer's capacity must satisfy this demand.

Traffic network services are directly demanded from the customer. Demand is unknown in advance. Because of server-client relationship, the capacity that can satisfy customer demand for traffic network becomes the demand for transport network. The capacity planning for traffic network is done according to demand forecasts by service providers and demand for transport network is declared to the physical network service providers periodically. Then, transport network demand is rather deterministic and updated periodically. In addition to the demand structure, functions performed by the two networks and their time scales are different [118].

In summary, from administrative and planning point of view, a modular modeling representation is necessary to handle the heterogeneity of telecommunication networks with regard to technologies, services, vendors, and areas/domains.

4.1.2 Technological Motivation

Telecommunication networks comprise of several technologies, which operate **interdependently**. The granularities of data streams that each technology uses are different from each other and a technology may use more than one granularity.

- Multiplexing: Process of combining small granularity signals to a coarse granularity signal.
- Demultiplexing: Process of decomposing a coarse granularity signal to a small granularity signals.

For example, in a multi-layer network ATM over SDH over WDM, virtual paths having bandwidth of 2 Mbit/s are used for ATM, virtual containers (VC) and STM are used for SDH technology whose bandwidths vary from 2Mbit/s to 140Mbit/s and lightpaths of 2.5 Gbit/s and 10Gbit/s are used for WDM. There is a "multiplexing hierarchy" between granularities of different facilities of a single technology as well as facilities of different technologies, i.e. VC-3 which is a link type from ATM can consist of combining either 7 of VC-2 links, or 21 of VC-12 links or 28 of VC-11 links while 3 of VC-3 links are combined to get 1 of VC-4 link.

Multiplexing procedure can be thought as placing small containers to a big container and the node hardware can be thought as an equipment that enables this. Multiplexing process is illustrated Figure 7 in the network topology. In the figure, red nodes have the hardware to make conversion from ATM 2 Mbit/s type links to VC-2 type links. Seven of VC-2 links are multiplexed into one VC-3 type links in blue nodes. Green node multiplexes three of VC-3 type links into one VC-4 type link. From the demultiplexing point of view, the nodes which include a multiplexer device from x type link to y type link also includes a demultiplexer from y type of link to x type of link.

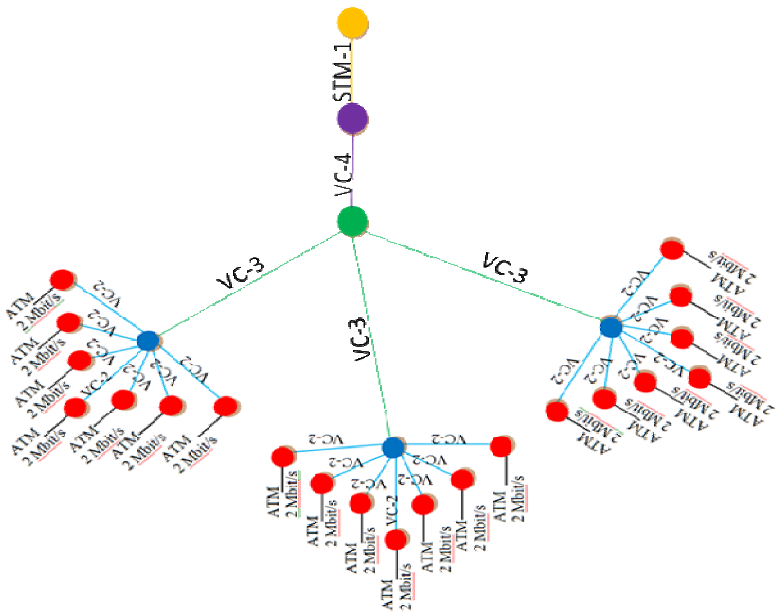


Figure 7. Illustration of Multiplexing Hierarchy in a Network

Multiplexing procedure highly resembles the concentrators in classical telecommunication networks. At this point revisiting the definition of multi-layer networks as multi-level and multi-technology networks in Section 4.1 is useful in order to link multi-layer network definition with “classical” single layer networks. From the point of view of Balakrishnan et al. [11], a multi-layer representation of a multi-level and multi-technology local access network resembles the multi-layer networks the most. Balakrishnan et al. [11] focus on local access networks, which are centered networks that collect flow from terminal nodes at a single root node. Hence, multiplexing is a concentration tool in [11] while the multi-layer networks involve multiplexing and demultiplexing together that address definition of grooming paths (see the next paragraph) and grooming paths are the main source of complexity in the multi-layer networks.

In the multi-layer networks each layer use different technology and each technology has its own protocol. Data is encapsulated into another protocol each time it is transmitted to a different layer. Multiplexing and demultiplexing procedure is used for encapsulation:

- Grooming paths: Formed by multiplexing the data at the beginning node and demultiplexing it again at the end node of the path and cannot be accessed until the end of the path. It follows that the data that has been multiplexed cannot be demultiplexed until the end of the grooming path.

Grooming paths are the main source of complexity of multi-layer networks; because grooming paths have a nested structure like "paths in paths in paths…" [117].

A grooming path in a layer addresses a direct link in the upper layer:

- Logical link is the link in the upper layer from the beginning node of the grooming path to the ending node and addressing the grooming path in the lower layer.

The upper layer which consists of logical links is also called logical layer or logical network, while the lower layer is called physical network or physical layer in a two-layer network.

The logical links and logical layer concept can be defined with a mailing system analogy. Suppose that a box is posted from point A to A' in Figure 8. The sender takes the packages to the mailing office at point B with his/her vehicle. At the mailing office, the box is packed into a larger package and sent to the distribution center at point C_1 using a vehicle such as a minivan or a truck. In distribution center, packages coming from various distribution centers are classified according to their destination and put into a larger container. This container is sent to other distribution centers C_2, C_3 and C_4 via different transportation vehicles such as plane, ship, train or trucks with other containers. Once the container reaches the distribution center at C4, it is unpacked and the package containing the box is sent to mailing office at B' using a minivan or a truck. At the mailing office, the package is unpacked and the box is delivered to the receiver at point A' by a motorcycle.

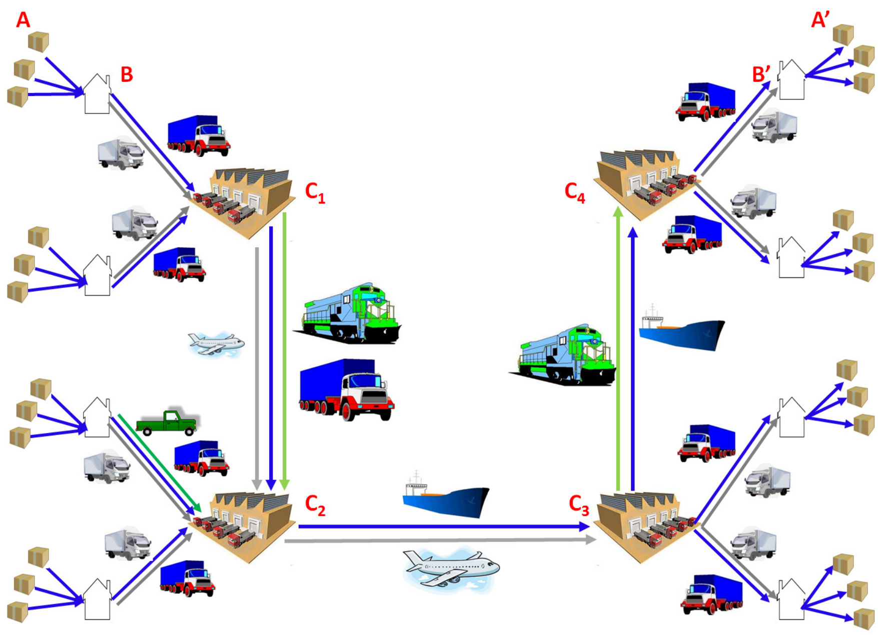


Figure 8. Mailing System Analogy

The sender at point A knows that there is a box sent to the receiver at point A' and the box sent from A cannot be accessed until it is delivered to A' during its journey. Likewise, the mailing office at point B knows that they sent a package to mailing office B' and this package cannot be accessed until it reaches its destination, mailing office at point B'. It is the same as the distribution centers on the route of the package and the containers that they send to each other. Then, in the mailing system, the relationships are defined among sender and receiver, mailing offices and distribution centers, i.e., sender does not care/know about the distributions centers that his/her box goes during its journey to the receiver. However, the box physically goes through the A-B-C_1-C_2-C_3-C_4-B'-A' path along the journey.

As seen in Figure 9, senders and receivers, mailing offices and distribution centers constitute different layers:

- At each layer, the box/package/containers are sent from one point to another, hence the size of items flow through the mailing system differs according to the layer. These sizes are analogous to the granularities of flow in multilayer telecommunication networks.
- Packing/unpacking processes should be performed at the interfaces of the layers, i.e., if a package is sent to a distribution center from a mailing office, it is packed by the distribution center into larger containers to be sent to another distribution center. These processes are analogous to multiplexing and demultiplexing processes in the multilayer networks.
- The sender only knows the box is sent to the receiver, then there is a logical link between sender and receiver. All senders and receivers constitute a multicommodity network that is analogous to the logical layer in multilayer telecommunication networks.

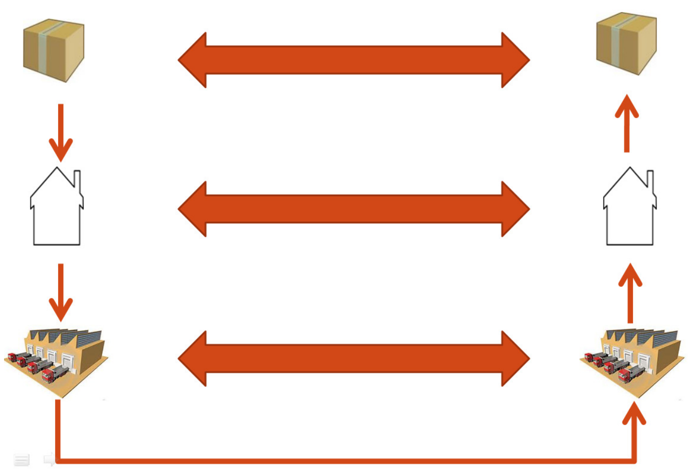


Figure 9. Layers in Mailing System

- The sending-receiving process is realized by a physical journey through A-B-C_1-C_2-C_3-C_4-B'-A' path and once an item is packed into a larger one, it cannot be accessed until it is unpacked. This is analogous to grooming paths in multilayer telecommunication networks.
- All the vehicles that are used for transmission constitute the physical components of the mailing system together with senders/receivers, mailing offices and distribution

centers, which is analogous to the physical layer of telecommunication networks that consists of node hardware and links between nodes.

The multi-layer network structure using logical links in a particular layer addressing paths in its lower layer makes possible to define the layer as the client for its lower layer. That is, the capacity needed to route the demand for a particular layer is the demand for its lower layer. Hence, the representation of multi-layer networks using logical links makes use of sequential optimization of multi-layer networks, i.e. the uppermost layer is solved optimally resulting in the capacity needed to route the demand. This capacity is taken as demand for the immediate lower layer and it is solved for optimality and so on.

Multiplexing and demultiplexing operations are done by several hardware modules located on the nodes. This hardware has capacity that is limited by the number of ports or cards that the hardware has. Thus, a grooming path has a capacity that is determined by the capacity of node hardware. Logical link capacity is in fact the capacity of the hardware located at origin and destination of the logical link.

The demand of upper layer is routed by paths in the lower layer and the amount of flow is restricted by the capacity of physical layer links as well as the capacity of logical link from origin to destination that is equal to the capacity of multiplexing/demultiplexing hardware at origin/destination nodes.

4.1.3 Design Motivation: Sequential Design vs. Integrated Design

Notion of logical link brings the complexity of multi-layer networks. This notion, however, makes possible to perform multi-layer network design sequentially from top to bottom for which designing each layer as a single layer by defining each layer's demand as the capacity of its the upper neighbor layer. Sequential design procedure is used for multi-layer network design until some recent studies propose the integrated multi-layer network design methods. Sequential design is tractable and computationally easier than the integrated design but there are some drawbacks of sequential design which are listed below [7]:

- Two logically link-disjoint paths found by sequential design do not need to be physically link-disjoint, and sequential design violates survivability conditions.
- If survivability constraints are taken into account, feasible solutions may not be identified by sequential approach. In Figure 10, logical and physical layers of a two-layered graph are presented and suppose that the numbers given on the links are the cost of installing that link and that 1 unit of flow has to be routed with 1+1 protection mechanism from C to B, i.e. two physically link-disjoint paths from C to B are needed to cope with this survivability mechanism. Sequential approach fails to find two physically link-disjoint paths in this example:
    - First, logical layer is solved to optimality first without any restriction on the selected paths in logical layer.
    - The optimal logical layer solution is to route the flow using the two parallel paths consisting of edges C-A and A-B since selecting direct logical link from C to B is more expensive.
    - After finding optimal solution in logical layer, physical layer is solved such that it must realize the optimal logical layer solution and meet the survivability constraint.

- However, the optimal logical layer solution cannot be routed by link disjoint paths on the physical network.

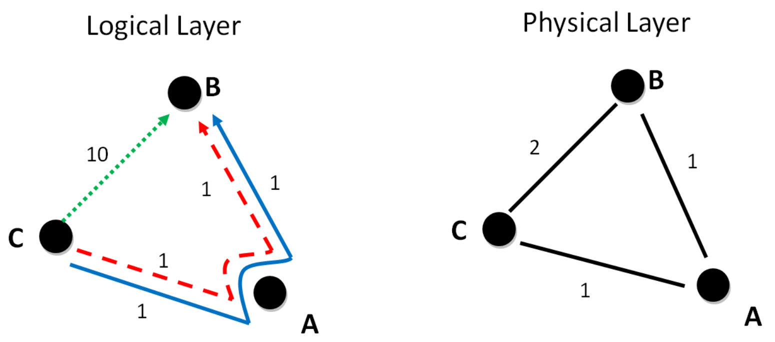


Figure 10. Feasible Solutions May Not Be Identified with Sequential Approach with Survivability Constraints

- The cost value found by sequential design may not be optimal. Suppose that the numbers given on the links are the cost of installing that link and that 1 unit of flow has to be routed from C to B in Figure 11. Let the logical link C-B is realized by physical link C-B, B-A is realized by physical link B-A, and logical link C-A is realized by physical links C-D and DA. Sequential design fails to find the optimal value:
  - First, optimal solution to logical layer is found without any information about physical layer. The optimal solution is to route 1 unit flow from C to B is to install C-A logical link, instead of using C-B and B-A links in logical layer.
  - Then, the optimal solution of logical layer is used to find a solution is physical layer. Since the C-A logical link is realized by C-D and D-A physical links, whose total cost is 11, the total cost found by the sequential design is 12. However, total cost of selecting logical links C-B and B-A is 4.

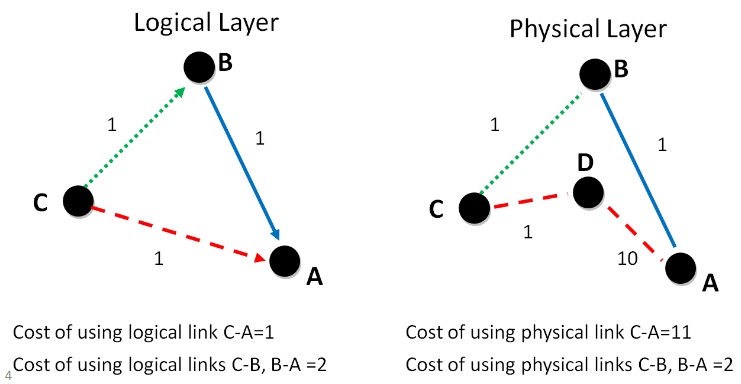


Figure 11. The Cost Value Found by Sequential Design May Not Be Optimal

- Coordination of routings in different layers in sequential design is important. If it cannot be done sufficiently, it may lead to excess capacity to be installed for some physical links and some links to be overused resulting in delays or increase in failure probability.

4.1.4 Grooming Trade-off

The cheapest solution for logical network is to route traffic by only one logical link from source to destination (without grooming) since logical link capacities are modular and the cost of a logical link depends on its capacity rather than its length [117].

- Trade-off 1 (valid for single layer networks too):
  - Trade-off is between adding a module with node capacity to a logical link and adding the origin node a grooming hardware that will send the excess flow to another layer.
- Trade-off 2 (only valid for multi-layer networks) : Indicates the importance of integrated solution of layers for a minimum cost network
  - Adding an over capacitated module or an extra module to a logical link may not be realizable for physical layer unless some spare capacity is added to the physical layer
  - The trade-off is between the cost of adding more capacity to the physical link and having more than one logical link in logical layer between origin-destination pairs by grooming (hence, adding hardware to the links).

4.2 Existing Multi-layer Graph Representation

The multi-layer representation used in the literature is based on the fact that the capacity of any particular layer, that meets its demand, becomes the demand for its lower layer. This representation uses "logical links" concept, i.e. existence of a logical link between a pair of nodes means that there exists a path between these node pair in the lower layer. Let's illustrate this logical link concept for an IP network. The IP network is comprised of several layers. These layers and their mapping to traffic and transport networks are shown in Figure 12.

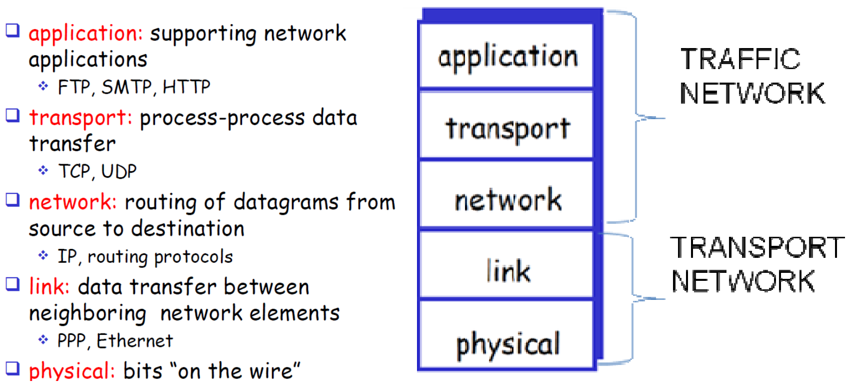


Figure 12. IP Network Stack [245]

Suppose that an e-mail is sent from A to B on an IP network. It follows that there must be a logical link from A to B at the very first layer. Once the email is sent, it is converted to one or more IP packages. At the nodes of the network, packages are processed from layer to layer and transmitted through the network. For the IP network, each work-in-process has a name like message, segment, datagram, and frame. Returning to our analogy of air travel example, these correspond to ticket, baggage, etc. Process of converting IP packages from layer to layer is called encapsulation. Each time the flow goes from upper layer to lower layer, it is encapsulated by the lower layers technology (i.e., multiplexing) and at destination node, the reverse process is done (i.e., demultiplexing). Logical and physical links can be represented as in Figure 13.

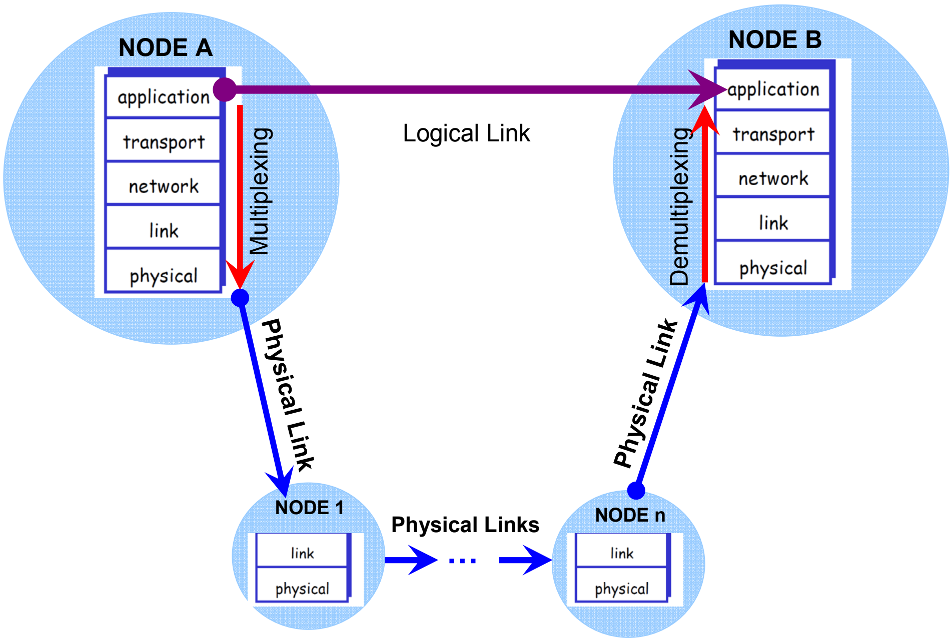


Figure 13. Multi-layer Routing

The existing multi-layer network representation is a multi-network model (MNM) that represents each layer by a distinct network. Each link in a network layer corresponds to a path in its lower layer. Capacity needed to route demand for each layer becomes the demand for its lower layer. In that sense, MNM representation is appropriate for sequential network design, since for each network layer the designer knows that the links in his/her layer are realized by the lower layer network somehow once they give their needed capacity to the network designer of the lower layer.

Let's give an example for MNM showing routing in multi-layer networks. Suppose that the network has 3 layers, the physical network consists of nodes $I=\{A, B, C, D\}$ and links $E=\{\{A, D\}, \{D, B\}, \{B, C\}\}$ .The node set is the same in all layers. The flow has to be routed from A to C in the uppermost layer. The different routing schemes are presented in Figure 14. In Figure 14, it is observed that although routing the flow in layer 1 (physical layer) is the same for all alternatives, routing in the other layers changes:

- In routing alternative 1 and 2, the flow is routed from A to C by a single logical link in layer 3.
- The logical link from A to C is realized by a logical link from A to C in layer 2 in routing alternative 1 while A-B-C path consisting of two links in layer 2 realizes the logical link A-C in layer 3 of routing alternative 2.
- In routing alternative 3, the flow is routed over a path instead of a direct link from A to C in layer 3. A-D in layer 3 is realized by a link A'-D' in layer 2 while path B-C in layer 3 is realized by a path D'-B'-C' in layer 2.

Alternative routings in Figure 14 show that even the physical layer is the same there are several routing alternatives in the upper layers. These variation leads alternative network designs.

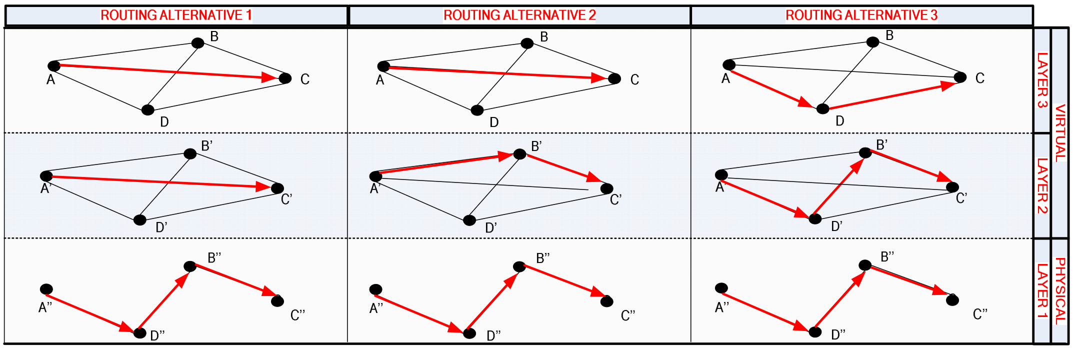


Figure 14. Alternative Routings for Demand From A to C

4.3 A New Graph Representation and Mathematical Model

4.3.1 A New Graph Representation

The physical topology of a multi-layer network given in Figure 14 is presented in Figure 15. The physical network consists of nodes having the devices for routing, switching and multiplexing/demultiplexing operations; and the links corresponding to fiber cables between the nodes.

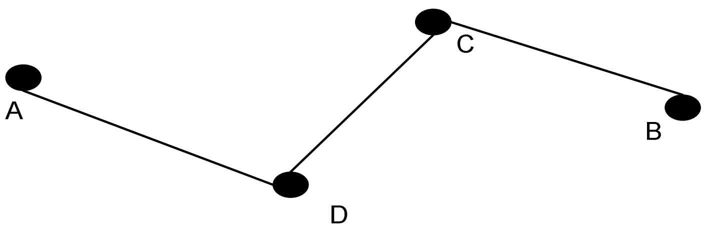


Figure 15. Physical Network

A multi-layer network is a multi-level and multi-technology network, that for optical networks, at most one grooming path corresponding to the given wavelength should traverse the link for any link and wavelength [117]. It implies that different lightpaths sharing a common fiber must have different wavelengths [7] and this physical network or the multi-layer network can be represented by a generalized multicommodity network flow problem:

The multi-layer network design problem is defined on an undirected graph $G = (I, E)$ where I is the set of nodes such that $I = \{i : i = 1,...,n\}$ where n is the number of potential node locations, and E is the set of links that are potential for installing fiber optic cables such that $E = \{\{i, j\} : i \in I \ and \ j \in I\}$. Let L is the set of logical layers such that physical layer is the base layer (layer zero) and $L = \{l : l = 1,...,| L |\}$.

- Nodes in set I represent the potential locations where hardware needed for bandwidth and wavelength conversion, and routing/switching processes are placed. Each node has several devices.
- There are two types of conversion done by hardware at nodes for optical networks i.e. networks who transmit signals via fiber optic cables:
  - Bandwidth conversion (multiplexing/demultiplexing): Signals with lower granularity are combined to get high granularity signals. This brings gain (loss) factors for flow routed through the network to convert high (low) granularity flow to low (high) granularity flow. A node can consist of more than one type of hardware that can convert one technology to other, so that gain/loss factors for a single node might not be unique. Gain/loss factors are denoted as $\gamma^{i}_{ll'}$ for hardware installed on node $i \in I$ that converts signals from l to l' where, $l, l' \in L$. These conversions are done according to the technology; hence the gain/loss factor is calculated according to multiplexing hierarchy of the network.
  - Wavelength conversion: Each lightpath has to be routed on a different wavelength in a single fiber. This is provided by either using wavelength conversion or two consecutive bandwidth conversions. Wavelength conversion does not imply gain/loss on the flow. Observe the situation presented in Figure 16(a). A-C link is routed by two virtual links in layer 2. This means, a wavelength conversion is needed at B'' node in layer 3 in order to change the wavelength. In WDM networks, there are two alternative solutions for this situation [7]:

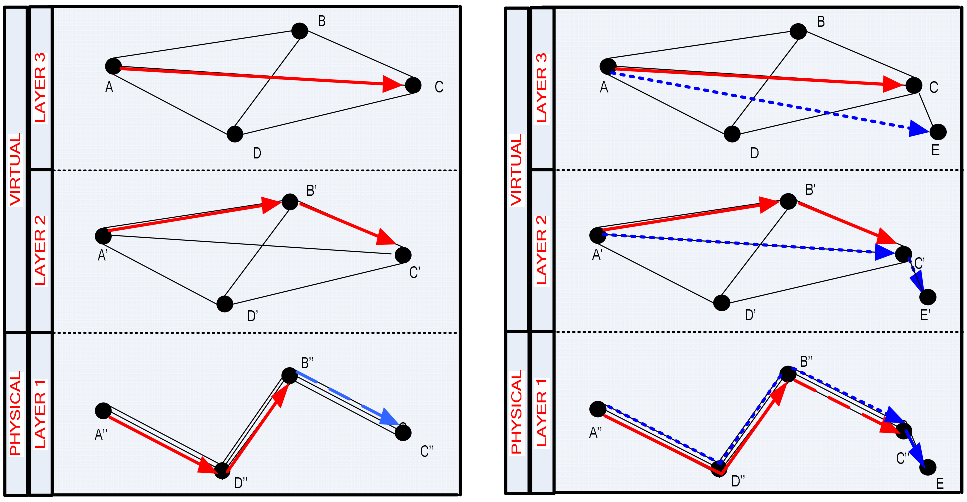
    > "To avoid that two lightpaths use the same wavelength on any fiber, lightpath signals can be sent to the EXC[26], converted into an electrical signal, and recreated using a different wavelength. Alternatively, if no EXC is required for grooming at that node, wavelength converters can be connected to the OXC[27] to perform this task at lower cost and without opto-electronic conversion."

  In Figure 16, the reason for virtual link of A-C to be routed by two lightpaths i.e. A'-B' and B'-C' instead of one single lightpath from A'-C', might be existence of another lightpath between A'-C' and adding the flow needed for routing demand of A-C exceeds the capacity of the fiber optic link. Here, there is a trade-off. In (b), A-C is routed using two lightpaths with an expense of an extra node hardware located at node B. Instead, fiber optic cables might be added to the physical network.

---

[26] EXC: Electrical Cross-Connect: EXC's are devices that provide translation between electrical and optical signals. They perform grooming and switching. EXC's are used in SDH networks and has similar function with routers in MPLS or switches in ATM.

[27] OXC: Optical Cross-Connect: These devices switch lightpath signals from an incoming port to an outgoing port. OXC devices are used in WDM. Different from EXC's, OXC's only perform switching. Different devices called multiplexer and demultiplexers are needed for grooming. Alternatively, OADMs (Optical Add Drop Multiplexer) combines switching and multiplexing/demultiplexing functions.



(a) (b)

Figure 16. Routing Example

- Some technologies have more than one type of facilities such that one technology may support more than one bandwidth. In this case, each bandwidth is represented by a single layer. Thus, layers are different either in terms of technology or in terms of bandwidth. Multi-level networks deal with having different bandwidths with single technology and in multi-level networks; location and node connection problems are solved by using variations of the Steiner tree problem. Multi-level problems mainly are solved for local access networks, that involve trees and bandwidth conversion is done in a single direction, i.e., concentration. Different from multi-level networks, multi-layer networks involve bandwidth conversion in both ways and solves telecommunication network topology, dimensioning, and routing problems jointly.
- Links represent the transmission environment between the nodes such as fiber optic cables, copper cables, etc. Suppose that we are dealing with WDM network consisting of point-to-point fiber optic cables. Each fiber can transmit up to 40 or 80 signals at the same time. The rule of thumb is that **a fiber optic cable cannot route two lightpaths with the same wavelength at the same time.**
- Demand of the network is the point-to-point communication requests of nodes.
- Flow between nodes is the amount of signals routed on the fiber optic cables of physical network.
- Flow and demands of the network is given in terms of base units of flow. For example, let routing unit (r^l) of the specific layer and on layer l. A flow of f base units means f / r^l units of flow for the technology given in the layer.
- The point-to-point communication demands constitute commodities and for each layer a different commodity type can be defined so that $K^l, l \in L$, is the set of commodities. For each commodity $k \in K^l$, $d_k \in Z_+$ is demand value in units of r^l, i.e., total amount of traffic demanded is d_k in base units; s^k is the source node and t^k is the sink node and, s^k and t^k are both in the same layer. s^k includes hardware to multiplex the demand into the technology of the physical layer and route the traffic to the next node, and t^k includes hardware to demultiplex the flow from the technology of the physical

network to the layer l's technology. For the same type of commodities, i.e. for $k \in K^l$, the demand values having the same origin and destination nodes may either be aggregated into one commodity or kept as parallel commodities, e.g. when the commodity has to be protected.

- Capacity of nodes is the capacity of the node hardware such as multiplexers, demultiplexers, wavelength converters, etc. The hardware is different in terms of technology. In addition, different types of hardware can be used for a particular technology. For example, hardware of the same technology may have different capacity and cost. A node may consist of different hardware having different technologies. For example, a node may consist of a multiplexer, a demultiplexer, and an OXC while it can have different types of EXC's that convert signals from one technology to other and perform grooming. Installable hardware on the nodes may vary according to the layer, technology and node.
- Cost of the hardware located at the nodes constitutes the node costs.
- Capacity of links is the number of wavelengths that can be transmitted simultaneously by a single fiber. Since fiber optic cables can transmit up to 40 or 80 wavelengths at the same time, there are two types of capacity modules installable on links.
- Cost of links is the corresponding cost of the capacity module installed on link.

With the above structure, a network flow model (NFM), which uses a single-mega network to model all of the network layers instead of distinct networks for each layer, can be applied to the multi-layer telecommunication networks. The nodes can be splitted such that each node denotes a single device belonging to one technology in order to represent the multi-layer telecommunication networks resembling Balakrishnan et al.'s multi-technology local access representation [137]. The network topology comparison is given for the existing multi-layer network representation that uses a multi-network model and the network flow representation that we propose, NFM, is given in Figure 17.

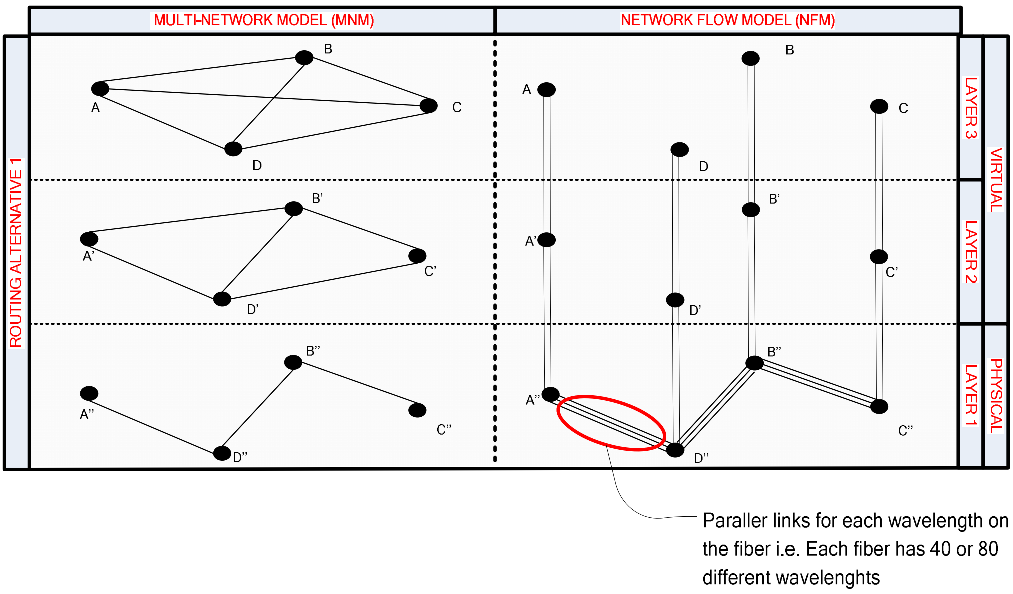


Figure 17. Comparison of Network Topology

Some examples of routing using The NFF instead of MNM are presented in Figure 18.

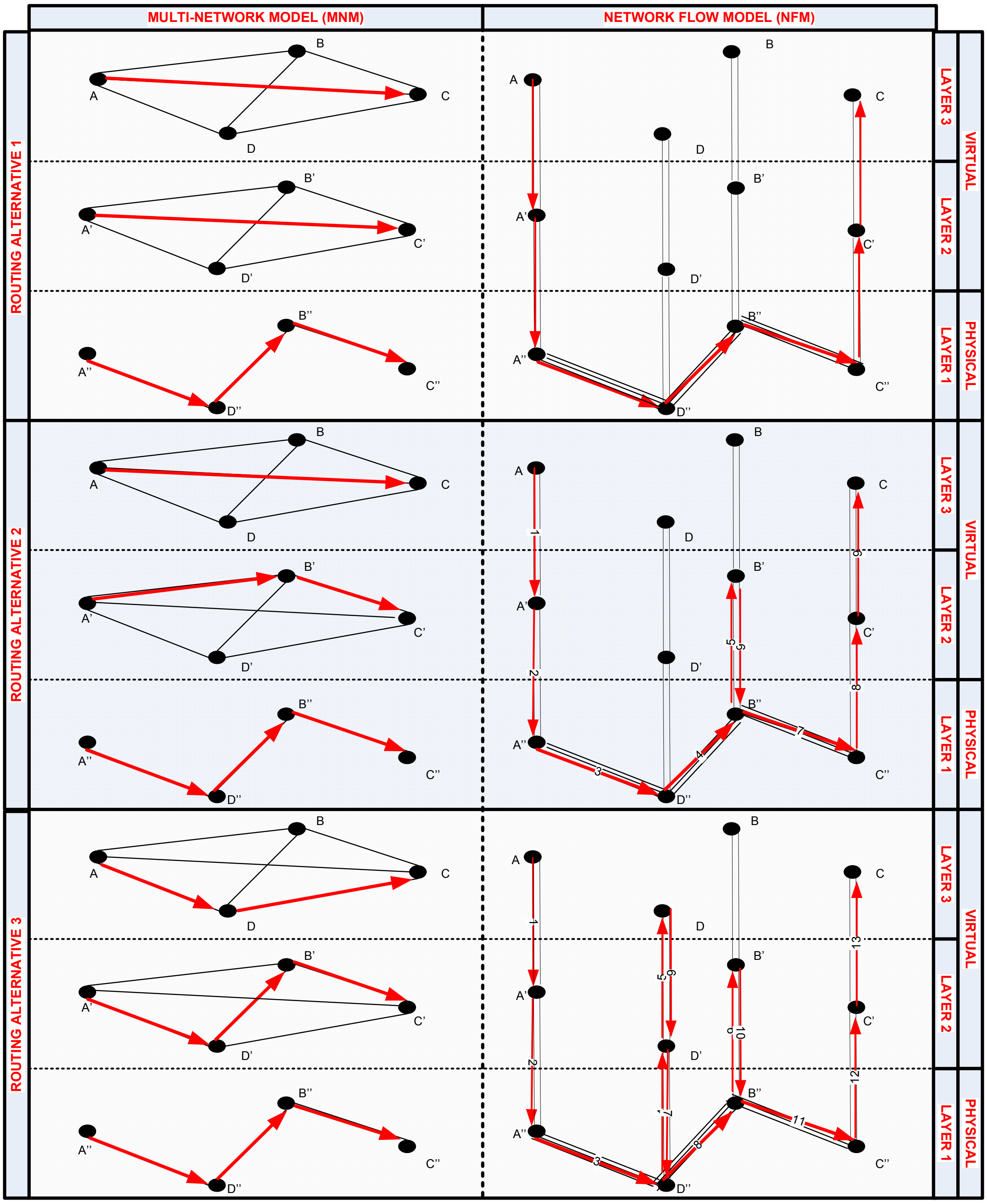
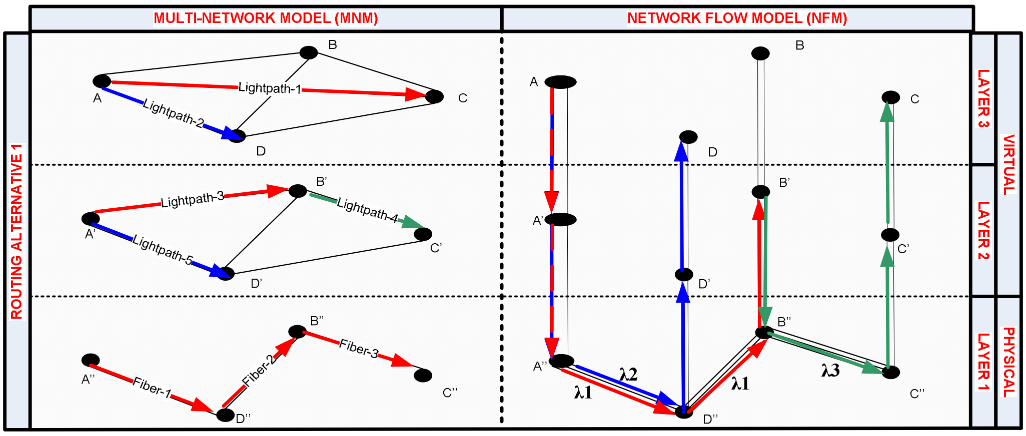


Figure 18. Comparison of Routing

Suppose the links are grouped as (i) processor links (i.e., links between nodes for different technologies such that A-A’, A’-A’’) and (ii) transmission links (i.e., links representing the fiber optic cables such that A’’-D’’, D’’-B’’ and B’’-C’’). The challenge using this new network representation is to apply the rule of thumb of optical networks. **A fiber optic cable cannot route two lightpaths with the same wavelength at the same time as**

presented in Figure 19**.** The consequences of this rule depend on the type of telecommunication network:



-Lightpath-1 is realized by lightpath 3 and Lightpath-4 in Layer 2
-Lightpath-2 is realized by lightpath 5 in Layer 2.
-Lightpath 3 is realized by Fiber-1 and Fiber-2 in Layer 3
-Lightpath 4 is realized by Fiber-3 in Layer 3
Each ligthpath consumes exactly one wavelength in a fiber:
-Lightpath 3 is routed on Fiber-1 and Fiber-2 via wavelength λ1
-Lightpath 5 is routed on Fiber-1 via wavelength λ2
-Lightpath 4 is routed on Fiber-3 via wavelength λ3
-λ1 must be different from λ2

Figure 19. Lightpath Routing

- For WDM networks (electro-optical networks i.e. links are optical while nodes are electrical), two flows having at least one uncommon processor link have to be routed using different wavelengths in the fibers they share. So that, two different flows having at least one uncommon processor link must be routed on different transmission links between the same nodes. The mathematical model has to guarantee that for each link, number of such kind of flows representing different lightpaths traversing a single link does not exceed the number wavelengths provided by a single fiber.
- For all optical networks, if the flow on the lowest layer goes to a node in another layer i.e. the flow goes from a transmission link to any processor link, its wavelength must be changed when the flow revisits transmission links. On the other hand, the wavelength of transmission links between two consecutive processor links in the routing path must have the same wavelength. This is called "lightpath routing" and introduces a very difficult problem of wavelength assignment to be solved together with routing and dimensioning problems. So that, for all optical networks, checking the number of lightpaths against the number of available wavelengths on the link is not enough. A conflict free wavelength assignment must be done with locating the wavelength converters. The routing and wavelength assignment problem is not valid if all nodes can make wavelength conversion. For the networks that have no wavelength conversion ability, it must be guaranteed that a lightpath uses the same wavelength from its beginning node to the end node and each lightpath in a fiber uses different wavelengths. Many WDM networks have sparse wavelength conversion ability. However, modeling sparse wavelength conversion increases difficulty of the formulation [128]. In that case, the wavelength assignment and wavelength converter

location problem is usually solved after a solution is found to the multi-layer network design problem [7].

The twelve-node "polska" network taken from SNDLIB is given as an example for G in Figure 20.

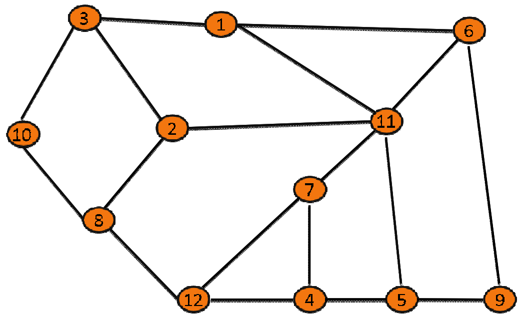


Figure 20. Polska Network

In the physical network in Figure 20, nodes include several devices with several cost and capacity levels, and several functions. So that node splitting is applied to separate these attributes and to assign them to links. Node splitting is applied to graph G and the original arcs in G are transformed from undirected links to directed arcs. Hence, a new graph is obtained.

Suppose we applied the transformations to $G=(N,E)$ in Figure 20. $G'=(I,A)$ is the transformed graph such that I is the set of nodes where $I=\{\{i,l,t\} \mid i \in N, l \in L$ and $t=1,2\} \cup N$ and A is the set of arcs where $A=\{(i,j) \mid i,j \in I\}$. $G'=(I,A)$ is presented in Figure 21. Note that, t indicates whether the node is a multiplexer node, i.e., $t=1$ or a demultiplexer node, i.e., $t=2$. Since the attributes of hardware are assigned to links now, links can be classified according to the function they represent.

4.4 Network Flow Formulation (NFF) for the MLNDP

We propose a mathematical formulation based on NFM, which we call the network flow formulation (NFF), since the NFM uses a single network and models all flows of the MLNDP using this network unlike the existing MNM representation that uses a distinct network for each layer. The sets, parameters and decision variables of NFF are presented below:

Sets:

N is the set of nodes in physical telecommunication network such that $N=\{i : i=1,\ldots,n\}$ where *n* is the number of potential node locations.

E is the set of potential links of physical telecommunication network such that $E=\{\{i,j\} : i \in I$ and $j \in I\}$

I is the set of nodes where $I=\{\{i,l,t\} \mid i \in N, l \in L$ and $t=1,2\} \cup N$

A is the set of arcs where $A=\{(i,j) \mid i,j \in I\}$.

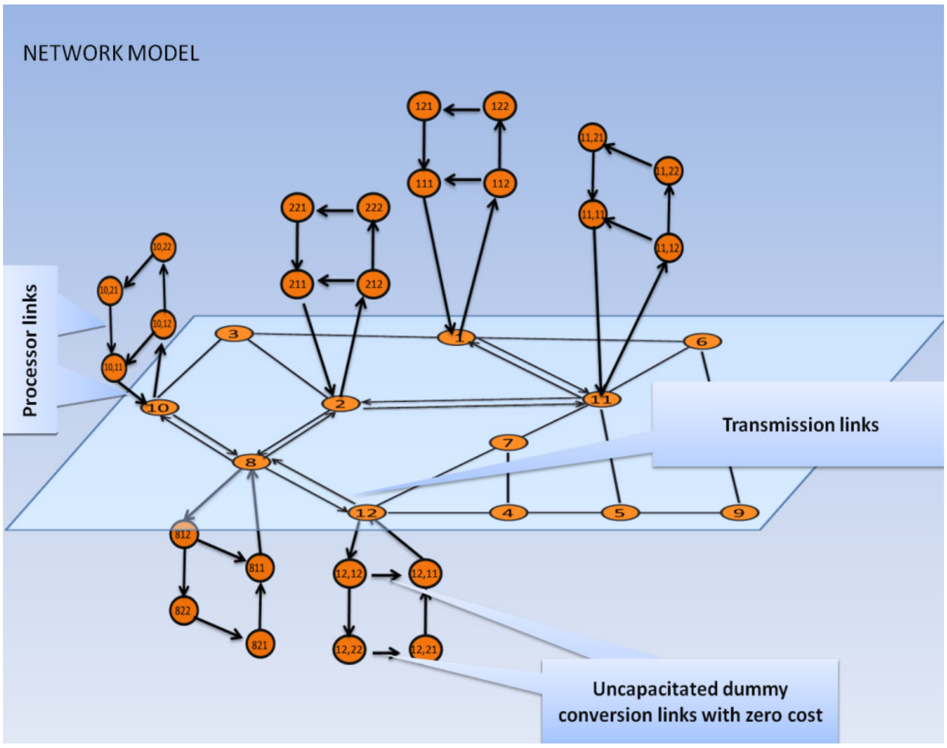


Figure 21. NFM Applied to Polska Network

L is the set of logical layers where $L = \{l : l = 1, ..., |L|\}$ and physical layer is the base layer (layer zero)

K is the set of commodities, that each commodity is characterized by source node s^k, sink node t^k, origination layer l^k, and demand value in base units d_i^k where $i \in I$

M_M is the set of available multiplexer devices that can be used at nodes to convert the signals from l to l' where $l > l'$, $l \in L \cup \{0\}$, and $l = 0$ is the physical layer

M_D is the set of available demultiplexer devices that can be used at nodes to convert the signals from l to l' where $l < l'$, $l \in L \cup \{0\}$, and $l = 0$ is the physical layer

M_R is the set of available routers/switches that can be used at node $i \in N$

M_F is the set of available link modules that can be installed at arc (i, j) where $i, j \in N$

Parameters:

γ_l is the conversion rate, taken as base unit equivalent of one routing unit of layer l ($\gamma_l > 1$)

$q_{ill'}^{1,m}$ is the capacity for multiplexers/demultiplexers installed at node i to convert the signals from l to l' where $l > l'$ if $m \in M_M$ and $l < l'$ if $m \in M_D$

$q_i^{2,m}$ is the capacity for routers/switches that can be used at node i, $\forall i \in N$ and $\forall m \in M_R$

$q_{ij}^{3,m}$ is the number of wavelengths in a single fiber of type $\forall m \in M_F$

Decision Variables:

f_{ij}^{pr} is the total number of wavelengths needed between nodes i and j such that $(i,j) \in A$ to carry flow that is firstly processed at node p ($p \neq j$) and secondly processed by r ($r \neq i$) where $p, r \in N, p \neq r$.

$x_{ill'}^{k}$ equals to 1 if the commodity $k \in K$ is routed over processor link at node i to convert the signals from l to l' where $l > l'$ and 0 otherwise.

$x_{ij}^{k,p,r}$ equals to 1 if commodity $k \in K$ is routed via transmission links between nodes i and j that is firstly processed at node p ($p \neq j$) and secondly processed by r ($r \neq i$) where $i, j, p, r \in N$ and $k \in K$, else 0.

$Y_{ill'}^{m}$ is the number of multiplexer/demultiplexer devices installed on node i at level l to convert signals from layer l to l' where $m \in M_M$ if $l > l'$ or $m \in M_D$.

W_i^m is the number of routers/switches installed on node i of capacity module type $m \in M_R$.

U^{pr} is the number of needed wavelengths for routing commodities that are firstly processed at node p ($p \neq j$) and secondly processed by r ($r \neq i$) where $p, r \in N, p \neq r$.

V_{ij}^{m} is the number of link modules of type $m \in M_F$ needed for routing the total number of wavelengths needed between nodes i and j

Let $G^O = (N, E)$ be the original graph composed of potential node locations $N = \{i : i = 1, ..., n\}$ and potential links $E = \{\{i, j\} : i \in I \ and \ j \in I\}$ between nodes $i \in N$. Let $G^T = (I, A)$ be the transformed graph composed of nodes $I = \{\{i, l, t\} \mid i \in N, l \in L \ and \, t = 1, 2\} \cup N$ and arcs $A = \{(i, j) \mid i, j \in I\}$. The transformed graph is presented in Figure 22.

The transformed graph can be decomposed into two different graphs representing processors (multiplexers and demultiplexers) of the telecommunication network installed at its nodes, and routers and fiber optical links. The latter part is presented in the box in Figure 22 and the former part is the remaining graph. Notice that, processor part of the network also decomposes into |N| distinct networks. This decomposition lets us to rewrite the problem in terms of |N| multicommodity flow problems.

After such decomposition, flow variables of the arcs that the red line intercepts in Figure 22 are no longer flow variables, they are demand and supply variables for both newly formed graphs.

Then the following change in the notation makes the formulation easier to understand:

- $x_{i11,i}^{k} = h_{i1}^{k}$ is the flow of commodity $k \in K$ on the multiplexers that convert signals from the lowermost logical layer to the physical layer.
- $x_{i,i12}^{k} = h_{i2}^{k}$ is the flow of commodity $k \in K$ on the demultiplexers that convert signals from the physical layer to the lowermost logical.

Using the graphs formed by the decomposition of network, we get the following formulation:

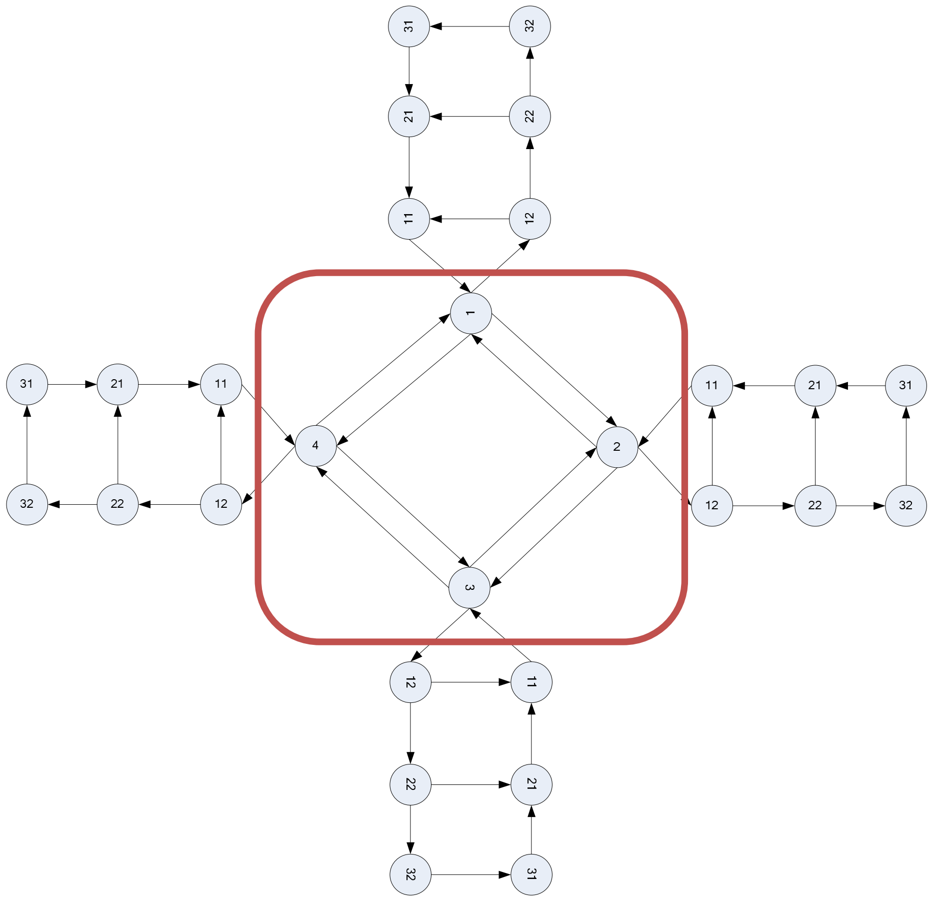


Figure 22. Transformed Graph

- Let $G_P^T = (I^P, A^P)$ where node set is $I^P = \{\{i, l, t\} \mid i \in N, l \in L \text{ and } t = 1,2\}$ and arc set is $A^P = \{(i, j) \mid i, j \in I^P\}$. Note that, in the transformed graph, demand and supply nodes of the commodities remain in this part of the network after decomposition. In addition to that, flow coming from the physical layer acts as supply and flow going to the physical layer acts as demand for them G_P^T. This demand and supply can be seen in Figure 23.

  Let φ^P be the node arc incidence matrix of G_P^T. Then, we can write the constraints flow balance constraints of this processor network as

  $$\varphi^P \boldsymbol{X}^k = \boldsymbol{b}^k \quad \forall k \in K \qquad (1.1)$$

  where

  - $\boldsymbol{X}^k$ is a column vector with length of $|A^P|$ and its elements are ordered by the same as arcs given in φ^P. Hence, the flow variable corresponding to arc $(ilt, il't)$ in $\boldsymbol{X}^k$ column vector is x_{ij}^k.

  - $\boldsymbol{b}^k$ is a column vector of length $|I^P|$.

  For $l > 1$:

$$\boldsymbol{b}^k(ilt) = \begin{cases} 1, & if\ d_{ilt} > 0 \\ -1, & if\ d_{ilt} < 0 \\ 0, & if\ d_{ilt} = 0 \end{cases}$$

For $l = 1$

$$\boldsymbol{b}^k(il1) = \begin{cases} 1 - h_{i1}^k, & if\ d_{ilt} > 0 \\ -h_{i1}^k, & if\ d_{ilt} = 0 \end{cases},$$

$$\boldsymbol{b}^k(il2) = \begin{cases} -1 + h_{i2}^k, & if\ d_{ilt} < 0 \\ h_{i2}^k, & if\ d_{ilt} = 0 \end{cases}$$

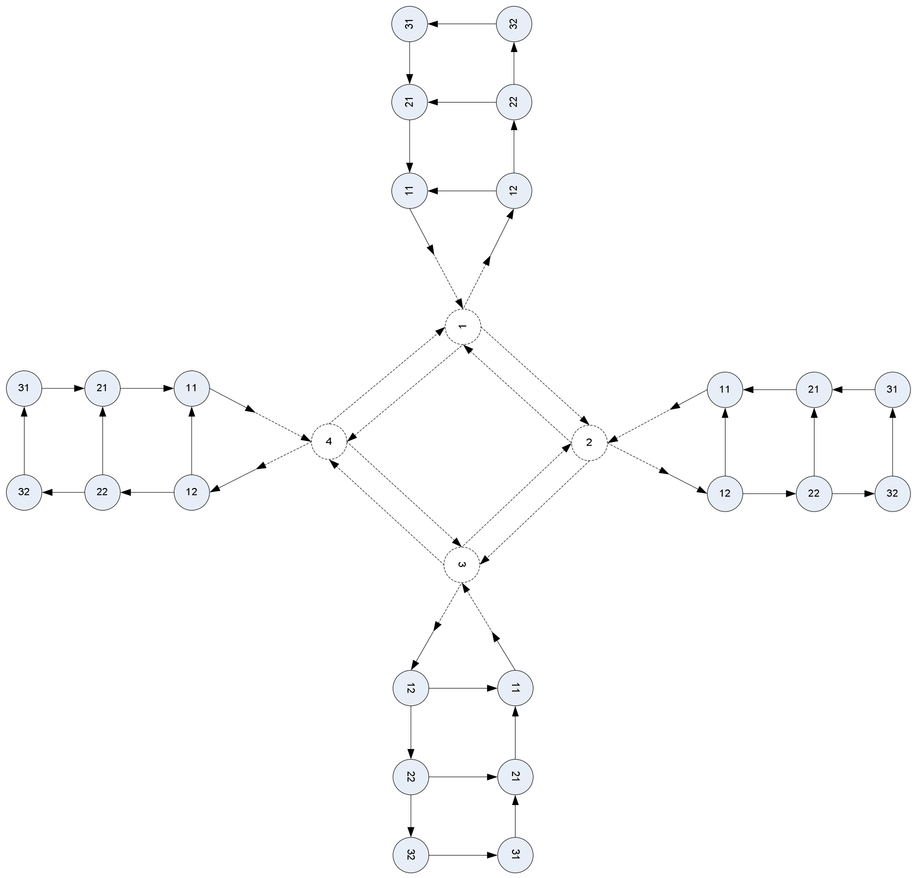


Figure 23. Processor Network

Capacity constraints for processors are

$$\sum_{k \in K} \boldsymbol{X}^k d_k \leq \sum_{m \in M} \boldsymbol{Q}^{1,m} \boldsymbol{Y}^m \qquad (1.2)$$

where

- d_k is the demand value of commodity $k \in K$,

- $\boldsymbol{Y}^m$ is a column vector with length equal to $|A^P|$ and entry of this column vector associated to arc $(ilt, il't') \in A^P$ is $y_{ill't}^m$ such that $t = 1\ if\ l = l'+1$ and $t = 2\ if\ l = l'-1$

- $\boldsymbol{Q}^{1,m}$ is a diagonal matrix with size equal to $|A^P| \text{x} |A^P|$ and diagonal entry of this matrix associated to arc $(ilt, il't') \in A^P$ is $q_{ll't}^{1,m}\gamma_l$ such that $t=1$ if $l=l'+1$ and $t=2$ if $l=l'-1$

- Let $G_T^T = (I^T, A^T)$ where node set is $I^T = N$ and arc set is $A^T = \{(i,j) \mid i,j \in N\}$. Flow on each arc $(i,j) \in A^T$ is indexed according to its first processor ($p \in I^T$) and last processor node in the transmission network ($r \in I^T$). Each commodity $k \in K$ is decomposed into sub-commodities between these processor nodes. Though, the value of demand of these sub-commodities are not known apriori, total demand value of commodities that share a common sink (source) node is equal to that particular node's supply (demand) value. Because, the amount of flow that is transmitted to virtual network at any node $i \in I^T$ is equal to sum of flows whose last processed node is i and amount of flow that is transmitted from virtual network to physical network at node $i \in I^T$ is equal to sum of flow whose first processed node is i. In addition, node $i \in I^T$ serves as a transshipment node for flows having i as neither last processed nor first processed node. Thus, there is another multicommodity flow problem in transmission network where all possible combinations of nodes serve as origin-destination pairs. The transmission network is illustrated in Figure 24.

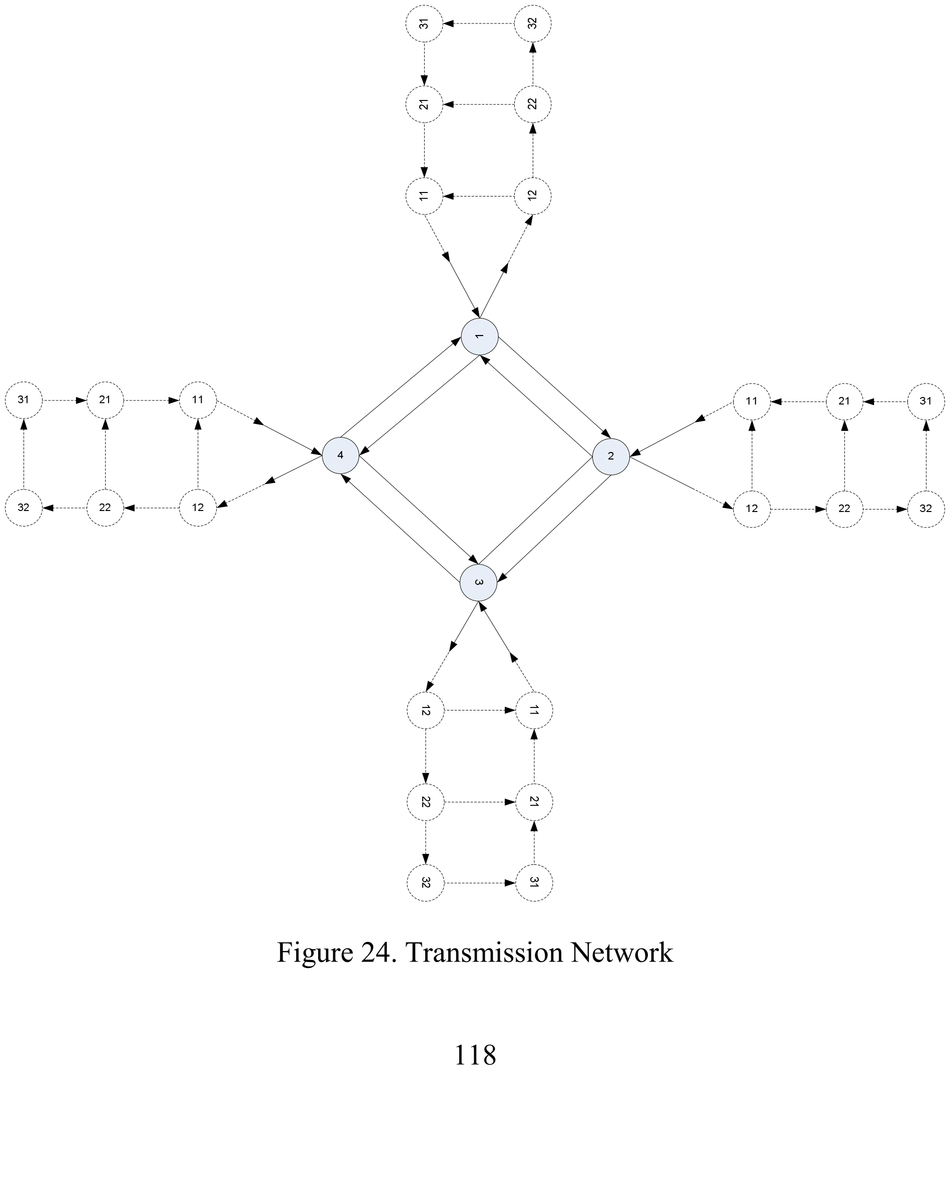


Figure 24. Transmission Network

Let $\boldsymbol{\varphi}^T$ be the node arc incidence matrix of G_T^T. The flow balance equation of the transmission network, G_T^T is:

$$\boldsymbol{\varphi}^T \boldsymbol{X}^{k,(p,r)} = \boldsymbol{b}^{k,(p,r)} \qquad \forall k \in K \ and \ \forall (p,r) \in N \times N : p \neq r \qquad (1.3)$$

where

- $\boldsymbol{X}^{k,(p,r)}$ is a column vector with length of $|A^T|$ and its elements are ordered by the same as arcs given in φ^T. Hence, the flow variable corresponding to arc (i,j) in $\boldsymbol{X}^{k,(p,r)}$ column vector is $x_{ij}^{k,p,r}$ for $\forall (i,j) \in A$ such that $i \neq r$ and $j \neq p$.
- $\boldsymbol{b}^{k,(p,r)}$ is a column vector of length $|I^T|$.

$$\boldsymbol{b}^{k,(p,r)}(i) = \begin{cases} -b_i^{kpr}, & if \quad p = i \\ b_i^{kpr}, & if \quad r = i \\ 0, & otherwise \end{cases}$$

We do not know the exact values of b_i^{kpr} variables explicitly, however we know their sum over p and r for each node $i \in N$ such that:

- Sum of flow leaving physical layer from the node $i \in N$ for a commodity $k \in K$ is sum of flow for that commodity whose last processed node is i in the transmission network:

$$\sum_{r:(p,r) \in N \times N} b_i^{kpr} = h_{i2}^k \qquad \forall i \in I \qquad (1.4)$$

- Sum of entering physical layer from the node $i \in N$ for a commodity $k \in K$ is sum of flow for that commodity whose first processed node is i in the transmission network:

$$\sum_{p:(p,r) \in N \times N} b_i^{kpr} = h_{i1}^k \qquad \forall i \in I \qquad (1.5)$$

The flows on transmission graph are upper-bounded by the number of wavelengths installed on the associated fiber optic cable. Hence,

$$\sum_{k \in K} \boldsymbol{X}^{k,(p,r)} d_k \leq \gamma_1 \boldsymbol{F}^{pr} \qquad \forall (p,r) \in N \times N : p \neq r \qquad (1.6)$$

where,

- $\boldsymbol{F}^{pr}$ is a column vector with length of $|A^T|$ and its elements are ordered by the same as arcs given in φ^T. Hence, the flow variable corresponding to arc (i,j) in $\boldsymbol{F}^{pr}$ column vector is f_{ij}^{pr} for $\forall (i,j) \in A$ such that $i \neq r$ and $j \neq p$.
- γ_1 is a scalar for converting amount of flow in physical layer to number of wavelengths.

- In the physical network, lightpaths are routed over the arcs. Each lightpath emanates from a processor node and terminates at another processor node. In between these nodes, the lightpath remains intact. So that, lightpaths to be routed between each node pair $(p,r) \in N \times N$ constitute a commodity and the lightpath routing problem becomes a multicommodity network flow problem in the transmission graph G_T^T where the flow variables are addressed by number of wavelengths routed on arc $(i,j) \in A^T$, f_{ij}^{pr} and commodity set is equal to $N \times N$. Demand for a commodity $(p,r) \in N \times N$ is the total number of wavelengths needed between nodes p and r denoted by u^{pr}. Then the flow conservation constraint is:

$$\boldsymbol{\varphi}^T \boldsymbol{F}^{pr} = \boldsymbol{U}^{pr} \qquad \forall (p,r) \in N \times N : p \neq r \qquad (1.7)$$

where $\boldsymbol{U}^{pr}$ is a column vector of length column vector with length of $|N|$ and its elements are ordered by the same as nodes given in φ^T such that

$$\boldsymbol{U}^{pr}(i) = \begin{Bmatrix} u^{pr}, & if\ p = i \\ -u^{pr}, & if\ r = i \\ 0, & otherwise \end{Bmatrix}$$

u^{pr} must be at least the total number of flows emanating from node p such that first processor node is p and last processor node is r:

$$\sum_{k \in K} \boldsymbol{X}^{k,(p,r)} \boldsymbol{\alpha}^T d_k \leq \gamma_1 \boldsymbol{U}^{pr} \qquad \forall (p,r) \in N \times N : p \neq r \quad (1.8)$$

where

- α is a column vector with length of $|A^T|$ and its elements are ordered by the same as arcs given in $\boldsymbol{\varphi}^T$ such that $\boldsymbol{\alpha}(ij) = 1$ if $i = p$ and $i \neq r$, else $\boldsymbol{\alpha}(ij) = 0$ for $\forall (i,j) \in A^T$.

The lightpaths are undirected, so that capacity constraint for the multicommodity flow problem for lightpath routing is:

$$\sum_{(p,r) \in N \times N} \boldsymbol{F}^{pr} \boldsymbol{\beta} \leq \sum_{m \in M^F} \boldsymbol{V}^m Q^{3,m} \qquad (1.9)$$

where

- $\boldsymbol{\beta}$ is a matrix with size of $|A^T|$x $|E|$ such that each row corresponds to an arc $\forall (i,j) \in A^T$ and each column corresponds to an edge $\forall \{i,j\} \in E$. $\boldsymbol{\beta}$ consists of zeros except for two entries each column that corresponds to edge $\{i,j\} \in E$ such that:
$\boldsymbol{\beta}(ij) = 1\ if\ i \neq r$ and $j \neq p$ and $\boldsymbol{\beta}(ji) = 1\ \ if\ i \neq p$ and $j \neq r$.
- $\boldsymbol{V}^m$ is a column vector with length equal to $|E|$.
- $\boldsymbol{Q}^{3,m}$ is a diagonal matrix with size equal to $|E|$x$|E|$ whose diagonal entry is $q_{ij}^{3,m}$

- Routers installed on the physical network nodes have a switching capacity:

$$\sum_{(p,r) \in N \times N} \sum_{k \in K} \boldsymbol{X}^{k,(p,r)} + \sum_{k \in K} \boldsymbol{H}_2^k \leq \sum_{m \in M} \boldsymbol{Q}^{2,m} \boldsymbol{W}^m \qquad (1.10)$$

- Multiplexers of type $m \in M$ that converts signals from the lowermost logical layer to the physical layer has a conversion capacity:

$$\sum_{k \in K} \boldsymbol{H}_1^k d_k \leq \sum_{m \in M_1^P} \boldsymbol{Q}_1^{1,m} \boldsymbol{Y}_1^m \qquad (1.11)$$

where

- $\boldsymbol{Y}_1^m$ is a column vector with length equal to $|N|$; each entry of this column vector is associated to node $i \in N$ and hence associated to arc $(i11, i) \in A$ such that $\boldsymbol{Y}_1^m(i) = y_{i11,i}^m$
- $\boldsymbol{Q}_1^{1,m}$ is a diagonal matrix with size equal to $|N|$x$|N|$; diagonal entry of this matrix associated to node $i \in N$ and hence associated to arc $(i11,1) \in A$ such that

$$\boldsymbol{Q}_1^m(i,i) = q_{i11,i}^m \gamma_l,$$

- Demultiplexers of type $m \in M$ that converts signals from the physical layer to the lowermost logical layer has a conversion capacity:

$$\sum_{k \in K} \boldsymbol{H}_2^k d_k \leq \sum_{m \in M_2^P} \boldsymbol{Q}_2^{1,m} \boldsymbol{Y}_2^m \tag{1.12}$$

  where

  - $\boldsymbol{Y}_2^m$ is a column vector with length equal to $|N|$ and each entry of this column vector is associated to a node $i \in N$ and hence associated to arc $(i, i12) \in A$ such that $\boldsymbol{Y}_2^m(i) = y_{i,i12}^m$

  - $\boldsymbol{Q}_2^{1,m}$ is a diagonal matrix with size equal to $|N|$x$|N|$ and diagonal entry of this matrix associated to node $i \in N$ and hence associated to arc $(i, i12) \in A$ such that

$$\boldsymbol{Q}_2^{1,m}(i,i) = q_{i,i12}^{1,m} \gamma_l,$$

- Adding nonnegativity and integrality constraints completes the formulation:

$$\boldsymbol{X}^{k,m}, \boldsymbol{X}^{k,(p,r)}, \boldsymbol{H}_1^{k,m}, \boldsymbol{H}_2^{k,m} \geq 0 \tag{1.13}$$

$$\boldsymbol{F}^{pr}, \boldsymbol{U}^m, \boldsymbol{V}^m, \boldsymbol{W}^m, \boldsymbol{Y}^m, \boldsymbol{Y}_1^m, \boldsymbol{Y}_2^m \in \mathbb{Z}^+ \tag{1.14}$$

The objective function of the problem is to minimize the installation cost of the network:

$$z = \sum_{m \in M^P} (\boldsymbol{C}^{1,m})^T \boldsymbol{Y}^m + \sum_{m \in M_1^P} (\boldsymbol{C}_1^{1,m})^T \boldsymbol{Y}_1^m + \sum_{m \in M_2^P} (\boldsymbol{C}_2^{1,m})^T \boldsymbol{Y}_2^m + \sum_{m \in M^R} (\boldsymbol{C}^{2,m})^T \boldsymbol{W}^m + \sum_{m \in M^F} (\boldsymbol{C}^{3,m})^T \boldsymbol{V}^m \tag{1.15}$$

Where the first term refers to the total installation cost of multiplexers and demultiplexers for conversion between logical layers, the second term is the total cost of multiplexers for conversion from lowermost logical layer to physical layer, the third term is the total cost of demultiplexers for conversion from physical layer to lowermost logical layer, fourth term is the total cost of routers, and last term is total cost of fiber optic cables.

Then the complete formulation is given below.

Minimize

$$z = \sum_{m \in M^P} (\boldsymbol{C}^{1,m})^T \boldsymbol{Y}^m + \sum_{m \in M_1^P} (\boldsymbol{C}_1^{1,m})^T \boldsymbol{Y}_1^m + \sum_{m \in M_2^P} (\boldsymbol{C}_2^{1,m})^T \boldsymbol{Y}_2^m + \sum_{m \in M^R} (\boldsymbol{C}^{2,m})^T \boldsymbol{W}^m + \sum_{m \in M^F} (\boldsymbol{C}^{3,m})^T \boldsymbol{V}^m \tag{1.15}$$

subject to

Flow Balance Constraints

$$\boldsymbol{\varphi}^P \boldsymbol{X}^k = \boldsymbol{b}^k \qquad \forall k \in K \tag{1.1}$$

$$\sum_{k \in K} \boldsymbol{X}^k d_k \leq \sum_{m \in M} \boldsymbol{Q}^{1,m} \boldsymbol{Y}^m \tag{1.2}$$

$$\boldsymbol{\varphi}^T \boldsymbol{X}^{k,(p,r)} = \boldsymbol{b}^{k,(p,r)} \qquad \forall k \in K, \forall (p,r) \in N \times N : p \neq r \tag{1.3}$$

$$\sum_{r:(p,r)\in N\times N} b_i^{kpr} = h_{i2}^k \qquad \forall k \in K, \forall i \in I \qquad (1.4)$$

$$\sum_{p:(p,r)\in N\times N} b_i^{kpr} = h_{i1}^k \qquad \forall k \in K, \forall i \in I \qquad (1.5)$$

$$\sum_{k\in K} \boldsymbol{X}^{k,(p,r)} d_k \le \gamma_1 \boldsymbol{F}^{pr} \qquad \forall (p,r) \in N \times N : p \ne r \qquad (1.6)$$

$$\boldsymbol{\varphi}^T \boldsymbol{F}^{pr} = \boldsymbol{U}^{pr} \qquad \forall (p,r) \in N \times N : p \ne r \qquad (1.7)$$

Capacity Constraints

$$\sum_{k\in K} \boldsymbol{X}^{k,(p,r)} \boldsymbol{\alpha}^T d_k \le \gamma_1 \boldsymbol{U}^{pr} \qquad \forall (p,r) \in N \times N : p \ne r \qquad (1.8)$$

$$\sum_{(p,r)\in N\times N} \boldsymbol{F}^{pr} \boldsymbol{\beta} \le \sum_{m\in M^F} \boldsymbol{V}^m Q^{3,m} \qquad (1.9)$$

$$\sum_{(p,r)\in N\times N} \sum_{k\in K} \boldsymbol{X}^{k,(p,r)} + \sum_{k\in K} \boldsymbol{H}_2^k \le \sum_{m\in M} \boldsymbol{Q}^{2,m} \boldsymbol{W}^m \qquad (1.10)$$

$$\sum_{k\in K} \boldsymbol{H}_1^k d_k \le \sum_{m\in M_1^P} \boldsymbol{Q}_1^{1,m} \boldsymbol{Y}_1^m \qquad (1.11)$$

$$\sum_{k\in K} \boldsymbol{H}_2^k d_k \le \sum_{m\in M_2^P} \boldsymbol{Q}_2^{1,m} \boldsymbol{Y}_2^m \qquad (1.12)$$

Nonnegativity and Integrality Constraints

$$\boldsymbol{X}^{k,m}, \boldsymbol{X}^{k,(p,r)}, \boldsymbol{H}_1^{k,m}, \boldsymbol{H}_2^{k,m} \ge 0 \qquad \forall k \in K, \forall m \in M_M, \forall m \in M_D, \; \forall (p,r) \in N \times N : p \ne r \qquad (1.13)$$

$$\boldsymbol{F}^{pr}, \boldsymbol{U}^m, \boldsymbol{V}^m, \boldsymbol{W}^m, \boldsymbol{Y}^m, \boldsymbol{Y}_1^m, \boldsymbol{Y}_2^m \in \mathbb{Z}^+ \qquad \forall (p,r) \in N \times N : p \ne r, \forall m \in M_F, \; \forall m \in M_R, \forall m \in M_M, \forall m \in M_D, \qquad (1.14)$$

4.4.1 Modifications on the NFF

In order to get a more compact formulation, we can eliminate some of the variables and constraints by replacement. Note that, a flow whose last processed node $r \in N$ cannot emanate from node r or a flow whose first processed node $p \in N$ cannot enter to node p unless there is a loop in the physical network. Since loops are not desired, such kind of flows is not allowed and therefore such variables are not defined in the formulation. Using this information, we can aggregate constraints (1.3)-(1.5) by substituting b_i^{kpr} values in constraints (1.4) and (1.5) with their equivalences in constraints (1.3). Then we get the following constraints:

$$\sum_{k\in K} X^{k,(p,r)}\alpha_1^T = \sum_{k\in K} H_1^k \qquad \forall (p,r)\in N\times N : p\neq r$$

$$\sum_{k\in K} X^{k,(p,r)}\alpha_2^T = \sum_{k\in K} H_2^k \qquad \forall (p,r)\in N\times N : p\neq r$$

$$\boldsymbol{\varphi}^T(\alpha_3^T \boldsymbol{X}^{k,(p,r)}) = \mathbf{0} \qquad \forall k\in K, \forall (p,r)\in N\times N : p\neq r$$

where

- α_1 is a column vector with length of $|A^T|$ and its elements are ordered by the same as arcs given in φ^T such that $\alpha(ij)=1$ if $i=p$ and $i\neq r$, else $\alpha(ij)=0$ for $\forall (i,j)\in A^T$.
- α_2 is a column vector with length of $|A^T|$ and its elements are ordered by the same as arcs given in φ^T such that $\alpha(ij)=1$ if $j=r$ and $j\neq p$, else $\alpha(ij)=0$ for $\forall (i,j)\in A^T$.
- α_3 is a column vector with length of $|A^T|$ and its elements are ordered by the same as arcs given in φ^T such that $\alpha(ij)=1$ if $i\neq r, i\neq p$ or $j\neq r, j\neq p$ else $\alpha(ij)=0$ for $\forall (i,j)\in A^T$.

Hence, we can replace $\sum_{k\in K} H_1^k$ ($\sum_{k\in K} H_2^k$) with $\sum_{k\in K} X^{k,(p,r)}\alpha_1^T$ ($\sum_{k\in K} X^{k,(p,r)}\alpha_2^T$) in the formulation and get rid of flow variables related to processor edges from physical layer to the lowermost virtual layer.

After those modifications, the resulting formulation in open form is presented below:
Minimize

$$\begin{aligned} z = &\sum_{i\in I}\sum_{l\in L\setminus\{0\}}\sum_{m\in M_M^{il,l-1}} c_{il,l-1}^{1,m} Y_{il,l-1}^m + \sum_{i\in I}\sum_{l\in L\setminus\{|L|-1\}}\sum_{m\in M_D^{il,l+1}} c_{il,l+1}^{1,m} Y_{il,l+1}^m \\ &+ \sum_{i\in I}\sum_{m\in M_R^i} c_i^{2,m} W_i^m + \sum_{(i,j)\in A}\sum_{m\in M_F} c_{ij}^{3,m} V_{ij}^m \end{aligned} \tag{2.1}$$

subject to

Flow Balance Constraints

$$x_{i,l,l-1}^k - x_{i,l+1,l}^k - x_{ill}^k = \begin{cases} 1, if\ s^k = i\, and\ l^k = l \\ 0, otherwise \end{cases} \quad \forall l\in L\setminus\{0,|L|-1\}, \forall i\in N\, and\, \forall k\in K \tag{2.2}$$

$$x_{i,l-1,l}^k - x_{i,l,l+1}^k - x_{ill}^k = \begin{cases} 1, if\ t^k = i\, and\ l^k = l \\ 0, otherwise \end{cases} \quad \forall l\in L\setminus\{0,|L|-1\}, \forall i\in N\, and\, \forall k\in K \tag{2.3}$$

$$x_{i,0,1}^k = \sum_{\substack{j\in N:\\ (j,i)\in A^T}}\sum_{\substack{p\in N\\ p\neq i}} x_{ji}^{k,p,i} \qquad \forall i\in N, \forall k\in K \tag{2.4}$$

$$x^k_{i,1,0} = \sum_{\substack{j\in N:\\(i,j)\in A^T}} \sum_{\substack{r\in N\\r\neq i}} x^{k,i,r}_{ij} \qquad \forall i\in N, \forall k\in K \tag{2.5}$$

$$\sum_{\substack{j\in N:\\(j,i)\in A^T,\\j\neq r}} x^{k,p,r}_{ji} = \sum_{\substack{j\in N:\\(i,j)\in A^T,\\j\neq p}} x^{k,p,r}_{ij} \qquad \forall k\in K, \forall i\in N \ and \ \forall (p,r)\in NXN \tag{2.6}$$

$$\sum_{\substack{j:(i,j)\in A^T,\\i\neq r, j\neq p}} f^{pr}_{ij} - \sum_{\substack{j:(j,i)\in A^T\\j\neq r, i\neq p}} f^{pr}_{ji} = \begin{cases} U^{pr}, & if \ i=p \\ -U^{pr}, & if \ i=r \quad \forall i\in I \ and \ \forall (p,r)\in N\times N \mid p\neq r \\ 0, & otherwise \end{cases} \tag{2.7}$$

Capacity Constraints

$$\sum_{k\in K} d^k x^k_{i,l,l-1} \le \gamma_l \sum_{m\in M_M} q^{1,m}_{i,l,l-1} Y^m_{i,l,l-1} \qquad \forall i\in N \ and \ \forall l\in L\setminus\{0\} \tag{2.8}$$

$$\sum_{k\in K} d^k x^k_{i,l,l+1} \le \gamma_l \sum_{m\in M_D} q^{1,m}_{i,l,l+1} Y^m_{i,l,l+1} \qquad \forall i\in N \ and \ \forall l\in L\setminus\{|L|-1\} \tag{2.9}$$

$$\sum_{k\in K} \sum_{j:(i,j)\in A^T} \sum_{\substack{(p,r)\in N\times N:\\p\neq r, p\neq j, r\neq i}} d^k x^{k,p,r}_{ij} + \sum_{k\in K} d^k x^k_{i,0,1} \le \sum_{m\in M_R} q^{2,m}_i W^m_i \qquad \forall i\in N \tag{2.10}$$

$$\sum_{\substack{(p,r)\in N\times N:\\p\neq r, p\neq j, r\neq i}} f^{p,r}_{ij} + \sum_{\substack{(p,r)\in N\times N:\\p\neq r, p\neq i, r\neq j}} f^{p,r}_{ji} \le \sum_{m\in M_F} q^{3,m}_{ij} V^m_{ij} \qquad \forall \{i,j\}\in E \tag{2.11}$$

$$\sum_{i:(p,i)\in A^T} \sum_{k\in K} d^k x^{k,p,r}_{pi} \le \gamma_1 U^{pr} \qquad \forall (p,r)\in N\times N \mid p\neq r \tag{2.12}$$

$$\sum_{k\in K} d^k x^{k,p,r}_{ij} \le \gamma_1 f^{p,r}_{ij} \qquad \forall i,j\in N \mid (i,j)\in A \, and \ \forall (p,r)\in N\times N \mid p\neq r \tag{2.13}$$

$$x^{kpr}_{ij} \le 1 \, \forall k\in K, \qquad \forall (i,j)\in A \ and \ \forall (p,r)\in NXN \mid p\neq r, p\neq j, r\neq i \tag{2.14}$$

$$x^k_{i,l,l'} \le 1 \qquad \forall l,l'\in L\setminus\{0,|L|-1\} \mid l'=\{l-1,l,l+1\}, \forall i\in N \, and \ \forall k\in K \tag{2.15}$$

Nonnegativity and Integrality Constraints

$$0\le x \tag{2.16}$$

$$U,V,W,Y,f\in \mathbb{Z}^+ \tag{2.17}$$

where

(2.2) and (2.3) are flow balance constraints for processor links such as multiplexers and demultiplexers, respectively.

(2.4) assures that for each node $i \in N$ only $x_{ij}^{k,p,i}$ can be transmitted to upper layers and the other flows are routed to another node.

(2.5) assures that for each node $i \in N$ only $x_{ij}^{k,i,r}$ can be transmitted from upper layers to the other nodes.

(2.6) assures that for each node $i \in N$, $i \neq p$ and $i \neq r$, $x_{ij}^{k,p,r}$ is transmitted to another node in layer 0.

(2.8)- (2.10) are capacity constraints for multiplexers, demultiplexers and routers respectively.

(2.12) is a set of constraints for the number of wavelengths needed between processor nodes (p,r) $p,r \in N, p \neq r$.

(2.13) give the lower bound for number of wavelengths needed between nodes i and j such that arc $(i,j) \in A^T$ carries flow that is firstly processed at node p ($p \neq j$) and secondly processed by r ($r \neq i$) where $p,r \in N, p \neq r$.

(2.11) is a set of constraints for the number of link modules needed to route the wavelengths.

(2.7) is a set of constraints the flow conservation constraints for number of wavelengths routed between (p,r) $p,r \in N, p \neq r$.

(2.14) - (2.15) impose upper bound on flow variables, since flow variables indicate the ratio of the commodity that is routed.

(2.16) and (2.17) are integrality and nonnegativity constraints.

We can still improve the formulation by removing equations (2.4) and (2.5) by replacing $x_{i,0,1}^k$ and $x_{i,1,0}^k$ with $\sum_{\substack{j \in N: \\ (j,i) \in A^T}} \sum_{\substack{p \in N \\ p \neq i}} x_{ji}^{k,p,i}$ and $\sum_{\substack{j \in N: \\ (i,j) \in A^T}} \sum_{\substack{r \in N \\ r \neq i}} x_{ij}^{k,i,r}$, respectively. Then we get the following formulation:

Minimize

$$\begin{aligned} z = &\sum_{i \in I} \sum_{l \in L \setminus \{0\}} \sum_{m \in M_M^{il,l-1}} c_{il,l-1}^{1,m} Y_{il,l-1}^m + \sum_{i \in I} \sum_{l \in L \setminus \{|L|-1\}} \sum_{m \in M_D^{il,l+1}} c_{il,l+1}^{1,m} Y_{il,l+1}^m \\ &+ \sum_{i \in I} \sum_{m \in M_R^i} c_i^{2,m} W_i^m + \sum_{(i,j) \in A} \sum_{m \in M_F} c_{ij}^{3,m} V_{ij}^m \end{aligned} \tag{3.1}$$

subject to

Flow Balance Constraints

$$x_{i,l,l-1}^k - x_{i,l+1,l}^k - x_{ill}^k = \begin{Bmatrix} 1, if \;\; s^k = i \, and \;\; l^k = l \\ 0, otherwise \end{Bmatrix} \qquad \forall l \in L \mid 1 < l <\mid L \mid, \quad \forall i \in N \, and \, \forall k \in K \tag{3.2}$$

$$\sum_{\substack{j \in N: \\ (i,j) \in A^T}} \sum_{\substack{r \in N \\ r \neq i}} x_{ij}^{k,i,r} - x_{i,2,1}^{k} - x_{i11}^{k} = \begin{cases} 1, if \ \ s^k = i \, and \ \ l^k = 1 \\ 0, otherwise \end{cases} \qquad \forall i \in N \, and \, \forall k \in K \tag{3.3}$$

$$x_{i,l-1,l}^{k} - x_{i,l,l+1}^{k} - x_{ill}^{k} = \begin{cases} 1, if \ \ t^k = i \, and \ \ l^k = l \\ 0, otherwise \end{cases} \qquad \forall l \in L \mid 1 < l <\mid L \mid, \quad \forall i \in N \, and \, \forall k \in K \tag{3.4}$$

$$\sum_{\substack{j \in N: \\ (j,i) \in A^T}} \sum_{\substack{p \in N \\ p \neq i}} x_{ji}^{k,p,i} - x_{i,1,2}^{k} - x_{i1}^{k} = \begin{cases} 1, if \ \ t^k = i \, and \ \ l^k = 1 \\ 0, otherwise \end{cases} \qquad \forall i \in N \, and \, \forall k \in K \tag{3.5}$$

$$\sum_{\substack{j \in N: \\ (j,i) \in A^T}} x_{ji}^{k,p,r} = \sum_{\substack{j \in N: \\ (i,j) \in A^T}} x_{ij}^{k,p,r} \qquad \forall k \in K, i \in N \ \ and \ \ \forall (p,r) \in NXN \tag{3.6}$$

$$\sum_{\substack{j:(i,j) \in A^T, \\ i \neq r, j \neq p}} f_{ij}^{pr} - \sum_{\substack{j:(j,i) \in A^T, \\ j \neq r, i \neq p}} f_{ji}^{pr} = \begin{cases} U^{pr}, & if \ \ i = p \\ -U^{pr}, & if \ \ i = r \\ 0, & otherwise \end{cases} \qquad \forall i \in I \ \ and \ \ \forall (p,r) \in N \times N \mid p \neq r \tag{3.7}$$

Capacity Constraints

$$\sum_{k \in K} d^k x_{i,l,l-1}^{k} \leq \gamma_l \sum_{m \in M_M} q_{i,l,l-1}^{1,m} Y_{i,l,l-1}^{m} \qquad \forall i \in N \ \ and \ \ \forall l \in L \mid 1 < l <\mid L \mid \tag{3.8}$$

$$\sum_{k \in K} \sum_{\substack{j \in N: \\ (i,j) \in A^T}} \sum_{\substack{r \in N \\ r \neq i}} d^k x_{ij}^{k,i,r} \leq \gamma_l \sum_{m \in M_M} q_{i,1,0}^{1,m} Y_{i,1,0}^{m} \qquad \forall i \in N \tag{3.9}$$

$$\sum_{k \in K} d^k x_{i,l,l+1}^{k} \leq \gamma_l \sum_{m \in M_D} q_{i,l,l+1}^{1,m} Y_{i,l,l+1}^{m} \qquad \forall i \in N \ \ and \ \ \forall l \in L \setminus \{\mid L \mid -1\} \tag{3.10}$$

$$\sum_{\substack{j \in N: \\ (j,i) \in A^T}} \sum_{\substack{p \in N \\ p \neq i}} d^k x_{ji}^{k,p,i} \leq \gamma_l \sum_{m \in M_D} q_{i,0,1}^{1,m} Y_{i,0,1}^{m} \qquad \forall i \in N \tag{3.11}$$

$$\sum_{k \in K} \sum_{j:(i,j) \in A} \sum_{\substack{(p,r) \in N \times N: \\ p \neq r, p \neq j, r \neq i}} d^k x_{ij}^{k,p,r} + \sum_{k \in K} \sum_{\substack{j \in N: \\ (j,i) \in A^T}} \sum_{\substack{p \in N \\ p \neq i}} d^k x_{ji}^{k,p,i} \leq \sum_{m \in M_R} q_i^{2,m} W_i^m \qquad \forall i \in N \tag{3.12}$$

$$\sum_{\substack{(p,r) \in N \times N: \\ p \neq r, p \neq j, r \neq i}} f_{ij}^{p,r} + \sum_{\substack{(p,r) \in N \times N: \\ p \neq r, p \neq i, r \neq j}} f_{ji}^{p,r} \leq \sum_{m \in M_F} q_{ij}^{3,m} V_{ij}^{m} \qquad \forall \{i,j\} \in E \tag{3.13}$$

$$\sum_{i:(p,i)\in A^T} \sum_{k\in K} d^k x_{pi}^{k,p,r} \leq \gamma_1 U^{pr} \qquad \forall (p,r)\in N\times N \mid p\neq r \qquad (3.14)$$

$$\sum_{k\in K} d^k x_{ij}^{k,p,r} \leq \gamma_1 f_{ij}^{p,r} \qquad \forall i,j\in N \mid (i,j)\in A^T \; and \; \forall (p,r)\in N\times N \mid p\neq r \qquad (3.15)$$

$$x_{ij}^{kpr} \leq 1 \qquad \forall k\in K, \forall (i,j)\in A^T \;\; and \;\; \forall (p,r)\in NXN \mid p\neq r, p\neq j, r\neq i \qquad (3.16)$$

$$x_{i,l,l'}^{k} \leq 1 \qquad \forall l,l'\in L\setminus\{0,|L|-1\} \mid l'=\{l-1,l,l+1\}, \forall i\in N \; and \;\; \forall k\in K \qquad (3.17)$$

Nonnegativity and Integrality Constraints

$$x \geq 0 \qquad (3.18)$$

$$U,V,W,Y,f \in \mathbb{Z}^+ \qquad (3.19)$$

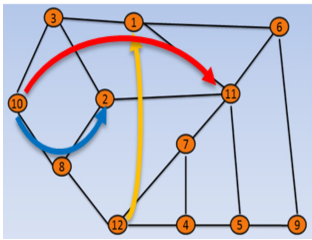
4.5 Discussion on NFF

Suppose we are given the twelve-node polska network of Figure 20 from SNDLIB [238] and we need to model ATM-over-SDH-over-WDM network using NFF. Traffic flows are taken in terms of base units in the network according to the NFM. Base unit equivalent of one unit flow for each layer changes according to the technology used or the type of flow granularity that the technology uses. For example, one unit flow for ATM network is equal to four units of base flow while; one unit flow for WDM network is equal to 4032 units of base flow. There can be different granularities within a technology, such as SDH that has different so-called "virtual containers" for transmitting flow. Each virtual container type has different base unit equivalents. In the example, we will use just one virtual container type for SDH. The base unit equivalents for one unit of flow for each technology in the example network are given in Table 15.

Table 15. Routing Units for Several Technologies

Layer	Base
ATM	4
STM-1 (SDH)	252
WDM	4032

Suppose the demands to be routed are those given in Figure 25.



Commodity	Source node (s)	Sink node (t)	Demand (base units)	Flow on ATM	Flow on STM-1	Flow on WDM
1	10	11	1200	$\lceil 1200/4 \rceil = 300$	$\lceil 1200/252 \rceil = 5$	$\lceil 1200/4032 \rceil = 1$
2	10	2	1000	$\lceil 1000/4 \rceil = 250$	$\lceil 1000/252 \rceil = 4$	$\lceil 1000/4032 \rceil = 1$
3	12	1	1500	$\lceil 1500/4 \rceil = 375$	$\lceil 1500/252 \rceil = 6$	$\lceil 1500/4032 \rceil = 1$

Figure 25. Traffic Demand

Two alter routing for these three commodities are presented in Figure 26.

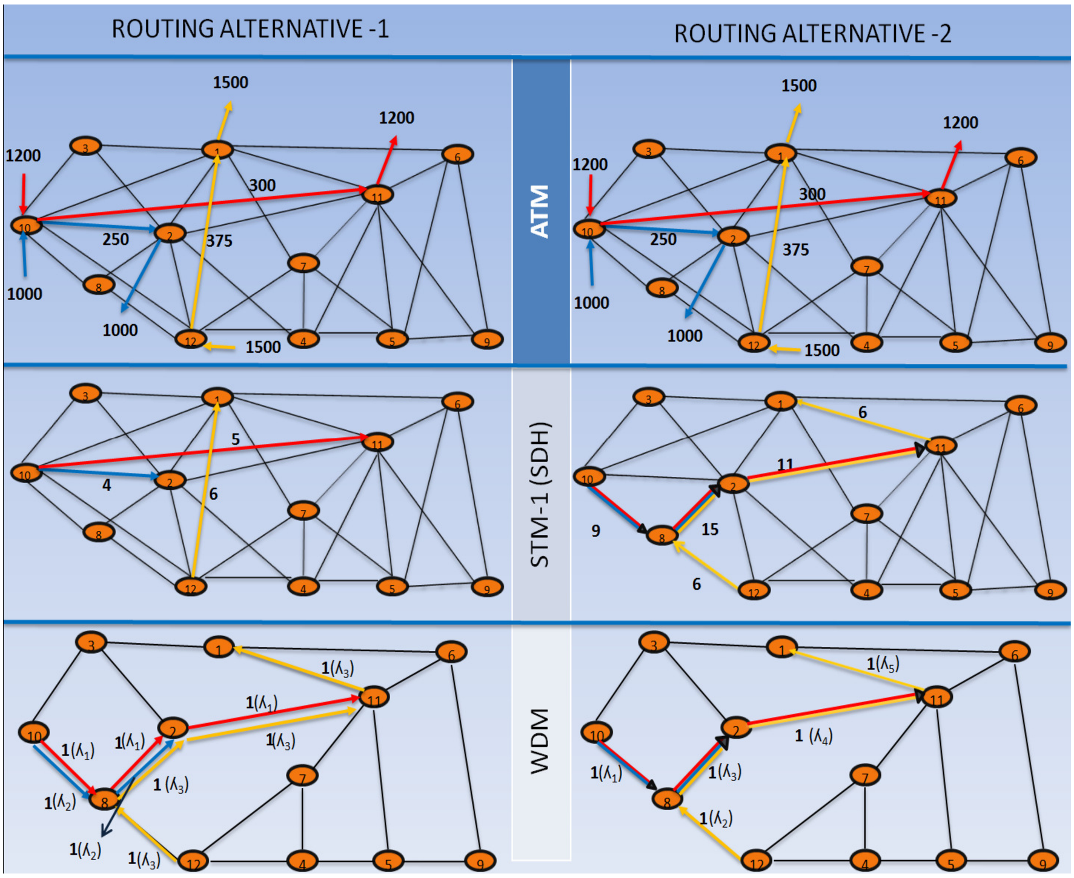


Figure 26. Feasible Routing Alternatives with the Conventional/Existing Multi-layer Network Representation

The corresponding routing for these three commodities with the NFM is shown in Figure 27.

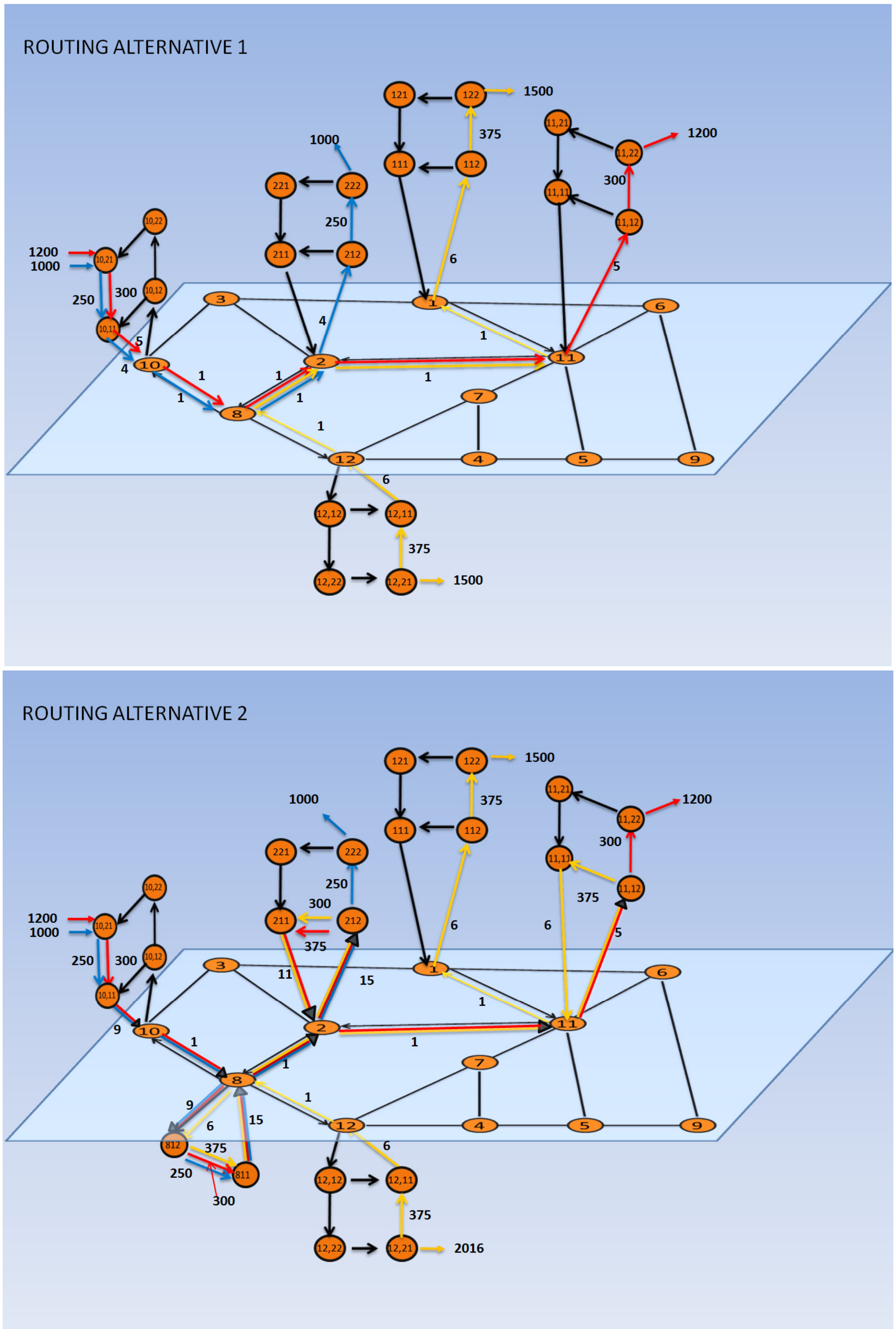


Figure 27. Feasible Routing Alternatives with NFF

Any routing in existing multi-layer network representation can be represented via the NFM in terms of the flows between node pairs and their layers. Thus, unlike the capacity formulation, the NFF gives optimal capacities as well as the corresponding routing of flows. Note that, the notion of "logical links" in the current modeling scheme is modeled in a different way in the NFM. Thus, the routing of flows is interpreted by the NFM in a different way than the current modeling scheme. The difference is clearly illustrated in Figure 28.

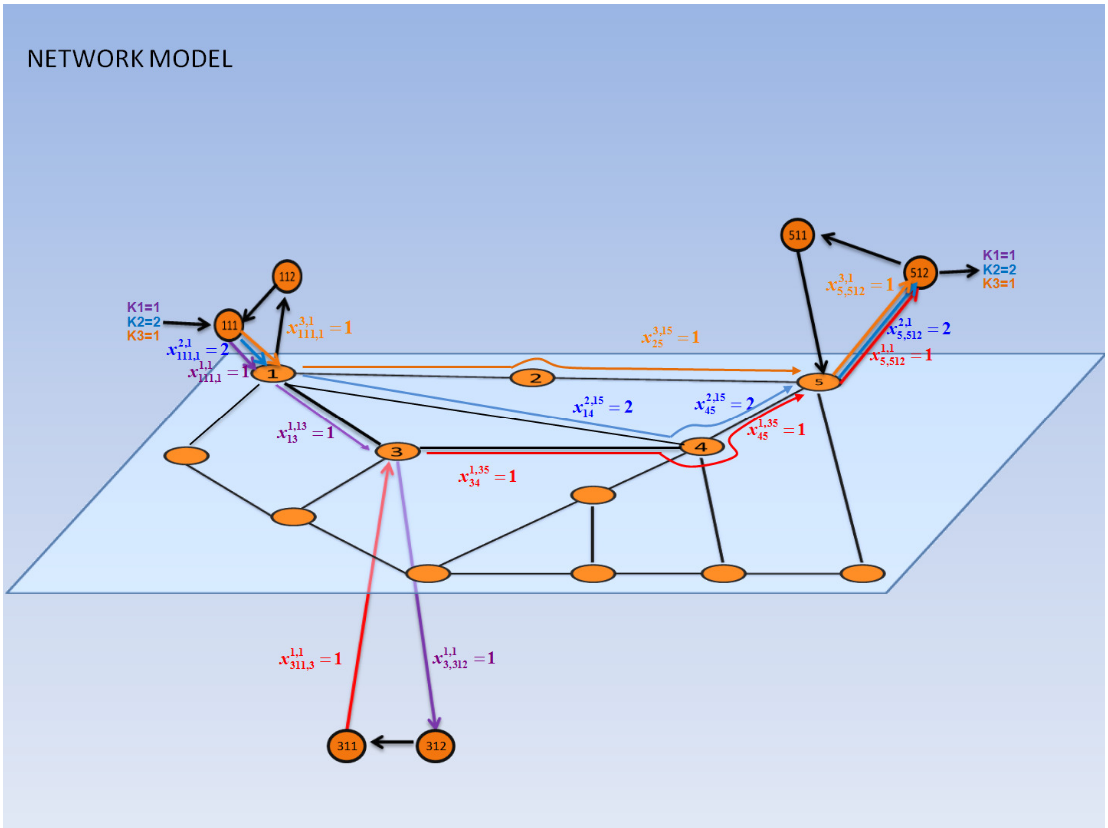


Figure 28. Illustration of NFF Output Routing

According to the instance illustrated in Figure 28:

- There are four logical links in the logical layer: two parallel links between (1, 5), one link between (1, 3) and one link between (3, 5). This information is provided in the physical layer flow variables (*p, r*) indexes such that *p* is the node that the flow is first processed at the upper layers and *r* is the node that the flow is last processed in the upper layers. Between nodes *p* and *r*, the flow is routed only on the transmission links. Flow between the same processing nodes (*p, r*) may be routed via a single lightpath depending on its amount however; two flows between different processing nodes cannot be routed via a single lightpath.
- In addition to (*p, r*), the layers that the traffic flows are known by the amount of flow on the processor links. Hence, capacity and cost of each logical link has a direct match in the NFF as the capacity and cost of interface cards that the logical links are terminated are modeled explicitly by the processor links in the NFM.
- In the NFF, the physical links that each logical link uses is not known a priori. In that sense, the NFF uses the implicit approach to model the logical layer. However, since flow on logical links and the corresponding physical links are given in the NFF's solution, the NFF gives the capacity of logical links explicitly just like the explicit formulation.

- Unlike other formulations using the implicit approach, multiple link failures and node failures can be modeled using the NFF since the physical paths and flows used by each commodity are known explicitly in the NFF.
- Having some loops in the physical layer may decrease the overall cost in the multi-layer networks. For example, suppose that there is a commodity to be routed from node A to C in Figure 29. In the logical layer, there are two logical links; one is between A and B, realized by the blue dashed physical links in the physical layer and the other is between B and C, realized by the green solid physical links in the physical layer. If there is spare capacity to route the commodity from A to B using the logical path A-B-C, routing the commodity without installing a direct logical link between A and C is cheaper than installing the logical link A-C. However, using the logical link A-B-C means, to route the commodity in the physical link on the path A-D-B-D-C. Hence, the physical link D-B is used twice. This situation is practically valid as long as one of the end points of the commodity is traversed more than once, since in that case some demands can reach their destination nodes several times before the path ends. Hence, the physical links that include loops involving one of the end points of any logical link are inadmissible and must be restricted in the mathematical formulation. Orlowski states that explicit flow formulation must be used to model such a restriction since formulations using implicit approach does not know correspondence between the physical paths and logical paths in the expense of increasing the number of variables [7]. In the NFF, using (*p, r*) indexes indicating the first and last nodes of the logical link together with the (*i, j*) indexes indicating the physical link makes it possible to model such a restriction without adding all physical paths corresponding to logical links a priori to the problem. Hence, the NFF acts like using the explicit approach in terms of modeling multi-layer routing without cycles while keeping the number of variables lower than explicit approach. Using Property 1 and Property 2 in the NFF prevent having such kind of practically irrelevant cycles without necessity to know the correspondence between logical links and physical links a priori.

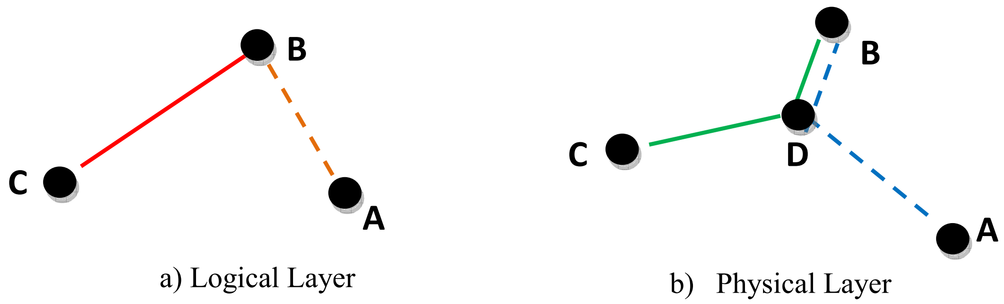


Figure 29. Loops in Physical Layer

Property 1: A commodity originates from its source node cannot turn back to its source node again as it is practically meaningless: $x_{ij}^{k,pr} = 0$ for $j = s^k$

Property 2: A commodity cannot emanate from its destination node as it means the commodity reaches its destination more than once and this is practically meaningless: $x_{ij}^{k,pr} = 0$ for $i = t^k$

- Although the NFF is an implicit edge-flow formulation, hop-limits and physical length constraints can be modeled for the logical links.
- A commodity must be multiplexed at its source node and demultiplexed at its end node. Then, the number of processor edges has a lower bound of greater than zero.

  Property 3: All commodities $k \in K$ with source node i and originate from layer l^k must be multiplexed at node i:

  $Y^m_{i,l^k,l^k-1} \geq \sum_{k:s^k=i} d^k / q^{1,m}_{i,l^k,l^k-1}$ for $\forall i \in I, \forall k \in K$

  Property 4: All commodities $k \in K$ with sink node i and terminate at layer l^k must be demultiplexed at node i:

  $Y^m_{i,l^k-1,l^k} \geq \sum_{k:t^k=i} d^k / q^{1,m}_{i,l^k-1,l^k}$ for $\forall i \in I, \forall k \in K$

In order to model multi-layer telecommunication network problems relevant for practical applications, it is crucial to be able to formulate practical side constraints such as general routing restrictions, i.e. geographical length of a fiber optic cable connection is important for its feasibility; single-path routing, i.e. transport networks use single-path routing in practice; node survivability; multiple link failures; and explicit node cost and capacity. In this respect, explicit formulation approach (EF) is used with edge-flow formulation (EFF) in the literature to develop practically relevant network models. (Section 3.10 is referred for more information about existing models and EFF-EF) However, the model increases exponentially with the EFF models that in some cases make it impossible to even construct the model in the electronic environment. The NFF beats the models other than EFF by being capable to model practical constraints and beats EFF by modeling these constraints without necessity to know all physical paths corresponding logical links a priori.

Telecommunication networks have more than two layers in practice. Since the networks are used by more than one service, technologies and granularity of traffic requests vary. Traffic grooming is used in order to cope with this heterogeneity in granularity and use the network resources more efficiently. Hence, traffic grooming is a practically relevant problem for telecommunication networks that has to be solved jointly with topology design and lightpath routing problems while traffic grooming problem is meaningful for more than two layers. However, edge-flow formulation with explicit approach is not computationally tractable for more than two layers:

- Suppose we have the 12 node polska network with 18 edges.
  - Number of links in the lowest layer (physical layer), layer 1 is 18.
  - Number of links in second layer is 2457, since there are a total of 2457 paths between the nodes of the network given 18 edges. In this layer, each pair of nodes have a minimum of 22 and a maximum of 59 parallel links in between and the graph in this layer is complete different from the physical layer. Hence, apart from parallel links between the node pairs, the graph has 66 distinct edges.
  - Second layer is a complete graph with comb(12,2)=66 different node pairs. Then, in third layer, the longest path has 65 hops and each hop has at least 22 and at most 59 alternatives since in second layer, each pair of nodes have a

minimum of 22 and a maximum of 59 parallel links in between. Then number of links in third layer has an order of magnitude of at least $22^{65} > 10^{87}$.

Then even for a 12-node-network, it is impossible to solve the edge-flow model using explicit approach for a network with more than 2 layers using the existing formulations. Therefore, it is impossible to find even a feasible solution with the existing formulations for a multi-layer telecommunication network that is practically relevant.

For the NFF, increasing the number of layers have a polynomial increase in the network size since adding an additional layer increases the number of nodes and edges by 2|N| and 3|N|, respectively. However, the number of logical layers increases exponentially with the number of nodes. That is, the NFF is both capable of modeling the practical side constraints and solving network instances having more than two layers.

Complexities of the NFF and the EFF-EF model differ from each other in the computational sense. The NFF has $O(|K||E||N|^2)$ constraints and $O(|K||E||N|^2)$ variables, whereas the EFF-EF has $O(|K||E||\mathcal{L}|)$ constraints and O(|L|) variables where $\mathcal{L}$ is the set of logical links and $|\mathcal{L}| >> |N|$ especially when the number of nodes increase.

Comparison of the NFF with the existing formulations is presented in Table 16. Detailed information about the existing formulations is provided in Section 3.10.

Table 16. Comparison of the NFF and Existing Formulations

Modeling Capability	NFF	EFF-EF	EFF-PF	ECF	IFF-EF	IFF-PF	ICF
Admissible physical paths for logical links	√	√	√			√	
General routing restrictions (bound on hops)	√		√			√	
Unsplittable flow (single-path routing)	√	√	√		√	√	
Single link failure by 1+1 protection	√	√	√		√	√	
Multiple link failures by 1+1 protection	√	√	√				
Node failures by 1+1 protection	√	√					
Single link failure by diversification[28]	√	√	√		√	√	

[28] EFF-EF and EFF-PF are equivalent [7]

Table 16 (Cont'd)

Modeling Capability	NFF	EFF -EF	EFF -PF	ECF	IFF-EF	IFF-PF	ICF
Multiple link failures by diversification[29]	√	√	√				
Node failures by diversification	√	√					
Single link failure by failure states	√	√	√	√	√	√	√
Multiple link failures by failure states	√	√	√	√	√	√	√
Node failures by failure states	√	√	√	√	√	√	√
Explicit node cost and capacity	√	√	√		√	√	
Routing costs	√	√	√		√	√	

The NFF after incorporating the properties 1-4 discussed in this section is presented below:

$$\min \quad z = \sum_{i\in I}\sum_{l\in L\setminus\{0\}}\sum_{m\in M_M^{il,l-1}} c_{il,l-1}^{1,m} Y_{il,l-1}^{m} + \sum_{i\in I}\sum_{l\in L\setminus\{|L|-1\}}\sum_{m\in M_D^{il,l+1}} c_{il,l+1}^{1,m} Y_{il,l+1}^{m} + \sum_{i\in I}\sum_{m\in M_R^{i}} c_i^{2,m} W_i^m + \sum_{(i,j)\in A}\sum_{m\in M_F} c_{ij}^{3,m} V_{ij}^m \tag{4.1}$$

subject to

$$x_{i,l,l-1}^k - x_{i,l+1,l}^k - x_{ill}^k = \begin{cases} 1, if \;\; s^k = i \, and \;\, l^k = l \\ 0, otherwise \end{cases} \quad \forall l \in L \mid 1 < l <\mid L \mid, \quad \forall i \in N \, and \, \forall k \in K \tag{4.2}$$

$$\sum_{\substack{j\in N:\\ (i,j)\in A \\ j\neq s^k}}\sum_{\substack{r\in N\\ r\neq i}} x_{ij}^{k,i,r} - x_{i,2,1}^k - x_{i11}^k = \begin{cases} 1, if \;\; s^k = i \, and \;\, l^k = 1 \\ 0, otherwise \end{cases} \quad \forall i \in N \, and \, \forall k \in K \mid i \neq t^k \tag{4.3}$$

$$x_{i,l-1,l}^k - x_{i,l,l+1}^k - x_{ill}^k = \begin{cases} 1, if \;\; t^k = i \, and \;\, l^k = l \\ 0, otherwise \end{cases} \quad \forall l \in L \mid 1 < l <\mid L \mid, \quad \forall i \in N \, and \, \forall k \in K \tag{4.4}$$

---

[29] EFF-PF is a strict relaxation of EFF-EF [7]

$$\sum_{\substack{j\in N:\\(j,i)\in A\\j\neq t^k}}\sum_{\substack{p\in N\\p\neq i}} x_{ji}^{k,p,i} - x_{i,1,2}^{k} - x_{i1}^{k} = \begin{Bmatrix} 1, if\ \ t^k = i\, and\ \ l^k = 1 \\ 0, otherwise \end{Bmatrix} \qquad \forall i\in N\, and\, \forall k\in K \mid i\neq t^k \tag{4.5}$$

$$\sum_{\substack{j:(i,j)\in A,\\i\neq r,j\neq p}} f_{ij}^{pr} - \sum_{\substack{j:(j,i)\in A\\j\neq r,i\neq p}} f_{ji}^{pr} = \begin{cases} U^{pr}, & if\ \ i=p \\ -U^{pr}, & if\ \ i=r \\ 0, & otherwise \end{cases} \qquad \forall i\in I\ \ and\ \ \forall(p,r)\in N\times N \mid p\neq r \tag{4.6}$$

$$\sum_{\substack{j\in N:(j,i)\in A,\\j\neq r,j\neq t^k,i\neq s^k}} x_{ji}^{k,p,r} = \sum_{\substack{j\in N:(i,j)\in A,\\j\neq p,j\neq s^k,i\neq t^k}} x_{ij}^{k,p,r}\ \forall k\in K, \forall i\in N\ \ and\ \ \forall(p,r)\in NXN \tag{4.7}$$

$$\sum_{k\in K} d^k x_{i,l,l-1}^{k} \leq \gamma_l \sum_{m\in M_M} q_{i,l,l-1}^{1,m} Y_{i,l,l-1}^{m}\ \ \forall i\in N\ \ and\ \ \forall l\in L \mid 1 < l <\mid L\mid \tag{4.8}$$

$$\sum_{k\in K}\sum_{\substack{j\in N:(i,j)\in A,\\i\neq t^k,j\neq s^k}}\sum_{\substack{r\in N\\r\neq i}} d^k x_{ij}^{k,i,r} \leq \gamma_l \sum_{m\in M_M} q_{i,1,0}^{1,m} Y_{i,1,0}^{m} \qquad \forall i\in N \tag{4.9}$$

$$\sum_{k\in K} d^k x_{i,l,l+1}^{k} \leq \gamma_l \sum_{m\in M_D} q_{i,l,l+1}^{1,m} Y_{i,l,l+1}^{m} \qquad \forall i\in N\ \ and\ \ \forall l\in L\setminus\{\mid L\mid -1\} \tag{4.10}$$

$$\sum_{k\in K}\sum_{\substack{j\in N:(j,i)\in A,\\j\neq t^k,i\neq s^k}}\sum_{\substack{p\in N\\p\neq i}} d^k x_{ji}^{k,p,i} \leq \gamma_l \sum_{m\in M_D} q_{i,0,1}^{1,m} Y_{i,0,1}^{m} \qquad \forall i\in N \tag{4.11}$$

$$\sum_{k\in K}\sum_{\substack{j:(i,j)\in A\\j\neq s^k,i\neq t^k}}\sum_{\substack{(p,r)\in N\times N:\\p\neq r,p\neq j,r\neq i}} d^k x_{ij}^{k,p,r} + \sum_{k\in K}\sum_{\substack{j\in N:(j,i)\in A,\\j\neq t^k,i\neq s^k}}\sum_{\substack{p\in N\\p\neq i}} d^k x_{ji}^{k,p,i} \leq \sum_{m\in M_R} q_i^{2,m} W_i^m \qquad \forall i\in N \tag{4.12}$$

$$\sum_{\substack{i:(p,i)\in A\\p\neq t^k,i\neq s^k}}\sum_{k\in K} d^k x_{pi}^{k,p,r} \leq \gamma_1 U^{pr} \qquad \forall(p,r)\in N\times N \mid p\neq r \tag{4.13}$$

$$\sum_{\substack{k\in K:\\t^k\neq i,s^k\neq j}} d^k x_{ij}^{k,p,r} \leq \gamma_1 f_{ij}^{p,r} \qquad \forall i,j\in N \mid (i,j)\in A\, and\ \ \forall(p,r)\in N\times N \mid p\neq r \tag{4.14}$$

$$\sum_{\substack{(p,r)\in N\times N:\\p\neq r,p\neq j,r\neq i}} f_{ij}^{p,r} + \sum_{\substack{(p,r)\in N\times N:\\p\neq r,p\neq i,r\neq j}} f_{ji}^{p,r} \leq \sum_{m\in M_F} q_{ij}^{3,m} V_{ij}^{m} \quad \forall\{i,j\}\in E \tag{4.15}$$

$$x_{ij}^{kpr} \leq 1\, \forall k\in K, \qquad \forall(i,j)\in A\ \ and\ \ \forall(p,r)\in NXN \mid p\neq r, p\neq j, r\neq i, i\neq t^k, j\neq s^k \tag{4.16}$$

$$x_{i,l,l'}^{k} \leq 1 \qquad \forall l,l'\in L\setminus\{0,\mid L\mid -1\} \mid l' = \{l-1,l,l+1\}, \forall i\in N\, and\ \ \forall k\in K \tag{4.17}$$

$$Y^{m}_{i,l^k,l^k-1} \geq \sum_{k:s^k=i} d^k \;/ q^{1,m}_{i,l^k,l^k-1} \qquad \forall i \in I,\ \forall k \in K \tag{4.18}$$

$$Y^{m}_{i,l^k-1,l^k} \geq \sum_{k:t^k=i} d^k \;/ q^{1,m}_{i,l^k-1,l^k} \qquad \forall i \in I,\ \forall k \in K \tag{4.19}$$

$$x \geq 0 \tag{4.20}$$

$$U,V,W,Y,f \in \mathbb{Z}^{+} \tag{4.21}$$

Although the NFM and the NFF are presented for optical networks in this thesis, the model can easily be adapted to other technologies including wireless networks that use radio frequencies instead of optical transmission at the physical layer.

4.6 Computational Experiments

In this study, computational experiments consist of three phases. First phase aims to compare the NFF with the existing MLNDP formulations in the literature. Second phase's purpose is to assess the behavior of different Benders decomposition algorithmic schemes to select the most promising algorithm to solve the MLNDP and fine tuning. In the third phase, extensive computational tests performed using test instances that are likely to be seen in real life problems to assess the performance of the selected Benders decomposition algorithm. The results of first phase of the computational experiments are reported in this section. Results for second and third phase are reported in Section 5.4.4 and Section 5.5, respectively.

The comparison among the existing formulations in the literature given in Section 3.10 shows that the EFF-EF (explicit flow formulation with edge flows) formulation is the most capable formulation in terms of modeling different side constraints and necessity for post processing the results. In that sense, the EFF-EF is the closest formulation to the NFF from the point of view of modeling capabilities. For this reason, a computational comparison of the NFF and the EFF-EF formulation is made to assess the performance of the NFF in terms of solution time and linear relaxation solution with respect to the EFF-EF.

Basically, "Polska" network from SNDLIB with 12 nodes, 18 edges and 66 commodities is used for comparison. A six-node network with seven edges and an eight-node network with 10 edges are produced by deleting some nodes and their neighboring edges from the polska network. In each instance, there is a commodity between all pair of nodes, i.e. commodity density is equal to 1. Test problems are given in Table 17.

The NFF is coded by MATLAB using the parallel computing features of MATLAB for network transformation. GAMS is used for solving the EFF-EF. During tests, it is observed that the CPLEX solver settings affect performance of the models. In order to make a fair comparison, models are written as .mps files and the solution is performed using the CPLEX interactive solver. Hence, the performance is isolated from the effects of the programs such as GAMS and MATLAB that are used to call CPLEX as a function.

Table 17. Test Problems

Name	# Layers	# Nodes	# Physical Links	# Logical Links	# Commodities
P1	3	12	18	2457	66
P2	2	12	18	2457	66
P3	2	8	10	158	28
P4	2	6	7	48	15

Both formulations for same instances are solved by IBM ILOG CPLEX 12.1 Interactive Solver on a computer with processor Intel Core i7-2720QM CPU @2.20GHz with 8 Gb RAM having Windows 7 Operating system.

First of all, solutions of two different formulations are compared for 6-node network since it is tractable. It is observed that the EFF-EF's solutions include some cycles that are not desirable operationally. The problem is illustrated in Figure 30. It is seen that the solutions of the two formulations are the same apart from these paths including the cycles.

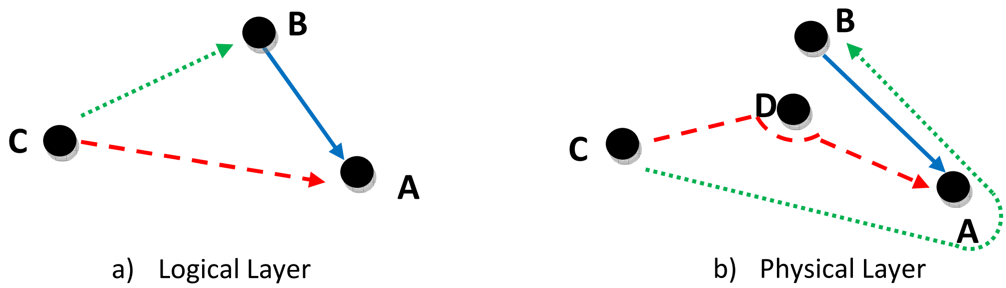


Figure 30. Loops in Physical Layer in EFF Solution

Suppose that there is a commodity k that is to be routed through the network illustrated in Figure 30 between nodes C and A; and in logical layer, link B-A which has slack capacity to route commodity k, has already been installed to route some other commodities. Then, using C-B or C-A in logical network with corresponding physical paths C-A-B and C-D-A, respectively, are alternative solutions with the same cost. Installing C-B logical link is not a desired solution operationally since the corresponding physical path to C-B, B-A logical path has a cycle including the sink node of commodity k. The EFF-EF tends to result in such weird routings if these type of routings are not prevented by additional constraints in the formulation [7]. However, in the NFF, assigning a very small routing cost to flow variables for transmission links prevents having such cycles. Assigning routing cost to flow variables does not work in the EFF-EF since flow variables are only defined in the logical layer.

The test results are given in Table 18. Results of Computational Experiments, the physical graphs of test instances are presented by "Problem" column. Number of layers, nodes, physical and logical links are given in "#Layer", "#Node", "#Physical Links" and "#Logical Links", respectively. "C"s in "Node Capacity" column indicate if the nodes are capacitated in the test instances. EFF-EF solutions are reported as “Multiple” in “Network Model” column and the NFF is reported as “Single” in this column. “Duration (sec)” column presents the duration until termination. “Termin” column reports how the model terminates, e.g. “Opt” means optimality, “Out of Memory” means out of memory error from the MIP solver and “Limit” means, the MIP solver terminates because of time limitation is reached. “#Row” and “#Column” columns lists the number of rows and number of columns of the MIP formed for test instances, respectively. Objective function values are reported by “Obj.” column and relative gap reported by MIP solver at termination is given in “Rel. Gap (%)” column. Linear programming relaxation gap and solution time are given by “LP Relax.” “Gap” and “Time (sec)” columns.

Before commenting on the results of the computational tests, the difference between the objective function values of the EFF-EF formulation and the NFF has to be emphasized. The EFF-EF counts the logical links whose cost is an approximate cost based on the network technology [118]. The NFF can explicitly use the amount of flow passing through the processors that make technology or granularity of flow changes. Since the EFF-EF basically counts the number of lightpaths and the NFF works with the total flow on processor links to find the number of processors, the cost computed by the EFF-EF can be greater than the cost calculated by the NFF for the same solution because of the aggregation done by the NFF.

Both formulations are able to find optimal solution in the six-node network while the NFF is faster than the EFF-EF formulation. For the eight-node network, the NFF finds the optimal solution while the EFF-EF formulation gives an out of memory error before reaching optimality. For the twelve-node network, time performances of two formulations are almost equal to each other although, the EFF-EF formulation gives an out of memory error before 2-hours running time. It is observed from the CPLEX log files that the number of nodes in the algorithm is very large (about 100 times more) when solving the EFF-EF formulation compared to the NFF.

For three-layer network, the EFF-EF formulation is computationally intractable for twelve-node polska network. However, the NFF’s solution performance is as good as solving the two-layer polska network.

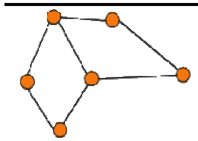
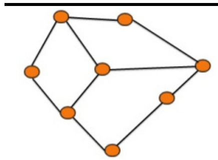
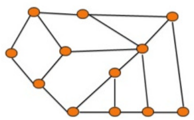
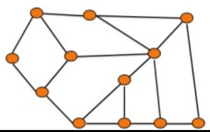
We observe that the NFF’s integrality gaps are consistently less than the EFF-EF formulation for the test instances. Let’s define $Z_{LP}(.)$ as the optimal objective function value of the linear programming (LP) relaxation and $F(.)$ is the feasible space of formulation $(.)$. Computational studies show that $Z_{LP}(\text{NFF}) \leq Z_{LP}(\text{EFF-EF})$ as conjecture:

Conjecture 1: $Z_{LP}(\text{NFF}) \leq Z_{LP}(\text{EFF-EF})$.

Conjecture 1 can be proved by firstly proving that $F(\text{NFF}) \subseteq F(\text{EFF-EF})$ and then showing that the strict equality does not hold by a counter example. Examples for such kind of proofs are presented in [246]. The main steps of this proof are provided in Appendix B.

We shall note that the three-layer network is given for the first time in the telecommunication literature according to the best knowledge of us.

Table 18. Results of Computational Experiments

Problem	#Layer	#Node	#Physical Link	#Logical Link	Node Capacity	Network Model	Duration (sec)	Termin.	# Row	#Column	Obj.	Rel. Gap (%)	LP Relax.	
													Gap	Time (Sec)
	2	6	7	48		Multiple	9	Opt.	147	1,496	1362	0	48	0.02
						Single	4	Opt.	3,805	7,045	1272	0	39	0.02
					C	Multiple	14	Opt.	153	1502	1422	0	46	0.02
						Single	3	Opt.	3,805	7,045	1332	0	38	0.01
	2	8	10	158		Multiple	3516	Out of Memory	394	9,017	2417	5	39	0.1
						Single	195	Opt.	15,322	33,242	2207	0	33	0.2
					C	Multiple	1999	Out of Memory	402	9,025	2497	15	30	0.1
						Single	146	Opt.	15,322	33,242	2287	0	32	0.2
	2	12	18	2457		Multiple	5794	Out of Memory	3,269	326,800	6542	11	22	13.7
						Single	7200	Limit	115,026	320,946	6308	9	17	10
					C	Multiple	6701	Out of Memory	3,281	326,812	6595	10	21	9
						Single	7200	Limit	115,026	320,946	6575	11	17	11
	3	12	18	$>10^{87}$		Single	7200	Limit	116,658	324,954	6848	8	16	11.2
					C	Single	7200	Limit	116,658	324,954	6968	8	16	14.4

CHAPTER 5

BENDERS DECOMPOSITION BASED ALGORITHMS TO SOLVE MULTI-LAYER TELECOMMUNICATION NETWORK DESIGN (MLND) PROBLEM

The computational tests in Section 4.6 show that, although the NFF performs better than EFF-EF, general-purpose integer programming solvers are not sufficient to solve the NFF for practically large networks. Thus, a tailored algorithm to solve the NFF is necessary. Benders decomposition algorithm, which is proposed by Benders in 1962 [247], is used to develop tailored solution algorithms. In this chapter, a brief introduction to Benders decomposition is made and literature review on the improvement techniques and variants of Benders decomposition is reported. The solution algorithms based on Benders decomposition, which are developed to solve NFF, and add-ons developed to improve these algorithms are presented. Results of preliminary computational experiments, which are performed to assess the behavior of developed algorithms and fine tuning, are reported. Using these results, most promising algorithm is selected and improvement opportunities of the algorithms are seen. The selected algorithm is improved to have an ultimate algorithm. Extensive computational experiments to assess the performance of this algorithm are done by using test instances that are likely to be real life problems and results are reported.

5.1 Benders Decomposition

The Benders decomposition [247] method, a resource directive decomposition method, has many successful applications for solving network design problems [248]. Main idea behind the Benders decomposition is to decompose the problem into a master problem with integer variables and a subproblem with continuous variables by temporarily holding a set of strategic resource variables constant [249]. In this section, a formal derivation of Benders reformulation of a mixed integer problem (MIP) is presented.

Consider the following problem:

$$\begin{aligned}
\text{(P)} \quad & \text{Min} \;\; cx + dy \\
& \text{s.t.} \\
& \quad Ax + By \geq b \\
& \quad Dy \geq e \\
& \quad x \geq 0, \\
& \quad y \geq 0 \text{ and integer}
\end{aligned}$$

where

- x is vector of continuous variables,

- y is vector of integer variables,
- A, B and C are coefficient matrices of appropriate size
- b and e are right hand side vectors of appropriate size
- c and d are row vectors of cost associated with x and y, respectively.

If we project integer variables y out of the problem P the problem can be expressed as:
$\min_{y^* \in Y}\{dy^* + \min_{x \geq 0}\{cx : Ax \geq b - By^*\}\}$ (5.1)
where

- $Y = \{y : Dy \geq e, y \geq 0 \text{ and integer}\}$

Note that the inner minimization problem in (5.1) is a linear program whose unboundedness for some $y \in Y$ implies unboundedness of P. Thus, assuming that the inner minimization is bounded, its dual can be written by the associating dual variables u and its dual can be replaced with it as its dual is either feasible or unbounded. The dual of the inner minimization problem in (5.1) is called Benders decomposition *subproblem (SP):*

$$\begin{array}{lll} (SP) & \max & u(b - By^*) \\ & s.t. & \\ & & uA \leq c \\ & & u \geq 0 \end{array} \qquad (5.2)$$

Then the problem P becomes:

$$\min_{y^* \in Y}\{dy^* + \max_{u \geq 0}\{u(b - By^*) : uA \leq c\}\} \qquad (5.3)$$

(4.3) reveals that the feasible space of (SP), $F = \{u : u \geq 0; uA \leq c\}$, is independent of the values of integer y variables. Then the following observations are made:

- SP is either bounded or unbounded. Infeasibility of SP implies the unboundedness of P and we assume that problem P is not unbounded. Then F is not empty and since SP is a linear program, F is composed of extreme rays ($r^1, r^2, \ldots, r^Q$) and extreme points ($u^1, u^2, \ldots, u^P$) where Q and P are numbers of extreme rays and extreme points, respectively.
- If SP is unbounded, then there is a direction r^q such that $r^q(b - By^*) > 0$ and r^q must be avoided in order to have a primal feasible solution for inner minimization in (5.1):
  $r^q(b - By^*) \leq 0 \quad q = 1, \ldots, Q$ (5.4)

  (5.4) are called "feasibility cuts".

- If (SP) is bounded, then the solution is one of the extreme points u^p ($p = 1, \ldots, P$). Since we are seeking the maximum value for (SP)'s over $y \in Y$, any solution for (SP) shall be less than or equal to objective function of SP with the optimal y values of the original problem. If we introduce an auxiliary continuous variable for objective value of SP in (5.3) as η, then having an optimal value for (SP) as u^p ($p = 1, \ldots, P$) restricts η as
  $$\eta \geq u^p(b - By^*) \qquad p = 1, \ldots, P \qquad (5.5)$$

(5.5) are called "optimality cuts".

Rewriting (5.3) with restrictions of feasibility and optimality cuts, we have the Benders Decomposition's *master problem (MP).*

$$
\begin{aligned}
(MP) \quad & \min\ dy + \eta \\
& s.t. \\
& \quad \eta \geq u^p(b - By^*) \qquad\qquad p = 1,\dots,P \\
& \quad r^q(b - By^*) \leq 0 \qquad\qquad q = 1,\dots,Q \\
& \quad y \in Y, \eta \geq 0
\end{aligned}
$$

The reformulation of (P) consisting of (SP) and (MP) is called the Benders Reformulation of (P) [250].

Benders decomposition algorithm involves iterative solution of (MP) and (SP) to solve (P) by generating necessary optimality and feasibility cuts during these iterations instead of generating them at once. So, (MP) is solved and an integer solution (y^*, η^*) is generated. (SP) is solved with this solution. If (SP) has an optimal solution and its objective function value is equal to η^* then the algorithm stops. Otherwise, if the solution is bounded (unbounded), associated optimality (feasibility) cut is added to (MP) and (MP) is solved again. Since (MP) is a relaxation to (P), its solution is a lower bound for (P). If (SP) is feasible given (y^*, η^*), then SP solution together with (y^*, η^*) is a feasible solution to (P), hence this solution gives an upper bound for (P). The procedure stops at optimality when upper bound is equal to the lower bound. [249–251] are referred for more detailed information on Benders reformulation and Benders decomposition method.

5.2 Literature Survey on Benders Decomposition

Since Benders decomposition method is proposed by Benders in 1962 [247], several variants of the algorithm and improvement methods are proposed. This section reports our literature survey on different variants and improvement methods for Benders decomposition mainly used to solve network design problems. In the literature review, we mainly focused on improving Benders cuts thorough cut selection methods and improving master problem solution. Costa's literature survey [252] is referred for more comprehensive survey on Benders decomposition methods for solving fixed-charge network design problems. Our findings during literature survey are presented in Table 19.

The literature review shows that classical Benders decomposition is improved either by changing the algorithm such as [5], [8], [253–255] or improving some of the subroutines of the algorithm such as cut selection [256–260], additional cut generation [124], [233] and model selection [256]. The applications of Benders decomposition consists of the combination of variants of algorithm and the improvement methods.

Table 19. Benders Decomposition Literature Survey

Paper	Improvement Method	Improvement	Applicability to Our Problem
[261]	When the subproblem suffers from degeneracy, alternative solutions exist for a single master problem solution and this brings the issue for selecting the one that violates master problem the most. Magnanti and Wong propose a method to generate "pareto optimal cuts" in this paper. Selecting a good model is also used as an improvement method.	No computational results given	N/A: Our problem's subproblem is a feasibility seeking problem; so that no optimal solutions found. If we add artificial routing cost to problem, we still need a cut selection method that considers both optimality and feasibility cuts.
[8]	Ɛ-optimal solution: MP is not solved to optimality and each iteration a feasible solution is generated. Hence, upper bound of the problem is found. Algorithm terminates when no feasible solution can be generated with an objective function value less than (best upper bound found so far)*(1- Ɛ).	No computational experiments regarding the improvement gained by the modification of the Benders decomposition.	Applicable: There are successful implementations of this algorithm for network design problems.
[124]	At each iteration of Benders Decomposition, violated bipartition cuts are generated and added to the master problem. This method is called Multi Cut Generation (MCG).	MCG significantly decreases the number of iterations in Benders decomposition (in test problems, MCG halves the number of iterations) and the time for solution (most of the test problem, MCG decreases the time for solution by an order of magnitude)	Applicable: Our problem consists of several multicommodity network flow problems

Table 19 (Cont'd)

Paper	Improvement Method	Improvement	Applicability to Our Problem
[253]	Solving LP relaxation of the master problem at the first k iterations instead of solving it as IP for all iterations.	Solves all tests instances to optimality in 5 minutes, while original algorithm cannot solve all off them to optimality. Number of iterations solving IP problem does not exceed 1 for the modified algorithm while it is almost 9 on the average for the original algorithm. In addition, for some instances, the solution times decrease dramatically, especially for those having very hard IP master problems	Applicable
[254]	Makes local branching search after solving master problem in order to find different feasible solutions. This provides better upper bounds and makes it possible to generate more optimality and feasibility cuts compared to the classical benders decomposition	Multicommodity fixed charge network flow problem is used for comparison. Improvement of local branching the solution time compared to benders decomposition including features of [256] and [253] is given in the following table.	Applicable: Master problem is expensive in terms of solution time, so that this can improve the solution time.
[262]	This method is a generalization of multicut generation method for MCMCF problem due to Gabrel et al. [124]. Propose covering cut bundle generation, a novel way to generate multiple cuts – proposed a generalized CCB algorithm which was proposed by [124] and [142]	For instances that need longer time to solve with classical BD, the improvement is more than others. For the former, the CPU time improvement exceeds 90% while for latter the improvement is still more than 15%	Applicable

Table 19 (Cont'd)

Paper	Improvement Method	Improvement	Applicability to Our Problem
[259]	Improves Magnanti and Wong's Pareto optimal cuts methods to find good cuts by proposing a method to compute core point independent from the problem.	In [263], it is shown than method proposed in [257], [258]is two times better than proposed method regarding solution times. In addition, method proposed in [263] finds solutions within %15 time of this method's solution times on average.	Not Applicable: We need a cut selection method to evaluate the effectiveness of both feasibility and optimality cuts.
[258], [257]	Provides a n alternative method to Pareto optimal cuts [261] in order to select the most efficient benders cuts using minimal infeasible subsystems and alternative polyhedron. Different from Pareto optimal cuts, this cut selection takes feasibility cuts into account when comparing the potential cuts and selecting most effective one.	In the technical report, the modified BD is much more effective on 9 out of 11 instances than the original one, with speedups of 1 to 2 orders of magnitude. In the paper, the tests are performed on multicommodity network design problem with and without routing costs, and network expansion problem. The solution time of original problem is 2 (3.5) times the modified algorithm for design problem without (with) routing cost. The difference can become 1 or 2 orders of magnitude for expansion problems.	Applicable: The proposed method can be used in separation of benders cuts in order to select most effective Bender's cuts at each iteration
[264]	an interior-point branch-and-cut algorithm for structured integer programs based on Benders decomposition and the analytic center cutting plane method (AC-CPM)	For the capacitated facility location problem, the proposed approach was on average 2.5 times faster than Benders-branch-and-cut and 11 times faster than classical Benders decomposition. For the multicommodity capacitated fixed charge network design problem, the proposed approach was 4 times faster than Benders-branch-and-cut while classical Benders decomposition failed to solve the majority of the tested instances.	Applicable

Table 19 (Cont'd)

Paper	Improvement Method	Improvement	Applicability to Our Problem
[255]	A genetic algorithm is used to find feasible solutions to master problem instead of solving an integer problem	It is shown that classical benders decomposition is improved by using GA to solve master problem. The upper bounds found by proposed algorithm in one-hour are better than the classical benders decomposition in 24 of 34 test problems. Since the problems are selected randomly from MIP library, in some problems the improvement in UB is very large (as high as 1/170 of the UB found by classical) while in some problems there is no improvement.	Applicable
[5]	Benders decomposition within branch and cut framework is used with MCG in [124] and alternative polyhedron is used [257] for cut selection.	Improved solution times for benders decomposition with single cut generation and multiple cut generation used by Gabrel et al. by 34%-2% for two-layered networks with up to 9 nodes.	Applicable: The capacitated formulation of multi-layer network design problem is solved.
[265]	Uses benders decomposition in a branch and cut algorithm just like [5]	No computational experiments presented for comparing classical benders decomposition and the proposed method.	Applicable

Table 19 (Cont'd)

Paper	Improvement Method	Improvement	Applicability to Our Problem
[263]	They used the following ideas from previous works and proposed a new cut generation scheme: - Pareto optimal cuts: however, the core point is problem depended [256]. -To make the core point problem depended, cut selection method of Papadakos [259] is used. -Another cut selection method: using MIS and alternative polyhedrons proposed by Fischetti et. al [257], [258] is used.	Proposed algorithm is compared by BD with the cut selection method of Papadakos [259] and MIS based cuts selection method of Fischetti et al. [257], [258]. It is seen that the proposed algorithm performs better than the two existing algorithms on tree of hubs location problem: -for small sized problems (proposed algorithm uses %15 of time used by Papadakos' and %30 of MIS algorithm) -for instances with more than 50 nodes, Papadakos' algorithm cannot find solution in 75 hours. Proposed algorithm performs 9 – 17 times better than the MIS.	Applicable

5.3 Benders Reformulation of Multi-layer Telecommunication Network Design (MLND) Problem

The multi-layer network design problem is decomposed into subproblem and master problem according to the Benders reformulation. The master problem generates the number of processors and fiber optic cables together with the number of wavelengths between nodes and number of wavelengths installed on edges. The subproblems use the numbers to calculate the edge capacities to find an optimal routing. Since, there is no routing cost, the subproblem is feasibility seeking problem and the master problem involves only feasibility cuts. The primal of subproblem and the master problem after such decomposition is presented below:

PRIMAL of SUBPROBLEM – (P-SP)

$$\min\ 0 \tag{6.1}$$

subject to

$$x^k_{i,l,l-1} - x^k_{i,l+1,l} - x^k_{ill} = \begin{cases} 1, if\ \ s^k = i\, and\ \ l^k = l \\ 0, otherwise \end{cases} \quad \forall l \in L \mid 1 < l < \mid L \mid, \ \forall i \in N\, and\, \forall k \in K \tag{6.2}$$

$$\sum_{\substack{j \in N: \\ (i,j) \in A \\ j \neq s^k}} \sum_{\substack{r \in N \\ r \neq i}} x^{k,i,r}_{ij} - x^k_{i,2,1} - x^k_{i11} = \begin{cases} 1, if\ \ s^k = i\, and\ \ l^k = 1 \\ 0, otherwise \end{cases} \quad \forall i \in N\, and\, \forall k \in K \mid i \neq t^k \tag{6.3}$$

$$x^k_{i,l-1,l} - x^k_{i,l,l+1} - x^k_{ill} = \begin{cases} 1, if\ \ t^k = i\, and\ \ l^k = l \\ 0, otherwise \end{cases} \quad \forall l \in L \mid 1 < l < \mid L \mid, \ \forall i \in N\, and\, \forall k \in K \tag{6.4}$$

$$\sum_{\substack{j \in N: \\ (j,i) \in A \\ j \neq t^k}} \sum_{\substack{p \in N \\ p \neq i}} x^{k,p,i}_{ji} - x^k_{i,1,2} - x^k_{i1} = \begin{cases} 1, if\ \ t^k = i\, and\ \ l^k = 1 \\ 0, otherwise \end{cases} \quad \forall i \in N\, and\, \forall k \in K \mid i \neq t^k \tag{6.5}$$

$$\sum_{\substack{j \in N:(j,i) \in A, \\ j \neq r, j \neq t^k, i \neq s^k}} x^{k,p,r}_{ji} = \sum_{\substack{j \in N:(i,j) \in A, \\ j \neq p, j \neq s^k, i \neq t^k}} x^{k,p,r}_{ij} \quad \forall k \in K, \forall i \in N\ \ and\ \ \forall (p,r) \in NXN \tag{6.6}$$

$$\sum_{k \in K} d^k x^k_{i,l,l-1} \leq \gamma_l \sum_{m \in M_M} q^{1,m}_{i,l,l-1} Y^m_{i,l,l-1} \quad \forall i \in N\ \ and\ \ \forall l \in L \mid 1 < l < \mid L \mid \tag{6.7}$$

$$\sum_{k \in K} \sum_{\substack{j \in N:(i,j) \in A, \\ i \neq t^k, j \neq s^k}} \sum_{\substack{r \in N \\ r \neq i}} d^k x^{k,i,r}_{ij} \leq \gamma_l \sum_{m \in M_M} q^{1,m}_{i,1,0} Y^m_{i,1,0} \quad \forall i \in N \tag{6.8}$$

$$\sum_{k\in K} d^k x^k_{i,l,l+1} \le \gamma_l \sum_{m\in M_D} q^{1,m}_{i,l,l+1} Y^m_{i,l,l+1} \qquad \forall i \in N \ and \ \forall l \in L \setminus \{|L|-1\} \tag{6.9}$$

$$\sum_{k\in K} \sum_{\substack{j\in N:(j,i)\in A,\\ j\neq t^k, i\neq s^k}} \sum_{\substack{p\in N\\ p\neq i}} d^k x^{k,p,i}_{ji} \le \gamma_l \sum_{m\in M_D} q^{1,m}_{i,0,1} Y^m_{i,0,1} \qquad \forall i \in N \tag{6.10}$$

$$\sum_{k\in K} \sum_{\substack{j:(i,j)\in A\\ j\neq s^k, i\neq t^k}} \sum_{\substack{(p,r)\in N\times N:\\ p\neq r, p\neq j, r\neq i}} d^k x^{k,p,r}_{ij} + \sum_{k\in K} \sum_{\substack{j\in N:(j,i)\in A,\\ j\neq t^k, i\neq s^k}} \sum_{\substack{p\in N\\ p\neq i}} d^k x^{k,p,i}_{ji} \le \sum_{m\in M_R} q^{2,m}_i W^m_i \qquad \forall i \in N \tag{6.11}$$

$$\sum_{\substack{i:(p,i)\in A\\ p\neq t^k, i\neq s^k}} \sum_{k\in K} d^k x^{k,p,r}_{pi} \le \gamma_1 U^{pr} \qquad \forall (p,r) \in N\times N \mid p \neq r \tag{6.12}$$

$$\sum_{\substack{k\in K:\\ t^k\neq i, s^k\neq j}} d^k x^{k,p,r}_{ij} \le \gamma_1 f^{p,r}_{ij} \qquad \forall i,j \in N \mid (i,j)\in A \, and \ \forall (p,r)\in N\times N \mid p\neq r \tag{6.13}$$

$$x^{kpr}_{ij} \le 1 \, \forall k \in K, \qquad \forall (i,j)\in A \ and \ \forall (p,r) \in NXN \mid p\neq r, p\neq j, r\neq i, i\neq t^k, j\neq s^k \tag{6.14}$$

$$x^k_{i,l,l'} \le 1 \qquad \forall l, l' \in L \setminus \{0, |L|-1\} \mid l' = \{l-1, l, l+1\}, \forall i \in N \, and \ \forall k \in K \tag{6.15}$$

$$x \ge 0 \tag{6.16}$$

MASTER PROBLEM – (MP)

$$\min \ z = \sum_{i\in I} \sum_{l\in L\setminus\{0\}} \sum_{m\in M^{il,l-1}_M} c^{1,m}_{il,l-1} Y^m_{il,l-1} + \sum_{i\in I} \sum_{l\in L\setminus\{|L|-1\}} \sum_{m\in M^{il,l+1}_D} c^{1,m}_{il,l+1} Y^m_{il,l+1} + \sum_{i\in I} \sum_{m\in M^i_R} c^{2,m}_i W^m_i + \sum_{(i,j)\in A} \sum_{m\in M_F} c^{3,m}_{ij} V^m_{ij} \tag{6.17}$$

subject to

$$\sum_{\substack{(p,r)\in N\times N:\\ p\neq r, p\neq j, r\neq i}} f^{p,r}_{ij} + \sum_{\substack{(p,r)\in N\times N:\\ p\neq r, p\neq i, r\neq j}} f^{p,r}_{ji} \le \sum_{m\in M_F} q^{3,m}_{ij} V^m_{ij} \qquad \forall \{i,j\} \in E \tag{6.18}$$

$$\sum_{\substack{j:(i,j)\in A,\\ i\neq r, j\neq p}} f^{pr}_{ij} - \sum_{\substack{j:(j,i)\in A\\ j\neq r, i\neq p}} f^{pr}_{ji} = \begin{cases} U^{pr}, & if \ i = p \\ -U^{pr}, & if \ i = r \\ 0, & otherwise \end{cases} \qquad \forall i \in I \ and \ \forall (p,r)\in N\times N \mid p\neq r \tag{6.19}$$

$$Y^m_{i,l^k,l^k-1} \ge \sum_{k:s^k=i} d^k / q^{1,m}_{i,l^k,l^k-1} \qquad \forall i \in I, \ \forall k \in K \tag{6.20}$$

$$Y^m_{i,l^k-1,l^k} \ge \sum_{k:t^k=i} d^k / q^{1,m}_{i,l^k-1,l^k} \qquad \forall i \in I, \ \forall k \in K \tag{6.21}$$

$$Feasibility \ \ cuts \tag{6.22}$$

$$U, V, W, Y, f \in \mathbb{Z}^{+} \tag{6.23}$$

5.4 Benders Decomposition Based Algorithms

Several Benders decomposition based solution algorithms are developed for solving the multi-layer network design problem.

5.4.1 Algorithm Frameworks

The first framework used is the original Benders decomposition algorithm as it was proposed by Benders in 1962 [247]. Since the model does not involve any routing cost, the primal subproblem is a feasibility seeking problem and its dual is always infeasible making the subproblem always unbounded. Hence, in its original form only feasibility cuts can be generated in this algorithm. The flow chart of the algorithm is given in Figure 31.

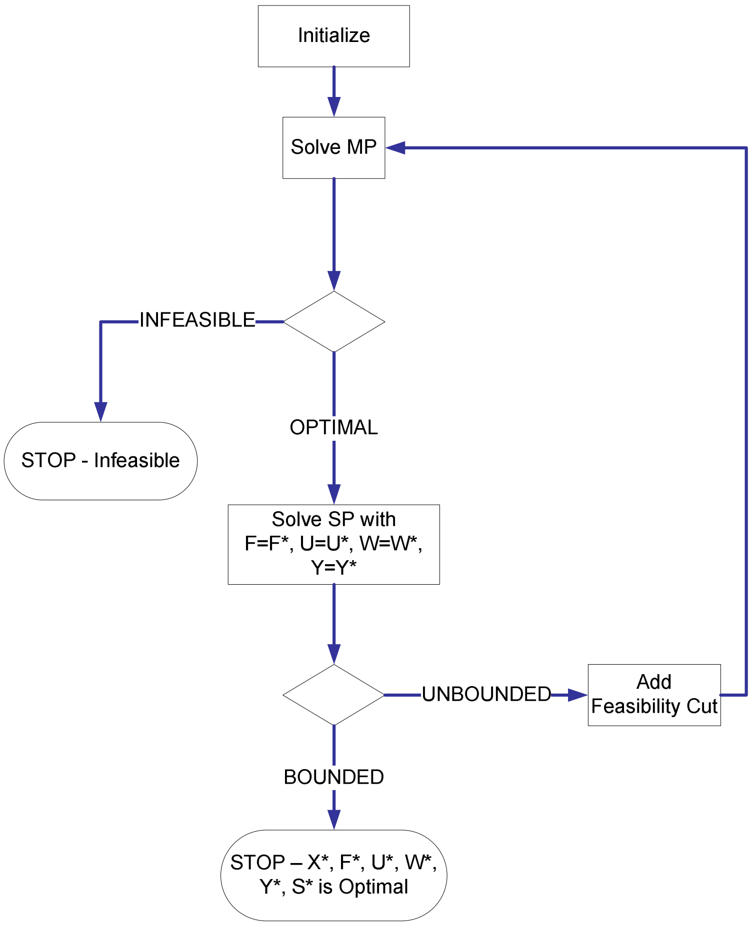


Figure 31. Original Benders Decomposition (O-BD_Feas)

In our first attempt to improve the original algorithm, we added artificial routing cost values (artRC) to the model and generated optimality cuts along with the feasibility cut. The second variant of the algorithm is given in Figure 32. We observe that optimality cuts are stronger than the feasibility cuts. In addition, artificial routing costs enable the algorithm to find some feasible solutions which are used to calculate upper bounds. Hence, in the second variant of the algorithm, calculating gap and terminating the algorithm when a solution of desired quality is obtained is possible.

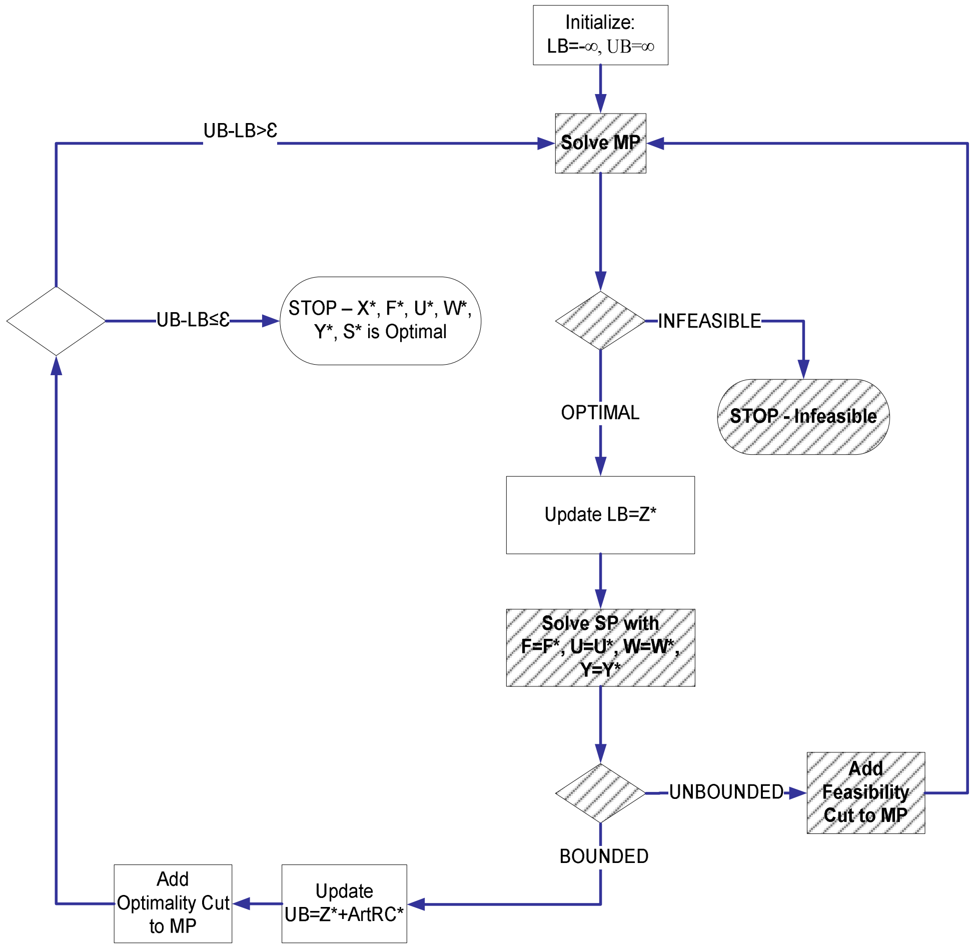


Figure 32. Original Benders Decomposition with Artificial Cost Values (O-BD_Opt)
(Shaded boxers are the same with O-BD_Feas algorithm)

In our preliminary tests, we saw that the original Benders decomposition framework is not good enough for solving moderate size multi-layer network design problem instances because of the complexity of the master problem. The master problem that we are dealing with is initially (before adding any feasibility or optimality cuts) is an integer multicommodity network flow problem. So that, we decided to improve the algorithm by simplifying this particular step. This led us to two different variants of Benders Decomposition.

The first variant involves solving the master problem by branch and cut such that the subproblem is solved using the value of each incumbent solution of the master problem in order to generate Benders cuts if necessary. It converges faster than the original algorithm, however, its convergence rate decreases as the optimality gap decreases. The algorithm is presented in Figure 33.

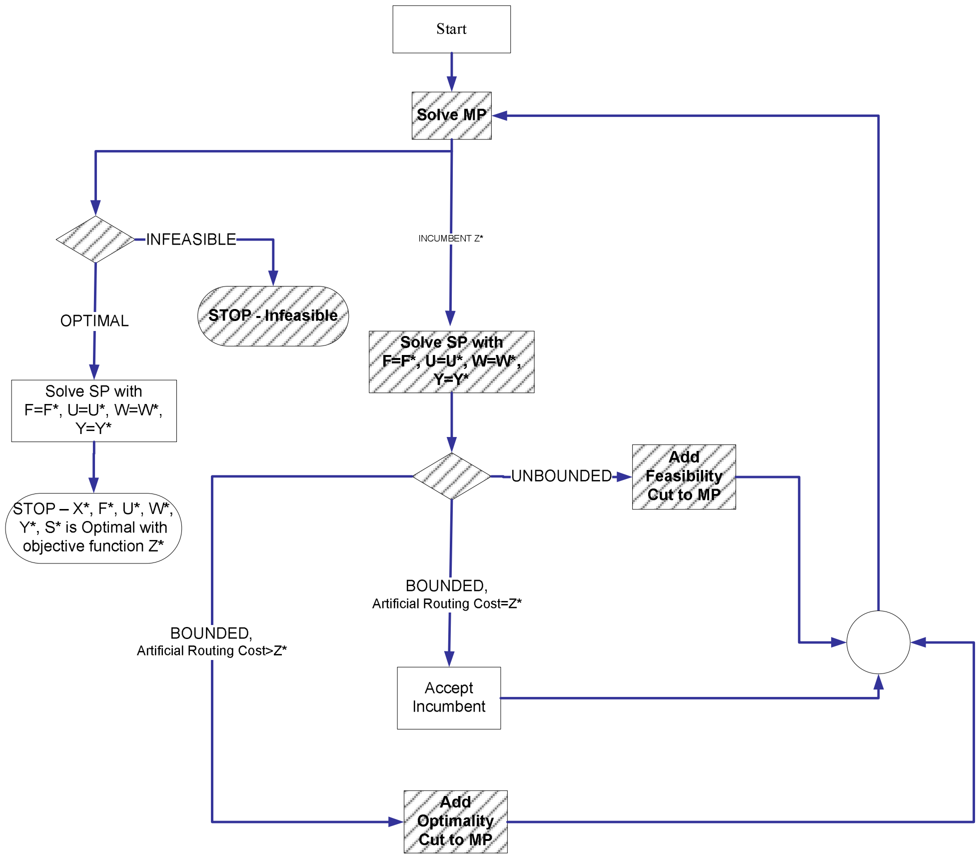


Figure 33. Branch and Cut - Benders Algorithm (B&C-BD) (Shaded boxes are the same with O-BD_Opt algorithm)

Geoffrion and Graves [8] proposed a variant of Benders decomposition method where the master problem is not solved to optimality. Hence, the master problem solution is no longer a lower bound as being in original Benders decomposition method, but the master problem solution together with the primal subproblem solution is an upper bound to the problem.

Since the issue is to improve the master problem solution time, we also implemented this variant. Geoffrion and Graves' Benders decomposition (GG-BD) algorithm changes the optimality step of the master problem into finding a feasible solution by changing the master problem to a feasibility seeking problem. This is done by moving the objective function to constraints with a right hand side of (1-ε) times the best upper bound found so far. Thus, this variant is an ε-optimal solution. The algorithm is presented in Figure 34.

In our variant of GG-BD algorithm, we generate a solution pool for the master problem and select a predetermined number of diverse solutions from that solution pool. For each feasible solution in the pool, we solve the subproblem, and then Benders cuts are added if necessary. This solution pool prevents the algorithm to get stuck at a local optimum to some extend by solving subproblem with more than one diverse master problem solutions. Hence, at each iteration the algorithm can generate more than one cut and its chance to find an optimal routing increases.

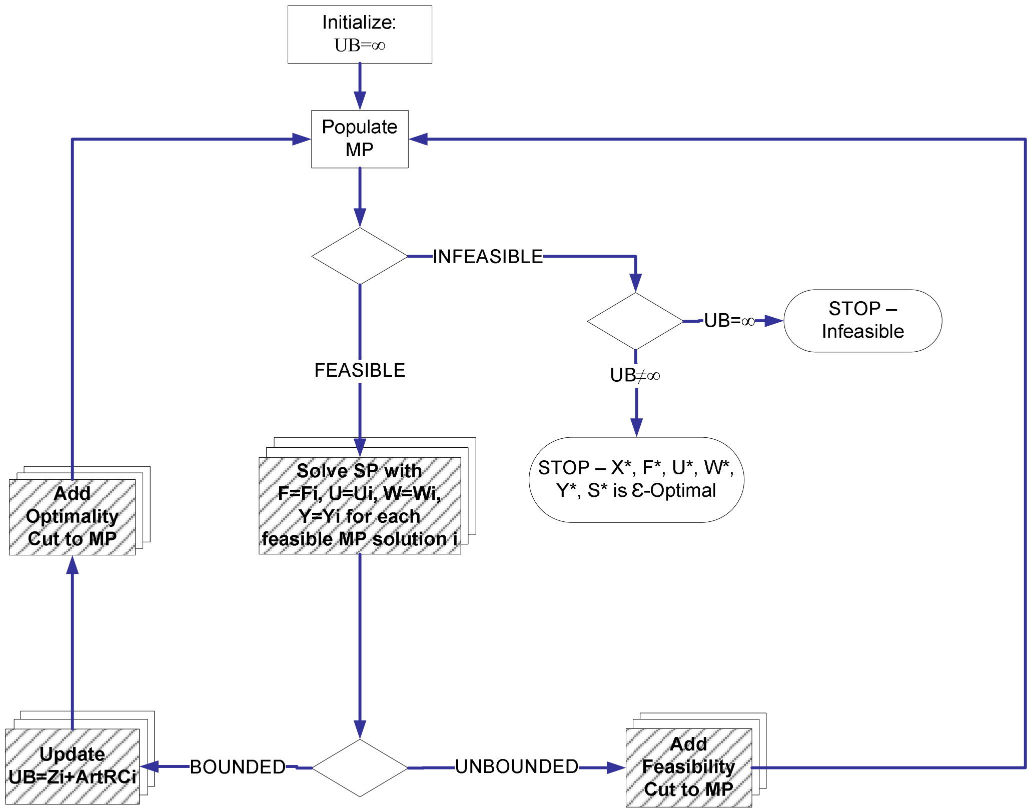


Figure 34. Geoffrion and Graves Benders decomposition Variant – (GG-BD) (Shaded boxes are the same with O-BD_Opt algorithm)

5.4.2 Selection of Benders Cuts

In the preliminary computational experiments, it is observed that the higher the number of Benders cuts, the slower the algorithm converges. If we use the original Benders Decomposition's cut generation, we generate a feasibility or optimality cut without any information about how much that cut is violated. Thus, we generate a number of Benders cuts that do improve neither the lower bound nor the upper bound. In the literature, there are a number of cut generation methods including [256–259]. Most methods in the literature are about finding the pareto optimal cuts involving effectiveness of optimality cuts. However, for our problem we need to assess the effectiveness of feasibility cuts together with the optimality cuts. We use the alternative polyhedron proposed by Fischetti et al. [257], [258] to generate the most violated Benders cut through each iteration. We change the primal subproblem as the following:

ALTERNATIVE POLYHEDRA -PRIMAL SUBPROBLEM – (AltP-SP)

$$\min \ \sigma \tag{6.1}$$

subject to

$$artRC - \sigma \leq \eta^* \tag{6.2}$$

$$x^k_{i,l,l-1} - x^k_{i,l+1,l} - x^k_{ill} = \begin{cases} 1, if\ \ s^k = i\, and\ \ l^k = l \\ 0, otherwise \end{cases} \quad \forall l \in L \mid 1 < l <\mid L \mid, \ \forall i \in N\, and\, \forall k \in K \tag{6.3}$$

$$\sum_{\substack{j \in N: \\ (i,j) \in A \\ j \neq s^k}} \sum_{\substack{r \in N \\ r \neq i}} x^{k,i,r}_{ij} - x^k_{i,2,1} - x^k_{i11} = \begin{cases} 1, if\ \ s^k = i\, and\ \ l^k = 1 \\ 0, otherwise \end{cases} \quad \forall i \in N\, and\, \forall k \in K \mid i \neq t^k \tag{6.4}$$

$$x^k_{i,l-1,l} - x^k_{i,l,l+1} - x^k_{ill} = \begin{cases} 1, if\ \ t^k = i\, and\ \ l^k = l \\ 0, otherwise \end{cases} \quad \forall l \in L \mid 1 < l <\mid L \mid, \ \forall i \in N\, and\, \forall k \in K \tag{6.5}$$

$$\sum_{\substack{j \in N: \\ (j,i) \in A \\ j \neq t^k}} \sum_{\substack{p \in N \\ p \neq i}} x^{k,p,i}_{ji} - x^k_{i,1,2} - x^k_{i1} = \begin{cases} 1, if\ \ t^k = i\, and\ \ l^k = 1 \\ 0, otherwise \end{cases} \quad \forall i \in N\, and\, \forall k \in K \mid i \neq t^k \tag{6.6}$$

$$\sum_{\substack{j \in N:(j,i) \in A, \\ j \neq r, j \neq t^k, i \neq s^k}} x^{k,p,r}_{ji} = \sum_{\substack{j \in N:(i,j) \in A, \\ j \neq p, j \neq s^k, i \neq t^k}} x^{k,p,r}_{ij} \quad \forall k \in K, \forall i \in N\ \ and\ \ \forall (p,r) \in NXN \tag{6.7}$$

$$\sum_{k \in K} d^k x^k_{i,l,l-1} - \sigma \leq \gamma_l \sum_{m \in M_M} q^{1,m}_{i,l,l-1} Y^m_{i,l,l-1} \quad \forall i \in N\ \ and\, \forall l \in L \mid 1 < l <\mid L \mid \tag{6.8}$$

$$\sum_{k \in K} \sum_{\substack{j \in N:(i,j) \in A, \\ i \neq t^k, j \neq s^k}} \sum_{\substack{r \in N \\ r \neq i}} d^k x^{k,i,r}_{ij} - \sigma \leq \gamma_l \sum_{m \in M_M} q^{1,m}_{i,1,0} Y^m_{i,1,0} \quad \forall i \in N \tag{6.9}$$

$$\sum_{k \in K} d^k x^k_{i,l,l+1} - \sigma \leq \gamma_l \sum_{m \in M_D} q^{1,m}_{i,l,l+1} Y^m_{i,l,l+1} \quad \forall i \in N\ \ and\ \ \forall l \in L \setminus \{\mid L \mid -1\} \tag{6.10}$$

$$\sum_{k \in K} \sum_{\substack{j \in N:(j,i) \in A, \\ j \neq t^k, i \neq s^k}} \sum_{\substack{p \in N \\ p \neq i}} d^k x^{k,p,i}_{ji} - \sigma \leq \gamma_l \sum_{m \in M_D} q^{1,m}_{i,0,1} Y^m_{i,0,1} \quad \forall i \in N \tag{6.11}$$

$$\sum_{k \in K} \sum_{\substack{j:(i,j) \in A \\ j \neq s^k, i \neq t^k}} \sum_{\substack{(p,r) \in N \times N: \\ p \neq r, p \neq j, r \neq i}} d^k x^{k,p,r}_{ij} + \sum_{k \in K} \sum_{\substack{j \in N:(j,i) \in A, \\ j \neq t^k, i \neq s^k}} \sum_{\substack{p \in N \\ p \neq i}} d^k x^{k,p,i}_{ji} - \sigma \leq \sum_{m \in M_R} q^{2,m}_i W^m_i \quad \forall i \in N \tag{6.12}$$

$$\sum_{\substack{i:(p,i) \in A \\ p \neq t^k, i \neq s^k}} \sum_{k \in K} d^k x^{k,p,r}_{pi} - \sigma \leq \gamma_1 U^{pr} \quad \forall (p,r) \in N \times N \mid p \neq r \tag{6.13}$$

$$\sum_{\substack{k \in K: \\ t^k \neq i, s^k \neq j}} d^k x^{k,p,r}_{ij} - \sigma \leq \gamma_1 f^{p,r}_{ij} \quad \forall i,j \in N \mid (i,j) \in A\, and\ \ \forall (p,r) \in N \times N \mid p \neq r \tag{6.14}$$

$$x_{ij}^{kpr} \leq 1 \; \forall k \in K, \forall (i,j) \in A \; and \; \forall (p,r) \in NXN \mid p \neq r, p \neq j, r \neq i, i \neq t^k, j \neq s^k \tag{6.15}$$

$$x_{i,l,l'}^{k} \leq 1 \qquad \forall l, l' \in L \setminus \{0, |L|-1\} \mid l' = \{l-1, l, l+1\}, \forall i \in N \; and \; \forall k \in K \tag{6.16}$$

$$x, \sigma \geq 0 \tag{6.17}$$

where

- artRC is the total artificial routing cost

Note that *AltP-SP* is always feasible. Feasibility and optimality cuts are generated using *AltP-SP* such that if the dual variable of (6.2) is zero, then we get a feasibility cut, otherwise we get an optimality cut. In original Benders Decomposition, if $\sigma^* = 0$, then we have an integer feasible solution and hence an upper bound.

Alternative polyhedron is implemented with B&C-BD and GG-BD so far. These algorithms are given in Figure 35 and Figure 36, respectively.

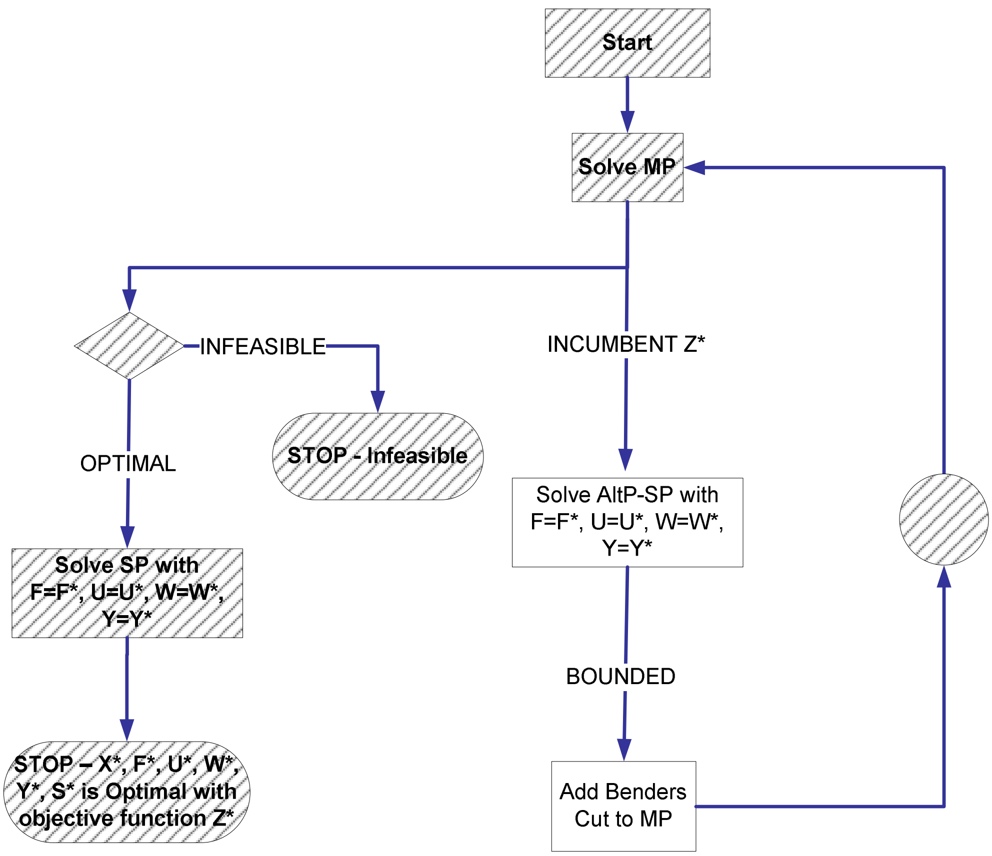


Figure 35. B&C-BD with Alternative Polyhedron (Shaded boxes are the same with B&C-BD algorithm)

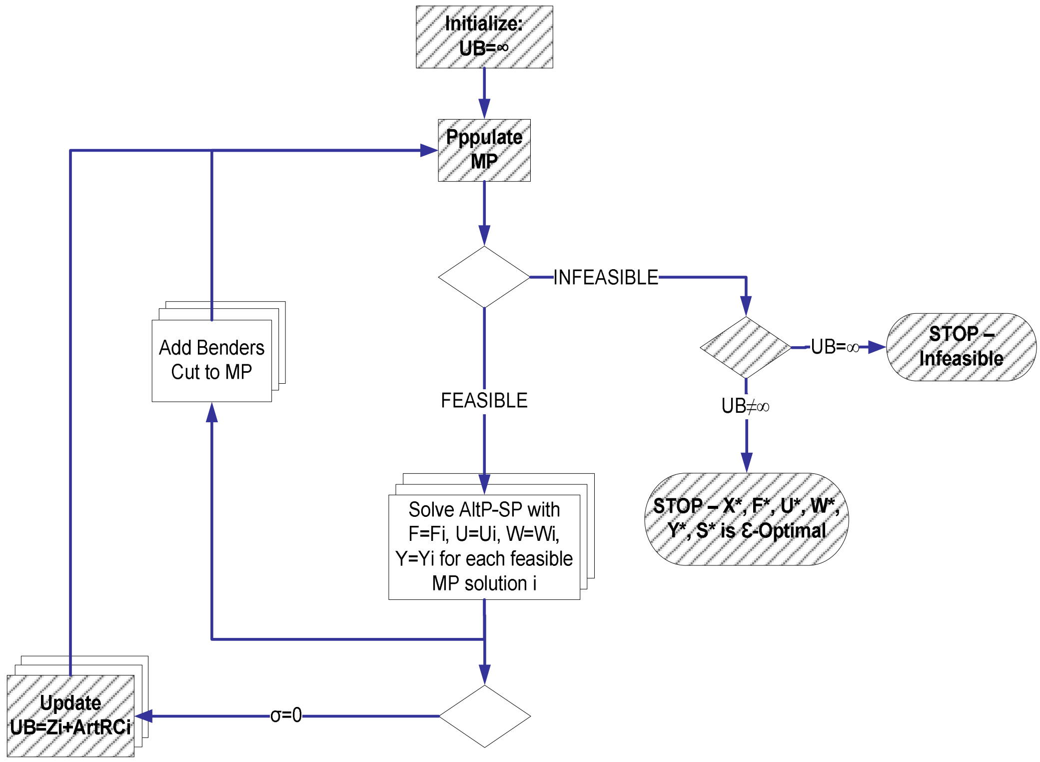


Figure 36. GG-BD with Alternative Polyhedron (Shaded boxes are the same with GG-BD algorithm)

5.4.3 Improving The Master Problem Solution

The results of development tests revealed that, although we improve the master problem's solution time by changing the algorithm variant or by improving the cut selection method, finding integer feasible solution to the master problem and the problem itself is difficult. Hence, they are important issues, especially for GG-BD variant. In order to fix these issues, we generated violated bipartition cuts and added them to master problem and added a repair heuristic for the master and sub problem solutions.

BIPARTITION CUTS

Adding bipartition cuts to the master problem together with Benders cuts is a widely used method for network design problems and reported to be successful in the literature [124], [142], [260], [265], [266]. In our problem, there is a lack of connection between the number of fiber optic cables and the number of processors in the master problem. Hence, a number of Benders iterations are needed to find a master problem solution that leads a feasible routing. We added one bipartition cut for each master problem solution. We used a variant of MAX-CUT-RATIO heuristic proposed by Gabrel et al. [124]. Since finding the most violated cut is an NP-Hard problem, an approximate solution to most violated cut problem is found by the MAX-CUT-RATIO heuristic. The violation is calculated by the ratio of the total capacity of the cut and the total demand between the node sets. The algorithm basically selects an edge randomly and finds a random cut that includes that edge. Then for each remaining node, swaps nodes between the node sets one at a time to find a more violated edge.

REPAIRING THE MASTER PROBLEM SOLUTION
Finding a feasible routing given an integer master problem solution is hard in our problem. This leads the algorithm to get stuck at a local optimum routing and so it cannot improve the flow for a while and hence the upper bound especially in GG-BD variant.

We try to overcome this issue by generating multiple integer feasible master problem solutions at each iteration but as the problem instance gets complicated, we need to increase the number of solutions in the solution pool. However, increasing number of solutions in solution pool increases the solution time of master problem significantly. Thus, repairing the infeasible solutions is necessary for improving the algorithm. The solution is repaired by using "feasopt" method of IBM ILOG Cplex. "feasopt" method is called and the amounts of infeasibilities for each capacity constraint of primal subproblem are taken from the method if the problem can be fixed without changing the flow variables. In order to achieve this, "feasopt" method adds two nonnegative variables in the form $x^{+} - x^{-}$ to each constraint that is to be repaired and solves the new LP to minimize the sum of $x^{+} - x^{-}$ terms. The infeasibilities found by feasopt are used to fix the RHS values of capacity constraints, which are in fact the number of processors, routers, fiber optic cables and lightpaths. In order to use "feasopt" method, a replica of the master problem is defined and called feascheck model. Each time an integer master problem solution is found, primal of subproblem is solved with feasopt. If the primal of subproblem can be repaired, the required capacity for processors and the lower bounds for f_{ij}^{pr} and u^{pr} variables in feascheck model are computed according to the flows in the repaired solution. Then, the feascheck model is solved to see if feasible capacities that meet the flow balance equations of the lightpath routing with the processor numbers and lower bounds obtained from the repaired solution can be found. If a feasible solution to feascheck is found, then the solution is feasible to the original problem; hence the solution is repaired.

Bipartition cuts and repair heuristic is implemented with GG-BD with alternative polyhedron algorithm so far. The flow chart of the algorithm is presented in Figure **37**.

5.4.4 Preliminary Computational Experiments for Assessing Benders Decomposition Based Algorithm Behavior

In the first phase of the computational experiments, the comparison of the NFF and EFF-EF, which is the most capable formulation in the literature, are presented. The results of these first phase computational experiments show that the NFF performs better than EFF-EF, however, in order to solve practically large networks, general-purpose integer programming solvers do not suffice. So that, tailored algorithms based on Benders decomposition are developed. The first phase of the algorithm is reported in Section 4.6. In this section, results of the second phase of the computational experiments, which aim to assess the behavior of the developed algorithms and fine tuning, are reported.

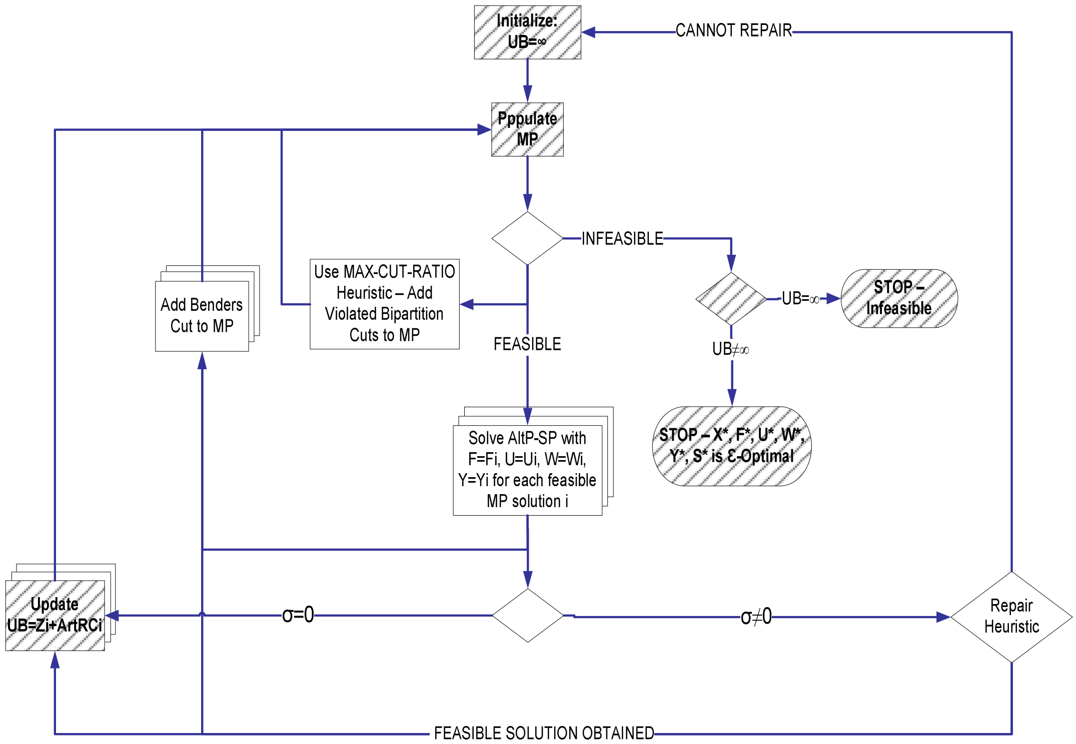


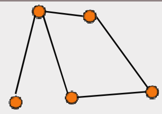
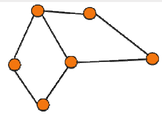
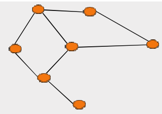
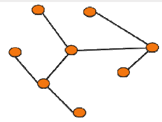
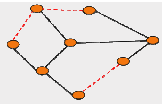
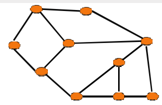
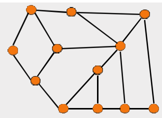
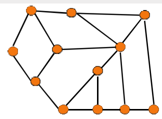
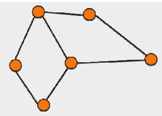
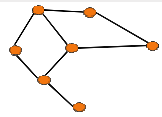
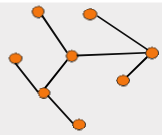
Figure 37 GG-BD with Alternative Polyhedron+Bipartition Cuts+Repair Heuristic (Shaded boxes are the same with GG-BD algorithm)

In order to assess the behavior of the algorithm variants and the add-ons of the algorithms such as alternative polyhedron, bipartition cuts and repair heuristic, we made computational experiments using 5 to 12 node networks with two and three layers. The purpose of this experimentation is to test the performance of the algorithm variants and add-ons on several networks with varying number of nodes, edges, commodities and layers to identify the advantages and disadvantages of the solution algorithms. Our aim is to observe problems of the algorithms, their strengths and weaknesses and improvement opportunities to come up with a most promising algorithm variant.

We used "Polska" network from SNDLIB [238] with 12 nodes, 18 edges and 66 commodities is for deriving our test instances. The test instances are presented in Table 20. The number of layers, nodes, edges and commodities in test instances are given in Table 20 in "#Layers","#Nodes","#Edges", and "#Comm." columns, respectively. The physical network used in each instance is presented in "Graph" column.

The test instances in Table 20 are derived by deletion of some nodes and their neighboring edges from the "polska" network. In each instance except for instances 5 and 7, there is a commodity between all pair of nodes, i.e. commodity density is equal to 1. Instance 5's and 7's commodity densities are less than 1; (i) instance 5 is same with instance 4 except for the graph topology as these instances are used to observe the effect of topology of nodes on the solution, (ii) instance 7 and instance 8 are used to compare the effect of number of commodities on the solution.

Table 20. Problem Test Instances

Problem	#Layers	#Nodes	#Edges	#Comm.	Graph
1	2	5	5	10	
2	2	6	7	15	
3	2	7	8	21	
4	2	8	7	28	
5	2	8	10	28	
6	2	10	14	45	
7	2	12	18	40	
8	2	12	18	66	
9	3	6	7	15	
10	3	7	8	21	
11	3	8	7	28	

The variants of algorithms together with the improvement methods are coded with IBM Cplex 12.5 Concert Technology using JAVA and the test instances are run on a computer with processor Intel Core i7-2720QM CPU @2.20GHz with 8 Gb RAM having Windows 7 operating system. The test results for two layer networks are presented in Table 21. In this table, algorithm framework is given as second column named as “Algorithm”. The add-ons to the algorithms for improvement are listed in the three consequent columns with names “Alt. Pol.”, “BP. Cut” and “Hr. Rp.” referring to alternative polyhedron, bipartition cuts and heuristic repair, respectively. These four columns together give the algorithm variant used, for example, the first row reports the O-BD_Feas algorithm while sixth row reports GG-BD algorithm with alternative polyhedron, bipartition cuts and heuristic repair. Number of the master problem and the subproblem iterations are reported in “#Iterations” “M” and “S” columns respectively. The number of feasibility. Optimality and bipartition cuts are also presented in “#Cuts” “F”, “O” and “B” columns. “#Rp. Sol.” Column reports the number of repaired master problem solutions by the repair heuristic. The relative gap of the best lower bound (upper bound) found with respect to the best solution of the problem instance found by MIP solution of the NFF is reported in the column GAP% - LB (GAP% - UB). The solution times are reported by “CPU” column in seconds.

Table 21. Test Results for 2-Layer Networks

P	Algorithm	Alt. Pol.	BP. Cut	Hr. Rp.	# Iteration		# Cuts			#Rp. Sol	GAP (%)		CPU (s)
					M	S	F	O	B		LB	UB	
1	O-BD_Feas				94	94	93				0		7
	O-BD_Opt				72	72	71	1				1	6
	B&C-BD					140	111	12				2	1
	B&C-BD	X				113	82	31				2	0.8
	GG-BD	X			83	132	27	105				5	4
	GG-BD	X	X	X	40	59	26	33	0	1		5	4
2	O-BD_Feas				173	173	172				0		62
	O-BD_Opt				194	194	192	2				0	112
	B&C-BD					623	581	26				3	16
	B&C-BD	X				544	384	160				0	14
	GG-BD	X			887	1496	1318	178				10	211
	GG-BD	X	X		177	414	82	332	210			21	72
	GG-BD	X	X	X	160	400	57	343	211	88		15	37
3	O-BD_Feas				492	492	492				2		1200
	O-BD_Opt											0	1200
	B&C-BD					1229	1185	34				15	1200
	B&C-BD	X				700	233	467				2	51
	GG-BD	X	X	X	225	483	85	398	204	142		10	168
4	O-BD_Feas				300	300	300				0		3600
	O-BD_Opt				212	212	212	0			NS[30]		3600
	B&C-BD					1522	1489	33				1	14400
						1207	1168	27				6	69
	B&C-BD	X				926	641	285				0,6	2145
						832	584	248				5	74
	GG-BD	X			1061	2088	130	1958				35	600
	GG-BD	X	X		976	1982	179	1803	576			36	1200
	GG-BD	X	X	X	40	93	63	30	123	21		0.5	32
5	O-BD_Feas				218	218	218					18	600
	B&C-BD	X				1658	1049	609				7	600
	GG-BD	X	X	X	228	735	84	651	370			18	600

[30] NS (No Solution): Algorithm cannot find any integer feasible solution

Table 21 (Con't)

P	Algorithm	Alt. Pol.	BP. Cut	Hr. Rp.	# Iteration		# Cuts			#Rp. Sol	GAP (%)[31]		CPU (s)
					M	S	F	O	B		LB	UB	
6	O-BD_Feas					242	242				13		7353
	B&C-BD _Feas					1090	1090				21		7200
	B&C-BD	X				3112	1965	1147			12	10	7200
	GG-BD	X	X	X	498	746	458	288	187	344		29	7200
7	O-BD_Feas				271	271	271				13		14400
	B&C-BD_Feas					285	285				41	NS	14400
	B&C-BD	X				481	353	128			26	NS	10000
	GG-BD		X	X	260	386	386	0	126	5		9	14400
8	B&C-BD	X				80	80				77	NS	14400
	GG-BD		X	X	370	511	511	0	150	6		27	14400

Test results for three layer networks are given in Table 22.

Table 22. Test Results for 3-Layer Networks

P	Algorithm	# Iteration		# Cuts		GAP (%)		CPU (s)
		M	S	F	O	LB	UB	
9	O-BD_Opt	241	241	239	2	0		197
	B&C-BD _Opt		572	530	20	0	0	78
	O-BD_Feas	207	207	206		0		235
	B&C-BD _Feas		845	821	23	0	0	147
10	O-BD_Opt	321	321	321		10		2400
	B&C-BD _Opt		1300	1256	25	6	9	2400
	O-BD_Feas	437	437	436		5		1200
	B&C-BD _Feas		1295	1308	13	1	2	1200
11	O-BD_Opt	210	210	210	0	7		7200
	B&C-BD _Opt		1326	1298	28	0	6	7200
	O-BD_Feas	277	277	277		-7		2400
	B&C-BD _Feas		1739	1722		-9	4	502

For two-layer networks we observe that:

- For small networks with 5-7 nodes, BC-BD with alternative polyhedron is the fastest variant. However, for 10 and 12 node networks, alternative polyhedron slows down the cut generation procedure and the BC-BD algorithm cannot perform well.
- B&C-BD algorithm performs worse than O-BD algorithm for the test problems. However, it decreases the optimality gap faster than O-BD at the first iterations of the algorithm.
- Using alternative polyhedron to generate Benders cuts increases solution performance of B&C-BD.

[31] NS (No Solution): Algorithm cannot find any integer feasible solution

- It is interesting that GG-BD algorithm with alternative polyhedron, bipartition cuts and repair heuristic seems like the most promising algorithm variant for 8-node instance while it is not for the 6-node instance. It is because 6-node instance has a cycle in its original graph while 8-node does not. If we change the 8-node instance with the instance 3 by adding the edges shown with dashed lines, all algorithms performances get worse including GG-BD. This is mainly because of the cycles in the transmission graphs forcing the algorithms to get stuck at a local optimum at subproblem and the bounds not to improve. In GG-BD, its main reason is the repair heuristic that uses Cplex's built-in functions. The repair heuristic needs to be improved and an improvement heuristic needs to be developed for solving this problem.
- Test instance 3 is solved by algorithms that seem most promising from the results of test instances 1 and 2.
    - GG-BD calculates an upper bound at the 4^{th} second, and it cannot improve the upper bound for the rest of 10 minutes more than 0.5 units.
    - B&C-BD with alternative polyhedron steadily converged to the optimal.
    - O-BD_Feas converges really slowly and cannot perform as good as other algorithms when the problem gets complicated.
    - GG-BD is promising since it can close the gap faster than B&C-BD, though B&C-BD can still be improved by using heuristic upper bounds and bipartition cuts.
- Algorithm variants with artificial routing cost cannot perform better than the variants without artificial routing costs for larger networks since artificial routing costs make primal subproblem a large scale minimum cost multicommodity network flow problem. Without artificial routing costs, the primal subproblem is only a feasibility seeking problem. It is obvious that finding a feasible solution needs less computation time than finding an optimal solution.
- Even the basic variants of the algorithms can solve three-layer instances once the two layer version can be solved by the algorithms. This is an expected outcome, since adding a layer increases the number of processor links and transmission links stay intact. The main component increasing the complexity of the problem is the number of parallel transmission links.

The weaknesses and strengths of the algorithm variants are summarized in Table **23**.

As a result of these computational tests, we can list the improvement opportunities for the algorithms as follows:

- Repair heuristic can be improved and an improvement heuristic that prevent GG-BD from getting stuck at local optimum routing can be developed.
- Feasible solutions found by repair heuristics can be injected to B&C-BD algorithm as incumbents.
- Bipartition cuts can be added to B&C-BD algorithm as user cuts to find better master solutions.

Table 23. Strengths and Weaknesses of Algorithm Variants

Algorithm	Strengths	Weaknesses
O-BD_Feas	Converges steadily to the optimal value; if there is enough time optimality is reached. SP problem is a feasibility seeking problem, hence easy to solve.	No upper bound information is generated. Feasibility cuts are weak; algorithm converges slowly.
O-BD_Opt	If an optimality cut is generated, converges faster than O-BD_Feas.	Optimality cuts are difficult to generate. Artificial routing costs make the SP problem a minimization problem although the original SP is a feasibility seeking problem. Added optimality cuts do not have enough benefit as the added computational burden of adding artificial cuts.
B&C-BD_Feas	Converges steadily to the optimal value like O-BD_Feas algorithm.	For networks with nodes 10 and 12, its performance is not better than O-BD_Feas algorithm.
B&C-BD_Opt	If optimality cuts are found, converges faster than B&C-BD_Feas. Converges faster than O-BD_Opt.	Optimality cuts are difficult to generate. Artificial routing costs make the SP problem a minimization problem although the original SP is a feasibility seeking problem. Added optimality cuts do not have enough benefit as the added computational burden of adding artificial cuts.
B&C-BD_AltPol	Converges faster than B&C-BD_Opt for smaller networks with 5, 6, 7 and 8 nodes.	For larger networks having 10 and 12 nodes, alternative polyhedron takes too long to find a violated cut for each subproblem iteration.
GG-BD	With a proper repair heuristic, it converges faster than all other algorithm variants	Needs a heuristic method to repair and improve integer solutions to find better upper bounds Has several algorithm parameters to be fine tuned including the number of feasible master solutions in the solution pool and the ε value for upper bound.

The weaknesses and strengths of the improvement methods are summarized in Table 24.

Table 24. Strengths and Weaknesses of Improvement Methods

Improvement Method	Strengths	Weaknesses
Alternative polyhedron	It finds efficient Benders cuts.	it is computationally difficult to solve the alternative polyhedron instead of subproblem
Bipartition cuts	Speeds up to find an integer feasible master problem solution	Separation is computationally expensive Needs solution pool management, else memory problems arise for large problems with node size 10 and 12.
Repair heuristic	Speeds up the algorithms by finding upper bounds	Current repair heuristic is based on feasopt method of Cplex and it is not capable of repairing most of the master problem solutions efficiently.

5.4.5 Algorithm Improvement

Although for smaller networks, it is outperformed by the algorithm variants, GG-BD with bipartition cuts and repair heuristic seems to be the most promising algorithm variant for finding good solutions for large networks. So, we improve the algorithm by adding an improvement step after repair heuristic to enhance the upper bounds we find. Finding an upper bound gets more difficult as the network gets larger, so we make a local search around the feasible integer solution found by using the master problem solution and the repair heuristic. We use an add-and-drop heuristic for improving the upper bound. The GG-BD algorithm with repair and improvement heuristic (GG-BD_IR) is presented in Figure **38**. The add-drop algorithm that is used with GG-BD_IR is given below:

Algorithm – ADD-DROP

1. Select an edge (i,j) that is already in use in the last found feasible solution., i.e., $V_{ij}^{m} > 0$ for $\exists\ m \in M_F$ and $Edge_{DROP} := (i,j)$
2. Find two partitions S and S' of the node set N such that $S \cup S' = N$ and each of these partitions induce a connected graph containing one end node of $Edge_{DROP}$ such that $i \in S,\ i \notin S',\ j \in S$ and $j \notin S'$.
3. List edges $(k,l) \in E : k \in S, l \in S'$. Randomly select an edge (i',j') from this list and set $Edge_{ADD} := (i',j')$.
4. Delete $Edge_{DROP} := (i,j)$ from the last feasible solution by setting $V_{ij}^{m} = 0$ for $\forall m \in M$ in feasibility checking model and $x_{ij}^{k,pr} = 0,$ for $\forall k \in K$ and $\forall (p,r) \in N \times N$ in primal of the subproblem.
5. Add $Edge_{ADD} = (i',j')$ to the last feasible solution by setting $V_{ij}^{m} = 1$ for $\forall m \in M$ in feasibility checking model
6. If there is no a feasible solution with a better objective function than the best upper bound after changes in steps 5 and 6, recover these changes.

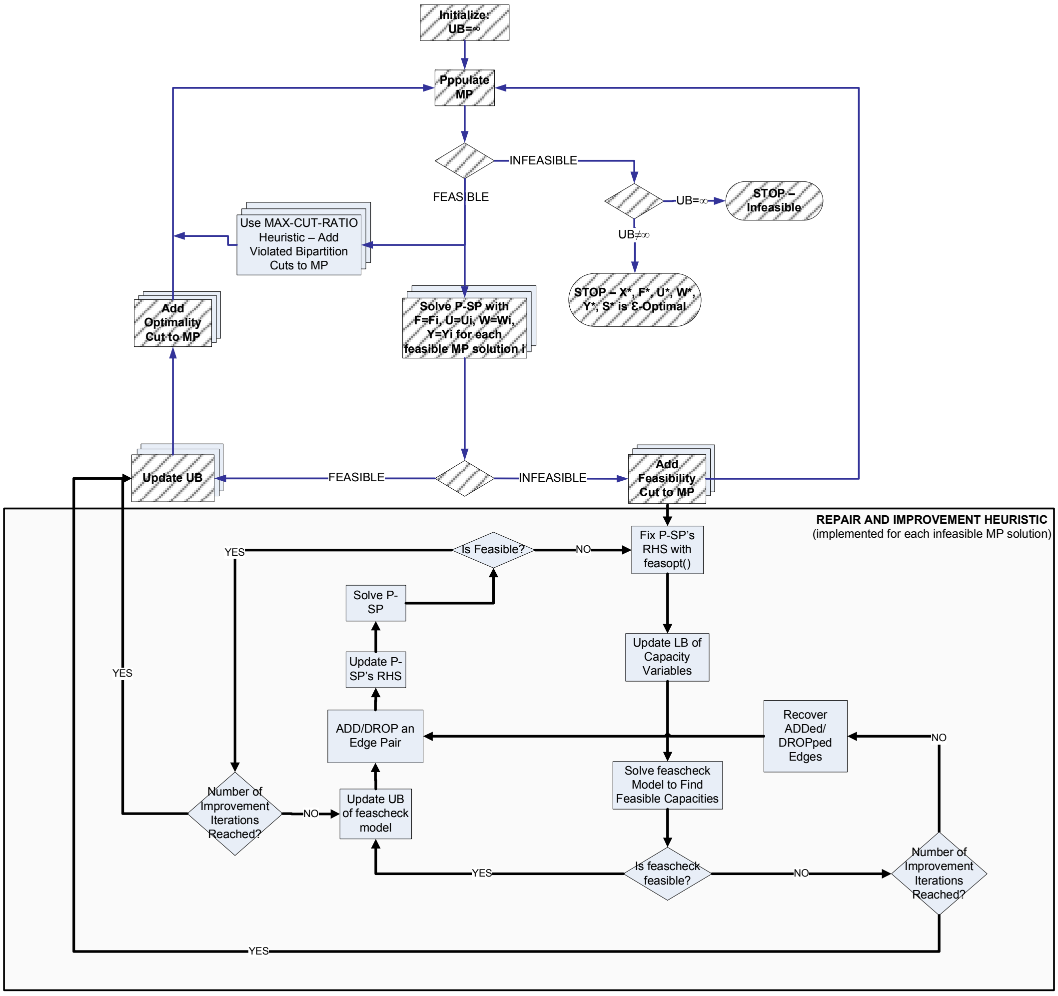


Figure 38. GG-BD_IR: GG-BD with Repair and Improvement Heuristic and Bipartition Cuts

5.5 Extensive Computational Experiments for Assessing Benders Decomposition Algorithms

The results of preliminary computational experiments, i.e., second phase of computational experiments, in Section 5.4.4 show that GG-BD with bipartition cuts and repair heuristic provides the most promising solutions. The results also reveal that GG-BD with bipartition cuts and repair heuristic still needs a mechanism to enhance the upper bounds. Therefore, an improvement algorithm is developed resulting in GG-BD with repair and improvement heuristics and bipartition cuts (GG-BD_IR). In the third phase of the computational experiments, experimentation is performed for assessing performance of this final algorithm, GG-BD_IR, against MIP solution of the NFF. In the literature, any library of test instances or any standard test instance set for multi-layer telecommunication networks does not exist. The single-layer test instances in SNDLIB are modified to be multilayer and are used in multilayer network design studies in the literature, namely in [5–7], [122], [267] and [264]. So, we consolidate the information about test instances given in several studies as [5–7], [122], [267] and [264], which include computational tests of multi-layer telecommunication networks, and single layer test instances provided in SNDLIB. The details of the computational experiments and information provided by these studies are given in Appendix C.

There are 25 network instances in SNDLIB. These are presented in Table 25. In the table, names of the instances in the library are given in "Instance" column. The number of nodes, edges and commodities are reported in "|V|", "|E|", and "|K|" column, respectively. "E" ("K") under density column presents the edge (commodity) density of the instances, i.e., the ratio of number of edges (commodities) in the instance to the number of node pairs in the instance.

Table 25. SNDLIB Test Instances

Instance	**|V|**	**|E|**	**|K|**	**Density**	
				E	**K**
dfn-bwin	10	45	90	1.00	2.00
pdh	11	34	24	0.62	0.44
di-yuan	11	42	22	0.76	0.40
dfn-gwin	11	47	110	0.85	2.00
abilene	12	15	132	0.23	2.00
polska	12	18	66	0.27	1.00
nobel-us	14	21	91	0.23	1.00
atlanta	15	22	210	0.21	2.00
newyork	16	49	240	0.41	2.00
nobel-germany	17	26	121	0.19	0.89
geant	22	36	462	0.16	2.00
ta1	24	55	396	0.20	1.43
france	25	45	300	0.15	1.00
janos-us	26	42	650	0.26	2.00
norway	27	51	351	0.15	1.00
sun	27	51	67	0.29	0.19
nobel-eu	28	41	378	0.11	1.00
india35	35	80	595	0.13	1.00
cost266	37	57	1332	0.09	2.00
janos-us-ca	39	61	1482	0.16	2.00
giul39	39	86	1471	0.23	1.99
pioro40	40	89	780	0.11	1.00
germany50	50	88	662	0.07	0.54
zib54	54	81	1501	0.06	1.05
ta2	65	108	1869	0.05	0.90

The SNDLIB instances are single layer instances, although, some of these test instances are commonly used in multilayer telecommunication network design studies as [5–7], [122], [267] and [264] after some modifications to make the instances multilayer. However, there is not any standard way to modify the test instances in the literature. Moreover, information provided about the multilayer instances derived from the SNDLIB is limited in these studies. So that, we consolidate the available information in the literature with the single layer network instances in SNDLIB to generate our multilayer test instances. The effort to consolidate the previous studies and SNDLIB instances results with a methodology to make single layer SNDLIB instances multi-layer as:

- Physical links are taken from SNDLIB ([5–7], [122], [267], [264]).
- The size of capacity module is selected as 40 for the physical layer, i.e., each fiber optic cable can carry up to 40 different lightpaths provided that they have different wavelengths. It is used as 4, 40 or 80 in [5–7], [122], [267], [264].
- The size of the logical capacity module is taken as 10Gbit/s independent from the distances of nodes. It is used as 2.5 Gbit/s, 10Gbit/s and 40Gbit/s in [5–7], [122], [267], [264].
- Physical costs are derived from the costs of the first available capacity module of the original problem as done in [122] and done for polska network in [7].
- In [5], [268] and [121], logical link costs are randomly generated. In [7], cost values are taken from the industry. We used the a cost model for WDM layer due to Gunkel et al. [263] to first observe the order of magnitude of cost of components with respect to the physical links. Gunkel et al. [269] gives a cost model for WDM layer which is valid for 2 layer case. In the cost model, normalized costs of equipments, sample link cost modules for several distances and sample node cost modules are presented. Using this study, we see that processor costs correspond to transponder costs, router costs correspond to EXC or OADM costs. Using the information in this cost model, mux/demux costs are uniformly generated within [%30, %80] range of the average physical link cost, i.e., average link cost of test instance given in the SNDLIB. With the same reasoning, router costs are uniformly generated within [%20, %30] range of the average physical link cost.

- Studies that use explicit approach to model logical links use hop constraints [122] and admissible physical paths for logical links [7] to decrease the size of feasible space. We use admissible physical paths to decrease the size of feasible space for instances with more than 20 nodes. Admissible physical paths are determined between each node pair according to the distance between nodes, i.e. k-shortest paths between nodes are determined as admissible paths and other paths are not allowed to be used. SNDLIB instances include the location of nodes either in coordinate plane as x-y coordinates or as longitude and latitude of the location of nodes on earth. This data is used to compute k-shortest paths between the node pairs.

The GG-BD_IR algorithm is coded with IBM Cplex 12.5 Concert Technology using JAVA. The NFF is solved by Cplex 12.5 MIP solver. The test instances are run on a computer with processor Intel Core i7-3720QM CPU @2.60GHz with 16 Gb RAM having Windows 7 operating system.

Table 26 reports the results of the NFF solved by Cplex 12.5 MIP solver. MIP solution of the NFF can find integer feasible solutions to only 8 of the 25 test instances. 17 test instances cannot be solved because of out of memory errors. For smaller test instances, the problem can be modeled by Cplex but MIP solver gives an out of memory error (NC). For larger instances, either Concert Technology cannot construct the model (NCT) or Java cannot handle the Cplex-Concert Technology operations during construction of the model due to physical memory limit of the computer (NJ).

Table 26. MIP Solutions for SNDLIB Test Instances

Instance	LB	UB	Rel Gap (%)	Time (sec)	# Integer Sol.
dfn-bwin	2,700,521	2,784,830	3.03	14,400	28
pdh	2,120,572	2,813,325	24.62	14,400	25
di-yuan	1,230,606	1,232,353	0.14	10,932	12
dfn-gwin	28,778	43,175	33.35	14,400	2
abilene	1,863,275	1,865,133	0.10	8,520	15
polska	4,958	6,523	23.99	14,400	35
nobel-us	135,520	238,862	43.26	14,400	7
atlanta	132,883,000	149,195,000	10.93	14,400	2
newyork	2,232,310	NA	NA	14,400	0
nobel-germany	70,315	NA	NA	14,400	0
geant	N/C	N/C	N/C	41	0
ta1	N/C	N/C	N/C	54	0
france	N/C	N/C	N/C	40	0
janos-us	N/CT	N/CT	N/CT	N/CT	0
norway	N/CT	N/CT	N/CT	N/CT	0
sun	N/S	N/S	N/S	14400	0
nobel-eu	N/C	N/C	N/C	40	0
india35	N/CT	N/CT	N/CT	N/CT	0
cost266	N/CT	N/CT	N/CT	N/CT	0
janos-us-ca	N/J	N/J	N/J	N/J	0
giul39	N/J	N/J	N/J	N/J	0
pioro40	N/J	N/J	N/J	N/J	0
germany50	N/J	N/J	N/J	N/J	0
zib54	N/J	N/J	N/J	N/J	0
ta2	N/J	N/J	N/J	N/J	0

Our first observation about the GG-BD_IR solution and Cplex MIP solver solution of the NFF is that GG-BD_IR can find good upper bounds faster than the MIP solver consistently for the test instances. Figure 39 presents the convergence rates of the MIP solver (NFF-MIP) and GG-BD_IR (GGBD) of the NFF for 12-node polska network and 14-node nobel-us network. It can be argued that, the MLNDP does not need a fast solution. However, fast convergence of GG-BD_IR shows that GG-BD_IR can still be successful in solving the NFF incorporated with several practical side constraints, although, MIP solver may fail to solve. In addition, GG-BD_IR has the potential to be used in another algorithm to find good upper bounds.

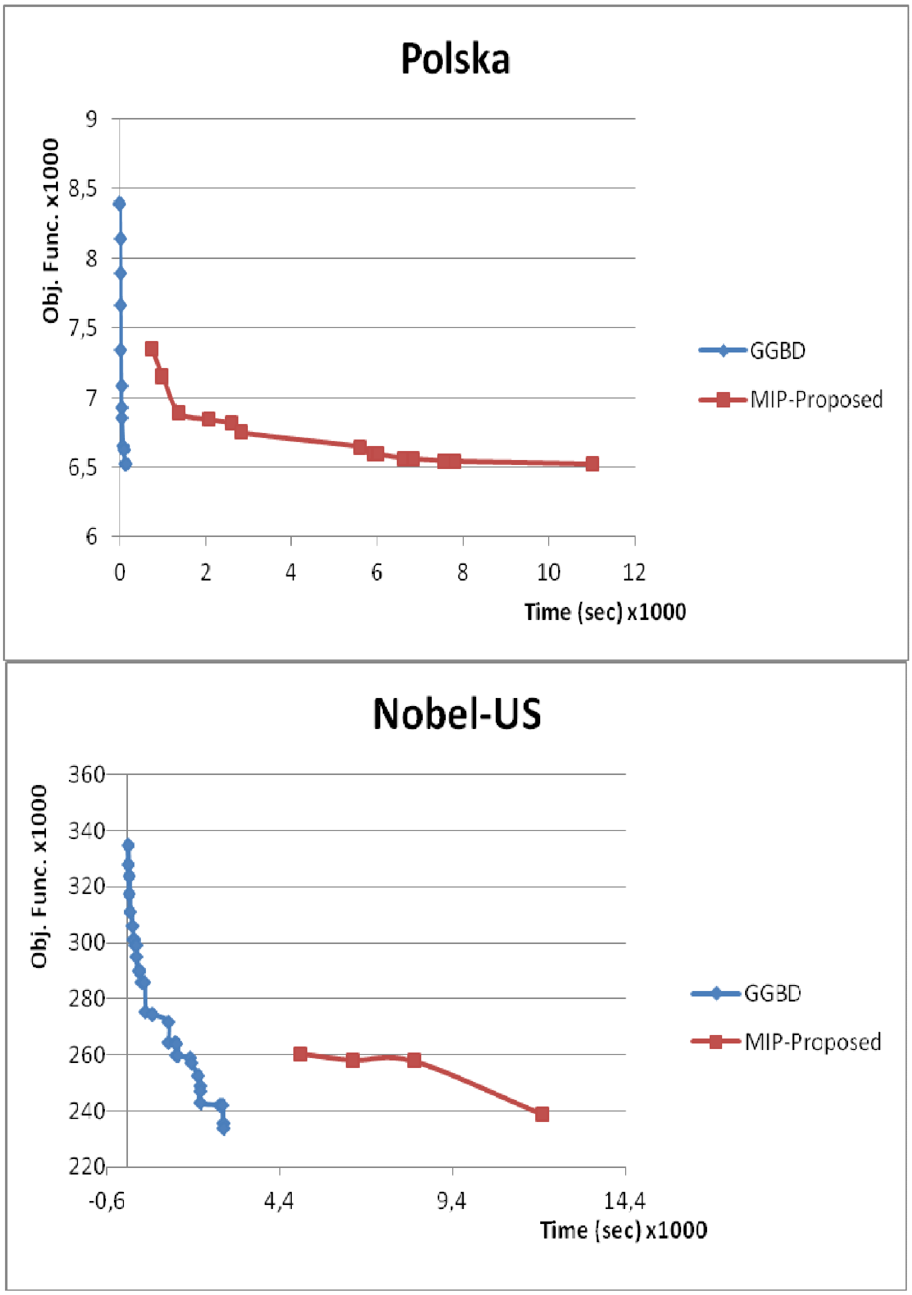


Figure 39. Convergence of GG-BD_IR and Cplex MIP Solver

The computational results are presented in Table 27, Table 28, Table 29 and Table 30. In these tables, instance names are presented in “Instance” column, with their number of nodes given in “Nodes” column. Instead of number of nodes, number of layers is reported in Table 30 by the “#Layer” column. “LB” (“UB”) column under “NFF – MIP” is the lower bound (upper bound) found by MIP solution of the NFF within “Time (sec)” amount of time. Relative gap between these upper and lower bounds are reported by “GAP (%)” column under “NFF – MIP”. “UB” column under “GG-BD_IR” is the upper bound found by GG-BD_IR solution of the NFF within “Time (sec)” amount of time. In The GG-BD_IR algorithm terminates when the master problem cannot find any solution which has an objective function less than *UB(1-Ɛ)*. If the algorithm does not terminate, we terminate the algorithm due to time limit. "GAP (%)" column under "GG-BD_IR" is Ɛ value if the GG-BD_IR algorithm terminated before the time limit, else it is the relative gap between the lower bound found by MIP solver and upper bound found by GG-BD_IR at the end of four hours. If any upper bound is found by the MIP solution of the NFF by MIP solver, "IMP (%)" column presents the improvement of this upper bound by GG-BD_IR algorithm, i.e., the relative gap between upper bounds of the NFF found by MIP solver and GG-BD_IR. The italic values under this column indicates the improvement of GG-BD_IR upper bound with respect to the first feasible solution found and reported, if no feasible solution can be found to the NFF by the MIP solver.

We solve test instances with less than 20 nodes without any restriction on the logical links with GG-BD_IR, i.e., all parallel logical links are admissible in the logical layer. The results are presented in Table 27. For the problems in Table 27, Ɛ is taken as 15% for newyork instance and 0.5% for all other instances. Time limit is taken as four hours (14,400 sec) for these instances.

Table 27. GG-BD_IR Results for Test Instances with Less Than 20 Nodes

Instance	**#Node**	**NFF-MIP**				**GG-BD_IR**			**IMP (%)**
		LB	**UB**	**Time (sec)**	**GAP (%)**	**UB**	**Time (sec)**	**GAP (%)**	
dfn-bwin	10	2,700,521	2,784,830 (2,962,971)	14,400	3.03	2,789,638	986	0.5	-0.17 (5.85)
pdh	11	2,120,572	2,813,325 (2986168)	14,400	24.62	2,811,741	265	0.5	0.06 (5.84)
di-yuan	11	1,230,606	1,232,353 (1246254)	14,400	0.14	1,232,359	902	0.5	0.00 (1.11)
dfn-gwin	11	28,778	43,175 (43,175)	14,400	33.35	39,446	3,361	0.5	8.64 (8.64)
polska	12	4,958	6,523 (N/A)	14,400	23.99	6,526	380	0.5	-0.05 (-)
nobel-us	14	135,520	238,862 (N/A)	14,400	43.26	233,751	2,815	0.5	2.14 (-)
newyork	16	2,232,310	NA	14,400	-	4,735,743	10,890	15	*29.78*
nobel-germany	17	70,315	NA	14,400	-	122,191	9,247	0.50	*21.70*(-)
abilene	12	1,863,275	1,865,133 (N/A)	439	0.10	1,874,427	89	0.50	-0.50 (-)
atlanta	15	132,883,000	149,195,000	14,400	10.93	158,901,287	14,400	16.37	-6.51

It is seen from Table 27 that GG-BD_IR finds better or same upper bound as MIP solver for 7 of the 8 test instances that MIP solver manages to find an upper bound. For the Atlanta instance, GG-BD_IR converges the best upper bound less than 7%. GG-BD_IR finds feasible solutions for the other 2 test instances that MIP solver cannot find any feasible solution. Four these instances, GG-BD_IR improves the initial feasible solutions more than 20% in four-hour time. These values are reported italic in "IMP (%)" column of Table 27.

As it is observed that the GG-BD_IR algorithm converges faster than the MIP solver, we also reported the upper bounds obtained by MIP solver at the termination time of the GG-BD_IR as the values in parenthesis under "NFF-MIP - UB" column. The improvement of the upper bound of the NFF by usage of GG-BD_IR instead of MIP solution at the end of the GG-BD_IR termination time is also given by "IMP (%)" column as values in parenthesis. The GG-BD_IR terminates due to $\mathcal{E}$-optimality for 8 of 10 test instances. For four of these test instances, the MIP solver cannot even find an upper bound to the NFF at the time of termination of the GG-BD_IR. For others, the GG-BD_IR's upper bound is better than the upper bound found by the MIP solver by 1-9%

The number of parallel logical links between the nodes is restricted for test instances more than 20 nodes, i.e., logical links associated to k-shortest paths in physical layer are admissible between each node. k is intact for all node pairs in a test instance, however, it decreases for the test instances with more nodes and commodities. These restricted test problems are also run with Cplex MIP solver. The results are reported in Table 28. For the problems in Table 28, $\mathcal{E}$ is taken as 1%. Time limit is taken as two hours for sun and nobel-eu instances and six hours (21,600 sec) for others.

Table 28. GG-BD_IR Results for Test Instances with 20-30 Nodes

Instance	k	#Node	NFF-MIP				GG-BD			Imp (%)
			LB	UB	Time (sec)	GAP (%)	UB	Time (sec)	GAP (%)	
geant	10	22	2,039,064	N/C	1,267	-	2,094,763	21,600	2.66	*0.00*
janos-us	10	26	32,207	N/C	3,438	-	63,950	12,748	1.00	*6.56*
norway	5	27	559,353	N/C	5,397	-	1,079,652	17,749	1.00	*50.30*
sun	5	27	326	850	7,200	61.65	801	519	1.00	5.76
nobel-eu	5	28	196,742	N/A	7,200	-	377,049	3,370	1.00	*22.82*
france	10	25	18,340	23,960	21,600	23.45	26,549	21,600	30.92	-10.81

As reported in Table 28, the GG-BD_IR improves the MIP solution of the NFF for 5 of 6 test instances. Cplex MIP solver gives out-of-memory error for geant, janos-us and norway instances before finding any feasible solution. The GG-BD_IR finds an upper bound with 2.66% optimality gap for geant in six-hour time limit. The GG-BD_IR terminates due to $\mathcal{E}$-optimality for janos-us and norway networks in less than five hours and for sun and nobel-us instances in less than one hour. The GG-BD_IR improves MIP solution of the NFF at the end of two hours by 5.76% for sun instance. MIP solver cannot even find any feasible solution for nobel-us instance within two hours.

Due to problem sizes for test instances with more than 30 nodes as presented in Table 35 in Appendix D, memory requirement increases to solve the test problems both with the GG-BD_IR and MIP solution for the NFF. Thus, we use a work station with Intel Zeon CPU X5650 @2.67GHz-2.66GHz with 48Gb RAM having Windows 7 operating system for these test instances. We solved 6 test instances with 30 to 50 nodes. The results are reported in Table 29**Error! Reference source not found.** with an $\mathcal{E}$ value of 1% and six hours time limit. At the end of six hours, the GG-BD_IR finds feasible solutions to the test instances though MIP solver cannot find any feasible solutions. It is seen that, the GG-BD_IR can find good upper bounds for 39-node janos-us-ca and 37-node cost266 test instances.

Table 29. GG-BD_IR Results for Test Instances with More Than 30 Nodes

Instance	#Node	NFF-MIP				GG-BD		
		LB	UB	Time (sec)	GAP (%)	UB	Time (sec)	GAP (%)
india35	35	9,641	N/A	21,600	-	16,128	21,600	40.22
cost266	37	33,533,100	N/A	21,600	-	37,908,860	21,600	11.54
janos-us-ca	39	425,528	N/A	21,600	-	454,539	21,600	6.38
giul39	39	1,256	N/A	21,600	-	5,599	21,600	77.57
pioro40	40	28,656	N/A	21,600	-	54,941	21,600	47.84
germany50	50	215,353	N/A	21,600	-	474,061	21,600	54.57

For test problems with high gaps in Table 29, we increased $\mathcal{E}$-value to see if the gap is due to bad lower bounds found by MIP solver. The GG-BD_IR terminated within six hours (20630 sec) using an $\mathcal{E}$-value of 20% for india35 with an upper bound of 16,286. Since increasing the $\mathcal{E}$-value makes it more difficult to find feasible master problem solutions, increasing $\mathcal{E}$-value does not help improving upper bounds for giul34, pioro40 and germany50 instances, which are the most difficult test instances. For these test instances, run time can be increased, or test instance specific improvement can be made, e.g. we can restrict the grooming nodes, for better solutions. In addition, increasing the number of feasible master problem solutions for master problem iterations and using parallel processing techniques for subproblem solutions can help exploring the master problem feasible space with larger $\mathcal{E}$-values, hence upper bounds can be improved for these problem instances.

We test the performance of the GG-BD_IR for networks with more than two layers. We used polska, newyork and nobel-eu instances without any restrictions on the logical links and solved three-layer and five-layer network design problems. The results are presented in Table 30. Upper bounds can be found for only polska instance within two hours with MIP solution of the NFF. For these instances, the GG-BD_IR algorithm terminates before the time limit is reached with an $\mathcal{E}$-value of 15% (20%) for newyork instance with three (five) layers and $\mathcal{E}$-value of 1% for other instances. The results indicate that we can find good upper bounds for three-layer and five-layer network design problems with up to 17 nodes without any restriction on the number of logical links in both of the logical layers.

Table 30. Test Results with 3- Layer and 5-Layer Network Instances

Instance	#Layer	#Node	NFF-MIP				GG-BD			Imp (%)
			LB	UB	Time (sec)	GAP (%)	UB	Time (sec)	GAP (%)	
polska	3	12	8,036	10,451	7,200	23.11	10,167	121	1.00	2.72
nobel-germany	3	17	124,680	N/A	7,200	-	183,183	678	1.00	*10.56*
newyork	3	16	4,047,286	N/A	7,200	-	6,473,519	9685	15	*32.0*
polska	5	12	20,330	21,861	7,200	7.00	21,752	739	1.00	0.50
nobel-germany	5	17	236,975	N/A	7,200	-	294,513	4,957	1.00	*7.09*
newyork	5	16	7,677,238	N/A	22,729	-	10,330,272	22,729	20	*15.26*

In summary, we can find good upper bounds for test instances with up to 30 nodes with the GG-BD_IR. For these test instances, we see that the GG-BD_IR can find good upper bounds even for the test instances that MIP solution of the NFF cannot find any feasible solutions. For the problems that MIP solution can find upper bounds to NFF, we observe that the GG-BD_IR converges much faster than MIP solver. The GG-BD_IR can solve three and five layer test instances with up to 17-nodes. The GG-BD_IR can find good upper bounds for 37 and 39 node test instances and can find feasible solutions to other four test instances with more than 30 nodes. Any feasible solutions to test instances with more than 30-nodes cannot be found by MIP solution of the NFF. To the best of our knowledge, test results to MLNDP with more than two layers are reported for the first time in the literature. In addition, 30-50 node network instance results found by a generic, i.e., does not include any test instance specific fine tuning, exact algorithm using a flow formulation with optimal routing decisions are reported for the first time in the literature.

CHAPTER 6

CONCLUSION

In this thesis, we study the telecommunication network design problems. First of all, we review the TNDP and network optimization problems to solve the TNDP to identify the state-of-the art of the literature about the TNDP and research questions. Our literature reviews reveal the necessity to find efficient exact and heuristic algorithms for solving practical larger multi-layer telecommunication network design problems. We address the MLNDP in an extensive way.

We propose a novel single-network representation (NFM) based mathematical formulation (NFF) for the MLNDP. Although our computational tests and examples are based on optical networks, the NFM and NFF can be used for when different technologies used to transmit signals including wireless networks. Our extensive computational tests are reported for comparing the NFF with the EFF-EF formulation, which is the most practically relevant MLNDP formulation in the literature. We develop tailored solution algorithms based on Benders decomposition to solve larger practical network instances. The weaknesses and strengths of alternative algorithms are illustrated using computational tests and the most promising algorithm is found. Algorithms' solution performances are compared with general purpose integer programming solver and original Benders decomposition algorithm.

Our main findings and contributions can be summarized as below:

- We provide an up to date survey and classification about telecommunication network design problems including new problem types to identify challenges and future research areas of the telecommunication network design problem.
- We provide a guide for OR researchers to match telecommunication network optimization problems to telecommunication design problems in solving telecommunication network design problems.
- Our survey on network optimization problems to solve the TNDP unifies the notation of available network optimization problems and provides a toolbox for OR researchers to study the TNDP.
- We propose a novel graph representation for the MLNDP using a single-mega network, called NFM. This NFM facilitates using the solution methods for single layer network optimization for multi-layer network design problems. In addition, matching of telecommunication hardware and network components can be done with the NFM easier than the existing multi-layer representation.
- A novel mathematical formulation, NFF, for the MLNDP based on the NFM is proposed, called the network flow formulation (NFF). The NFF has $O(|K||E||N|^2)$

constraints and $O(|K||E||N|^2)$ variables, whereas explicit flow formulation with edge flows (EFF-EF), the most capable one regarding modeling practical side constraints, has $O(|K||E||L|)$ constraints and $O(|L|)$ variables where L is the set of logical links and $|L|>>|N|$ especially when the number of nodes increase. Then, the NFF decreases the complexity of the existing EFF-EF formulation without degrading its modeling capabilities. It is very important as the NFF can solve the MLNDP with three and more layers which were not computationally tractable with EFF-EF until now.

- Computational experiments reveal that the NFF's LP relaxation is tighter than the EFF-EF's LP relaxation.
- We develop and implement different variants and improvement techniques for Benders decomposition based algorithms. We compare the algorithms to identify their weaknesses and strengths and to figure out improvement opportunities to come up with the most promising algorithm. The computational tests show that for the large networks, the Ɛ-optimal Benders algorithm framework with improvement and repair heuristic and bipartition cuts, which we call GG-BD_IR, outperforms the original Benders decomposition and Benders decomposition algorithm within branch and cut framework.
- We consolidate all available test problems in the literature and provide extensive test results.
  - We test the GG-BD_IR algorithm with 22 test instances provided in SNDLIB.
  - Test instances have 10-50 nodes with two layers and 12-17 nodes with three and five layers. There are not any limitations in the physical links that can be used to route the logical links for the test instances with 10-17 nodes.
  - We provide a method for making single-layer test instances multi-layer.
  - The computational tests reveal that the GG-BD_IR can find good upper bounds to the test instances faster than MIP solution of the NFF, if any feasible solution can be found using MIP solver. Thus, the GG-BD_IR algorithm is promising to solve MNLDP incorporated with practical side constraints that are unlikely to be solved by general-purpose MIP solvers. In addition, the GG-BD_IR algorithm can be used to generate upper bounds within other algorithms since it is fast.
  - The GG-BD_IR finds upper bounds to 12 test instances that MIP solver cannot even find a feasible solution to the NFF.
  - To the best of our knowledge, results of computational tests with instances having 30-50 nodes that are solved a generic algorithm based on flow formulation, i.e., an algorithm that is not test instant specific fine tuned and includes optimal physical layer routings in the solution, are reported for the first time in the literature.
  - Three and five layer test results are reported for test instances with up to 17 nodes. Together with the NFF based on the novel NFM and the GG-BD_IR algorithm, we can solve three and five layer network instances up to 17 nodes without any restriction on the logical links and find feasible solutions. We can achieve this without any degradation of the modeling capabilities of the EFF-EF which is the most practically relevant formulation in the literature. To the best of our knowledge, this is the first time in the literature.

The future research directions are listed below:

- In the thesis, we focused on a general model for the MLDNP network design. However, the model can be enhanced with technology specific information about hardware and link cost and capacity, and relevant practical side constraints.
- Survivability can be incorporated with the NFF and its performance can be compared with the existing formulations.
- The Benders decomposition algorithms can be further improved by using metaheuristics or column generation to solve primal subproblem of the Benders reformulation of the problem to solve larger networks. Benders decomposition methods within branch and cut framework can be further improved by injecting heuristic upper bounds to the algorithm. Local search methods can be used to enhance the upper bounds found in the GG-BD_IR algorithm framework instead of the existing improvement heuristic.
- The GG-BD_IR algorithm solution performance can be improved by using parallel computing facilities. In the GG-BD_IR, the master problem is not solved to optimality and a number of feasible solutions are generated each iteration. These feasible solutions are used with primal subproblem to find feasibility cuts and upper bounds to the problem. The master problem can be solved at a server and found integer feasible solutions can be solved at clients in such a parallel processing setting. Using parallel computing increases the effectiveness of feasible space search as more feasible master solutions are generated. In addition, parallelizing subproblem solutions increases the efficiency of searching the feasible space.
- Some of the Benders decomposition algorithm variants can be combined in a single algorithm. For example, good feasible solution with the GG-BD_IR can be found and injected to the BC-BD as an initial solution. Performance of combining several Benders decomposition algorithm variants can be investigated.
- The polyhedral structure of the NFF can be studied and valid inequalities can be investigated.

REFERENCES

[1] "Cisco Visual Networking Index: Forecast and Methodology, 2012–2017," 2013. [Online].Available: http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-481360_ns827_Networking_Solutions_White_Paper.html . [Accessed: 30-Sep-2013].

[2] T. Lehman, X. Yang, N. Ghani, F. Gu, C. Guok, I. Monge, and B. Tierney, "Multilayer networks: An architecture framework," *IEEE Communications Magazine*, 2011.

[3] B. Lardeux, A. Knippel, and J. Geffard, "Efficient algorithms for solving the 2-layered network design problem," in *INOC 2003*, 2003, pp. 367–373.

[4] A. Knippel and B. Lardeux, "The multi-layered network design problem," *European Journal of Operational Research*, vol. 183, no. 1, pp. 87–99, Nov. 2007.

[5] B. Fortz and M. Poss, "An improved Benders decomposition applied to a multi-layer network design problem," *Operations Research Letters*, vol. 37, no. 5, pp. 359–364, Sep. 2009.

[6] A. M. C. A. Koster, S. Orlowski, C. Raack, G. Baier, and T. Engel, "Single-layer Cuts for Multi-layer Network Design Problems," in *Telecommunications Modeling, Policy, and Technology*, vol. 21, no. August, E. Raghavan, S. and Golden, Bruce and Wasil, Ed. Springer US, 2008, pp. 1–23.

[7] S. Orlowski, "Optimal Design of Survivable Multi-layer Telecommunication Networks," Ph.D. Thesis, Berlin Technical University, 2009.

[8] A. M. Geoffrion and G. W. Graves, "Multicommodity Distribution System Design by Benders Decomposition," vol. 20, no. 5, pp. 822–844, 1974.

[9] B. Gavish, "Topological Design of Centralized Computer Networks formulations and Algorithms," *Networks*, vol. 12, pp. 355–377, 1982.

[10] B. Gavish, "Topological Design of Telecommunication Networks - Local Access Design Methods," *Annals of Operations Research*, vol. 33, pp. 17–71, 1991.

[11] A. Balakrishnan, T. L. Magnanti, A. Shulman, and R. T. Wong, "Models for Planning Capacity Expansion in Local Access Telecommunication Networks," *Annals of Operations Research*, vol. 33, p. 239, 1991.

[12] T. Carpenter and H. Luss, "Telecommunications access network design," in *Network*, 1st ed., R. Mauricio and P. M. Pardalos, Eds. Springer, 2006.

[13] D. Alevras, M. Grotschel, and R. Wessaly, "Cost Efficient Network Synthesis from Leased Lines," *Annals of Operations Research*, vol. 76, pp. 1–20, 1998.

[14] R. S. Barr, B. M. S. Kingsley, and R. A. Patterson, “Telecommunications network grooming,” in *Handbook of Optimization in Telecommunications*, M. G. C. Resende and P. M. Pardalos, Eds. Springer, 2006.

[15] K. Zhu and B. Mukherjee, “A Review of Traffic Grooming in WDM Optical Networks Architectures and Challenges,” *Optical Networks Magazine*, vol. 4, no. 2, pp. 55–64, 2003.

[16] H. Zang, J. P. Jue, and B. Mukherjee, “A Review of Routing and Wavelength Assignment Approaches for Wavelength- Routed Optical WDM Networks,” *Optical Networks Magazine*, no. January, pp. 47–60, 2000.

[17] J. S. Choi, N. Golmie, F. Lapeyrere, F. Mouveaux, and D. Su, “A Functional Classification of Routing and Wavelength Assignement Schemes in DWDM networks Static Case,” in *VII Int. Conf. on Optical Communication and Networks*, 2000, pp. 1–8.

[18] R. Dutta and G. N. Rouskas, “A Survey of Virtual Topology Design Algorithms for Wavelength Routed Optical Networks,” 1999.

[19] J. G. Klincewicz, “Hub location in backbone/tributary network design: a review,” *Location Science*, vol. 6, pp. 307–335, 1998.

[20] D. Medhi, “Network restoration,” in *Handbook of Optimization in Telecommunications*, M. G. C. Resende and P. M. Pardalos, Eds. Springer, 2006.

[21] M. Minoux, “Network Synthesis and Optimum Network Design Problems: Models, Solution Methods and Applications,” *Network*, vol. 19, pp. 313–360, 1989.

[22] H. P. L. Luna, “Network planning problems in telecommunications,” in *Handbook of Optimization in Telecommunications*, M. G. C. Resende and P. M. Pardalos, Eds. Springer, 2006.

[23] A. Forsgren and M. Prytz, “Telecommunications network design,” in *Handbook of Optimization in Telecommunications*, M. G. C. Resende and P. M. Pardalos, Eds. Springer, 2006.

[24] P. Cortes and J. Munuzuri, “A brief review of the state of the art in Operational Research in Telecommunications,” *OR Insight*, vol. 21, no. 2, pp. 25–38, 2008.

[25] S. Van Hoesel, “Optimization in Telecommunication Networks,” *Statistica Neerlandica*, vol. 58, no. 2, pp. 180–205, 2004.

[26] B. Gavish, C. Li, and D. Simchi-Levi, “Analysis of heuristics for the design of tree networks,” *Annals of Operations Research*, vol. 36, pp. 77–86, 1992.

[27] E. Gourdin, M. Labbe, and H. Yaman, “Telecommunication and Location,” in *Facility Location*, Z. Drezner and H. W. Hamacher, Eds. Springer, 2002.

[28] X. Chen and B. Chen, “Cost-Effective Designs of Fault-Tolerant Access Networks in Communication Systems,” *Networks*, vol. 53, no. 4, pp. 382–391, 2009.

[29] S. Soni, S. Narasimhan, and L. J. Leblanc, “Telecommunication Access Network Design With Reliability Constraints,” *IEEE Transactions on Reliability*, vol. 53, no. 4, pp. 532–541, 2004.

[30] S. Chamberland, “Global Access Network Evolution,” *IEEE/ACM Transactions on Networking*, vol. 18, no. 1, pp. 136–149, 2010.

[31] L. F. Frantzeskakis and H. Luss, “The Network Redesign Problem for Access Telecommunications Networks,” *Naval Research Logistics*, vol. 46, pp. 488–506, 1999.

[32] C. D. Randazzo and H. P. L. Luna, “A Comparison of Optimal Methods for Local Access Uncapacitated Network Design,” *Annals of Operations Research*, vol. 106, pp. 263–286, 2001.

[33] F. S. Salman, R. Ravi, and J. N. Hooker, “Solving the Capacitated Local Access Network Design Problem,” *INFORMS Journal on Computing*, vol. 20, no. 2, pp. 243–254, Sep. 2007.

[34] I. Ljubic, P. Putz, and J.-J. Salazar-González, “A Heuristic Algorithm for a Prize-Collecting Local Access Network Design Problem,” in *5th International Network Optimization Conference (INOC)*, 2011, pp. 139–144.

[35] M. C. Goldstein, “Design of Long Distance Telecommunication Networks - The Telpak problem,” *IEEE Transactions On Circuit Theory*, vol. 20, no. 3, pp. 186–192, 1973.

[36] M. C. Goldstein and B. Rothfarb, “The one terminal TELPAK problem,” *Operations Research*, vol. 19, no. 1, pp. 156–169, 1971.

[37] G. Zhou, Z. Cao, J. Cao, and Z. Meng, “A Centralized Network Design Problem with Genetic Algorithm Approach,” *Computer Intelligence and Security*, vol. 4456, pp. 123–132, 2007.

[38] L. R. Esau and K. C. Williams, “On teleprocessing system design. Part 2,” *IBM System Journal*, vol. 5, no. 3, pp. 142–147, 1966.

[39] B. Gavish, “Augmented Lagrangean Based Algorithms for Centralized Network Design,” *IEEE Transactions on Communications*, vol. 33, no. 12, pp. 1247–1257, 1985.

[40] B. Gavish and K. Altinkemer, “A parallel savings heuristic for the topological design of local access tree networks,” in *IEEE INFOCOM*, 1986, pp. 130–139.

[41] F. Gzaraa and J. Goffin, “Exact Solution of the Centralized Network Design Problem on Directed Graphs,” *Networks*, vol. 45, pp. 181–192, 2005.

[42] A. Kershenbaum, R. Boorstyn, and R. Oppenheim, “Second Order Greedy Algorithms for Centralized Teleprocessing Network Design,” *IEEE Transactions on Communications*, vol. C–28, no. 10, pp. 1835–1838, 1980.

[43] M. Trampont, C. Destré, and A. Faye, “Solving a continuous local access network design problem with a stabilized central column generation approach,” *European Journal of Operational Research*, vol. 214, no. 3, pp. 546–558, 2011.

[44] K. Altinkemer, “Parallel Savings Heuristics for Designing Multi-center Tree Networks,” *IEEE*, pp. 762–769, 1989.

[45] I. Ljubic, P. Putz, and J.-J. Salazar-González, “Exact Approaches to the Single-Source Network Loading Problem,” *Networks*, vol. 59, no. 1, pp. 89–106, 2012.

[46] S. Chamberland and B. Sanso, “Heuristics for the Topological Design Problem of Two-Level Multitechnology Telecommunication Networks with Modular Switches,” Canada, 1997.

[47] M. Henningsson, K. Holmberg, and D. Yuan, “Ring network design,” in *Handbook of Optimization in Telecommunications*, M. G. C. Resende and P. M. Pardalos, Eds. Springer, 2006.

[48] M. Gerla and L. Kleinrock, “On the Topological Design of Distributed Computer Networks,” *IEEE Transactions on Communications*, vol. 25, no. 1, pp. 48–60, Jan. 1977.

[49] A. Kershenbaum, P. Kermani, and G. A. Grover, “MENTOR: an algorithm for mesh network topological optimization and routing,” *IEEE Transactions on Communications*, vol. 39, no. 4, pp. 503–513, Apr. 1991.

[50] K. Altinkemer and Z. Yu, “Topological Design of Wide Area Communication Networks,” *Annals of Operations Research*, vol. 36, pp. 365–382, 1992.

[51] A. Amiri and H. Pirkul, “Routing and capacity assignment in backbone communication networks,” *European Journal of Operational Research*, vol. 117, pp. 15–29, 1997.

[52] A. Amiri and H. Pirkul, “Routing and capacity assignment in backbone communication networks under time varying traffic conditions,” *European Journal of Operational Research*, vol. 117, no. 1, pp. 15–29, Aug. 1999.

[53] A. Bley and T. Koch, “Integer Programming Approaches to Access and Backbone IP Network Planning,” *Modeling, Simulation and Optimization of Complex Systems*, pp. 87–110, 2008.

[54] X. Ma, S. Kim, and K. Harfoush, “Towards realistic physical topology models for Internet backbone networks,” *2009 6th International Symposium on High Capacity Optical Networks and Enabling Technologies (HONET)*, pp. 36–42, Dec. 2009.

[55] P. Datta, M. Sridharan, and A. K. Somani, “A Simulated Annealing Approach for Topology Planning and Evolution of Mesh-Restorable Optical Networks,” in *Design of Reliable Communicatin networksetworks*, 2003, pp. 1–18.

[56] D. Din, “Heuristic and Simulated Annealing Algorithms for Wireless ATM Backbone Network Design Problem,” *Journal of InformationScience and Engineering*, vol. 24, pp. 483–501, 2008.

[57] S. Pierre and A. Elgibaoui, “A tabu-search approach for designing computer-network topologies with unreliable components.,” *IEEE Transactions on Re*, vol. 46, no. 3, pp. 350–359, 1997.

[58] A. Konak and A. E. Smith, “A Hybrid Genetic Algorithm Approach for Backbone Design of Communication Networks,” in *Proceedings of the 1999 Congress on Evolutionary Computation CEC*, 1999.

[59] B. Al-bassam, A. Alheraish, and S. H. Bakry, “A tutorial on using genetic algorithms for the design of network topology,” *Networks*, no. September 2005, pp. 253–262, 2006.

[60] B. Dengiz, F. Altiparmak, and A. E. Smith, “Local Search Genetic Algorithm for Optimal Design of Reliable Networks,” *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 3, pp. 179–188, 1997.

[61] K. Watcharasitthiwat and P. Wardkein, “Reliability optimization of topology communication network design using an improved ant colony optimization,” *Computers & Electrical Engineering*, vol. 35, no. 5, pp. 730–747, Sep. 2009.

[62] B. K. Chen and F. Tobagi, “Network Topology Design to Optimize Link and Switching Costs,” *2007 IEEE International Conference on Communications*, pp. 2450–2456, Jun. 2007.

[63] T. L. Magnanti, P. Mirchandani, and R. Vachani, “Modeling and Solving the Two-Facility Capacitated Network Loading Problem,” *Operations Research*, vol. 43, no. 1, pp. 142–157, 1995.

[64] A. Altin, “Robust Network Design Under Polyhedral Traffic Uncertainty,” Ph.D. Thesis, Bilkent University, 2007.

[65] K. Ho and K. W. Cheung, “Generalized Survivable Network,” *IEEE/ACM Transactions on Networking*, vol. 15, no. 4, pp. 750–760, Aug. 2007.

[66] K. Ho, M. Zhou, and K.-W. Cheung, “A New Approach for Designing the Next Generation Survivable Backbone Network,” in *12th International Telecommunications Network Strategy and Planning Symposium*, 2006, pp. 1–6.

[67] C. Hsu, J. Wu, S. Wang, and C. Hong, “Survivable and delay-guaranteed backbone wireless mesh network design,” *Journal of Parallel and Distributed Computing*, vol. 68, no. 3, pp. 306–320, Mar. 2008.

[68] M. Grotschel, C. L. Monma, and M. Stoer, “Computational Results with a Cutting Plane Algorithm for Designing Communication Networks with Low-Connectivity Constraints,” *Operations Research*, vol. 40, no. 2, pp. 309–330, 1992.

[69] C. S. K. Vadrevu and M. Tornatore, “Survivable IP topology design with re-use of backup wavelength capacity in optical backbone networks,” *Optical Switching and Networking*, vol. 7, no. 4, pp. 196–205, Dec. 2010.

[70] H. Liu and F. Tobagi, “Physical topology design for all-optical networks,” *Optical Switching and Networking*, vol. 5, no. 4, pp. 219–231, Oct. 2008.

[71] T. Thomadsen, “Hierarchical Network Design,” Ph.D Thesis, The Technical University of Denmark, 2005.

[72] T. Thomadsen and T. Stidsen, “The generalized fixed-charge network design problem,” *Computers & Operations Research*, vol. 34, pp. 997 – 1007, 2007.

[73] E. Rosenberg, “Hierarchical Topological Network Design,” in *Handbook of Optimization in Telecommunications*, vol. 13, no. 6, M. G. C. Resende and P. M. Pardalos, Eds. Springer, 2006, pp. 1402–1409.

[74] S. Chopra and C. Tsai, “A branch-and-cut approach for minimum cost multi-level network design,” *Discrete Mathematics*, vol. 242, pp. 65–92, 2002.

[75] S. Gollowitzer, L. Gouveia, and I. Ljubic, “The Two Level Network Design Problem with Secondary Hop Constraints,” in *5th International Network Optimization Conference (INOC)*, 2011, pp. 71–76.

[76] S. Gollowitzer, L. Gouveia, and I. Ljubic, “Enhanced formulations and branch-and-cut for the two level network design problem with transition facilities,” *European Journal Of Operational Research*,vol. 225, pp. 211–222, 2013.

[77] S. Gollowitzer, L. Gouveia, and I. Ljubic, “A Node Splitting Technique for Two Level Network Design Problems with Transition Nodes,” in *5th International Network Optimization Conference (INOC)*, 2011, pp. 57–70.

[78] I. Godor and M. Gabor, “Cost-optimal topology planning of hierarchical access networks,” *Computers & Operations Research*, vol. 32, pp. 59 – 86, 2005.

[79] V. Grout, S. Cunningham, and R. Picking, “Practical Large-Scale Network Design with Variable Costs for Links and Switches,” in *Handbook of Optimization in Telecommunications*, vol. 7, no. 7, M. G. C. Resende and P. M. Pardalos, Eds. Springer, 2006, pp. 113–125.

[80] T. V. Do, T. T. Nguyen, H. T. Tran, G. Kalvach, and B. Varga, “Topology optimization of an overlay ATM network in an SDH infrastructure,” *Computer Networks*, vol. 34, no. 1, pp. 199–210, Jul. 2000.

[81] A. Proestaki and M. C. Sinclair, “Design and dimensioning of dual-homing hierarchical multi-ring networks,” *IEEE Proceedings - Communications*, vol. 147, no. 2, pp. 96–104, 2000.

[82] M. Labbe, G. Laporte, I. R. Martin, and J. J. S. Gonzalez, “The Ring Star Problem: Polyhedral Analysis and Exact Algorithm,” *Networks*, vol. 43, no. 3, pp. 177– 189, 2004.

[83] J. Petrek and V. Siedt, “A Large Hierarchical Network Star-Star Topology Design Algorithm,” *European Transactions on Telecommunication*, vol. 12, no. 6, pp. 1–12, 2001.

[84] T. Koch and R. Wessaly, “Hierarchical Infrastructure Planning in Networks,” 2004.

[85] G. R. Mateus, F. R. B. Cruz, and H. P. L. Luna, "An algorithm for hierarchical network design," *Location Science*, vol. 2, no. 3, pp. 149–164, 1994.

[86] K. Park, K. Lee, S. Park, and H. Lee, "Telecommunication Node Clustering with Node Compatibility and Network Survivability Requirements," *Management Science*, vol. 46, no. 3, pp. 363–374, 2000.

[87] A. Balakrishnan, T. L. Magnanti, and P. Mirchandani, "A Dual-Based Algorithm for Multi-Level Network Design," *Management*, vol. 40, no. 5, pp. 567–581, 1994.

[88] A. Balakrishnan, T. L. Magnanti, and P. Mirchandani, "Modeling and Heuristic Worst Case Performance Analysis of the Two Level Network Design Problem," *Management Science*, vol. 40, no. 7, pp. 846–867, 1994.

[89] L. Gouveia and J. Telhado, "An Augmented Arborescence Formulation for the Two-Level Network Design Problem," *Annals of Operations Research*, vol. 106, pp. 47–61, 2001.

[90] T. Thomadsen and J. Larsen, "A hub location problem with fully interconnected backbone and access networks," *Computers & Operations Research*, vol. 34, pp. 2520 – 2531, 2007.

[91] H. Pirkul and V. Nagarajan, "Locating concentrators in centralized computer networks," *Annals of Operations Research*, vol. 36, pp. 247–262, 1992.

[92] P. Mirchandani, "The Multi Tier Tree Problem," *INFORMS Journal on Computing*, vol. 8, no. 3, pp. 202–218, 1996.

[93] F. R. B. Cruz, G. R. Mateus, and J. M. Smith, "A Branch-and-Bound Algorithm to Solve a Multi-level Network Optimization Problem," *Journal of Mathematical Modelling and Algorithms*, pp. 37–56, 2003.

[94] S. Chang and B. Gavish, "Lower Bounding Procedures for Multiperiod Telecommunications Network Expansion Problems," *Operations Research*, vol. 43, no. 1, pp. 43–57, 1995.

[95] M. Corte-real and L. Gouveia, "Network flow models for the local access network expansion problem," *Computers and Operations Research*, vol. 34, pp. 1141–1157, 2007.

[96] J. Freindenfelds, *Capacity Expansion*. New York, USA: ELsevier North Holland, 1981.

[97] H. Luss, "Operations Research and Capacity Expansion Problems A Survey," *Operations Research*, vol. 30, no. 5, pp. 907–947, 1982.

[98] A. Shulman and R. Vachani, "A Decomposition Algorithm for Capacity Expansion of Local Access Networks," *IEEE Transactions on Communications*, vol. 41, no. 7, pp. 1063–1073, 1993.

[99] A. Shulman and R. Vachani, "An Algorithm for Capacity Expansion of Local Access Networks," in *IEEE INFOCOM*, 1990, pp. 221–229.

[100] I. Saniee, "An Efficient Algorithm for the Multiperiod Capacity Expansion of one Location in Telecommunications," *Operations Research*, vol. 43, no. 1, pp. 187–190, 1995.

[101] A. Balakrishnan, T. L. Magnanti, and R. T. Wong, "A Decomposition Algorithm for Expanding Local Access Telecommunications Networks," *Operations Research*, vol. 43, pp. 58–76, 1993.

[102] M. Corte-Real and L. Gouveia, "A node rooted flow-based model for the local access network expansion problem," *European Journal of Operational Research*, vol. 204, no. 1, pp. 20–34, Jul. 2010.

[103] G. Cho and D. X. Shaw, "Limited Column Generation for Local Access Telecommunication Network Expansion and Its Extensions – Formulations, Algorithms and Implementation," Technique Report, Purdue University, 1993.

[104] G. Cho and D. X. Shaw, "Models and Implementation Techniques for Local Access Telecommunication Network Design," in *Industrial Applications of Combinatorial Optimization*, G. Yu, Ed. 1998.

[105] O. E. Flippo, A. W. J. Kolen, A. M. C. A. Koster, and R. L. M. J. Van De Leensel, "A dynamic programming algorithm for the local access telecommunication network expansion problem," *European Journal of Operational Research*, vol. 127, pp. 189–202, 2000.

[106] M. Gendreau, J.-Y. Potvin, A. Smires, and P. Soriano, "Multi-period capacity expansion for a local access telecommunications network," *European Journal Of Operational Research*, vol. 172, pp. 1051–1066, 2006.

[107] R. Kouassi, M. Gendreau, J.-Y. P. Patrick, and P. Soriano, "Heuristics for multi-period capacity expansion in local telecommunications networks," *Journal of Heuristics*, vol. 15, no. 4, pp. 381–402, 2007.

[108] R. L. M. J. van de Leensel, O. E. Flippo, and A. M. C. A. Koster, "A Dynamic Programming Algorithm for the ATM Network Installation Problem on a Tree," *European Journal of Operations Research*, vol. 127, pp. 189–200, 2000.

[109] N. Zadeh, "On Building Minimum Cost Communication Networks Over Time," *Networks*, vol. 4, pp. 19–34, 1974.

[110] N. Christofides and P. Brooker, "Optimal expansion of an existing network," *Mathematical Programming*, vol. 6, no. June 1973, pp. 197–211, 1974.

[111] R. Sivaraman, "Capacity Expansion in Contemporary Telecommunication Networks," Ph.D. Thesis, MIT, 2007.

[112] S. Chamberland and B. Sanso, "Topological Expansion of Multiple-Ring Metropolitan Area Networks," *Networks*, vol. 36, no. 4, pp. 210–224, 2000.

[113] D. Saha and B. Mukherjee, "Managing the Topological Expansion," *International Journal of Network Management*, no. August, pp. 206–211, 1996.

[114] R. K. Ahuja, J. L. Batra, S. K. Gupta, and A. P. Punnen, “Optimal Expansion of Capacitated Transshipment Networks,” *European Journal Of Operational Research*, vol. 89, pp. 176–184, 1996.

[115] L. J. Leblanc and S. Narasimhan, “Topological expansion of metropolitan area network,” *Computer Networks and ISDN Systems*, vol. 26, pp. 1235–1248, 1994.

[116] D. Bienstock and O. Gunluk, “Capacitated Network Design - Polyhedral Structure and Computation,” *Informs journal of computing*, vol. 8, pp. 243–259, 1996.

[117] S. Orlowski and R. Wessaly, “An Integer Programming Model for Multi-Layer Network Design,” Technical Report, Zuse Institute Berlin, 2004.

[118] M. Pioro and D. Medhi, *Routing, FLow and Capacity Design in Communication and Computer Networks*. Morgan Kaufmann Publishers, 2004.

[119] M. Plante and B. Sanso, “A Typology for Multi-Technology, Multi-Service,” *Telecommunication Systems*, vol. 1, pp. 39–73, 2002.

[120] S. Borne, E. Gourdin, B. Liau, and A. R. Mahjoub, “Design of survivable IP-over-optical networks,” *Annals of Operations Research*, pp. 41–73, 2006.

[121] I. Katib and D. Medhi, “Network protection design models , a heuristic , and a study for concurrent single-link per layer failures in three-layer networks,” *Computer Communications*, vol. 36, no. 6, pp. 678–688, 2013.

[122] S. Mattia, “Solving Survivable Two-Layer Network Design Problems by Metric Inequalities,” *Informatica*, 2010.

[123] G. Dahl, A. Martin, and M. Stoer, “Routing Through Virtual Paths in Layered Telecommunication Networks,” *Operations Research*, vol. 47, no. 5, pp. 693–702, 1999.

[124] V. Gabrel, A. Knippel, and M. Minoux, “Exact solution of multicommodity network optimization problems with general step cost functions,” *Operations Research Letters*, vol. 25, pp. 15–23, 1999.

[125] E. Kubilinskas, “Design of Multi-layer Telecommunication Networks”, Ph.D Thesis, Lund University, 2008.

[126] S. Orlowski, A. M. C. A. Koster, C. Raack, and R. Wessaly, “Two-layer Network Design by Branch-and-Cut featuring MIP-based Heuristics,”Technical report, ZIB-Report 06-47, Zuse Institute Berlin, 2006.

[127] S. Raghavan and D. Stanojevic, “Branch and Price for WDM Optical Networks with No Bifurcation of Flow,” *INFORMS Journal on Computing*, vol. 23, no. 1, pp. 56–74, 2011.

[128] D. Stanojevic, “Optimization Of Contemporary Telecommunications Networks : Generalized Spanning Trees And Wdm Optical Networks,”, Ph.D. Thesis, University of Maryland, College Park, 2005.

[129] K. Zhu and B. Mukherjee, "Traffic grooming in an optical WDM mesh network," *IEEE Journal on Selected Areas in Communications*, vol. 20, no. 1, pp. 122–133, 2002.

[130] H. Zhu, H. Zang, K. Zhu, and B. Mukherjee, "A Novel Generic Graph Model for Traffic Grooming in Heterogeneous WDM Mesh Networks," *IEEE/ACM TRANSACTIONS ON NETWORKING*, vol. 11, no. 2, pp. 285–299, 2003.

[131] B. Mukherjee, "Cost-effective WDM backbone network design with OXCs of different bandwidth granularities," *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 9, pp. 1452–1466, Nov. 2003.

[132] H. Zhu, "Optical WDM Networks: Traffic Grooming in Mesh Networks and Metro Networks using ROADMs," Ph.D. Thesis University of Califormia Davis, 2005.

[133] C. Papagianni, C. Pappas, N. Lefkaditis, and I. S. Venieris, "Particle Swarm Optimization for the Multi Level Capacitated Minimum Spanning Tree," in *Proceedings of the International Multiconference on Computer Science and Technology*, 2009, pp. 765 – 770.

[134] A. Martin, M. C. De Souza, M. J. F. Souza, and T. A. M. Toffolo, "GRASP with hybrid heuristic-subproblem optimization for the multi-level capacitated minimum spanning tree problem," *Journal of Heuristics*, vol. 15, pp. 133–151, 2009.

[135] S. Voß, "Steiner tree problems in telecommunications," in *Handbook of Optimization in Telecommunications*, no. 1835, M. G. C. Resende and P. M. Pardalos, Eds. Springer, 2006.

[136] F. R. B. Cruz, J. M. Smith, and G. R. Mateus, "Algorithms for a Multi-level Network Optimization Problem," *European Journal Of Operational Research*, 1998.

[137] A. Balakrishnan, T. L. Magnanti, A. Shulman, and R. T. Wong, "Models for Planning the Evolution of the Local Telecommunication Networks," *Annals of Operations Research*, vol. 33, pp. 239–284, 1991.

[138] H. Yaman, "Concentrator location in telecommunications," *4OR*, vol. 2, pp. 175–177, 2004.

[139] B. Gendron, T. G. Crainic, and A. Frangioni, "Multicommodity capacitated network design," in *Telecommunications network planning*, B. Sanso and P. Soriano, Eds. Kluwer Academic Publishers, 1998, pp. 1–19.

[140] L. Gouveia and P. Martins, "The Capacitated Minimal Spanning Tree Problem An experiment with a hop-indexed model," *Annals of Operations Research*, vol. 86, pp. 271–294, 1999.

[141] H. Kerivin and A. R. Mahjoub, "Design of Survivable Networks: A Survey," *Networks*, vol. 46, no. 1, pp. 1–21, 2005.

[142] M. Minoux, "Discrete Cost Multicommodity Network Optimization Problems and Exact Solution Methods," *Annals of Operations Research*, vol. 106, pp. 19–46, 2001.

[143] M. Minoux, "Multicommodity network flow models and algorithms in telecommunications," in *Handbook of Optimization in Telecommunications*, M. G. C. Resende and P. M. Pardalos, Eds. Springer, 2006.

[144] A. Ouorou, P. Mahey, and J.-P. Vial, "A Survey of Algorithms for Convex Multicommodity Flow Problems," *Management Science*, vol. 46, no. 1, pp. 126–147, 2000.

[145] D. Alevras, M. Grötschel, and R. Wessäly, "Capacity and survivability models for telecommunication networks,"Z=Technical Report, Zuse Institute Berlin, 1997.

[146] L. Graham and P. Hell, "On the History of the Minimum Spanning Tree Problem," *Annals of the History of Computing*, vol. 7, no. 1, pp. 43–57, 1985.

[147] M. Grotschel, C. L. Monma, and M. Stoer, "Design of survivable networks.pdf," in *Handbooks in OR&MS Vol, 7*, 1995, pp. 617–672.

[148] A. Amberg, W. Domschke, and S. Voß, "Capacitated Minimum Spanning Trees: Algorithms Using Intelligent Search," *Combinatorial Optimization: Theory and Practice*, vol. 1, no. 1, 1996.

[149] C. Bazlamacci and K. S. Hindi, "Minimum-weight spanning tree algorithms A survey and empirical study Cu," *Computers and Operations Research*, vol. 28, pp. 767–785, 2001.

[150] L. A. Hall and T. L. Magnanti, "A Polyhedral Intersection Theorem for Capacitated Spanning Trees," *Mathematics of Operations Research*, vol. 17, no. December, pp. 398–410, 1992.

[151] C. Papadimitriou, "The complexity of the capacitated tree problem," *Networks*, vol. 3, pp. 217–230, 1978.

[152] B. Gavish, "Formulations and Algorithms for the Capacitated Minimal Directed Tree Problem," *Journal of the Association for Computing Machinery*, vol. 30, no. 1, pp. 118–132, 1983.

[153] E. Uchoa, R. Fukasawa, J. Lysgaard, A. Pessoa, M. Poggi, and D. Andrade, "Robust branch-cut-and-price for the Capacitated Minimum Spanning Tree problem over a large extended formulation," *Mathematical Programming*, vol. 112, pp. 443–472, 2008.

[154] L. Gouveia, "A 2n Constraint Formulation for the Capacitated Minimal Spanning Tree Problem," *Operations Research*, vol. 43, no. 1, pp. 130–141, 1995.

[155] P. M. Camerini, G. Galbiati, and F. Maffoli, "Complexity of spanning tree poblems: Part 1," *European Journal Of Operational Research*, vol. 5, pp. 346–352, 1980.

[156] P. M. Camerini, G. Galbiati, and F. Maffoli, "On the complexity of finding multi-constrained spanning trees," *Discrete Applied Mathematics*, vol. 5, pp. 39–50, 1983.

[157] L. Gouveia and P. Martins, "The capacitated minimum spanning tree problem : revisiting hop-indexed formulations," *Computers & Operations Research*, vol. 32, pp. 2435 – 2452, 2005.

[158] L. Gouveia and P. Martins, "A hierarchy of hop-indexed models for the capacitated minimum spanning tree problem," *Networks*, vol. 35, pp. 1–16, 2000.

[159] L. A. Hall, "Experience with a cutting plane algorithm for the capacitated spanning tree problem," *Informs journal of computing*, vol. 8, pp. 219–234, 1996.

[160] B. Hu, M. Leitner, and G. Raidl, "Combining Variable Neighborhood Search with Integer Linear Programming for the Generalized Minimum Spanning Tree Problem," *Journal of Heuristics*, vol. 43, no. October, 2008.

[161] J. Han, Z. Sun, J. Huai, and X. Li, "An Efficient Node Partitioning Algorithm for the Capacitated Minimum Spanning Tree Problem," in *6th IEEE/ACIS International Conference on Computer and Information Science*, 2007, no. Icis, pp. 2–7.

[162] P. Martins, "Enhanced second order algorithm applied to the capacitated minimum spanning tree problem," *Computers & Operations Research*, vol. 34, pp. 2495 – 2519, 2007.

[163] M. Karnaugh, "A New Class of Algorithms for Multipoint Network Optimization," *IEEE Transactions on Communications*, vol. COM-24, no. 5, pp. 500–505, 1976.

[164] T. Oncan, "Design of capacitated minimum spanning tree with uncertain cost and demand parameters," *Information Sciences*, vol. 177, pp. 4354–4367, 2007.

[165] M. Reimann and M. Laumanns, "Savings based ant colony optimization for the capacitated minimum spanning tree problem," *Computers & Operations Research*, vol. 33, no. 6, pp. 1794–1822, Jun. 2006.

[166] R. Jothi and B. Raghavachari, "Approximation Algorithms for the Capacitated Minimum Spanning Tree Problem and its Variants in Network Design," *ACMTransactions on Algoritms*, vol. 1, no. 2, pp. 265–282, 2005.

[167] M. C. De Souza, C. Duhamel, and C. C. Ribeiro, "A GRASP heuristic for the capacitated minimum spanning tree problem using a memory-based local search strategy," *Applied Optimization*, no. 2002, pp. 1–24, 2002.

[168] R. K. Ahuja, J. B. Orlin, and D. Sharma, "Multi-exchange neighborhood structures for the capacitated minimum spanning tree problem," *Math. Program., Ser.A*, vol. 97, pp. 71–97, 2001.

[169] R. Patterson, H. Pirkul, and E. Rolland, "A Memory Adaptive Reasoning Technique for Solving the Capacitated Minimum Spanning Tree Problem," *Journal of Heuristics*, vol. 180, pp. 159–180, 1999.

[170] Y. Sharaiha, M. Gendreau, G. Laporte, and I. Osman, "A tabu search algorithm for the capacitated shortest spanning tree problem," *Networks*, vol. 29, pp. 161–171, 1997.

[171] K. Altinkemer and B. Gavish, "Heuristics with Constant Error Guarantees for the Design of Tree Networks," *Management Science*, vol. 34, no. 3, pp. 331– 341, 1988.

[172] R. L. Sharma, "Design of an Economical Multidrop Network Topology with Capacity Constraints," *IEEE Transactions on Communications*, vol. 31, no. 4, pp. 590–591, 1983.

[173] A. Kershenbaum and W. Chou, "A unified algorithm for designing multidrop teleprocessing networks," *IEEE Transactions on Communications*, vol. Com-22, 1972.

[174] I. Gamvros, S. Raghavan, and B. Golden, "An Evolutionary Approach to the Multi-Level Capacitated Minimum Spanning Tree Problem," in *Telecommunications network design and management*, vol. 10, H. G. Anandalingam and S. Raghavan, Eds. Berlin: Kluwer Academic Publishers, 2003, pp. 99–125.

[175] I. Gamvros, B. Golden, and S. Raghavan, "The Multilevel Capacitated Minimum Spanning Tree Problem," *INFORMS Journal on Computing*, vol. 18, no. 3, pp. 348–365, Jan. 2006.

[176] M. Goemans and Y. S. Myung, "A Catalog of Steiner Tree Formulations," *Networks*, vol. 23, pp. 19–28, 1993.

[177] T. Polzin and S. V. Daneshmand, "A comparison of Steiner tree relaxations," *Discrete Applied Mathematics*, vol. 112, pp. 241–261, 2001.

[178] T. L. Magnanti and R. T. Wong, "Network design and transportation planning models and algorithms," *Transportation Science*, 1984.

[179] Y. P. Aneja, "An integer linear programming approach to the Steiner problem in graphs," *Networks*, vol. 10, pp. 167–178, 1980.

[180] A. Lucena and J. E. Beasley, "A Branch and Cut Algorithm for the Steiner Problem in Graphs," *Networks*, vol. 31, pp. 39–59, 1998.

[181] T. Polzin and S. V. Daneshmand, "Approaches to the Steiner Problem in Networks," *Algorithmics*, vol. 6, pp. 81–103, 2009.

[182] C. W. Duin and A. Volgenant, "Reduction tests for the steiner problem in graphs," *Networks*, vol. 19, pp. 549–567, 1989.

[183] C. W. Duin and A. Volgenant, "An edge elimination test for the steiner problem in graphs," *Operations Research Letters*, vol. 8, no. 2, pp. 79–83, 1989.

[184] A. Balakrishnan and N. R. Patel, "Problem Reduction Methods and a Tree Generation Algorithm for the Steiner Network Problem," *Networks*, vol. 17, no. 1, pp. 65–85, 1987.

[185] E. Uchoa, M. P. De Aragao, and C. C. Ribeiro, "Preprocessing Steiner Problems from VLSI Layout," *Networks*, vol. 40, no. 1, pp. 30-50, 2002.

[186] C. W. Duin, "Steiner's problem in graphs," Universuty of Amsterdam, 1993.

[187] T. Polzin and S. V. Daneshmand, "Improved algorithms for the Steiner problem in networks," *Discrete Applied Mathematics*, vol. 112, no. February 1999, pp. 263–300, 2001.

[188] T. Polzin and S. V. Daneshmand, "Extending Reduction Techniques for the Steiner Tree Problem," in *Algorithms*, R. Mohring and R. Raman, Eds. Berlin: Springer, 2002, pp. 795–807.

[189] T. Polzin, "Algorithms for the Steiner Problem in Networks," Ph.D. Thesis, Universitat I at des Saarlandes, 2003.

[190] S. Chopra, E. R. Gorres, and M. R. Rao, "Solving the Steiner tree problem on a graph using branch and cut," *ORSA Journal on Computing*, vol. 4, pp. 320–335, 1992.

[191] A. Lucena, "Steiner problem in graphs Lagrangian relaxation and cutting planes," *Bulletin of the Committee on Algorithms*, vol. 21, pp. 2–7, 1992.

[192] T. Polzin and S. V. Daneshmand, "Steiner trees and minimum spanning trees in hypergraphs," *Operations Research Letters*, vol. 31, pp. 12–20, 2003.

[193] S. V. Daneshmand, "Algorithmic Approaches to the Steiner Problem in Networks,"Ph.D.Thesis, Universitat Mannheim, 2003.

[194] B. Fuchs, W. Kern, P. Rossmanith, S. Richter, and X. Wang, "Dynamic Programming for Minimum Steiner Trees," *Theory of Computing Systems*, vol. 6, pp. 1–10, 2006.

[195] S. Voß, "Modern heuristic search methods for the Steiner tree problem in graphs," in *Advances in Steiner Trees*, Z. Du, J. M. Smith, and J. H. Rubinstein, Eds. Boston: Kluwer, 2000, pp. 283–323.

[196] C. W. Duin and S. Voß, "The Pilot Method : A Strategy for Heuristic Repetition with Application to the Steiner Problem in Graphs," *Operations Research*, vol. 34, pp. 181–191, 1999.

[197] C. W. Duin and S. Voß, "Steiner tree heuristics - a survey," in *Operations Research Proceedings*, and H. C. T. H. Dyckhoff, U. Derigs, M. Salomon, Ed. Berlin: Springer, 1994, pp. 485–496.

[198] S. L. Martins, M. G. C. Resende, C. C. Ribeiro, and P. M. Pardalos, "A Parallel GRASP for the Steşner Tree Problem in Graphs," *Journal of Global Optimization*, vol. 17, pp. 267–283, 2000.

[199] C. C. Ribeiro, E. Uchoa, and R. F. Werneck, "A Hybrid GRASP with Perturbations for the Steiner Problem in Graphs," *INFORMS Journal on Computing*, vol. 14, no. 3, pp. 228–246, Jul. 2002.

[200] J. D. Kramer, “Min-Cost Multicommodity Network Flows: A Linear Case for the Convergence and Reoptimization of Multiple Single Commodity Network Flows.,” M.Sc. Thesis, North Carolina State University, 2009.

[201] M. D. Grigoriadis and W. W. White, “A Partitioning Algorithm for the Multicommodity Network FLow Problem,” *Mathematical Programming*, vol. 3, no. 2, pp. 157–177, 1972.

[202] J. L. Kennington and R. V Helgason, “Minimum cost network flow algorithms,” in *Handbook of Optimization in Telecommunications*, no. 1962, M. G. C. Resende and P. M. Pardalos, Eds. Springer, 2006.

[203] J. E. Mitchell, K. Farwell, and D. Ramsden, “Interior point methods for large-scale linear programming,” in *Handbook of Optimization in Telecommunications*, M. G. C. Resende and P. M. Pardalos, Eds. Springer, 2006.

[204] A. Frangioni and G. Gallo, “A Bundle Type Dual-Ascent Approach to Linear Multicommodity Min-Cost Flow Problems,” *INFORMS Journal on Computing*, vol. 11, no. 4, pp. 370–393, Jan. 1999.

[205] J. Castro, “A specialized interior-point algorithm for multicommodity network flows,” *SIAM Journal of Optimization*, vol. 10, no. 3, pp. 852–877, 2000.

[206] J. Castro, “A Parallel Implementation of an Interior-Point Algorithm for Multicommodity Network Flows,” *Lecture Notes in Computer Science*, vol. 1981, pp. 301–315, 2001.

[207] P. Chardaire and A. Lisser, “Simplex and interior point specialized algorithms for solving nonoriented multicommodity flow problems,” *Operations Research*, vol. 50, no. 2, pp. 260–276, 2002.

[208] I. C. Choi and D. Goldfarb, “Solving multicommodity network flow problems by an interior point method,” *SIAM Proceedings in Applied Mathematics*, vol. 46, pp. 58–69, 1990.

[209] G. L. Schulz and R. R. Meyer, “An interior-point method for block-angular optimization,” *SIAM Journal of Optimization*, vol. 1, no. 4, pp. 583–601, 1991.

[210] A. A. Assad, “Multicommodity Network Flows - A Survey,” *Networks*, vol. 8, pp. 37–91, 1978.

[211] J. L. Kennington, “Multicommodity flows: A state-of-the art survey of linear models and solution techniques,” *Operations Research*, vol. 26, pp. 209–236, 1978.

[212] J. W. Mamer and R. D. Mcbride, “A Decomposition-Based Pricing Procedure for Large-Scale Linear Programs : An Application to the Linear Multicommodity Flow Problem,” *Management Science*, vol. 46, no. 5, pp. 693–709, 2000.

[213] M. Minoux, *Mathematical Programming. Theory and Algorithms*. John Wiley & Sons, 1995.

[214] J. L. Kennington and M. Shalaby, "An effective subgradient procedure for minimal cost multicommodity flow problems," *Management Science*, vol. 23, no. 9, pp. 994–1004, 1977.

[215] D. Medhi, "Bundle-Based Decomposition for Large Scale Convex Optimization: Error Estimate," *Mathematical Programming*, vol. 66, pp. 79–101, 1994.

[216] A. Frangioni, "Dual-ascent methods and multicommodity flow problems," Ph. D. Thesis, University of Pisa, 1997.

[217] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows Theory Algorithms and Applications*. Prentice-Hall Inc., 1993.

[218] T. Larsson and D. Yuan, "An Augmented Lagrangian Algorithm for Large Scale Multicommodity Routing," *Computational Optimization and Applications*, vol. 27, pp. 187–215, 2004.

[219] F. Alvelos and J. M. V. De Carvalho, "An Extended Model and a Column Generation Algorithm for the Planar Multicommodity Flow Problem," *Networks*, vol. 50, no. 1, pp. 3–16, 2007.

[220] J. Castro and N. Nabona, "An implementation of linear and nonlinear multicommodity network flows," *European Journal Of Operational Research*, vol. 2217, no. 95, 1996.

[221] F. Ortega and L. A. Wolsey, "A branch-and-cut algorithm for the single-commodity, uncapacitated, fixed-charge network flow problem.pdf," *Networks*, vol. 41, no. 3, pp. 143–158, 2003.

[222] R. L. Rardin and L. A. Wolsey, "Valid inequalities and projecting the multicommodity extended formulation for uncapacitated fixed charge network flow problems," *European Journal of Operational Research*, vol. 71, no. 1, pp. 95–109, 1993.

[223] T. G. Crainic, M. Gendreau, and J. M. Farvolden, "A Simplex-Based Tabu Search Method for Capacitated Network Design," *INFORMS Journal on Computing*, vol. 12, no. 3, pp. 223–236, Jul. 2000.

[224] T. G. Crainic, A. Frangioni, and B. Gendron, "Bundle based relaxation methods for multicommodity capacitated fixed charge network design," *Discrete Applied Mathematics*, vol. 112, pp. 73–99, 2001.

[225] M. Chouman, T. G. Crainic, and B. Gendron, "A Cutting-Plane Algorithm for Multicommodity Capacitated Fixed-Charge Network Design," Technical Report, CIRRELT-2009-20, Montreal, 2009.

[226] F. Say and C. Bazlamacci, "Minimum concave cost multicommodity network design," *Telecommunication Systems*, no. 2007, pp. 181–203, 2008.

[227] F. Say, "Minimum Concave Cost Multicommodity Netwoek Design,"M.Sc. Thesis, Middle East Technical University, 2005.

[228] B. A. J. Yaged, “Minimum cost routing for static network models,” *Networks*, vol. 1, pp. 139–172, 1971.

[229] C. Bazlamacci and K. S. Hindi, “Enhanced Adjacent Extreme-point Search and Tabu Search for the Minimum Concave-cost Uncapacitated Transshipment Problem,” *Journal of the Operational Research Society*, vol. 47, no. 9, pp. 1150–1165, 1996.

[230] T. L. Magnanti and P. Mirchandani, “Shortest paths, single origindestination network design, and associated polyhedra,” *Networks*, vol. 23, pp. 103–121, 1993.

[231] F. Barahona, “Network Design Using Cut Inequalities,” *SIAM Journal of Optimization*, vol. 6, no. 3, pp. 823–837, 1996.

[232] V. Gabrel, A. Knippel, and M. Minoux, “A Comparison of Heuristics for the Discrete Cost Multicommodity Network Optimization Problem,” *Journal of Heuristics*, vol. 9, no. 1996, pp. 429–445, 2003.

[233] B. Fortz and M. Labbe, “Polyhedral approaches to the design of survivable networks,” in *Handbook of Optimization in Telecommunications*, M. G. C. Resende and P. M. Pardalos, Eds. Springer, 2006.

[234] O. Gunluk, “A branch-and-cut algorithm for capacitated network design problems,” *Mathematical Programming Programming*, vol. 86, pp. 17–39, 1999.

[235] A. M. C. A. Koster, S. Orlowski, C. Raack, G. Baier, and T. Engel, “Single Layer Cuts for Multilayer Network Design Problems,” *Discrete Mathematics*.

[236] A. Bley, R. Klahne, U. Menne, C. Raack, and R. Wessaly, “Multi-layer network design a model-based optimization approach.”, Proceedings of PGTS, Berln, Germany, 2008.

[237] E. Kubilinskas, P. Nilsson, and M. Pioro, “Design Models for Robust Multi-Layer Next Generation Internet Core Networks Carrying Elastic Traffic,” *Journal of Network and Systems Management*, vol. 13, no. 1, pp. 57–76, Mar. 2005.

[238] S. Orlowski, M. Pioro, A. Tomaszewski, and R. Wessaly, “SNDlib 1.0–Survivable Network Design Library,” Technical Report, ZIB-07-15, Zuse Instutute Berlin, 2007.

[239] R. K. Ahuja and J. B. Orlin, “Multi-Exchange Neighborhood Structures for the Capacitated Minimum Spanning Tree Problem Multi-Exchange Neighborhood Structures for the Capacitated Minimum Spanning Tree Problem,” *Operations Research*, no. September 1998, 2001.

[240] E. Uchoa, R. Fukasawa, J. Lysgaard, A. Pessoa, M. Poggi de Aragao, D.Andrade,. "Robust Branch Cut-And-Price For The Capacitated Minimum Spanning Tree Problem Over A Large Extended Formulation," Math. Program, vol.112, pp.443–472, 2008.

[241] K. Altinkemer and H. Pirkul, "Heuristics with Constant Error Guarantees for the Multicenter Capacitated Minimum Spanning Tree Problem," *Journal of Information & Optimization Sciences*, vol. 13, no. 1, pp. 49–71, 1992.

[242] W. Liu, "A lower bound for the Steiner tree problem in directed graphs," *Networks*, vol. 20, no. 6, pp. 765–778, 1990.

[243] T. Koch and A. Martin, "Solving Steiner Tree Problems in Graphs to Optimality," *Networks*, vol. 32, pp. 207–232, 1998.

[244] J. E. Mitchell, K. Farwell, and D. Ramsden, "Interior point methods for large-scale linear programming," in *Handbook of Optimization in Telecommunications*, M. G. C. Resende and P. M. Pardalos, Eds. Springer, 2006.

[245] J. F. Kurose and W. Ross, *Computer Networking A Top Down Approach Featuring the Internet*. Addison-Wesley, 2000.

[246] O. Solyalı and H. Süral, "The one-warehouse multi-retailer problem: reformulation, classification, and computational results," *Annals of Operations Research*, vol. 196, no. 1, pp. 517–541, 2012.

[247] J. F. Benders, "Partitioning procedures for solving mixed variables programming problems," *Numerische Mathematik*, vol. 4, pp. 238–252, 1962.

[248] A. M. Costa, "A survey on benders decomposition applied to fixed-charge network design problems," *Computers & Operations Research*, vol. 32, no. 6, pp. 1429–1450, Jun. 2005.

[249] T. L. Magnanti and R. T. Wong, "Discrete Location Theory," 1st Editio., P. B. Mirchandani and R. L. Francis, Eds. USA: Wiley, 1989, pp. 209–262.

[250] G. L. Nemhauser and L. A. Wolsey, *Integer and Combinatorial Optimization*. Wiley, 1988.

[251] Z. C. Taşkın, "Benders Decomposition," in *Encyclopedia of Operations Research and Management Science*, J. J. Chochran, Ed. Wiley, 2010.

[252] A. M. Costa, J. Cordeau, and G. Laporte, "Models and branch-and-cut algorithms for the Steiner tree problem with revenues, budget and hop constraints," *Networks*, vol. 53, no. 2, pp. 141-159, 2009.

[253] D. Mcdaniel and M. Devine, "A Modified Benders' Partitioning Algorithm for Mixed Integer Programming," *Management Science*, vol. 24, no. 3, pp. 312–319, 1977.

[254] W. Rei, J.-F. Cordeau, M. Gendreau, and P. Soriano, "Accelerating Benders Decomposition by Local Branching," *INFORMS Journal on Computing*, vol. 21, no. 2, pp. 333–345, Nov. 2008.

[255] C. A. Poojari and J. E. Beasley, "Improving benders decomposition using a genetic algorithm," *European Journal of Operational Research*, vol. 199, no. 1, pp. 89–97, Nov. 2009.

[256] T. L. Magnanti and R. T. Wong, “Accelerating Benders Decomposition: Algorithmic Enhancement and Model Selection Criteria,” *Operations Research*, vol. 29, no. 3, pp. 464–484, 1981.

[257] M. Fischetti, D. Salvagnin, and A. Zanette, “A note on the selection of Benders’ cuts,” *Mathematical Programming*, vol. 124, no. 1–2, pp. 175–182, May 2010.

[258] M. Fischetti, D. Salvagnin, and A. Zanette, “Minimal Infeasible Subsystems and Benders cuts,” Technical Report, 2008.

[259] N. Papadakos, “Practical enhancements to the Magnanti–Wong method,” *Operations Research Letters*, vol. 36, no. 4, pp. 444–449, Jul. 2008.

[260] G. K. D. Saharidis, M. Minoux, and M. G. Ierapetritou, “Accelerating Benders method using covering cut bundle generation,” *International Transactions in Operational Research*, vol. 17, no. 2, pp. 221–237, Mar. 2010.

[261] T. L. Magnanti and R. T. Wong, “Accelerating Benders Decomposition: Algorithmic Enhancement and Model Selection Criteria,” *Operations Research*, vol. 29, no. 3, pp. 464–484, 1981.

[262] G. K. D. Saharidis and M. G. Ierapetritou, “Improving benders decomposition using maximum feasible subsystem (MFS) cut generation strategy,” *Computers & Chemical Engineering*, vol. 34, no. 8, pp. 1237–1245, Aug. 2010.

[263] E. M. de Sá, R. S. de Camargo, and G. de Miranda, “An improved Benders decomposition algorithm for the tree of hubs location problem,” *European Journal of Operational Research*, vol. 226, no. 2, pp. 185–202, Apr. 2013.

[264] J. Naoum-Sawaya and S. Elhedhli, “An interior-point Benders based branch-and-cut algorithm for mixed integer programs,” *Annals of Operations Research*, Nov. 2010.

[265] Q. Botton, B. Fortz, L. Gouveia, and M. Poss, “Benders Decomposition for the Hop-Constrained Survivable Network Design Problem,” *INFORMS Journal on Computing*, pp. 1–14, 2011.

[266] B. Fortz and M. Poss, “Operations Research Letters,” *Operations Research Letters*, vol. 37, pp. 359–364, 2009.

[267] E. Kubilinskas and M. Pi, “Design Models for Robust Multi-Layer Next Generation Internet Core Networks Carrying Elastic Traffic,” *Network*, vol. 13, no. 1, 2005.

[268] M. Poss, “Models and algorithms for network design problems,” Ph.D. Thesis, Universite Libre de Bruxelles, 2011.

[269] M. Gunkel, R. Leppla, M. Wade, A. Lord, D. Schupke, G. Lehmann, C. Fürst, T. E. S. Gmbh, D.- Darmstadt, A. Park, M. Heath, I. Ip, L. Technologies, N. Systems, and D.- Nuremberg, “A Cost Model for the WDM Layer,” in *Photonics in Switching*, 2006, pp. 3–8.

[270] B. Gavish and K. Altinkemer, “Backbone Network Design Tools with Economic Tradeoffs,” *ORSA Journal on Computing*, vol. 2, pp. 236–252, 1990.

APPENDIX – A

TELECOMMUNICATION NETWORK STRUCTURE

Let's see how an e-mail is delivered from sender to receiver using an example. Suppose, Alex is leaving in Seattle and sends an e-mail to his friend Berk living in Ankara that are given as point A and point B in Figure 40.

Figure 40. Sending an E-mail from A to B – 1

Since Alex is sending an e-mail and Berk is receiving it, there should be two computers at points A and B. Then, e-mail is sent from one computer to another, but how does the e-mail go from one computer to another? As shown in Figure 41, there are two e-mail servers that enable this delivery.

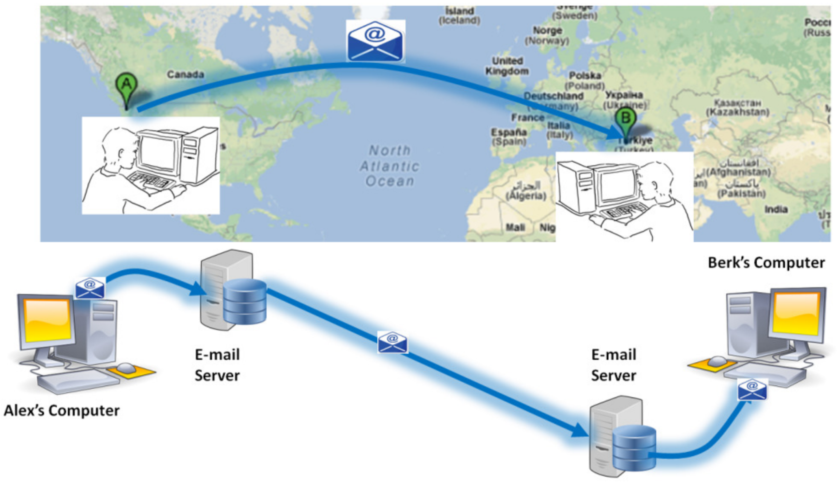


Figure 41. Sending an E-mail from A to B – 2

Let's look at the journey of this e-mail on telecommunication network components. Both Alex and Berk live in a building, such as a house or an office as shown in Figure 42.

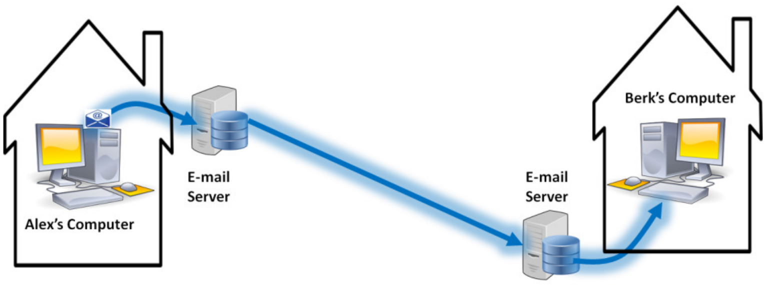


Figure 42. Sending an E-mail from A to B – 3

They both have their neighbors living in other buildings at the same region of their houses as presented in Figure 43.

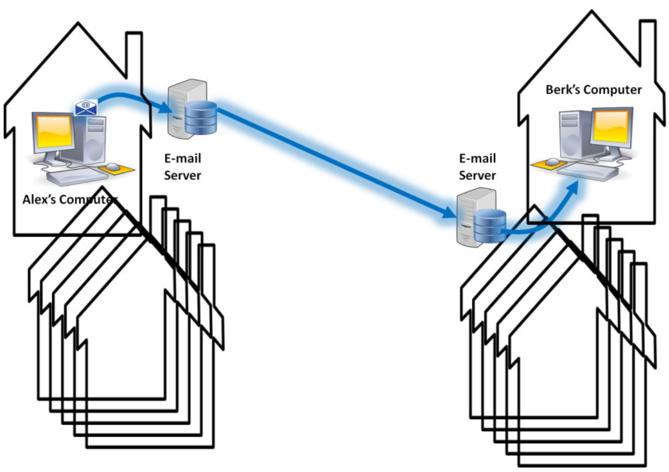


Figure 43. Sending an E-mail from A to B – 4

Alex's neighbors constitute a network and they are all connected to a common terminal box. Terminal box is connected to a regional network regional network is connected to USA backbone network as presented in Figure 44.

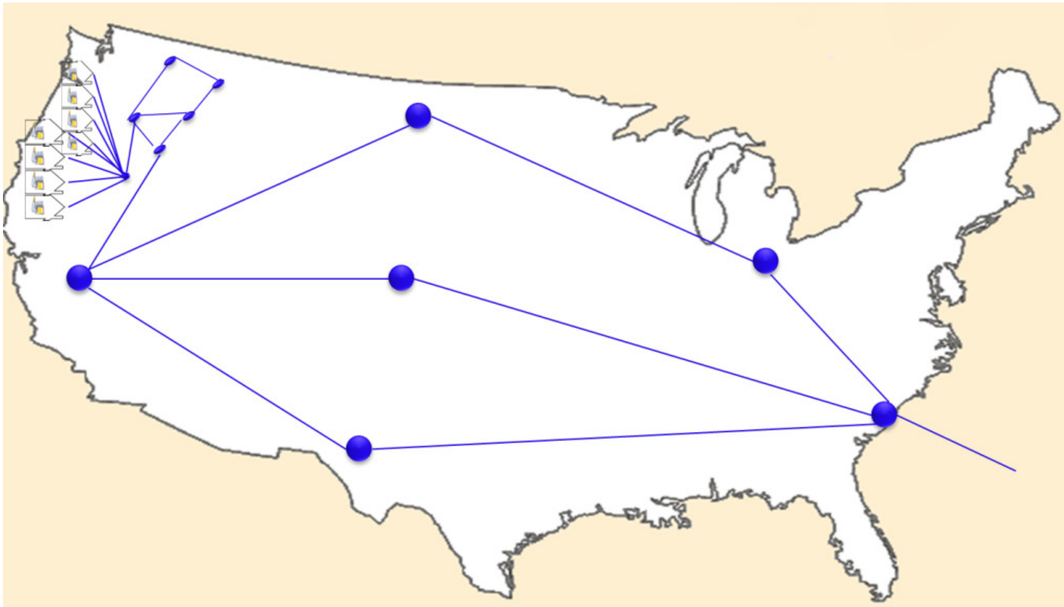


Figure 44. USA Backbone Network

The USA backbone network is connected to the Europe backbone network through a transatlantic leased line. The regional network that Berk's computer is connected to via a terminal box is connected to the Europe Backbone as presented in Figure 45.

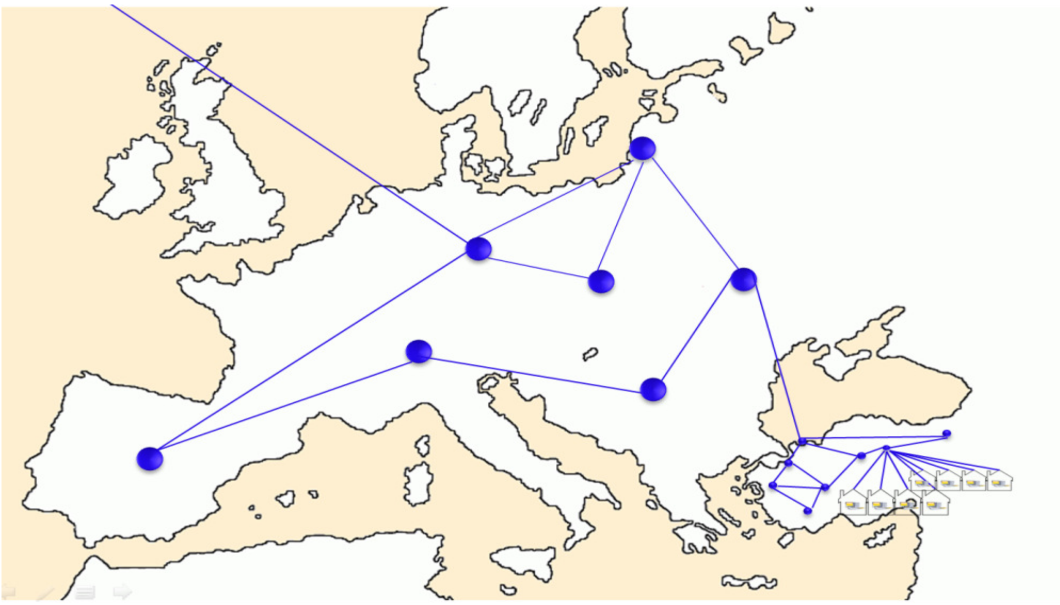


Figure 45. Europe Backbone Network

The physical routing of the e-mail through these different telecommunication network types is presented in Figure 46.

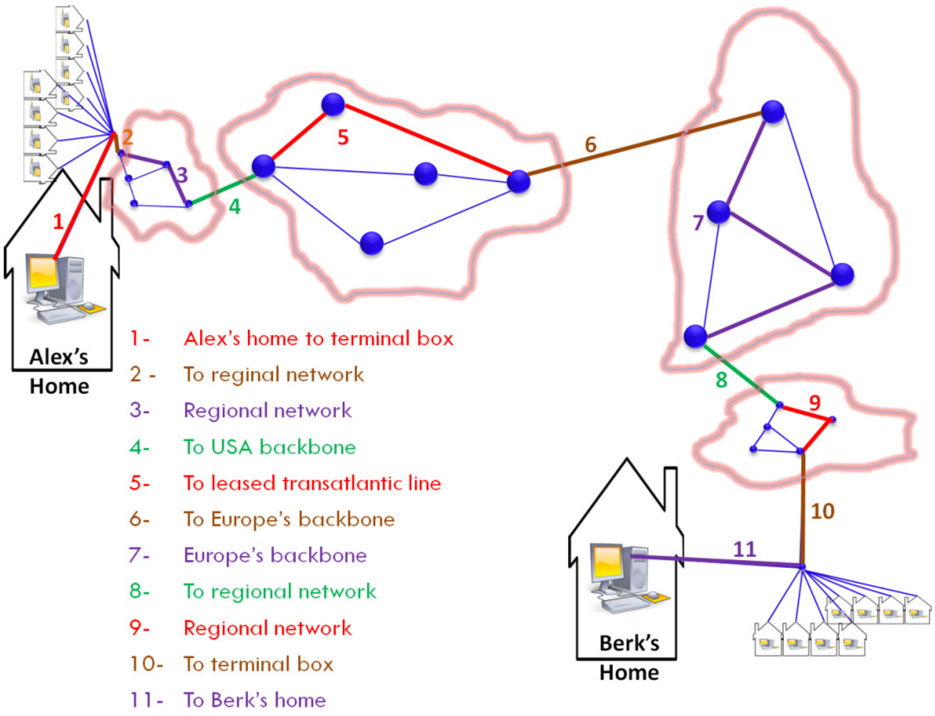


Figure 46. Physical Routing of an E-mail

In real life, a number of terminal boxes are connected to a node in regional network and a number of regional networks are connected to a single backbone network node. Then, the general structure of the telecommunication networks is an hierarchical network structure as given in Figure 47.

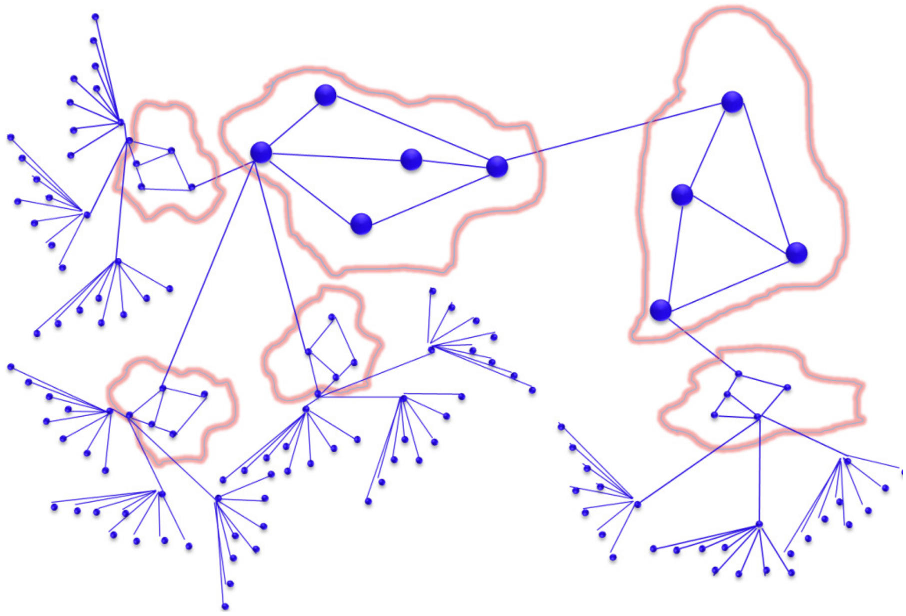


Figure 47. General Structure of Telecommunication Networks

Thus, telecommunication networks consist of different type of networks that serve to regions of different sizes. The capacity and transmission rate of network components differ for these networks, i.e., capacity and transmission rate requirement of a regional network link or node is much less than a backbone link or node regarding the demand of networks arising from the population of the regions. This constitutes the multi-level structure of telecommunication networks as present in Figure 48.

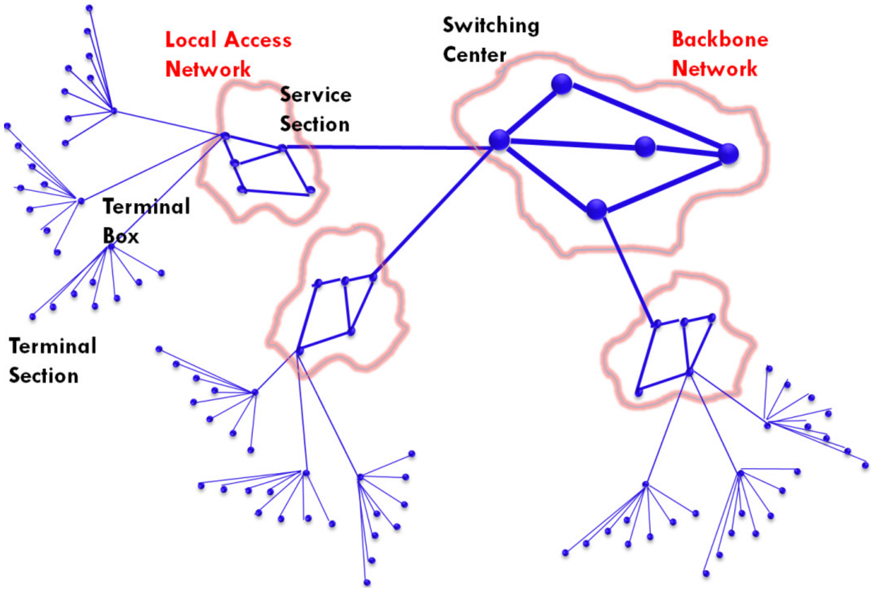


Figure 48. Multi-Level Network Structure

In telecommunication networks, the signals are packed into larger packages as they go to higher levels. This process is called concentration (multiplexing in optical networks). Concentration is done by node hardware. Concentration process is presented in Figure 49.

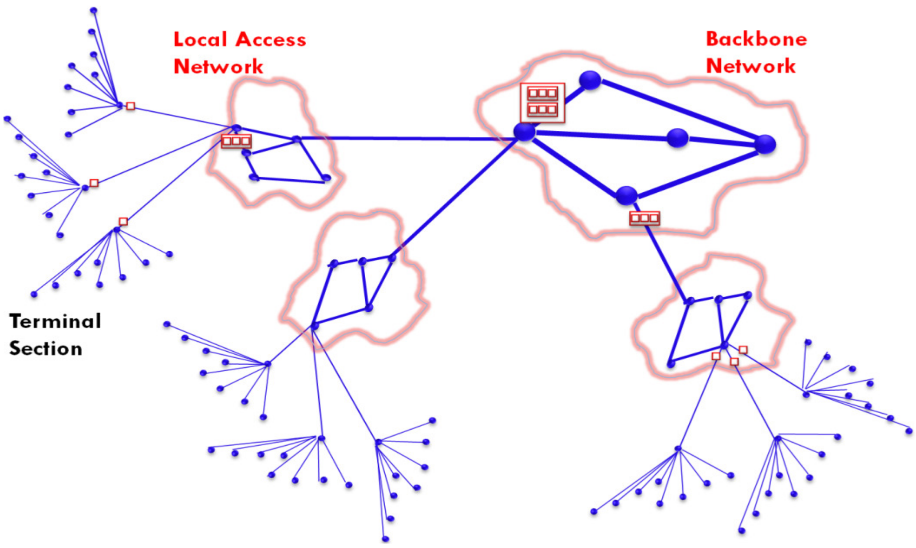


Figure 49. Concentration in Multi-Level Telecommunication Networks

As Figure 50 presents, a telecommunication network consists of nodes and links that correspond to telecommunication hardware in nodes and signal transmitting connections in the links.

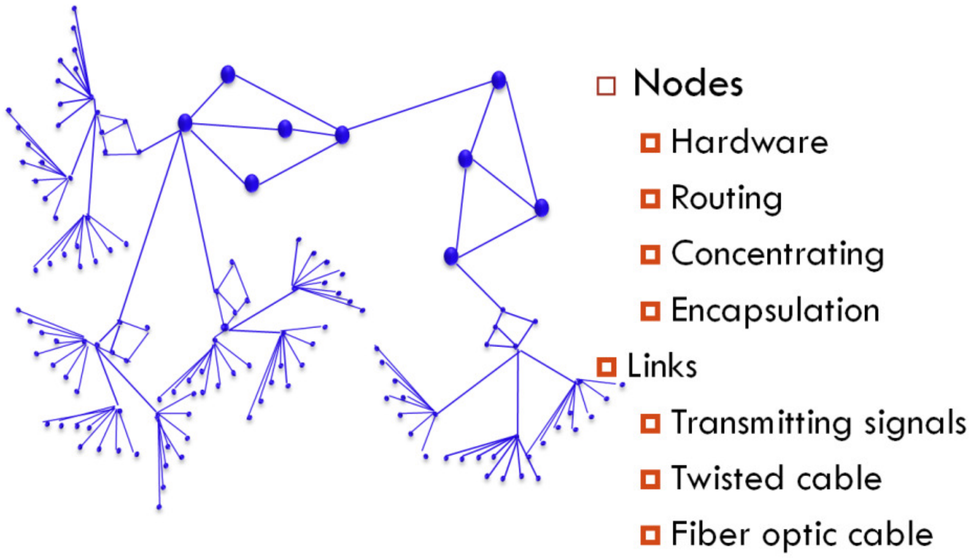


Figure 50. Components of Telecommunication Networks – 1

Let's magnify the section given in Figure 51 and look closer to the telecommunication network components.

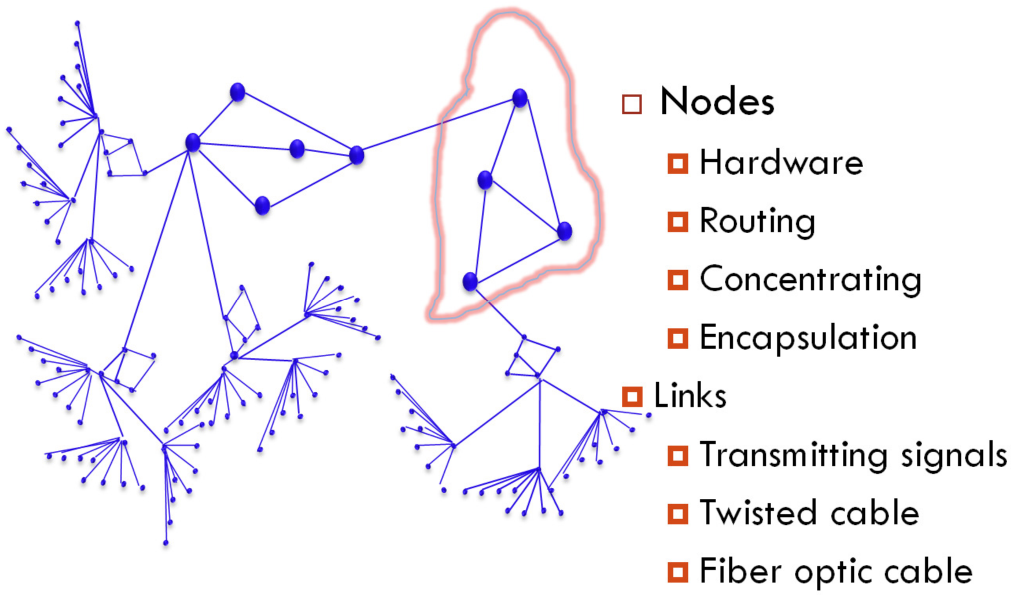


Figure 51. . Components of Telecommunication Networks – 2

Telecommunication network components are basically links and nodes. Nodes are responsible for transmission of signals between the nodes. The links can be of different technologies including twisted cable, fiber optic cable, etc. Specifications including cost, capacity, transmission rate, maximum transmission distance, and quality of service of the links change according to the technology used. In a single network, more than one technology can be used as shown in Figure 52.

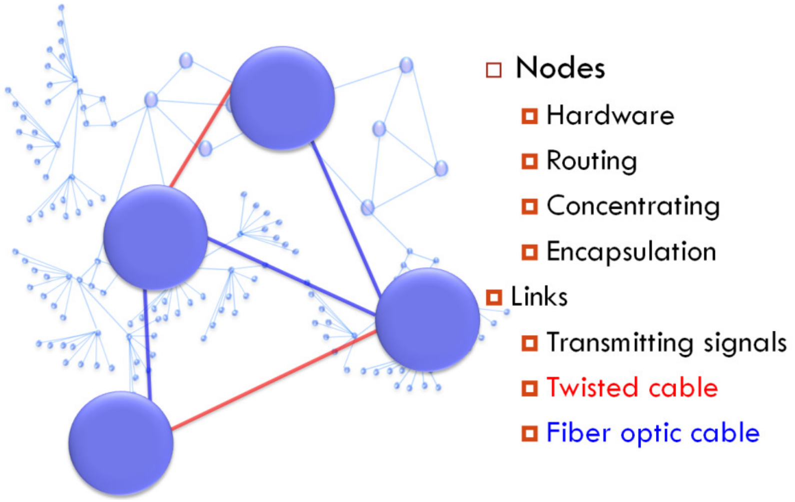


Figure 52. Links in Telecommunication Networks

Nodes in telecommunication networks are responsible for different processes including concentration, routing/switching, encapsulation, multiplexing/demultiplexing, wavelength conversion, etc. that change from technology to technology. A single node in telecommunication networks consists of several devices like line card, transponders, converters etc. to perform these processes. The nodes are presented in Figure 53.

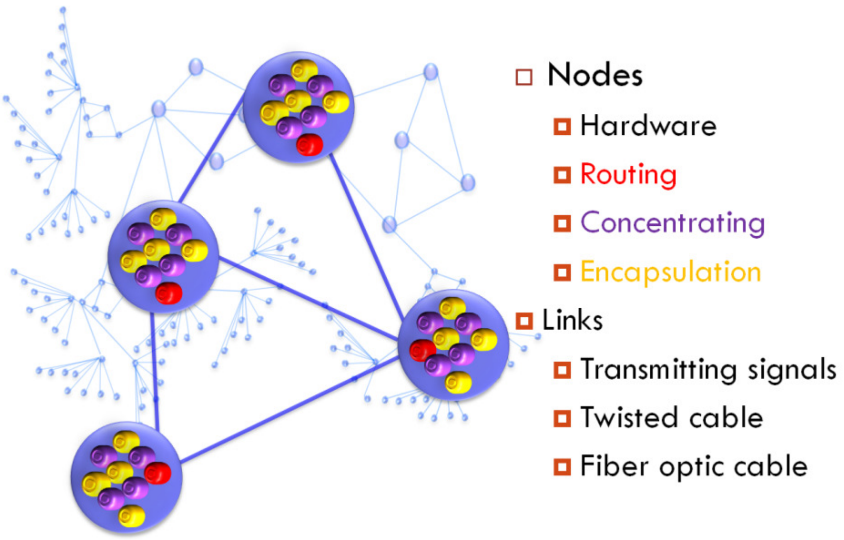


Figure 53. Nodes in Telecommunication Networks

In a telecommunication network, more than one technology works interdependently. The interfaces of the technologies are the node components that are also responsible for technology conversion, i.e., encapsulation. Then there are some logical links between these node components. An abstraction is presented in Figure 54. Multi-technology and multi-level telecommunication networks are called multi-layer networks.

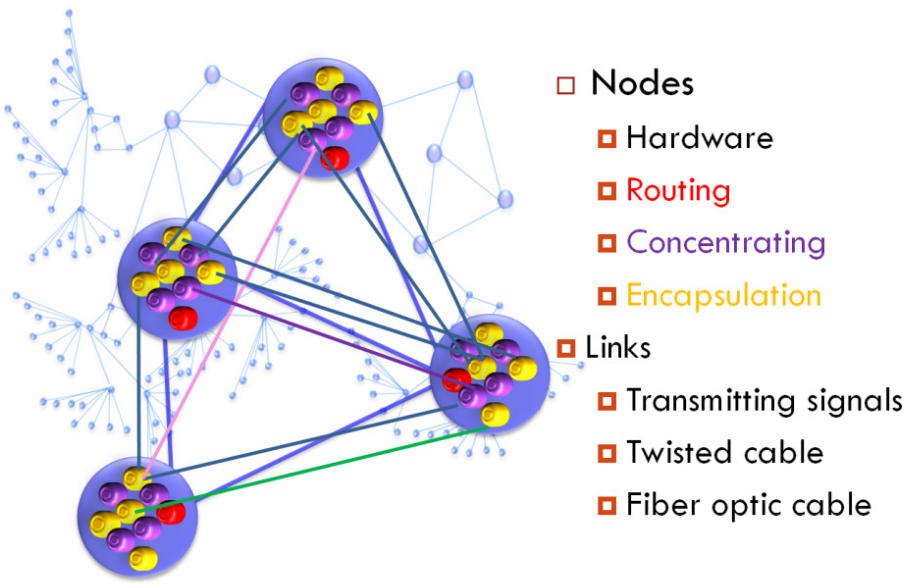


Figure 54. Logical Connections Between Nodes

APPENDIX – B

PROOF OF CONJECTURE 1: MAIN STEPS

Conjecture 1: $Z_{LP}(\text{NFF}) \leq Z_{LP}(\text{EFF-EF})$.

In order to prove Conjecture 1, $F(\text{NFF}) \subseteq F(\text{EFF-EF})$ is proved as a first step. Then a counter example is shown that strict equality between the feasible spaces does not hold as the last step.

In order to show that $F(\text{NFF}) \subseteq F(\text{EFF-EF})$, a feasible solution $(x, f, U, V, W, Y) \in F(\text{NFF})$ is taken an a feasible solution $(\overline{f}, y, \overline{x}) \in F(\text{EFF-EF})$ is constructed.

Suppose (x, f, U, V, W, Y) is a feasible solution to LP relaxation of NFF, i.e., $(x, f, U, V, W, Y) \in F(\text{NFF})$. Then, in this solution there is at least one path from s^k to t^k for each commodity $k \in K$. Total amount of flow on these paths for each commodity is d^k. Without loss of generality, assume that there is one module type for multiplexers, demultiplexers and fiber optic cables and nodes are uncapacitated.

Any $x_{ij}^{k,pr}$ value in this feasible solution is the fraction of demand d^k, $k \in K$, flowing on physical arc $(i, j) \in A$. $x_{ij}^{k,pr} > 0$ means that there is a path between nodes p and r that uses arc $(i, j) \in A$ and routes $a \leq d^k x_{ij}^{k,pr}$ amount of commodity $k \in K$, i.e., the physical edges used by this path has at least a amount of flow of commodity $k \in K$. Note that, a commodity $k \in K$ can be routed by more than one physical path between nodes p and r that uses the physical arc $(i, j) \in A$. Using $x_{ij}^{k,pr} > 0$ values and flow balance equations in NFF, the physical paths between each node pair p and r, carrying flow of commodity $k \in K$ and the amount of flow between p and r on each of these physical paths can be identified.

For any commodity $k \in K$ and any node pair $(p, r) \in N \times N$, arcs $(i, j) \in A$ such that $x_{ij}^{k,pr} > 0$ in the feasible solution (x, f, U, V, W, Y) induce a graph $G' = (N', A')$ such that $N' = \{i : x_{ij}^{k,pr} > 0, i \in N\}$ and $A' = \{(i, j) : x_{ij}^{k,pr} > 0, (i, j) \in A\}$. Any feasible solution of a single commodity network flow problem on $G' = (N', A')$ with link capacities of $x_{ij}^{k,pr}$ for $\forall (i, j) \in A'$, and a commodity with source and sink nodes as (p, r) and amount of demand D gives which physical paths are used to route the commodity using the logical links between nodes p and r. Note that, D is the total amount of flow of commodity $k \in K$ routed between nodes p and r in base units:

$$D = \sum_{\substack{j:(p,j)\in A \\ p\neq t^k, j\neq s^k}} d^k x_{pj}^{k,pr}$$

Hence, feasible $x_{ij}^{k,pr}$ values give the physical paths corresponding to the parallel logical links between (p,r) in EFF-EF and their amount of flow. So, using feasible $x_{ij}^{k,pr}$ values, we can identify the $f_{l,pr}^{k}$ values where l is the logical link corresponding to a physical path between (p,r) that is found as a feasible solution to the single commodity flow problem explained above. Note that, in NFF all flow is in terms of base units. However, in EFF-EF logical link flows are in terms of layer 1's units. So, a conversion parameter γ is used to convert base units to the layer 1's routing unit.

In order to write an affine transformation between $f_{l,pr}^{k}$ and $x_{ij}^{k,pr}$ variables, we can use the following iterative process to find a feasible solution to the single commodity network flow problem defined on $G' = (N', A')$.

Algorithm 1– Find Physical Paths Between Nodes p and r Given Feasible $x_{ij}^{k,pr}$ Values

Initiate:
Set number of paths between nodes p and r to zero: $nPaths := 0$

Set number of iterations to one: $iter := 1$
While $A^{k,pr} \neq \varnothing$

find minimum flow value between nodes p and r: $minF = \min_{(i,j)\in A^{k,pr}} (d^k x_{ij}^{k,pr})$

record the arc with the minimum flow: $minA = \arg\min_{(i,j)\in A^{k,pr}} (x_{ij}^{k,pr})$

find a path $Piter$ from node p to node r using $minA$ that routes $minF$ amount of flow

decrement the capacities by used capacity in $Piter$:

$$x_{ij}^{k,pr} := x_{ij}^{k,pr} - \min_{(i,j)\in A^{k,pr}} (x_{ij}^{k,pr}),\ \forall (i,j) \in P_{iter}$$

assign set $A_{iter}^{k,pr} = \{(i,j) : x_{ij}^{k,pr} = 0 \ and \ (i,j) \in P_{iter}\}$

update set of arcs $A^{k,pr}$ that still has capacity, $x_{ij}^{k,pr} > 0$: $A^{k,pr} := A^{k,pr} \setminus A_{iter}^{k,pr}$

Increment iteration count: $iter := iter + 1$

Increment number of paths between nodes p and r: $nPaths := nPaths + 1$

end while

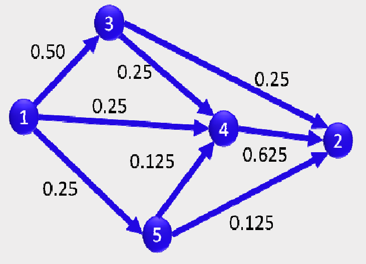
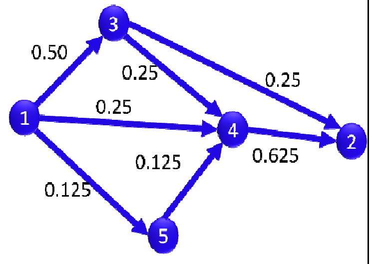
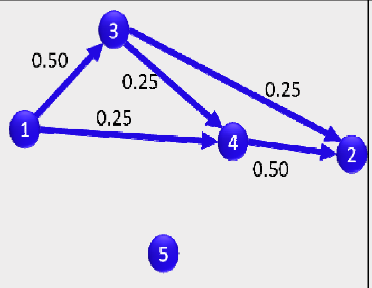
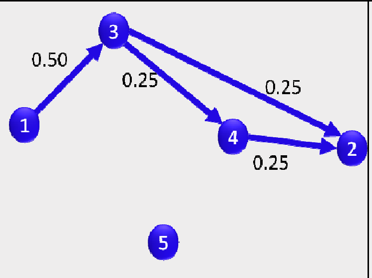
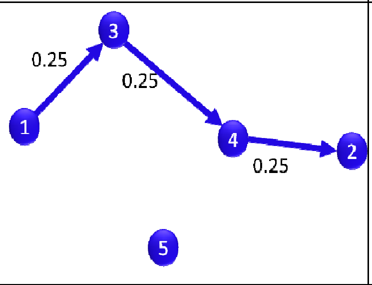
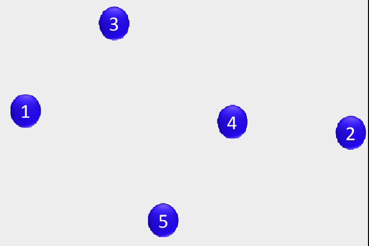
Example:

Let's say, we have a feasible solution to NFF with the following x values for commodity 1 with a demand value of 1 unit between nodes 1 and 2:

$$x_{42}^{1,12} = 0.625,\ x_{13}^{1,12} = 0.50,\ x_{15}^{1,12} = x_{54}^{1,12} = x_{32}^{1,12} = x_{34}^{1,12} = 0.25,\ x_{15}^{1,12} = x_{54}^{1,12} = 0.125.$$

The results of Algorithm 1 are presented in Table 31.

Table 31. Algorithm 1 Iteration Results

Iter	Graph	Computations
1		$A^{1,12} = \{(1,3),(1,4),(1,5),(3,2),(3,4),(4,2),(5,4),(5,2)\}$ $minF = 0.125, minA = (5, 2)$ $P_1 = \{(1,5),(5,2)\}$ $A_1^{1,12} = \{(5,2)\}$ $nPaths = 1$
2		$A^{1,12} = \{(1,3),(1,4),(1,5),(3,2),(3,4),(4,2),(5,4)\}$ $minF = 0.125$ $minA = (1,5)$ $P_2 = \{(1,5),(5,4),(4,2)\}$ $A_2^{1,12} = \{(1,5),(5,4)\}$ $nPaths = 2$
3		$A^{1,12} = \{(1,3),(1,4),(3,2),(3,4),(4,2)\}$ $minF = 0.25$ $minA = (1, 4)$ $P_3 = \{(1,4),(4,2)\}$ $A_3^{1,12} = \{(1,4)\}$ $nPaths = 3$
4		$A^{1,12} = \{(1,3),(3,2),(3,4),(4,2)\}$ $minF = 0.25$ $minA = (3, 2)$ $P_4 = \{(1,3),(3,2)\}$ $A_4^{1,12} = \{(3,2)\}$ $nPaths = 4$
5		$A^{1,12} = \{(1,3),(3,4),(4,2)\}$ $minF = 0.25$ $minA = (3, 4)$ $P_5 = \{(1,3),(3,4),(4,2)\}$ $A_5^{1,12} = \{(1,3),(3,4),(4,2)\}$ $nPaths = 5$
6		$A^{1,12} = \{\}$

Therefore, for any logical link $l : l \in L_{pr}$, if $\exists P_{iter} = E_l : iter = \{1,...,nPaths\}$, then

$$\overline{f}_{l,pr}^{k} = \frac{\min\limits_{(i,j)\in A^{k,pr} \setminus \bigcup\limits_{t=1:iter-1} A_t^{k,pr}} \left\{d^k x_{ij}^{k,pr}\right\} - \sum\limits_{m=1:iter-1} \begin{Bmatrix} \min\limits_{(i,j)\in A_m^{k,pr}} \left\{d^k x_{ij}^{k,pr}\right\},\ if \ \arg\min\limits_{(i,j)\in A^{k,pr} \setminus \bigcup\limits_{t=1:iter-1} A_t^{k,pr}} \left\{d^k x_{ij}^{k,pr}\right\} \in P_m \\ 0,\ otherwise \end{Bmatrix}}{\gamma}$$

else, $\overline{f}_{l,pr}^{k} = 0$.

In NFF, using the $x_{ij}^{k,pr}$ variables, a lightpath routing is done via f_{ij}^{pr} variables in order to find how many lightpaths are needed to route the flow on the physical arc (i,j) between nodes p and r. f_{ij}^{pr} is equal to at least the number of different physical paths between nodes p and r that use the arc (i,j) regardless of the size of flows. Because, different physical paths between nodes p and r correspond to different logical links and they can be routed on the same fiber as long as they use different wavelengths.

If we are given feasible f_{ij}^{pr} variables, we can find feasible number of logical links, y^l, in EFF-EF using the logical links that we obtained by using Algorithm 1. Then,

Let number of logical links needed to route flow from node p to r be

$$y_l^{pr} = \min\limits_{(i,j)\in A^{k,pr} \setminus \bigcup\limits_{t=1:p-1} A_t^{k,pr}} \left\{f_{ij}^{pr}\right\} - \sum\limits_{m=1:p-1} \begin{Bmatrix} \min\limits_{(i,j)\in A_m^{k,pr}} \left\{f_{ij}^{pr}\right\}, & if \ \arg\min\limits_{(i,j)\in A^{k,pr} \setminus \bigcup\limits_{t=1:p-1} A_t^{k,pr}} \left\{f_{ij}^{pr}\right\} \in P_m \\ 0, & otherwise \end{Bmatrix}$$

if any logical link $l : l \in L_{pr}$, if $\exists P_p = E_l : p = \{1,...,nPaths\}$, else, $y_l^{pr} = 0$.

Therefore, for any logical link $l : l \in L_{pr}$, if $\exists P_p = E_l : p = \{1,...,nPaths\}$, then $y_l = y_l^{pr} + y_l^{rp}$, else, $y_l = 0$.

The number of physical links in NFF, $V_{ij}, \{i,j\} \in E$ corresponds to number of physical links x_e, $e \in E$. Then, $\overline{x}_e = V_{ij}, \qquad \forall e = \{i,j\}$

Let's show that the constructed solution of EFF-EF satisfies the constraints (21.1)- (21.3) in Section 3.10.1.

Constraint (21.1) is flow balance constraints of logical layer:

$$\sum_{j\in I}\sum_{l\in L_{ij}} f_{l,ij}^{k} - \sum_{j\in I}\sum_{l\in L_{ij}} f_{l,ji}^{k} = \begin{cases} \overline{d}^{k}, & if\ i = s^k \\ -\overline{d}^{k}, & if\ i = t^k \\ 0, & o.w. \end{cases}, \qquad i \in I \ and \ k \in K \qquad (21.1)$$

where $\overline{d}^{k}$ is demand in terms of first layers routing unit. Algorithm 1is used to allocate the total flow in physical links to logical links flows realized as physical paths between the end nodes of logical links. Then, Algorithm 1 we know that:

$$\sum_{l \in L_{ij}} \overline{f}_{l,ij}^{k} = \frac{\sum_{(i,r)\in A} d^k x_{ir}^{k,ij}}{\gamma}$$

If we substitute $\sum_{l \in L_{ij}} \overline{f}_{l,ij}^{k} = \frac{\sum_{(i,r)\in A} d^k x_{ir}^{k,ij}}{\gamma}$ with $\sum_{l \in L_{ij}} \overline{f}_{l,ij}^{k}$ in (21.1) and divide both sides with d^k / γ, we get the following:

$$\sum_{j\in I}\sum_{(i,r)\in A} x_{ir}^{k,ij} - \sum_{j\in I}\sum_{(p,i)\in A} x_{pi}^{k,ji} = \begin{cases} 1 & if\ i = s^k \\ -1 & if\ i = t^k \\ 0 & o.w. \end{cases}, \qquad i \in I\ and\ k \in K$$

Recall constraints (4.3) and (4.5) of NFF presented in Section 4.5:

$$\sum_{\substack{j\in N:\\(i,r)\in A\\ j\neq s^k}}\sum_{\substack{j\in N\\ j\neq i}} x_{ir}^{k,i,j} - x_{i,2,1}^{k} - x_{i11}^{k} = \begin{Bmatrix} 1, if\ \ s^k = i\, and\ \ l^k = 1 \\ 0, otherwise \end{Bmatrix} \qquad \forall i \in N\, and\, \forall k \in K \mid i \neq t^k \tag{6.18}$$

$$\sum_{\substack{p\in N:\\(p,i)\in A\\ j\neq t^k}}\sum_{\substack{j\in N\\ j\neq i}} x_{pi}^{k,j,i} - x_{i,1,2}^{k} - x_{i1}^{k} = \begin{Bmatrix} 1, if\ \ t^k = i\, and\ \ l^k = 1 \\ 0, otherwise \end{Bmatrix} \qquad \forall i \in N\, and\, \forall k \in K \mid i \neq t^k \tag{4.5}$$

If $i = s^k$, then

$\sum_{\substack{j\in N:\\(i,r)\in A\\ j\neq s^k}}\sum_{\substack{j\in N\\ j\neq i}} x_{ir}^{k,i,j} = 1 + x_{i,2,1}^{k} + x_{i11}^{k}$ from (4.3) and $\sum_{\substack{p\in N:\\(p,i)\in A\\ j\neq t^k}}\sum_{\substack{j\in N\\ j\neq i}} x_{pi}^{k,j,i} = x_{i,1,2}^{k} + x_{i1}^{k}$ from (4.5)

Then, $\sum_{\substack{j\in N:\\(i,r)\in A\\ j\neq s^k}}\sum_{\substack{j\in N\\ j\neq i}} x_{ir}^{k,i,j} - \sum_{\substack{p\in N:\\(p,i)\in A\\ j\neq t^k}}\sum_{\substack{j\in N\\ j\neq i}} x_{pi}^{k,j,i} = 1$

If $i = t^k$, then

$\sum_{\substack{j\in N:\\(i,r)\in A\\ j\neq s^k}}\sum_{\substack{j\in N\\ j\neq i}} x_{ir}^{k,i,j} = x_{i,2,1}^{k} + x_{i11}^{k}$ from (4.3) and $\sum_{\substack{p\in N:\\(p,i)\in A\\ j\neq t^k}}\sum_{\substack{j\in N\\ j\neq i}} x_{pi}^{k,j,i} = 1 + x_{i,1,2}^{k} + x_{i1}^{k}$ from (4.5)

Then, $\sum_{\substack{j\in N:\\(i,r)\in A\\ j\neq s^k}}\sum_{\substack{j\in N\\ j\neq i}} x_{ir}^{k,i,j} - \sum_{\substack{p\in N:\\(p,i)\in A\\ j\neq t^k}}\sum_{\substack{j\in N\\ j\neq i}} x_{pi}^{k,j,i} = -1$

If $i \neq s^k, i \neq t^k$, then

$\sum_{\substack{j\in N:\\(i,r)\in A\\ j\neq s^k}}\sum_{\substack{j\in N\\ j\neq i}} x_{ir}^{k,i,j} = x_{i,2,1}^{k} + x_{i11}^{k}$ from (4.3) and $\sum_{\substack{p\in N:\\(p,i)\in A\\ j\neq t^k}}\sum_{\substack{j\in N\\ j\neq i}} x_{pi}^{k,j,i} = x_{i,1,2}^{k} + x_{i1}^{k}$ from (4.5)

Then, $\sum_{\substack{j\in N:\\(i,r)\in A\\j\neq s^k}}\sum_{\substack{j\in N\\j\neq i}} x_{ir}^{k,i,j} - \sum_{\substack{p\in N:\\(p,i)\in A\\j\neq t^k}}\sum_{\substack{j\in N\\j\neq i}} x_{pi}^{k,j,i} = 0$

Constraints (21.2) are capacity constraints for logical links:

$$\sum_{k\in K}(f_{l,ij}^k + f_{l,ji}^k) \leq Uy_l \qquad l=(i,j)\in L \qquad (21.2)$$

From construction:

(i) If, for any logical link $l : l \in L_{ij}$, if $\exists P_{iter} = E_l : iter = \{1,...,nPaths\}$, then $\overline{f}_{l,ij}^k = X_{pr}^{k,ij}$, else $\overline{f}_{l,ij}^k = 0$, where:

$$X_{pr}^{k,ij} = \frac{\min\limits_{(p,r)\in A^{k,ij}\setminus \bigcup\limits_{t=1:iter-1} A_t^{k,ij}}\left\{d^k x_{pr}^{k,ij}\right\} - \sum\limits_{m=1:iter-1}\begin{Bmatrix}\min\limits_{(p,r)\in A_m^{k,ij}}\left\{d^k x_{pr}^{k,ij}\right\}, \; if \; \arg\min\limits_{(p,r)\in A^{k,ij}\setminus \bigcup\limits_{t=1:iter-1} A_t^{k,ij}}\left\{d^k x_{pr}^{k,ij}\right\}\in P_m \\ 0, \quad otherwise\end{Bmatrix}}{\gamma}$$

(ii) In addition, we know that for any logical link $l : l \in L_{ij}$, if $\exists P_{iter} = E_l$: $iter = \{1,...,nPaths\}$, then
$y_l = y_l^{ij} + y_l^{ji}$ else, $y_l = 0$ where,

$$y_l^{ij} = \min\limits_{(p,r)\in A^{k,ij}\setminus \bigcup\limits_{t=1:p-1} A_t^{k,ij}}\left\{f_{pr}^{ij}\right\} - \sum\limits_{m=1:iter-1}\begin{Bmatrix}\min\limits_{(p,r)\in A_m^{k,ij}}\left\{f_{pr}^{ij}\right\}, \; if \; \arg\min\limits_{(p,r)\in A^{k,ij}\setminus \bigcup\limits_{t=1:iter-1} A_t^{k,ij}}\left\{f_{pr}^{ij}\right\}\in P_m \\ 0, \quad otherwise\end{Bmatrix}$$

(iii) Substituting (i) and (ii) in constraint (21.2), we get:
$\sum_{k\in K}(X_{pr}^{k,ij} + X_{pr}^{k,ji}) \leq U(y_l^{ij} + y_l^{ji}) \; l=(i,j)\in L$

(iv) We know that the following inequality is true from constraint 4.14 of NFF from Section 4.5:

$$\sum_{\substack{k\in K:\\t^k\neq i, s^k\neq j}} d^k x_{ij}^{k,p,r} \leq \gamma_1 f_{ij}^{p,r} \qquad \forall i,j\in N \mid (i,j)\in A \, and \, \forall (p,r)\in N\times N \mid p\neq r$$

(v) Note that, γ_1 is a conversion parameter to compute the number of logical links in NFF. In NFF, there is no logical link capacity. But instead, the capacity for multiplexer and demultiplexers are used for more accurate logical layer cost. In order to compare NFF and EFF-EF, we can assume that all multiplexer and demultiplexer capacities in NFF are equal to the logical link capacity in EFF-EF, which is U. Then, $\gamma_1 = \gamma U$ since all operations in NFF is held in base units.

The $X_{pr}^{k,ij}$ and y_l^{ij} values are computed using the Algorithm 1 given the feasible solution $(x,f,U,V,W,Y)\in F(\text{PF})$, then the $A_t^{k,ij}$ sets used to compute these values are the same for a single logical link $l : l\in L_{ij}$, if $\exists P_{iter} = E_l$: $iter = \{1,...,nPaths\}$. Then, from (iv) and (v), $(x, f, U, V, W, Y) \in F(\text{PF})$ satisfies the inequality given as (iii) and hence the constraint (21.2).

Constraints (21.3) are capacity constraints for physical links:

$$\sum_{l \in L_e} y_l \le B x_e \qquad e \in E \qquad (21.3)$$

From construction we know that the total number of logical links between nodes p and r that use edge $e = \{i, j\}$ is $f_{ij}^{pr} + f_{ji}^{pr}$. Then, total number of logical links that use edge $e = \{i, j\}$ is the sum of this term over all node pairs which is equal to $\sum_{l \in L_e} y_l$ in EFF-EF:

$$\sum_{l \in L_e} y_l = \sum_{\substack{(p,r) \in N \times N: \\ p \ne r, p \ne j, r \ne i}} f_{ij}^{pr} + f_{ji}^{pr} = \left(\sum_{\substack{(p,r) \in N \times N: \\ p \ne r, p \ne j, r \ne i}} f_{ij}^{pr} + \sum_{\substack{(p,r) \in N \times N: \\ p \ne r, p \ne j, r \ne i}} f_{ji}^{pr} \right)$$

Note that, number of lightpaths that can be routed using a single fiber optic cable is B in EFF-EF and q_{ij}^3 in NFF. Then from construction, $B\overline{x}_e = q_{ij}^3 V_{ij}$, for $e = \{i, j\}$. Then from constraint 4.15 of NFF from Section 4.5, which is given below, we can conclude that the feasible solution $(x, f, U, V, W, Y) \in F(\text{PF})$ satisfies Constraint (21.3) of EFF-EF.

$$\sum_{\substack{(p,r) \in N \times N: \\ p \ne r, p \ne j, r \ne i}} f_{ij}^{p,r} + \sum_{\substack{(p,r) \in N \times N: \\ p \ne r, p \ne i, r \ne j}} f_{ji}^{p,r} \le q_{ij}^{3,m} V_{ij}^m \qquad \forall \{i, j\} \in E$$

From the steps of proof provided above, we see that, number of total logical links that use a specific edge is available in the NFF solution. However, in the NFF, the total bandwidth needed for multiplexers and demultiplexers can be computed explicitly. We prefer to use the latter instead of the number of logical links in our model as we can incorporate different cost values to the multiplexers and demultiplexers instead of providing just an approximate cost for a logical link as it is done in the EFF-EF.

Another important result of this proof is that the NFF results in a solution with the same detail level as the EFF-EF, although unlike the EFF-EF NFF doesnot need to have all the physical paths realizing the logical links in advance. Hence, the NFF model is more compact then the EFF-EF regarding the size of variables.

In order to complete the proof of Conjecture 1, a counter example solution for strict equality between $F(\text{NFF})$ and $F(\text{EFF-EF})$ is found.

APPENDIX – C

COMPUTATIONAL TESTS FOR MUTLI-LAYER NETWORK DESIGN PROBLEMS IN LITERATURE

In this chapter, the details of computational test instances of studies on multilayer telecommunication network design problem are provided.

Orlowski [7] and Koster et al. [6] reported the instances in SNDLIB given in Table 32 and a 67-node network which is not available.

Table 32. Test Instances Solved in [7]

Instance	|V|	|E|	|K|	Density	
				E	K
polska	12	18	66	0.27	1.00
nobel-us	14	21	91	0.23	1.00
nobel-germany	17	26	121	0.19	0.89
nobel-eu	28	41	378	0.11	1.00
germany50	50	88	662	1.33	10.03

Orlowski report that they used real world data with the graphs given in the graph. The data was taken from their partner in the project, Nokia-Siemens, hence there are only few explanations about how they created the instances which is not enough to use in our test instances.

- For polska network, logical capacities are STM -1 and STM4 capacity units. 4 logical units can share a physical unit (4 channels in each fiber). Physical layer and demands are taken from SNDLIB. No node cost is used. They limited the number of admissible paths between each node pair with 50.
- Nobel-us, nobel-germany and germany50 are used by realistic data provided by Nokia-Siemens (cost and capacity of logical and physical links depend on the length of the link, Dwivedi-Wagner population model which uses number of hosts in the cities to estimate the demand between cities are used to generate demand data. Thus, only available information about those instances is the physical layer that is taken by the SNDLIB.
- For nobel-us network, physical link cost is set to zero. The logical layer consists of two logical links between each pair of nodes.
- For germany 50 network and the 67-node network, logical layer consists of two-three logical links between each pair of nodes.
- For nobel-germany network, logical layer consists of four-five logical links between each pair of nodes. For nobel-eu network, physical capacity and demands are taken from SNDLIB. Physical layer costs are used as given in SNDLIB. Logical link

capacities are taken as 2.5 Gbit/s and 40 channels per physical link. The logical layer consists of two logical links between each pair of nodes.

Us67 network is not available in the SNDLIB and on the internet. However, they report that the g50 network is more challenging than us67 network since the logical layer density of us67 is lower (because they limited the number of admissible logical links between each pair of nodes in the logical layer with 2 or 3 in us67).

Fortz and Poss [5] and Poss [268] reported computational tests using 35 randomly generated test instances with 8, 9 and 10 nodes and the test instances from SNDLIB given in Table 33. Note that, they used less complex capacity formulation to model the multilayer network design problem and capacity formulation cannot model many practical side constraints as reported in Chapter 3.10.

Table 33. Test Instances Solved in [5] and [268]

Instance	**|V|**	**|E|**	**|K|**	**Density**	
				E	**K**
pdh	11	34	24	0.62	0.44
di-yuan	11	42	22	0.76	0.40
dfn-gwin	11	47	110	0.85	2.00
polska	12	18	66	0.27	1.00
nobel-us	14	21	91	0.23	1.00
atlanta	15	22	210	0.21	2.00

Fortz and Poss and Poss [5], [268] took logical link capacity a 64 and physical link capacity as 128, demands are randomly generated between 0 and 63, and cost of edges are based on link lengths for the randomly generated instances.

Mattia [122] reported computational tests using the test instances derived by the SNDLIB instances given in Table 34.. Note that she used capacity formulation, which is less complex but not capable of modeling many practical side constraints as reported in Section 3.10, to model the multilayer telecommunication network design problem.

Mattia used physical networks and derived logical networks of these instances as given in SNDLIB. Logical layer with logical links that use up to 3 hops (3 physical links), up to 5 hops and all available logical links are generated for each instance except for cost 266 and nobel –eu. For these two larger network instances, logical layer is derived by logical links that use up to 3 hops (3 physical links) and up to 5 hops. In addition, physical costs are derived from the costs of the first available capacity module of the original problem in the SNDLIB. She used original demands but, during optimization, she replaced commodities (i, j, d_{ij}) and (i, j, d_{ji}), if any, by a unique commodity $(i, j, d_{ij} + d_{ji})$ since the graphs are undirected. The size of the physical capacity module is taken as 8 in all instances and the size of the logical capacity module is randomly taken as a value depending on the mean demand of each instance.

Table 34. Test Instances Solved in [122]

Instance	**|V|**	**|E|**	**|K|**	**Density**	
				E	**K**
pdh	11	34	24	0.62	0.44
di-yuan	11	42	22	0.76	0.40
polska	12	18	66	0.27	1.00
nobel-us	14	21	91	0.23	1.00
atlanta	15	22	210	0.21	2.00
nobel-germany	17	26	121	0.19	0.89
nobel-eu	28	41	378	0.11	1.00
cost266	37	57	1332	0.09	2.00

APPENDIX – D

SIZE OF SNDLIB TEST INSTANCES

The size of the master and primal of subproblem for the SNDLIB test instances used to assess performance of GG-BD_IR are presented in Table 35.

Table 35. Problem Sizes for Master and Sub Problem

Instance	*k*	Master Problem		Sub Problem	
		#Row	#Col	#Row	#Col
dfn-bwin	all	946	6,735	73,290	480,510
pdh	all	1245	6365	30619	122750
di-yuan	all	1253	7829	30051	139745
dfn-gwin	all	1,258	8,744	120,017	779,624
abilene	all	1,600	3,513	180,906	371,214
polska	all	1,603	4,182	92,868	222,570
nobel-us	all	2,570	6,839	208,110	518,903
atlanta	all	3,173	8,329	587,907	1,476,666
newyork	all	3,890	21,015	835,046	4,366,898
nobel-germany	all	4,651	12,881	510,649	1,341,776
geant	10	10,201	9,284	2,168,636	3,681,284
france	10	15,046	10,676	1,093,031	2,760,334
janos-us	5	16,943	8,804	3,228,262	4,845,334
norway	5	19,006	9,264	3,547,293	5,506,884
sun	5	19,006	9,532	361,901	543,555
nobel-eu	5	21,210	11,257	2,511,308	3,657,350
india35	2	41,731	8,135	2,655,805	3,816,565
cost266	2	49,342	11,134	9,393,685	12,197,758
janos-us-ca	2	57,860	12,756	12,179,915	15,669,870
giul39	2	57,885	10,555	8,812,933	12,445,125
pioro40	2	62,490	11,379	5,135,890	7,157,015
germany50	2	122,589	21,060	9,005,636	11,678,301

CURRICULUM VITAE

PERSONAL INFORMATION

Surname, Name: Yüksel Ergün, İnci
Nationality: Turkish (TC)
Date and Place of Birth: 15 June 1982, İzmir
Marital Status: Married
Phone: +90 312 592 2509
Fax: +90 312 592 3030
email: iyuksel@aselsan.com.tr

EDUCATION

Degree	Institution	Year of Graduation
MS	METU Industrial Engineering	2007
BS	Bilkent University, Industrial Engineering	2004
High School	Bursa Ali Osman Sönmez Science High School	2000

WORK EXPERIENCE

Year	Place	Enrollment
2004-Present	ASELSAN Inc.	Senior Expert Engineer

FOREIGN LANGUAGES

Advanced English, Basic French

PUBLICATIONS

Conference Proceedings

1. İ. Yüksel, Nur Evin Özdemirel, Levent Kandiller, "Atış Kontrol Sistemi Hata Analizi ile Tek Atışta Vuruş İhtimalinin Hesaplanması (Single Shot Hit Probability Computation For Air Defense Based On Error Analysis)", National Modeling and Simulation Conference (USMOS), Ankara, Turkey, 2007.
2. T. Gülez, İ. Yüksel, M. Avcı Özgün, D. Arslan, S. Arslan, Ö. Kırca, S. Meral, S. Batun, Z. Kirkizoğlu, F. Soykan, L. Acun, Ü. Kıyak, C. Somer, "Tehdit Değerlendirme ve Silah Tahsisi Algoritması: Bir Uygulama (Threat Evaluation and Weapon Assignment Algorithm: An Application)", National Modeling and Simulation Conference (USMOS), Ankara, Turkey, 2007.

3. İ. Yüksel, T. Gülez, M. Avcı Özgün, D. Arslan, S. Arslan, Ö. Kırca, "Tehdit Değerlendirme ve Silah Tahsisi Algoritması: Dinamik Uygulama (Threat Evaluation and Weapon Assignment Algorithm: Dynamic Application)", Defence Technologies Conference, Ankara, Turkey, 2008.
4. İnci Yüksel Ergün, "Sistem Mühendisliği ve Sistem Düşüncesi: Süreç Uygulaması (System Engineering and Systems Thinking: Process Application)", Defence Technologies Conference, Ankara, Turkey, 2010.
5. E. Lappi, I. Yüksel-Ergün, P. Hörling, T. Lappi, "Future Indirect Firing Cost Analysis", Scythe: Proceedings and Bulletin of the International Data Farming Community, Issue 11, October, 2011.
6. Ç. Çolakoğlu, İ. Yüksel-Ergün, M. Günay, M. Avcı-Özgün, "Alçak Irtifa Karaya Konuşlu Hava Savunma Sistemlerinin Birlikte Çalişabilirliğine Yönelik NIAG Çalişmalari ve LLAPI Entegrasyon Projesi (NIAG Studies for Low Altitude Ground Based Air Defense Systems Interoperability and LLAPI Integration)", Defence Technologies Conference, Ankara, Turkey, 2012.

Conference Presentations

1. İ. Yüksel, T. Gülez, M. Avcı Özgün, D. Arslan, S. Arslan, Ö. Kırca, S. Meral, S. Batun, Z. Kirkizoğlu, "Threat Evaluation and Weapon Allocation: An Application", INFORMS Annual Conference, Seattle, Washington, USA, 2007.
2. İnci Yüksel Ergün, "Sistem Mühendisliği ve Sistem Düşüncesi: Süreç Uygulaması (System Engineering and Systems Thinking: Process Application)" INCOSE Turkey Meeting of System Engineers, ODTÜ, Ankara, Turkey, 2009.
3. İ. Yuksel Ergun, O. Kirca, H. Sural, "Telekomünikasyon Ağ Tasarımı Problemleri (Telecommunication Network Planning Problems)", Operations Research and Industrial Engineering Congress, İstanbul, Turkey, 2010.
4. İ. Yuksel Ergun, O. Kirca, H. Sural, "Telekomünikasyon Ağ Tasarımı Problemi (Telecommunication Network Planning Problem)", Operations Research and Industrial Engineering Congress, İstanbul, Turkey, 2012.
5. İ. Yuksel-Ergun, O. Kirca, H. Sural, "A Decomposition Based Solution Approach for Multilayer Telecommunication Network Design Problem", International IIE Conference/ Operations Research and Industrial Engineering Congress, İstanbul, Turkey, 2012.
6. İ. Yuksel-Ergun, O. Kirca, H. Sural, "The Multilayer Telecommunication Network Design Problem", EURO/INFORMS 26. European Conference on Operational Research, Rome, Italy, 2013.

TRAINING and CERTIFICATES

1. Project Management Processes, Makro Consulting (Şekip Karahan) 06-08 September 2004, Ankara.

2. Using MS PROJECT 2002 Software, Makro Consulting (Demir Özkaya), 04-05 November 2004, Ankara.
3. Written Communication and Reporting (in Turkish), Strata (Kaan Eran), 24 November 2004, Ankara.
4. Systems Engineering, Project Performance International Australia (Robert Calligan), 04-08 September 2006, Ankara.
5. Introduction to Capability Maturity Model Integration v1.2, Wayne Littlefield, 10-12 September 2007, Ankara (Certificate from Carnegie Mellon University).
6. WAID 2007 – Artificial Intelligence and Data Mining Workshop, 3 November 2007, Seattle Washington, USA.
7. Requirements Methodology, Telelogic (Adrian Ditchburn), 05-06 May 2008, Ankara.
8. DOORS for Requirement Management, Telelogic (Naseem Ul Haq), 14-15 May 2008, Ankara.
9. Applying Systems Thinking to Problem Solving Workshop, Prof. Joseph Kanser, 10 June 2008, Utrecht, Hollanda.
10. Robust Product Design, SATEM, 2 January 2009, Ankara.
11. Effective Presentation Techniques (in Turkish), Derin Eğitim, 27 January 2009, Ankara.
12. Quality Costs, KALDER, 29 December 2009, Ankara.
13. Business Writing, DOOR Training and Consulting (Prof. Dr. İsmet Barutçugil), 26-27 April 2010, Ankara.
14. Process Management and Improvement, KALDER, 12-13 May 2010, Ankara.
15. CMMI SCAMPI A and SCAMPI B Training, Wayne Littlefield, Ankara, 2012

PARTICIPATED INTERNATIONAL WORK GROUPS

1. NATO Research and Technology Organization Modeling and Simulation Group – NMSG 088 Data Farming in Support of NATO (2010-2012) – Participated in Design of Experiments Subgroup
2. NATO Industrial Advisory Group – Study Group 160 on NATO Ground Based Air Defence Systems Threat Evaluation and Weapon Assignment Interoperability (2011-2012) – Team Leader of Architectural Solutions Subgroup